

# Click-Conversion Multi-Task Model with Position Bias Mitigation for Sponsored Search in eCommerce

Yibo Wang

ywang633@uic.edu

University of Illinois Chicago  
Chicago, IL, USA

Yanbing Xue

yanbing.xue@walmart.com

Walmart eCommerce  
Sunnyvale, CA, USA

Bo Liu

bo.liu1@walmart.com

Walmart eCommerce  
Sunnyvale, CA, USA

Musen Wen

musen.wen@walmart.com

Walmart eCommerce  
Sunnyvale, CA, USA

Wenting Zhao

wzhao41@uic.edu

University of Illinois Chicago  
Chicago, IL, USA

Stephen Guo

sguo@indeed.com

Indeed  
Sunnyvale, CA, USA

Philip S. Yu

psyu@uic.edu

University of Illinois Chicago  
Chicago, IL, USA

## ABSTRACT

Position bias, the phenomenon whereby users tend to focus on higher-ranked items of the search result list regardless of the actual relevance to queries, is prevailing in many ranking systems. Position bias in training data biases the ranking model, leading to increasingly unfair item rankings, click-through-rate (CTR), and conversion rate (CVR) predictions. To jointly mitigate position bias in both item CTR and CVR prediction, we propose two position-bias-free CTR and CVR prediction models: Position-Aware Click-Conversion (PACC) and PACC via Position Embedding (PACC-PE). PACC is built upon probability decomposition and models position information as a probability. PACC-PE utilizes neural networks to model product-specific position information as embedding. Experiments on the E-commerce sponsored product search dataset show that our proposed models have better ranking effectiveness and can greatly alleviate position bias in both CTR and CVR prediction.

## CCS CONCEPTS

• Applied computing → Electronic commerce; • Computing methodologies → Machine learning.

## KEYWORDS

Multi-task Learning; Sponsored Product Search; Position Bias

## ACM Reference Format:

Yibo Wang, Yanbing Xue, Bo Liu, Musen Wen, Wenting Zhao, Stephen Guo, and Philip S. Yu. 2023. Click-Conversion Multi-Task Model with Position Bias Mitigation for Sponsored Search in eCommerce. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*, July 23–27, 2023, Taipei, Taiwan. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3539618.3591963>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*SIGIR '23, July 23–27, 2023, Taipei, Taiwan*

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9408-6/23/07...\$15.00

<https://doi.org/10.1145/3539618.3591963>

## 1 INTRODUCTION

The phenomenon of position bias is commonly observed in many ranking and information retrieval systems, including digital advertising and recommender systems. Position bias is the tendency of users to pay greater attention to higher-ranked items, regardless of their actual relevance to the query. For example, the eye-tracking study [13] and [14] show that in web search, the highest-ranked items receive the most attention; the study of digital library recommendation system [5] discovers that items shown at the top positions are more often clicked regardless of their actual relevance.

A common practice from machine-learned ranking (MLR) models is to use implicit user feedback, such as click/no-click as training data labels. However, due to inherited position bias, these models tend to exhibit lower predicted click-through rates (CTR) for lower-ranked items. This bias in prediction can then affect subsequent training data collection, leading to a persistent and cyclical effect.

The accumulative position bias of the training data will skew the MLR model, leading to increasingly unfair item rank prediction [6]. To address this position bias problem, various approaches have been proposed, including factorization models [4], inverse propensity scores [9, 15, 19], deep neural network-based CTR prediction models [8, 17, 21], and more. Pioneering approaches such as [4] proposed a factorization model that decouples CTR into position-normalized CTR and position bias, which are then estimated by an Expectation-Maximization (EM) algorithm framework. To remove the position bias for learning to rank models, a propensity-weighted empirical risk minimization framework is proposed in [15]. In [10], an unbiased LambdaMART model is proposed, which jointly estimates the biases at click positions and unclick positions and learns an unbiased ranker. Several literatures dedicate to the propensity estimation, such as [1, 2, 18]. Recently, increasing research studies correcting position bias in the deep-ranking models. In [12], the authors propose to combine recurrent neural network and survival analysis techniques to model unbiased user behaviors. [3] designs a neural-based Context-Aware Click Model with an examination predictor able to automatically learn the position bias during training.

