Mitigating Position Bias in Cascading User Behaviors for Recommendation

Yan Liu
Nanyang Technological University
College of Computing and Data Science
Singapore
yan010@e.ntu.edu.sg

Yingpeng Du
Nanyang Technological University
College of Computing and Data Science
Singapore
yingpeng.du@ntu.edu.sg

Hongzhi Liu
School of Software and Microelectronics
Peking University
Beijing, China
liuhz@pku.edu.cn

Thang-son Nguyen
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore
nguyen_thanh_son@a-star.edu.sg

Erik Cambria
Nanyang Technological University
College of Computing and Data Science
Singapore
cambria@ntu.edu.sg

***Abstract*—Recommender systems aim to learn users' intrinsic preferences from historical behaviors. However, user behaviors are often strongly influenced by item display positions, which introduces exposure bias and can mislead preference learning. Existing position-aware methods still face two key limitations. (1) Some approaches collect "unbiased" data via randomized shuffling or swapping, yet this may degrade recommendation quality and harm user experience. (2) Methods trained on biased logs usually rely on single-type behaviors, making it difficult to decouple true preference from position effects. In this paper, we propose a position-aware causal model for multi-behavior recommendation. By analyzing the recommendation and interaction process with a causal graph, we show that display position affects both users' willingness to interact and the observation of items, leading to cascading behavioral effects (e.g., position $\rightarrow$ click $\rightarrow$ purchase). To alleviate position bias without sacrificing user experience, we adopt causal inference via backdoor adjustment to block the spurious influence of position on willingness and observation. Meanwhile, we leverage multi-behavior cascades to better disentangle users' intrinsic preferences from position effects. To further combat sparsity, we develop a cascading deconfounding framework that exploits rich auxiliary behaviors to capture and reduce position-induced bias. Extensive experiments on four real-world datasets demonstrate that our approach consistently outperforms state-of-the-art baselines.**

***Index Terms*—Multi-behaviors recommendation, Causal inference learning, Cascading deconfounding framework**

I. Introduction

Recommender systems (RSs), which aim to provide personalized service to users, can enhance both users' satisfaction and platforms' profits [1]–[3]. Most of the existing recommendation methods attempt to learn users' preferences from their historical behaviors, e.g., purchases and clicks in e-commerce. However, besides users' intrinsic preferences, users' behaviors are also affected by the display position of RSs. Taking users' click behaviors as an example, we conduct an A/B test in an online service APP. For users in group A, we keep the original recommendation order of items.

For users in group B, we deliberately swap the items of ranking $1 \sim 3$ with the items of ranking $4 \sim 6$ in recommendation lists. However, the online experiment shows that the click ratios of different display positions in group B (Figure 1(b)) still share a similar trend compared with group A (Figure 1(a)). That is, users' click behavior is profoundly affected by the display position besides their preferences. We argue that this unexpected but reasonable phenomenon is caused by two main reasons: **1) Users' behavioral willingness, which is defined as how willing are people to try [4], always fades with the increase of display position**. Users usually scan the recommendation results from the top to the bottom and their willingness usually depends on whether their needs have been satisfied. **2) The possibility of an item being observed will decline with the increase of its display position**. The display position of items can affect their observation and make the top-ranked items easy to observe by users. Therefore, some user behaviors are caused by the effect of display position rather than the users' intrinsic preferences.

Recently, several methods attempt to explore "unbiased" data to alleviate the effects of display position [5]. However, collecting sufficient unbiased data typically relies on random shuffling or swapping strategies [6], [7], which may degrade recommendation quality and harm user experience [8].

Alternatively, other approaches directly exploit "biased" data and aim to decouple user preferences from position effects. However, these methods often rely on single-typed behaviors and struggle to achieve effective disentanglement [9], [10]. Specifically, as shown in Figure I(a), they incorporate position information during training (i.e., preference + position effect), but remove it entirely at prediction time, considering only user preferences. Therefore, it indicates that the user prefers item 3 (i.e., preference score 0.8) to item 2 (i.e., preference score 0.7), which is wrongly predicted because the user ultimately purchases item 2.

To overcome this limitation, we propose to explore and

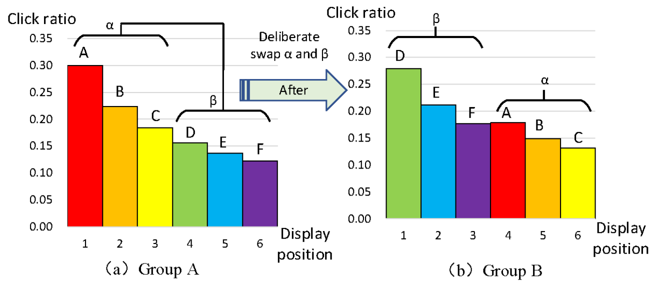


Fig. 1. An online A/B test shows users' click ratio varying with the display position.

TABLE I

AN ILLUSTRATIVE EXAMPLE OF POSITION-AWARE RECOMMENDATION WITH SINGLE-BEHAVIORS AND CASCADING BEHAVIORS. ITEMS 1-4 ARE LOCATED IN DISPLAY POSITIONS 1-4.

