Adapting Interactional Observation Embedding for Counterfactual Learning to Rank

Mouxiang Chen[1,3*], Chenghao Liu[2*], Jianling Sun[1,3], Steven C.H. Hoi[2]
[1]Zhejiang University, [2]Salesforce Research Asia,
[3]Alibaba-Zhejiang University Joint Institute of Frontier Technologies
{chenmx,sunjl}@zju.edu.cn,{chenghao.liu,shoi}@salesforce.com

ABSTRACT

Counterfactual Learning to Rank (CLTR) becomes an attractive research topic due to its capability of training ranker with click logs. However, CLTR inherently suffers from a large amount of bias caused by confounders, variables that affect both the observation (examination) behavior and click behavior. Recent efforts to correct bias mostly focus on position bias, which assumes that each observation in a ranking list is isolated and only depends on the position. Though effective, users often engage with documents in an interactive manner. Ignoring the interactions between observations/clicks would incur a large interactional observation bias no matter how much data is collected.

In this work, we leverage the embedding method to develop an Interactional Observation-Based Model (IOBM) to estimate the observation probability. We argue that while there exist complex observed and unobserved confounders for observation/click interactions, it is sufficient to use the embedding as a proxy confounder to uncover the relevant information for the prediction of the observation propensity. Moreover, the embedding could offer an alternative to the fully specified generative model for observation and decouples the complex interaction structure of observations/clicks. In our IOBM, we first learn the individual observation embedding to capture position and click information. Then, we learn the interactional observation embedding to uncover their local interaction structure. To filter out irrelevant information and reduce contextual bias, we utilize query context information and propose the intra-observation attention and the inter-observation attention, respectively. We conduct extensive experiments on two LTR benchmark datasets, demonstrating that the proposed IOBM consistently achieves better performance over the baseline models in various click situations and verifying its effectiveness of eliminating interactional observation bias.

CCS CONCEPTS

• **Information systems → Learning to rank**.

KEYWORDS

counterfactual learning to rank, causal inference, neural network

ACM Reference Format:
Mouxiang Chen[1,3*], Chenghao Liu[2*], Jianling Sun[1,3], Steven C.H. Hoi[2]. 2021. Adapting Interactional Observation Embedding for Counterfactual Learning to Rank. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462901

1 INTRODUCTION

Learning to Rank (LTR) with implicit feedback from user behavior (e.g. click, dwell time) has attracted increased research interest with the introduction of counterfactual inference approaches [1, 29]. Using logged feedback for LTR is attractive not only because they are cheap and relatively easy to acquire at scale [27] but also because relevance annotations from experts are impractical, unethical, or impossible in some domains [23]. On the other hand, learning from implicit feedback inherently contains a large amount of bias from user behavior and the ranker used during logging. Simply ignoring it and treating the click as a relevance signal would result in suboptimal performance [28, 45].

Recently, there has been remarkable progress in using Counterfactual Learning to Rank (CLTR) to remove bias. The key idea is to model the probability of a user observing an item in a displayed ranking. By reweighting the clicked documents based on the reciprocal of their observation propensities, the *inverse propensity score* (IPS) method could provide a principled approach to an unbiased estimate of the ranking objective. Most of the existing works have focused on position bias [27], in which users tend to observe documents at the top of rankings and, consequently documents displayed at the top position have a high chance to be clicked.

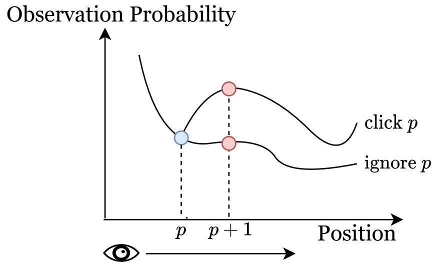


Figure 1: An illustrative example of interactional observation bias.

Algorithms that correct for position bias typically assume that observation is isolated in a ranking list and only depends on the position, which is the confounder that affects both observation

[*] denotes equal contribution.
Chenghao Liu and Jianling Sun are corresponding authors.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462901

and click events. While simplifying assumptions may widen its applicability, it ignores the fact that there are interactions between observations/clicks, which is another type of confounder and play a critical role in CTLR. To illustrate this problem in a more intuitive way, we present an example in Figure 1. Here, we aim to estimate the user observation probability at position $p + 1$, once the observation and click events at position p end. It is obvious that observation probability is quite different when we adopt different decisions at position p because of the dependencies between observations/clicks. In some domains, these dependencies would dominate the click decision since they reflect user's current tendency to continue browsing. If we simply assume that each observation is isolated from other observation and click behaviors, it will incur a large interactional observation bias no matter how much data is collected.