The existing methods primarily consider single-task objectives like CTR prediction. However, in E-commerce, typical ranking models can have multiple objectives, such as maximizing both CTR and CVR. Deep Multifaceted Transformers (DMT) [7] learns both CTR and CVR predictors by modeling multiple user behaviors simultaneously with bias mitigation and treats CTR and CVR prediction

as two parallel tasks. However, item impression, click, and conversion processes have sequential dependencies and are all affected by position bias. Such dependencies are not leveraged in DMT.

In this paper, two models, Position Aware Click-Conversion (PACC) and PACC with Position Embedding (PACC-PE) are proposed to model the sequential relationship and jointly mitigate position bias in both CTR prediction and CVR prediction. PACC is based on the following two assumptions: (1). whether an item will be seen is only related to its position; (2). after an item is seen, whether it will be clicked/purchased is independent of its position. PACC is built upon the probability decomposition presented in §2.2. PACC-PE is a variant of PACC, which adopts a neural network to model the product-specific position information as embedding. Compared to PACC, learning the product-specific position embedding with a neural network enables PACC-PE to achieve richer information and superior performance than PACC.

Specifically, our work has the following contributions:

- We propose to jointly learn position-bias-free CTR and CVR prediction models in a multi-task learning framework. By mitigating position bias, the proposed models achieve comparable performance as state-of-the-art models on CTR prediction and significant performance improvement on CVR prediction regarding weighted Mean-Reciprocal-Rank (MRR) [19], MRR, position-wise AUC (PAUC) [11], and AUC.
- We conduct experiments on real-world E-commerce sponsored product searches. Our proposed models achieve better ranking effectiveness and greatly mitigate position bias.

## 2 METHODOLOGY

In this section, we first introduce our problem formulation and then present assumptions and theoretical basis. Finally, detailed explanations of our proposed PACC and PACC-PE models are provided.

### 2.1 Notation and Problem Formulation

We assume the training set to be  $T = \{(f_i, p_i) \rightarrow (y_i^{ctr}, y_i^{cvr})\}_{i=1}^N$ , where  $f_i$  is features other than position of sample  $i$ ,  $p_i$  is the position of sample  $i$ ,  $y_i^{ctr} \in \{0, 1\}$  is the click label of sample  $i$ ,  $y_i^{cvr} \in \{0, 1\}$  is the conversion label of sample  $i$ , and  $N$  is the number of samples in  $T$ . Therefore, our models estimate the two probabilities:  $P(y_i^{ctr} = 1|f_i, p_i)$ , the probability of an item to be clicked according to features and position, and  $P(y_i^{cvr} = 1|f_i, p_i)$ , the probability of an item to be purchased according to features and position. Besides, we use  $s_i \in \{0, 1\}$  to represent whether an item is seen by the user or not.

### 2.2 Assumptions and Theoretical Basis

PACC is based on two assumptions: 1) whether an item will be seen by the user is only related to item position, which is  $P(s_i|f_i, p_i) = P(s_i|p_i)$ ; 2) if an item is already seen, whether it will be clicked/purchased is independent of item position, which is  $P(y_i^{ctr}|f_i, p_i, s_i = 1) = P(y_i^{ctr}|f_i, s_i = 1)$  and  $P(y_i^{cvr}|f_i, p_i, s_i = 1) = P(y_i^{cvr}|f_i, s_i = 1)$ .

In addition to these two assumptions, two facts underlie PACC: 1) an item has to be seen before it can be clicked, which is  $P(y_i^{ctr} = 1|f_i, p_i) = P(y_i^{ctr} = 1 \cap s_i = 1|f_i, p_i)$ ; 2) an item has to be clicked and seen before it can be purchased, which is  $P(y_i^{cvr} = 1|f_i, p_i) = P(y_i^{cvr} = 1 \cap y_i^{ctr} = 1|f_i, p_i)$  and  $P(y_i^{cvr} = 1|f_i, p_i, y_i^{ctr} = 1) = P(y_i^{cvr} = 1 \cap s_i = 1|f_i, p_i, y_i^{ctr} = 1)$ .