(a) Explore on single-behaviors for position-aware recommendation

Item	Preference	Position effect	Click label
Item 1	-0.4	0.4	0
Item 2	0.7	0.3	1
Item 3	0.8	0.2	1
Item 4	-0.1	0.1	0

(b) Explore on cascading behaviors for position-aware recommendation

Item	Preference	Position effect	Click label	Purchase label
Item 1	-0.4	0.4	0	0
Item 2	1.0 $\uparrow$	0.25 $\downarrow$	1	1
Item 3	0.5 $\downarrow$	0.25 $\uparrow$	1	0
Item 4	-0.1	0.1	0	0

make use of the cascading effects behind users' multi-behaviors as shown in Figure I (b). Specifically, we consider that the user's decision to click and purchase item 2 is primarily driven by their preference for it, so the preference score should be higher while the position effect should be lower. Conversely, we consider that the user clicking item 3 is primarily driven by the display position of item 3, so the preference score should be lower while the position effect should be higher. Therefore, the cascading effects behind users' multi-behaviors contribute to effectively clarifying the users' preferences and the effect of display position. Finally, users' target behaviors are quite sparse w.r.t. display positions, making it hard to accurately estimate weights (e.g., propensity score) for unbias learning [11]–[14]. Exploring the multi-behaviors of users can alleviate the sparsity problem of deconfounding learning. Specifically, the data of auxiliary behaviors (e.g., click) is always much richer than that of target behaviors(e.g., purchase), which can be seen as intermediate variables between the display position and the target behavior. Therefore, we can use these auxiliary behaviors to block the bad effects of the display position on the target behaviors.

In summary, we propose a Position-aware Cascading-behavior causal model (PCcam) for personalized recommendation. We propose to analyze how the position confounder affects users' multi-behaviors in real-world scenarios with a causal graph. It indicates that the display position can affect users' willingness and items' observation, and further affects users' behaviors in a cascading way. Firstly, we propose to conduct causal inference learning based on the backdoor adjustment strategy, which can block the effects of display position on users' willingness and items' observation. Secondly, we propose to explore and make use of the cascading effects behind users' multi-behaviors (e.g., position $\rightarrow$ click $\rightarrow$ purchase) to decouple the users' preference and the effect of display position. Thirdly, we propose a cascading deconfounding framework to capture and alleviate the effects of display position which rich auxiliary behaviors. Experimental results on four real-world data sets show that our model consistently outperforms state-of-the-art methods.

II. RELATED WORK

Effects of display position in recommender systems: Debias position for recommendation has been a hot topic in recent years [9], [15]–[17]. Several position-aware models focused on exploring the propensity score for debiasing [5]. These models randomly shuffle [6] or swap [7] ranking results, and then collect users' behaviors in different positions to make the unbiased estimation of the truth propensities. However, these randomization strategies may decrease the ranking quality and harm users' experience dramatically [8]. To address this limitation, several methods directly explore "biased" data without collecting "unbiased" data, which aims at removing the effect of display position on users' behaviors. One class of existing methods utilizes position information for model training and removes the whole effect of display position for prediction [9], [10]. The other class of existing methods estimate weights (e.g., propensity score) of training samples for unbias learning [11]–[14]. However, users' target behaviors may be quite sparse w.r.t. display positions, hindering accurate estimation for unbiased learning.

Causal modeling tried to alleviate the confounder's bad impact, which simultaneously affects both the intermediate and the outcome variables. A standard approach is to conduct the causal inference analysis [18] by the backdoor adjustment and intervention strategies, which has been utilized for items' popularity [19], [20], content [21], [22], social network [23], [24], and videos' duration [25] in recommendation. For example, Zhang et al. [19] proposed a causal framework that performs deconfounded training and causally intervenes the popularity bias during recommendation inference. However, these are unexplored problems for multi-behaviors based recommendation, e.g., how the display position affects users' multi-behaviors in a confounding way.

III. PROBLEM DEFINITION

In this paper, the random variables and their specific values are denoted by uppercase characters (e.g., U) and lowercase characters (e.g., u), respectively. Calligraphic font characters (e.g., $\mathcal{U}$) denote the sample space of corresponding variables. Let $\mathcal{U} = \{u_1, \cdots, u_m\}$ and $\mathcal{I} = \{i_1, \cdots, i_n\}$ denote the sets of m users and n items, respectively. In this paper, we focus on two types of users' behaviors, i.e., click $\mathcal{T}_C = \{(u, i, c)|u \in \mathcal{U}, i \in \mathcal{I}\}$ and purchase $\mathcal{T}_A = \{(u, i, c, a)|u \in \mathcal{U}, i \in \mathcal{I}\}$,

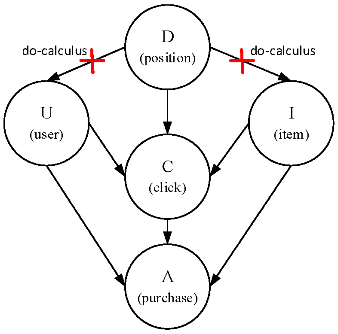


Fig. 2. A causal graph $\mathcal{G}$ describes the recommendation process.

where c and a denote the label of click and purchase respectively. For each interaction between user u and item i, we suppose to know the display position $d(u,i) \in \mathbb{N}_+$ of item i shown in the recommendation list for user u.