The probability of an item being clicked is represented as:

$$\begin{aligned} P(y_i^{ctr} = 1|f_i, p_i) &= P(y_i^{ctr} = 1|f_i, p_i, s_i = 1) \cdot P(s_i = 1|f_i, p_i) \\ &= P(y_i^{ctr} = 1|f_i, s_i = 1) \cdot P(s_i = 1|p_i). \end{aligned} \quad (1)$$

The probability of an item to be purchased is represented as:

$$\begin{aligned} P(y_i^{cvr} = 1|f_i, p_i) &= P(y_i^{cvr} = 1 \cap y_i^{ctr} = 1|f_i, p_i) \\ &= P(y_i^{cvr} = 1|f_i, y_i^{ctr}, s_i) \cdot P(y_i^{ctr} = 1|f_i, p_i). \end{aligned} \quad (2)$$

## 2.3 Proposed Framework

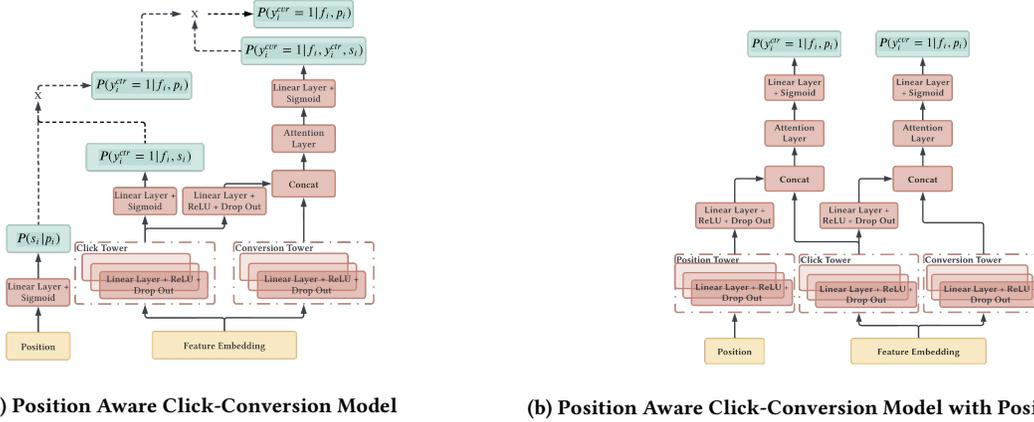
**2.3.1 Overview.** Figure 1 illustrates the architecture of our proposed models. In both models, CTR prediction and CVR prediction share the same feature embedding but have different model architectures and parameters. A neural network is utilized to adaptively learn what and how much information to transfer from CTR prediction to CVR prediction. Position information is modeled separately and combined in innovative ways with the click-conversion multi-task model. With this novel framework, our proposed models can jointly alleviate the position bias of both CTR and CVR prediction.

**2.3.2 Position Aware Click-Conversion Model.** The model architecture of PACC is shown in Fig. 1a. Given the input feature  $f_i$  embedded as  $v_i$ , for task  $k \in \{ctr, cvr\}$ , the output of  $k$  Tower is defined as  $T_k = g_t^k(v_i)$ , where  $g_t^k(\cdot)$  is three linear layers each followed by a ReLU activation function and a drop-out layer. Then  $T_{ctr}$  is fed into a linear layer with a sigmoid function to calculate  $P(y_i^{ctr}|f_i, s_i = 1)$ , which is the probability of an item to be clicked after it is seen.  $T_{ctr}$  is also fed into a linear layer followed by a ReLU activation function and a drop-out layer, whose output is  $INFO_{ctr}$ . Then  $T_{cvr}$  and  $INFO_{ctr}$  are concatenated and fed into an attention layer as  $A_{cvr} = g_a^{cvr}([T_{cvr}; INFO_{ctr}])$ , where  $g_a^{cvr}$  is the function of the attention layer and  $A_{cvr}$  is the output. Then  $A_{cvr}$  is fed into a linear layer with a sigmoid function to calculate  $P(y_i^{cvr}|f_i, y_i^{ctr} = 1, s_i = 1)$ , which is the probability of an item to be purchased if it is already seen and clicked.

$P(s_i|p_i)$ , the probability of an item to be seen given the position, is modeled using a linear layer and a sigmoid function. Then  $P(s_i|p_i)$  is multiplied by  $P(y_i^{ctr}|f_i, s_i = 1)$  for  $P(y_i^{ctr} = 1|f_i, p_i)$  and  $P(y_i^{ctr} = 1|f_i, p_i)$  is multiplied by  $P(y_i^{cvr}|f_i, y_i^{ctr} = 1, s_i = 1)$  for  $P(y_i^{cvr}|f_i, p_i)$  according to Eq.(1) and Eq.(2).

**2.3.3 Position Aware Click-Conversion Model with Position Embedding.** The architecture of PACC-PE is shown in Fig. 1b. Given  $v_i$  as the shared feature embedding of sample  $i$ , for task  $k \in \{pos, ctr, cvr\}$ , the output of  $k$  Tower is defined as  $T_k = g_t^k(v_i)$ , where  $g_t^k(\cdot)$  encodes  $v_i$  through three linear layers each followed by a ReLU activation function and a drop-out layer.  $T_{pos}$  is then fed into a linear layer followed by a ReLU activation function and a drop-out layer, whose output is  $INFO_{pos}$ . Then  $T_{ctr}$  and  $INFO_{pos}$  are concatenated and fed into an attention layer as  $A_{ctr} = g_a^{ctr}([T_{ctr}; INFO_{pos}])$ , where  $g_a^{ctr}$  is the function of the attention layer and  $A_{ctr}$  is the output. Then  $A_{ctr}$  is fed into a linear layer with a sigmoid function to calculate  $P(y_i^{ctr}|f_i, p_i)$ , which is the probability of an item to be clicked.  $P(y_i^{cvr}|f_i, p_i)$  is obtained similar as  $P(y_i^{ctr}|f_i, p_i)$ .

The difference between PACC and PACC-PE is that PACC models position information into a scalar while PACC-PE models product-specific position information into an embedding. Thus, PACC-PE focuses on representing product-related position information, which



(a) Position Aware Click-Conversion Model

(b) Position Aware Click-Conversion Model with Position Embedding

**Figure 1: An overview of PACC and PACC-PE.** The left figure shows the overall structure of the PACC model which utilizes the assumptions and probability decomposition in §2.2 to estimate the probability of an item being clicked and the probability of an item being purchased based on position and feature embedding. The right figure illustrates the PACC-PE model that integrates neural networks to model product-specific position information as embedding.

is richer and more useful. Besides, PACC-PE has high fault tolerance compared with PACC, where subsequent tasks will be greatly affected if the probability of a former task is predicted wrongly.

**2.3.4 Loss Function.** The loss function of our proposed models is

$$\mathcal{L}(\theta) = \mathcal{L}_{CTR}(\theta) + \mathcal{L}_{CVR}(\theta) + \mathcal{L}_{res}(\theta), \quad (3)$$

where  $\mathcal{L}_{CTR}(\theta)$  is the binary cross entropy loss of the CTR prediction task,  $\mathcal{L}_{CVR}(\theta)$  is the binary cross entropy loss of the CVR prediction task and  $\mathcal{L}_{res}(\theta)$  is a restriction loss. The restriction loss is based on the fact that an item has to be clicked before it can be purchased, which is defined as

$$\mathcal{L}_{res}(\theta) = \sum_{i=0}^N \max(P(y_i^{ctr} | f_i, p_i) - P(y_i^{cvr} | f_i, p_i), 0). \quad (4)$$

## 3 EXPERIMENTS

### 3.1 Dataset and Data Preprocessing

The training and testing query-item pair data are collected using Walmart<sup>1</sup> sponsored ads logs. The dataset is a real-world dataset with position bias, containing 4.2M training samples, 1.1M validation samples, and 7.5M testing samples.

The extracted features are categorized into three types: categorical features, numeric features, and text features. The categorical features are transformed into one-hot vectors; the numeric features are normalized; and the text features are embedded using BERT [16] to calculate cosine similarity scores and element-wise product between query and ad item. The element-wise product obtained using  $BERT_{base}$  has a dimension of 768, whereas the dimensions of other features are much smaller. To prevent the element-wise product from overwhelming other features, we use PCA to reduce its dimensionality to 5. In addition, the position feature is transformed into a one-hot format.