Our goal is to learn a function $f_\theta(u,i)$ to predict the probability that a user u will interact with an item i for target behavior (i.e., purchase) based on users' historical multi-behaviors. Therefore, we can generate a recommendation list for each user according to the predicted probability.

IV. Methodology

A. Causal graph for multi-behavior based recommendation

Figure 2 shows a causal graph that illustrates the recommendation and interaction process, which is a directed acyclic graph with nodes denoting random variables and edges denoting causal effects. The nodes and edges in this causal graph $\mathcal{G}$ are explained as follows,

- Node D denotes the display position on which the websites or APPs exhibit items to users. Nodes U and I denote the users and the items in the system, respectively. Nodes C and A denote the click and purchase behavior labels respectively.
- Edge $D \rightarrow U$ represents that the display position affects the behavioral willingness of users, i.e., whether the user is willing to interact (e.g., click or purchase) with RSs, which always fades with the increase of display position.
- Edge $D \rightarrow I$ represents that the display position affects the observation of items. Intuitively, the item displayed in the higher position is more likely to be observed by users.
- Edge $D \rightarrow C$ denotes the display position's direct effect on users' click behaviors generated by RSs, e.g., the recommender system tends to rank more relevant items in the higher position.
- Edge $C \rightarrow A$ denotes the necessary condition for users' purchase behaviors, i.e., users may purchase an item that users prefer after they clicked it.

In the causal graph as shown in Figure 2, the display position D of RSs is a confounder that affects users' willingness and items' observation besides users' multi-behaviors. Position D can affect both click behavior C and purchase behaviors A through several paths. The path $D \rightarrow C \rightarrow A$ indicates that the display position directly affects click and further affects purchase, which should be captured since the RSs should rank more relevant items in the higher position. The remaining paths $D \rightarrow (U/I) \rightarrow C \rightarrow A$ and $D \rightarrow (U/I) \rightarrow A$ represent the display position's effect on the users' click and purchase behaviors through affecting users' behavioral willingness $P(U|D)$ and the items' observation $P(I|D)$. However, such effects on users and items can amplify the users' behaviors on the higher-ranked items rather than their intrinsic preferences, which may mislead the user preference learning for recommendation.

B. Causal inference learning

To learn users' intrinsic preferences with consideration of the potential effects of display position, we propose to conduct causal inference learning without sacrificing users' satisfaction. Specifically, we remove the effects of display position on users and items with an intervention strategy $do(\cdot)$. The do-calculus $do(\cdot)$ aims to modify variables with the desired situation in an intervention way, which can be used to remove the causal effects of their parent nodes in a causal graph [26].

To alleviate the bad effects of display position, we propose to cut off the paths from display position to users U (i.e., $D \rightarrow U$) and items I (i.e., $D \rightarrow I$) as shown in Figure 2, i.e., $do(U=u, I=i)$. Specifically, we propose to learn the causal inference $P_{\mathcal{G}}(A|do(U=u,I=i))$ to model users' intrinsic preferences for target behaviors prediction,

$$
\begin{aligned}
P_{\mathcal{G}}(A|do(U=u,I=i)) &\overset{(1)}{=} \sum\nolimits_{d\in\mathcal{D}} P(A|u,i,d)P(d) \\
&\overset{(2)}{=} \sum\nolimits_{d\in\mathcal{D},c\in\mathcal{C}} P(A,c|u,i,d)P(d) \\
&\overset{(3)}{=} \sum\nolimits_{d\in\mathcal{D},c\in\mathcal{C}} P(A|u,i,d,c)P(c|u,i,d)P(d) \\
&\overset{(4)}{=} \sum\nolimits_{d\in\mathcal{D},c\in\mathcal{C}} P(A|u,i,c)P(c|u,i,d)P(d) \\
&\overset{(5)}{=} \sum\nolimits_{c\in\mathcal{C}} P(A|u,i,c)[\underbrace{\sum\nolimits_{d\in\mathcal{D}} P(c|u,i,d)P(d)]}_{P_{\mathcal{G}}(c|do(U=u,I=i))}
\end{aligned} \tag{1}
$$

where Step (1) is established according to the theory of backdoor adjustment [26], which is an implementation of the do-calculus in a probabilistic way. Step (2) follows the law of total probability; Step (3) follows the conditional probability formula; Step (4) is established because (U,I,C) blocks the path from D to A in the Bayes network; Step (5) is established according to the theory of backdoor adjustment.

Therefore, the causal inference $P_{\mathcal{G}}(A|do(U=u,I=i))$ is decomposed into two key factors: the deconfounded click probability $P_{\mathcal{G}}(c|do(U=u,I=i))$ for user click modeling and the click-aware purchase probability $P(A|u,i,c)$ for user purchase prediction. Specifically, $P_{\mathcal{G}}(c|do(U=u,I=i))$ can make use of the rich click data for better deconfounding learning. $P(A|u,i,c)$ further blocks the paths from display position D to purchase behavior A, which can alleviate the bad effects of display position D.

C. Deconfounding learning

1) Deconfounded click model $P_{\mathcal{G}}(C|do(U,I))$: To alleviate the sparsity problem of deconfounding learning, we utilize auxiliary behaviors for better deconfounding learning. On the one hand, the data of auxiliary behaviors (i.e., click) is always much richer than that of target behaviors (i.e., purchase), based on which we can conduct deconfounding learning more effectively. On the other hand, the auxiliary behaviors are intermediate variables between the display position and the target behavior as shown in Figure 2. Therefore, in theory, we can use these cascading relations to block the bad effects of the display position on the target behaviors.