### 3.2 Evaluation Metrics

Since the standard Mean Reciprocal Rank (MRR) assumes that the clicked/purchased items are relevant while ignoring the position

<sup>1</sup><https://www.walmart.com>

**Table 1: Ranking Performance Comparison Results with Based Models**

Models	CTR				CVR			
	Weighted MRR	MRR	PAUC	AUC	Weighted MRR	MRR	PAUC	AUC
DMT [7]	39.65	<b>39.68</b>	88.43	87.75	42.38	42.31	56.60	83.82
PAL [8]	<b>39.67</b>	39.44	88.67	88.02	42.52	43.37	55.22	80.82
ATM [20]	39.40	39.25	88.89	89.04	40.64	43.18	60.39	89.56
PACC	39.56	39.35	88.96	88.28	43.73	43.53	<b>60.79</b>	<b>90.05</b>
PACC-PE	39.43	39.43	<b>92.16</b>	<b>91.68</b>	<b>47.44</b>	<b>47.44</b>	60.72	89.28

bias, we use weighted MRR [19] to evaluate ranking effectiveness, which is formulated as  $MRR^m = \frac{1}{\sum_{i=1, \tilde{y}_i^m=1}^N w_i^m} \sum_{i=1, \tilde{y}_i^m=1}^N w_i^m \frac{1}{p_i}$ ,

where  $m \in \{ctr, cvr\}$ ,  $N$  is the number of samples,  $w_i^m = \frac{1}{P(s_i | f_i, p_i)}$  is a weight of  $i$ -th sample,  $p_i$  is the position of the  $i$ -th sample and  $\tilde{y}_i^m$  is the predicted label.  $w_i^m$  is defined through the reciprocal of  $P(s_i | p_i)$  which reflects the impact of position.

For PACC,  $w_i^m$  can be calculated in a straightforward way since  $P(s_i | p_i) = P(s_i | f_i, p_i)$  is obtained after training.

To compute  $P(s_i | f_i, p_i)$  for PACC-PE as mentioned in [15], we first swap  $p_i$  with  $p_i = r$  for item  $i$ . The probability of this item being clicked at position  $p_i = r$  is

$$P(y_i^{ctr} = 1 | f_i, p_i = r) = P(y_i^{ctr} = 1 | f_i, s_i = 1) \cdot P(s_i | f_i, p_i = r). \quad (5)$$

The probability of this item being clicked at position  $p_i$  is

$$P(y_i^{ctr} = 1 | f_i, p_i) = P(y_i^{ctr} = 1 | f_i, s_i = 1) \cdot P(s_i | f_i, p_i). \quad (6)$$

With Eq. (5) and Eq. (6), the ratio  $\frac{P(s_i | f_i, p_i)}{P(s_i | f_i, p_i = r)} = \frac{P(y_i^{ctr} = 1 | f_i, p_i)}{P(y_i^{ctr} = 1 | f_i, p_i = r)}$  is obtained. Thus, the weight  $w_i^{ctr}$  for PACC-PE is

$$w_i^{ctr} = \frac{P(y_i^{ctr} = 1 | f_i, p_i = r)}{P(y_i^{ctr} = 1 | f_i, p_i)} \cdot \frac{1}{P(s_i | f_i, p_i = r)}, \quad (7)$$

which is proportional to  $\frac{P(y_i^{ctr} = 1 | f_i, p_i = r)}{P(y_i^{ctr} = 1 | f_i, p_i)}$  since  $P(s_i | p_i = r)$  is a constant for a fixed position  $r$ .  $w_i^{cvr}$  can be calculated similarly.

In addition to the weighted MRR, we also apply the widely-used evaluation metrics: standard MRR, AUC, and position-wise AUC (PAUC) [11] to evaluate model performance.

### 3.3 Experimental Results and Analysis

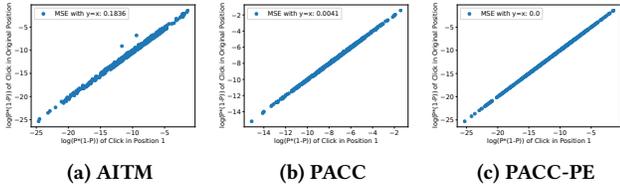
**3.3.1 Ranking Effectiveness.** We compare the weighted MRR, MRR, PAUC, and AUC of our proposed models and the baselines PAL [8], AITM [20], and DMT [7] models. PAL is a single-task position bias mitigation model; AITM is a multi-task model; DMT is a multi-task position bias mitigation model. The evaluation results are reported in Table 1.