To learn the deconfounded click model $P_{\mathcal{G}}(C|do(U,I))$, we need to estimate $P(C|U,I,D)$ first according to Equation (1).

Estimate $P(C|U,I,D)$**.** To learn how likely the user will click the item $P(C=1|u,i,d)$ with given user $U=u$, item $I=i$, and display position $D=d$, we propose to use the sigmoid cross-entropy loss as the objective function, i.e.

$$\min \frac{1}{|\mathcal{T}_1|}\sum\nolimits_{(u,i,d,c)\in\mathcal{T}_1} \text{CrossEntropy}(f_1(u,i,d),c) \qquad (2)$$

where $f_1(u,i,d)=P_{\Theta_1}(C=1|u,i,d)$ denotes the position-aware click model with learnable parameters Θ_1. $\mathcal{T}_1 = \{(u,i,d,c)\}$ denotes the training data set for the click model learning, and $c\in\{0,1\}$ denotes the click label. In this paper, we decouple the position-aware click model $f_1(u,i,d)$ into a position-free user-item click model and a position-aware prediction model by a linear function,

$$f_1(u,i,d)=f_{\theta_1}(u,i)+g(u,d) \qquad (3)$$

where $f_{\theta_1}(u,i) =< W_u^1, H_i^1 >$ adopts matrix factorization [27] for the position-free user-item click modeling. W_u^1 and H_i^1 denote the embedding of user u and item i, respectively. The position-aware model $g(u,d)$ aims to capture the effects of display position on click behaviors,

$$g(u,d)=K_u\cdot g'(d) \qquad (4)$$

where $g'(d)$ is a linear transform $g'(x)=\text{linear}(x)$ of the position-aware information, e.g., $g'(d)=d$. To capture users' personalized behavioral willingness, we adopt learnable parameters (e.g., $K_u\geq 0$) to model their behavioral patterns.

Estimate $P_{\mathcal{G}}(C|do(U,I))$**.** To predict the user's deconfounded click behaviors, we estimate $P_{\mathcal{G}}(C=1|do(U=u,I=i))$ with the following reduction,

$$\begin{aligned}
&P_{\mathcal{G}}(C=1|do(U=u,I=i))=\sum\nolimits_{d\in\mathcal{D}}P(C=1|u,i,d)P(d)\\
&\quad=\sum\nolimits_{d\in\mathcal{D}}P(d)f_{\theta_1}(u,i)+\sum\nolimits_{d\in\mathcal{D}}P(d)g(u,d)\\
&\quad=f_{\theta_1}(u,i)+E_D(g(u,d))
\end{aligned} \qquad (5)$$

where $E_D(g(u,d))$ denotes the expectation of $g(u,d)$ on sample space $\mathcal{D}$. The recommendation is essentially a ranking task, so $E_D(g(u,d))$ can be ignored because it will not change the ranking of items for the given user u.

2) Click-aware purchase model $P(A|U,I,C)$*:* Conducting decoupling learning with single-typed behaviors, as described in Equation (2), makes it hard to decouple the users' preferences and the effect of display position effectively. To this end, we propose to capture the cascading pattern of users' behaviors for decoupling learning in a fine-tuning way. Specifically, we propose a click-aware purchase model $P(A|U,I,C)$ to predict users' purchase behaviors based on their click behaviors, which can be learned with the sigmoid cross-entropy loss,

$$\min \frac{1}{|\mathcal{T}_2|}\sum\nolimits_{(u,i,c,a)\in\mathcal{T}_2} \text{CrossEntropy}(f_2(u,i,c),a) \qquad (6)$$

where $f_2(u,i,c)=P_{\Theta_2}(A=1|u,i,c)$ denote the click-aware purchase model with parameters Θ_2. $\mathcal{T}_2=\{u,i,c,a\}$ denotes the training data set for the click-aware purchase model learning, where $c\in\{0,1\}$ and $a\in\{0,1\}$ denote click label and purchase label, respectively. To better decouple the users' preferences and the effect of display position, we propose to model users' purchase behavior $f_2(u,i,c)$ with consideration of users' click willing $\hat{c}=f_1(u,i,d)$, i.e.,

$$\begin{aligned}
f_2(u,i,\hat{c})&=f_{\theta_2}(u,i)+\mu\cdot\hat{c}\\
&=f_{\theta_2}(u,i)+\mu\cdot f_{\theta_1}(u,i)+\mu\cdot g(u,d)
\end{aligned} \qquad (7)$$

where $f_{\theta_2}(u,i) =< W_u^2, H_i^2 >$ denotes the purchase preference score. μ denotes a trade-off coefficient. With the combined learning of the deconfounded click model and the click-aware purchase model, the user's intrinsic click willing on item i can be refined based on the user's cascading behaviors in a fine-tuning way.

D. User Behavior Prediction

With the learned deconfounded click model $P_{\mathcal{G}}(C|do(U,I))$ and click-aware purchase model $P(A|U,I,C)$, we can predict users' target behaviors (i.e. purchase behaviors) through causal inference, i.e.,

$$\begin{aligned}
&P(A=1|do(U=u,I=i))\\
&=\sum\nolimits_{c\in\{0,1\}}P(A=1|u,i,c)P(c|do(U=u,I=i))\\
&\overset{Eq(7)}{=} f_{\theta_2}(u,i)+\mu\sum\nolimits_{c\in\{0,1\}}c\cdot P(c|do(U=u,I=i))\\
&\overset{Eq(5)}{=} f_{\theta_2}(u,i)+\mu\cdot f_{\theta_1}(u,i)
\end{aligned}$$

where $f_{\theta_1}(u,i)$ denotes the deconfounded click score and $f_{\theta_2}(u,i)$ denotes the deconfounded purchase score for recommendation.

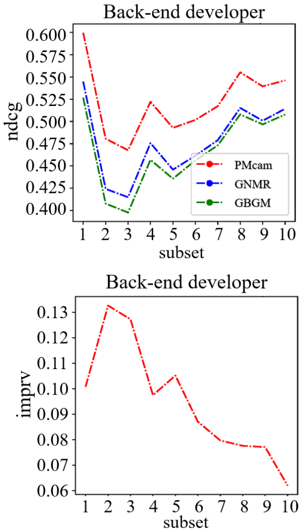
V. Experiments

We conduct several experiments to study the following research questions. **RQ1**: Whether the proposed method outperforms state-of-the-art methods? **RQ2**: Whether the proposed method alleviates the bad effect of display position for user preference modeling via causal inference learning? **RQ3**: Whether the proposed method benefits from capturing the cascading effects behind users' behaviors for recommendation? **RQ4**: Whether the proposed method benefits from exploiting the rich auxiliary click behaviors among display positions for deconfounding learning?

Fig. 3. Performance of different methods. * indicates statistically significant improvement of the proposed method to every baseline model on an independent-samples t-test ($p < 0.01$).

Data	Back-end developers		Software tester		Project manager		Website designer	
Metric	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
LightGCN	0.2701	0.2303	0.3643	0.3345	0.2561	0.1813	0.3206	0.2924
MCBPR	0.2807	0.2375	0.3596	0.3270	0.2680	0.1935	0.3433	0.3065
CMF	0.2315	0.1921	0.3022	0.2708	0.2373	0.1693	0.2792	0.2489
NMTR	0.2423	0.1940	0.3132	0.2740	0.2754	0.1895	0.3004	0.2585
GNMR	0.3102	0.2621	0.3909	0.3553	0.2929	0.2112	0.3559	0.3193
GHCF	0.1972	0.1577	0.2896	0.2634	0.2173	0.1519	0.2271	0.1979
GB_GMN	0.2919	0.2437	0.3761	0.3407	0.2976	0.2158	0.3562	0.3201
PAL	0.2714	0.2282	0.3446	0.3131	0.2653	0.1897	0.3304	0.2951
XPA	0.2292	0.1905	0.2912	0.2627	0.2287	0.1630	0.2646	0.2311
MRS	0.3145	0.2691	0.3811	0.3449	0.3135	0.2357	0.3570	0.3230
PRISM	0.2224	0.1859	0.3004	0.2717	0.2237	0.1567	0.2674	0.2361
IPL	0.2764	0.2297	0.3517	0.3198	0.2637	0.1905	0.3340	0.2985
PP4RNR	0.2700	0.2268	0.3211	0.2926	0.2493	0.1787	0.3049	0.2727
ReSN	0.2566	0.2205	0.3334	0.3125	0.2326	0.1681	0.2844	0.2594
PCcam	**0.3370***	**0.2933***	**0.3998***	**0.3651***	**0.3446***	**0.2739***	**0.3760***	**0.3418***
Imprv	7.17%	9.00%	2.27%	2.76%	9.92%	16.22%	5.32%	5.83%

Fig. 4. Deconfounding analysis.



A. Experimental setup

1) Datasets: We adopt four data sets in an online recruitment APP for the experiments, as there are no available public data sets with both multi-behavior and display position information. This APP aims to provide users with proper jobs. The users can click a displayed job title to scan the detailed description of it, and further apply it if they want the job. We regard the job application of users as the purchase behavior in E-commerce. We collect data in different job categories of information technology, i.e., back-end developer, software tester, project manager, and website designer. For these data sets, we filter out users and items that have less than 5 records and focus on users with more than one type of behavior as in [28]. The statistics of the four data sets are summarized in Table II. We adopt the same data split strategy as in [29], [30], i.e., randomly select 60% of historical interactions of each user for the training set and the remaining are used as the test set.

TABLE II
STATISTICS OF THE EXPERIMENTAL DATA SETS

Dataset	#Users	#Items	#Click	#Purchase
Back-end developer	9,990	35,155	1,042,488	365,684
Software tester	8,486	19,878	1,554,517	698,013
Project manager	6,006	17,001	284,827	78,781
Website designer	47,772	13,778	706,923	286,109

Hyper-parameter settings and evaluation metrics: For a fair comparison, all methods are optimized by the Adam optimizer with the same latent space dimension (i.e., 64), batch size (i.e., 1024), the default choice of initial learning rate (i.e., 0.001), and the regularization coefficient (i.e., 0.0001) for all the experiments. We adopt click-to-purchase ratio $x = CVR(d)$ for position-aware prediction model implementation in Equation (4). For the unique hyper-parameters of the baseline methods, we tune them based on authors' suggestions if they exist, otherwise we tune them to their best. To evaluate the performance of the proposed method and the baseline methods, we adopt two widely used evaluation metrics for recommendation, including NDCG and Hit Ratio (HR) in the top-k recommendation, and we use $k = 10$ empirically. The experimental results are recorded as the average of the ten runs.