For CTR prediction, all models perform similarly in terms of weighted MRR and MRR. PACC-PE outperforms the best baseline by 3.27%/2.64% in terms of PAUC/AUC with p-value = 0.0072 and confidence level > 99%. For CVR prediction, PACC-PE significantly outperforms all other baseline models regarding weighted MRR and MRR, increasing weighted MRR/MRR by 4.92%/4.07% with p-value = 0.0088 and confidence level > 99%. PACC also outperforms other baseline models.

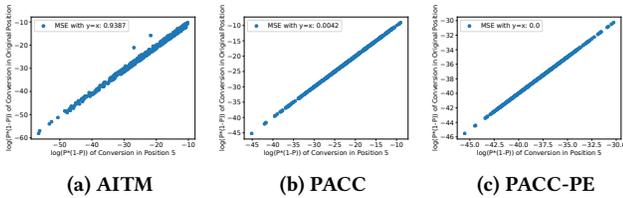
The substantial improvement in CVR prediction of our proposed models indicates that jointly mitigating position bias for both CTR and CVR and considering the sequential dependencies in order between click and purchase is effective for performance improvement.

**3.3.2 Position Bias.** To generally evaluate the ability of our proposed model in mitigating position bias, we randomly select 500 query-item pairs and visualize their probability of being clicked and purchased before and after swapping positions. To better interpret the predicted probabilities, the log odds function is used to project probabilities in log odds space. From Fig 2 and Fig 3, the sample points of PACC and PACC-PE fit better on  $y = x$  compared to AITM, indicating that swapping item positions with position 1 has less influence on the predicted probabilities of being clicked and purchased by PACC and PACC-PE.

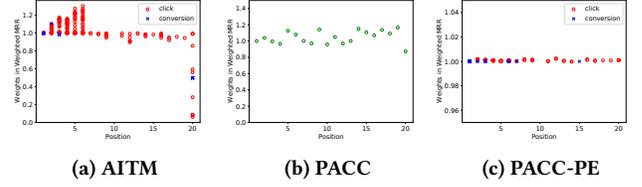
**3.3.3 Position Bias on Different Positions.** To evaluate the ability of our proposed models in mitigating position bias on different positions, we investigate the differences in model prediction changes for items at different positions due to swapping positions with position



**Figure 2: The log odds of probabilities of being clicked of 500 randomly selected query-item pairs. The y-axis is the probability of being clicked at the original position; the x-axis is the probability of being clicked at position 1.**



**Figure 3: The log odds of probabilities of being purchased of 500 randomly selected query-item pairs. The y-axis is the probability of being purchased at the original position; the x-axis is the probability of being purchased at position 1.**



**Figure 4: Visualization of the impact of position swapping demonstrated by 300 randomly selected examples. Red  $\circ$  are clicked query-item pairs. Blue  $\times$  are purchased query-item pairs. Green  $\circ$  are the impact of swapping positions at each position in PACC.**

1. The smaller the position bias, the smaller the prediction changes due to swapping positions. The impact of swapping item positions is visualized in Fig 4. In Fig 4a and Fig 4c,  $\frac{P(y_i^{ctr}=1|f_i, p_i=1)}{P(y_i^{ctr}=1|f_i, p_i)}$  and  $\frac{P(y_i^{cvr}=1|f_i, p_i=1)}{P(y_i^{cvr}=1|f_i, p_i)}$  are used to measure the impact of swapping item positions on CTR prediction and CVR prediction, respectively. As in Fig 4c, swapping item positions almost has no effect on both CTR and CVR prediction for PACC-PE on all positions, indicating that PACC-PE has the ability to mitigate position bias on all positions. In Fig 4a swapping item positions has a large impact on the top few items and the last item in the ranking list for AITM. For items on the top, swapping them to position 1 improves the probability of being clicked and purchased, while for the last items swapping them to position 1 decreases the probability of being clicked and purchased. This phenomenon is counter-intuitive because intuitively swapping items to position 1 should increase the probability of being clicked and purchased. One possible explanation is that some items are safe choices, but not preferred. For these items, if they are ranked high, users will not click on them. But if they are ranked low, users may click them given no better items at the end of their browsing. In Fig 4b,  $\frac{P(s_i|p_i=1)}{P(s_i|p_i)}$  is used to measure the impact of swapping item positions. The impact of swapping item positions for PACC is low compared to AITM, demonstrating the ability of PACC to alleviate position bias. For items at most positions, swapping positions to position 1 increases the probability of being clicked and purchased a little bit.