We compare the PCcam model with various baselines including single-behavior based methods (LGCN [31]), multi-behavior based methods (MCBPR [32], CMF [33], NMTR [34], GNMR [35], GHCF [36], and MBGMN [28]), and debias based methods (PAL [9], XPA [37], MRS [10], PRISM [38], IPL [39], PP4RNR [40], and ReSN [41]).

2) Overall Performance: Table 2 shows the performances of different methods. To make the table more notable, we bold the best results and underline the best baseline results in each case. We can get the following observations: 1) The proposed method PCcam outperforms all the baselines significantly in all cases, which confirms the effectiveness of the proposed model. Such performance improvement comes from position deconfounding learning and exploring users' multi-behaviors in a cascading way, which can learn users' intrinsic preferences more effectively. It achieves the greatest improvement on the most sparse data set Project manager, owing to PCcam can effectively exploit the auxiliary clicks for better deconfounding learning (RQ1). 2) Among the baselines, multi-behavior based methods (e.g., GNMR, GB_GMN, and MRS) achieve good performance, which indicates effectively exploring multi-behavior information can alleviate the sparsity problem in single-behavior methods. Some multi-behavior based methods perform even worse than single-behavior methods. These models may be misled by the click behaviors, whose records are affected by the bad effect of display position. 3) Among the baselines, MRS achieves the best performance in most cases, which demonstrates the necessity of exploring multi-behavior information and debias learning for display position.

TABLE III
PERFORMANCE OF ABLATION METHODS.

Data	Back-end developer		Software tester	
Metric	HR	NDCG	HR	NDCG
w/o DP	0.3306	0.2892	0.3886	0.3547
w/o CR-UI	0.3173	0.2775	0.3905	0.3547
w/o CR-U	0.2726	0.2277	0.3519	0.3169
w/o CR-I	0.2762	0.2314	0.3583	0.3223
w/o FT	0.3312	0.2914	0.3834	0.3509
w/o MG	0.2806	0.2352	0.3584	0.3256
w/o CD	0.3297	0.2876	0.3909	0.3568
PCcam	**0.3370***	**0.2933***	**0.3998***	**0.3651***

In addition, PAL achieves good performance among single-behavior based methods (NCF and LGCN), which also demonstrates the necessity of debias learning for display position. The underperformance of XPA compared to baseline methods could be attributed to its specialization for search sessions, which may be not applicable to recommendation scenarios.

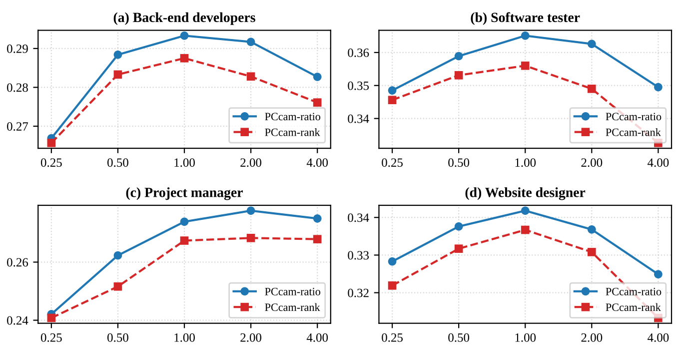


Fig. 5. Performances of PCcam-DP and PCcam-CP vary with μ.

3) Deconfounding Analysis: To investigate whether our method can alleviate the bad effects of the position confounder during the model training phase, we conduct experiments on different test sets that are affected by similar display positions, where each test set collects items that are displayed to users on the same screen in APPs. Specifically, group i contains the items that are shown on the i-th sliding window on a mobile phone. Figure 4 shows the performance of the proposed methods as well as the two baseline methods (GNMR and GBGMN) which fail to model the display position on different subsets. First, the proposed method PCcam significantly outperforms the two baseline methods in all subsets, which confirms the effectiveness of the proposed method for recommendation. Second, the proposed method PCcam shows a larger improvement in the high-position groups which suffer from more influence by the display position in most cases. This confirms the effectiveness of the proposed method for display position deconfounding learning.

B. Ablation study

To evaluate the effectiveness of the model design and the motivations of this paper, we take some special cases of the proposed method as comparisons: **w/o DP**: We do not conduct the Deconfounding learning to remove the bad effect of the display Position by causal learning, that is, removing the position-aware model $g(u, d)$, i.e., $f_1(u, i, d) = f_{\theta_1}(u, i)$. **w/o CR-U/I/UI**: We do not explore the Causal Relationship between click and purchase behaviors, that is, we remove the path from click to purchase in the causal graph, i.e., $f_2(u, i, c) = f_{\theta_2}(u, i)$. Following a similar idea as the existing multi-task methods do, we share the embeddings of users (-U) or items (-I) or users&items (-UI) for multi-task learning [10]. **w/o FT**: We do not conduct the Fine-Tuning learning for the click-aware purchase model learning, which adopts the click label $\{0, 1\}$ in Equation (6) instead. **w/o MG**: We do not perform the Multi-Goal causal learning, only exploiting purchase behavior for deconfounding learning, i.e., only conducting causal learning $f(u, i, d) = f(u, i) + g(u, d)$ based on purchase behaviors. **w/o CD**: We do not conduct the Click-level Deconfounding learning, that is, only remaining the target behavior deconfounding learning.