One difference between PACC-PE and PACC from Fig 4c and Fig 4b is that for PACC, position bias of different items at the same position is the same, while for PACC-PE position bias of different items at the same position is different. This difference makes PACC-PE more flexible to different items and conveys richer information.

## 4 CONCLUSION

To jointly mitigate position bias that exists in both item CTR and CVR prediction, we propose two position-bias-free CTR and CVR prediction models: Position Aware Click-Conversion and PACC with Position Embedding. In PACC, the position is modeled as a probability while in PACC-PE position is modeled into embedding. Our experiments and analyses illustrate that our proposed models achieve better ranking effectiveness than the state-of-the-art models and effectively mitigate position bias in all positions. Besides, PACC-PE outperforms PACC in ranking effectiveness and position debias due to the rich information by modeling product-specific position information as embedding.

## REFERENCES

- [1] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating position bias without intrusive interventions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 474–482.
- [2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 385–394.
- [3] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. A context-aware click model for web search. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 88–96.
- [4] Ye Chen and Tak W Yan. 2012. Position-normalized click prediction in search advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 795–803.
- [5] 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).
- [6] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*. 87–94.
- [7] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. 2020. Deep Multifaceted Transformers for Multi-Objective Ranking in Large-Scale E-Commerce Recommender Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (Virtual Event, Ireland) (CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 2493–2500. <https://doi.org/10.1145/3340531.3412697>
- [8] Huifeng Guo, Jinkai Yu, Qing Liu, Ruiming Tang, and Yuzhou Zhang. 2019. PAL: a position-bias aware learning framework for CTR prediction in live recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 452–456.
- [9] Ruocheng Guo, Xiaoting Zhao, Adam Henderson, Liangjie Hong, and Huan Liu. 2020. Debiasing grid-based product search in e-commerce. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2852–2860.
- [10] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. 2019. Unbiased lambdamart: an unbiased pairwise learning-to-rank algorithm. In *The World Wide Web Conference*. 2830–2836.
- [11] Jianqiang Huang, Ke Hu, Qingtao Tang, Mingjian Chen, Yi Qi, Jia Cheng, and Jun Lei. 2021. Deep position-wise interaction network for CTR prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1885–1889.
- [12] Jiarui Jin, Yuchen Fang, Weinan Zhang, Kan Ren, Guorui Zhou, Jian Xu, Yong Yu, Jun Wang, Xiaoqiang Zhu, and Kun Gai. 2020. A deep recurrent survival model for unbiased ranking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 29–38.
- [13] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately Interpreting Clickthrough Data as Implicit Feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Salvador, Brazil) (SIGIR '05)*. Association for Computing Machinery, New York, NY, USA, 154–161. <https://doi.org/10.1145/1076034.1076063>
- [14] 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 Trans. Inf. Syst.* 25, 2 (apr 2007), 7–es. <https://doi.org/10.1145/1229179.1229181>
- [15] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (Cambridge, United Kingdom) (WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 781–789. <https://doi.org/10.1145/3018661.3018699>
- [16] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.
- [17] Xiaoliang Ling, Weiwei Deng, Chen Gu, Hucheng Zhou, Cui Li, and Feng Sun. 2017. Model ensemble for click prediction in bing search ads. In *Proceedings of the 26th international conference on world wide web companion*. 689–698.
- [18] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2020. Cascade model-based propensity estimation for counterfactual learning to rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2089–2092.
- [19] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM)*. 610–618.
- [20] Dongbo Xi, Zhen Zi Chen, Peng Yan, Yinger Zhang, Yongchun Zhu, Fuzhen Zhuang, and Yu Chen. 2021. Modeling the Sequential Dependence among Audience Multi-step Conversions with Multi-task Learning in Targeted Display Advertising. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2021).
- [21] Honglei Zhuang, Zhen Qin, Xuanhui Wang, Michael Bendersky, Xinyu Qian, Po Hu, and Dan Chary Chen. 2021. Cross-positional attention for debiasing clicks. In *Proceedings of the Web Conference 2021*. 788–797.