The ablation study results are shown in Table III. From the results, we have the following conclusions: 1) The performance gap between PCcam and the variant **w/o DP** illustrates the advantage of removing the bad effect of display position on users' behavioral willingness and items' observation. It demonstrates the effectiveness of the proposed deconfounding method via causal inference learning (RQ2). 2) PCcam consistently outperforms its variant **w/o CR** regardless of different embeddings sharing strategies, i.e., users (-U), items (-I), or both (-UI). This indicates the necessity to capture the click-to-purchase causal relation, which contributes to better behavior prediction compared to traditional multi-task strategies. The variant **w/o FT** outperforms the variant **w/o CR** in most cases, but it does not perform as well as PCcam. This indicates the effectiveness of exploring the cascading effects behind users' multi-behaviors in a fine-tuning way, which contributes to better decoupling the users' preferences and the effect of display position. (RQ3) 3) The variant **w/o MG** achieves bad performances among these variants, which indicates the single-behavior method severely suffers from the data sparsity problem. In addition, the performance of the variant **w/o CD** compared to the proposed method also confirms the importance of exploiting auxiliary click behaviors which are richer than purchase behaviors among display positions for deconfounding (RQ4).

C. Hyper-parameter Analysis

In this section, we investigate how the click and purchase trade-off coefficient μ influences purchase behavior prediction and how the implementation of $g'(x)$ in Equation (4) influences the proposed method. Figure 5 shows the performances of PCcam-DP (i.e., direct position $g'(d) = d$) and PCcam-CP (i.e., click-to-purchase ratio $g'(d) = CVR(d)$) vary with coefficient μ. First, both methods achieve the best performance when $\mu = 1$, which means the equal importance of click willingness prediction and purchase preference modeling. Second, PCcam-CP outperforms PCcam-DP consistently, so we suggest adopting the click-to-purchase ratio as the position-aware model in the proposed method.

VI. Conclusion

In this paper, we propose a position-aware causal model for multi-behavior based recommendation. By analyzing the causal graph of the recommendation process, we find that the position affects users' behavioral willingness and items' observation. To this end, we propose a method to alleviate the bad effects of display position, and further explore and make use of the cascading effects behind users' multi-behaviors on user preference learning. Experimental results on real-world data sets show our model consistently outperforms state-of-the-art methods, and ablation studies prove the effectiveness of the proposed method and verify our motivations.

Acknowledgments

This research is supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (Award I2301E0026), administered by A*STAR. The work is also supported by the Ministry of Education, Singapore under its MOE Academic Research Fund Tier 2 (MOE-T2EP20123-0005).

References

[1] P. Kazienko and E. Cambria, "Towards responsible recommender systems," *IEEE Intelligent Systems*, vol. 39, no. 3, pp. 5–12, 2024.
[2] Y. Li, S. Wang, Q. Pan, H. Peng, T. Yang, and E. Cambria, "Learning binary codes with neural collaborative filtering for efficient recommendation systems," *Knowledge-Based Systems*, vol. 172, pp. 64–75, 2019.
[3] C. Angulo, Z. Falomir, D. Anguita, N. Agell, and E. Cambria, "Bridging cognitive models and recommender systems," *Cognitive Computation*, vol. 12, no. 2, pp. 426–427, 2020.
[4] I. Ajzen, "The theory of planned behavior," *Organizational behavior and human decision processes*, vol. 50, no. 2, pp. 179–211, 1991.
[5] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and debias in recommender system: A survey and future directions," *arXiv preprint arXiv:2010.03240*, 2020.
[6] X. Wang, M. Bendersky, D. Metzler, and M. Najork, "Learning to rank with selection bias in personal search," in *SIGIR*, 2016, pp. 115–124.
[7] T. Joachims, A. Swaminathan, and T. Schnabel, "Unbiased learning-to-rank with biased feedback," in *WSDM*, 2017, pp. 781–789.
[8] X. Wang, R. Zhang, Y. Sun, and J. Qi, "Doubly robust joint learning for recommendation on data missing not at random," in *ICML*. PMLR, 2019, pp. 6638–6647.
[9] H. Guo, J. Yu, Q. Liu, R. Tang, and Y. Zhang, "Pal: a position-bias aware learning framework for ctr prediction in live recommender systems," in *RecSys*, 2019, pp. 452–456.
[10] Z. Zhao, L. Hong, L. Wei, J. Chen, A. Nath, S. Andrews, A. Kumthekar, M. Sathiamoorthy, X. Yi, and E. Chi, "Recommending what video to watch next: a multitask ranking system," in *RecSys*, 2019, pp. 43–51.
[11] Q. Ai, K. Bi, C. Luo, J. Guo, and W. B. Croft, "Unbiased learning to rank with unbiased propensity estimation," in *SIGIR*, 2018, pp. 385–394.
[12] R. Guo, X. Zhao, A. Henderson, L. Hong, and H. Liu, "Debiasing grid-based product search in e-commerce," in *SIGKDD*, 2020, pp. 2852–2860.
[13] A. Agarwal, K. Takatsu, I. Zaitsev, and T. Joachims, "A general framework for counterfactual learning-to-rank," in *SIGIR*, 2019, pp. 5–14.
[14] A. Vardasbi, M. de Rijke, and I. Markov, "Cascade model-based propensity estimation for counterfactual learning to rank," in *SIGIR*, 2020, pp. 2089–2092.
[15] H. Oosterhuis, "Doubly robust estimation for correcting position bias in click feedback for unbiased learning to rank," *TOIS*, vol. 41, no. 3, pp. 1–33, 2023.
[16] Y. Zhao, Z. Liu, T. Cai, H. Zhang, C. Zhuang, and J. Gu, "Mitigate position bias with coupled ranking bias on ctr prediction," *arXiv preprint arXiv:2405.18971*, 2024.
[17] J. Lin, W. Liu, X. Dai, W. Zhang, S. Li, R. Tang, X. He, J. Hao, and Y. Yu, "A graph-enhanced click model for web search," in *SIGIR*, 2021, pp. 1259–1268.
[18] P. Wu, H. Li, Y. Deng, W. Hu, Q. Dai, Z. Dong, J. Sun, R. Zhang, and X.-H. Zhou, "On the opportunity of causal learning in recommendation systems: Foundation, estimation, prediction and challenges," *arXiv preprint arXiv:2201.06716*, 2022.
[19] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, and Y. Zhang, "Causal intervention for leveraging popularity bias in recommendation," in *SIGIR*, 2021, pp. 11–20.
[20] W. Wang, F. Feng, X. He, X. Wang, and T.-S. Chua, "Deconfounded recommendation for alleviating bias amplification," in *SIGKDD*, 2021, pp. 1717–1725.
[21] Y. Chen, W. Ye, G. Xv, C. Lin, and X. Zhu, "Tccm: Time and content-aware causal model for unbiased news recommendation," in *CIKM*, 2023, pp. 3778–3782.
[22] D. Rajagopal, E. Cambria, D. Olsher, and K. Kwok, "A graph-based approach to commonsense concept extraction and semantic similarity detection," in *WWW*, 2013, pp. 565–570.
[23] Q. Li, X. Wang, Z. Wang, and G. Xu, "Be causal: De-biasing social network confounding in recommendation," *TKDD*, vol. 17, no. 1, pp. 1–23, 2023.
[24] L. Oneto, F. Bisio, E. Cambria, and D. Anguita, "Statistical learning theory and ELM for big social data analysis," *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 45–55, 2016.
[25] X. He, Y. Zhang, F. Feng, C. Song, L. Yi, G. Ling, and Y. Zhang, "Addressing confounding feature issue for causal recommendation," *TOIS*, vol. 41, no. 3, pp. 1–23, 2023.
[26] J. Pearl, *Causality*. Cambridge university press, 2009.
[27] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[28] L. Xia, Y. Xu, C. Huang, P. Dai, and L. Bo, "Graph meta network for multi-behavior recommendation," in *SIGIR*, 2021, pp. 757–766.
[29] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.
[30] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," in *IJCAI*, 2017, pp. 1725–1731.
[31] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020, pp. 639–648.
[32] B. Loni, R. Pagano, M. Larson, and A. Hanjalic, "Bayesian personalized ranking with multi-channel user feedback," in *RecSys*, 2016, pp. 361–364.
[33] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi, "Improving user topic interest profiles by behavior factorization," in *WWW*, 2015, pp. 1406–1416.
[34] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T.-S. Chua, and D. Jin, "Neural multi-task recommendation from multi-behavior data," in *ICDE*. IEEE, 2019, pp. 1554–1557.
[35] L. Xia, C. Huang, Y. Xu, P. Dai, M. Lu, and L. Bo, "Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling," in *ICDE*. IEEE, 2021, pp. 1931–1936.
[36] C. Chen, W. Ma, M. Zhang, Z. Wang, X. He, C. Wang, Y. Liu, and S. Ma, "Graph heterogeneous multi-relational recommendation," in *AAAI*, vol. 35, no. 5, 2021, pp. 3958–3966.
[37] H. Zhuang, Z. Qin, X. Wang, M. Bendersky, X. Qian, P. Hu, and D. C. Chen, "Cross-positional attention for debiasing clicks," in *WWW*, 2021, pp. 788–797.
[38] D. Loveland, M. Ju, T. Zhao, N. Shah, and D. Koutra, "On the role of weight decay in collaborative filtering: A popularity perspective," in *SIGKDD*, 2025, pp. 1975–1986.
[39] Y. Liu, Q. Cao, H. Shen, Y. Wu, S. Tao, and X. Cheng, "Popularity debiasing from exposure to interaction in collaborative filtering," in *SIGIR*, 2023, pp. 1801–1805.
[40] W. Chen and Y. Chen, "Pp4rnr: Popularity-and position-aware contrastive learning for retrieval-driven news recommendation," in *CIKM*, 2024, pp. 3684–3688.
[41] S. Lin, C. Gao, J. Chen, S. Zhou, B. Hu, Y. Feng, C. Chen, and C. Wang, "How do recommendation models amplify popularity bias? an analysis from the spectral perspective," in *WSDM*, 2025, pp. 659–668.