Unfortunately, it is rather challenging to model the interactional observation mechanism, although it naturally helps estimate observation probability. First, the model space of a full interaction structure of observations/clicks would increase combinatorially and then incur a large variance in the training process. While defining the interaction structure in prior could reduce model space, it faces the risk of building a misspecified model. Second, the data representing observation behavior is not accessible from user's implicit feedback (unobserved confounder), which impedes observation propensity estimation from training data.

To tackle these issues, in this work, we leverage the embedding method and propose an Interactional Observation-Based Model (IOBM) to estimate the observation probability. We argue that while there exist complex observed and unobserved confounders, it is sufficient to exploit the embedding as a substitute (observed) confounder to uncover the information relevant for the prediction of the observation propensity. Meanwhile, the embedding method offers an alternative to the fully specified generative model w.r.t. observation and decouples the complex interaction structure of observations/clicks. Specifically, we first learn the individual observation embedding to capture the click and position information from specific user observation events. Then we learn the interactional observation embedding which uncovers the hidden relationship among observations. To filter out irrelevant noises and reduce contextual bias, the intra-observation attention and inter-observation attention are further applied based on query context information. The proposed IOBM is a plug-and-play model, which can be seamlessly integrated into any IPS-based CLTR frameworks. We conduct comprehensive empirical evaluations on two LTR benchmark datasets, which shows that the proposed IOBM consistently outperforms the baseline methods in various click situations and verifies its effectiveness of eliminating interactional observation bias.

2 INTERACTIONAL OBSERVATION-BASED MODEL (IOBM)

2.1 Problem Formulation

Given a set of queries Q, where each query q is sampled from a data generating distribution, a ranking system R generates a ranking list of documents π_q retrieved for each query q, sorted by their predicted relevance score. The goal of learning to rank is to optimize S by minimizing the empirical risk $\mathcal{R}$:

$$\mathcal{R}(Q,S) = \frac{1}{|Q|}\sum_{q\in Q}\Delta(\pi_q \mid q),$$

where $\Delta(\pi_q \mid q)$ is a loss for a single query. Assuming that we already know the true relevance r_p^q for query-document pair (q, d) in position p (full information setting), it commonly takes the form of:

$$\Delta_{\text{full-info}}(\pi_q \mid q) = \sum_{p\in\{1,\ldots,|\pi_q|\}} \delta\left(\pi_q[p] \mid \pi_q\right)\cdot r_p^q,$$

where $\delta(d \mid \pi_q)$ can be any specific metric of interest for d in the ranking list π_q. The relevance score r_p^q denotes how relevant a document d locating at p related to a query q, which is typically obtained by human annotaion. However, they are expensive and even impossible to obtain, especially in a large massive dataset.

In contrast, the click logs from users are an attractive proxy of relevance signals, which are cheap and timely to collect at scale. Nevertheless, unlike the full-info setting, r_p^q is unknown in this partial information setting and click signal c_p^q in position p is usually biased [39], which could not reflect the real relevance value exactly. For example, higher-ranked documents are more likely to be observed and clicked (known as position bias).

To address this problem, researchers proposed to use the examination hypothesis to model user's click behavior, which assumes that in a certain position p, clicks ($c_p^q = 1$) appearing on the relevant results ($r_p^q = 1$) should also be examined ($o_p^q = 1$):

$$\Pr(c_p^q = 1 \mid q, p) = \Pr(o_p^q = 1 \mid q, p)\cdot\Pr(r_p^q = 1 \mid q, p, o_p^q = 1). \quad (1)$$

Obviously, click signals are biased towards observation probability $\Pr(o_p^q = 1 \mid q, p)$. Therefore, [29] proposed to eliminate the bias via the inverse propensity scoring (IPS) method:

$$\Delta_{\text{IPS}}(\pi_q \mid q) = \sum_{p\in\{1,\ldots,|\pi_q|\}} \frac{\delta\left(\pi_q[p] \mid \pi_q\right)\cdot c_p^q}{\Pr(o_p^q = 1 \mid q, p)}.$$

[1, 29] proved that $\Delta_{\text{IPS}}(\pi_q \mid q)$ is an unbiased estimate of $\Delta_{\text{full-info}}(\pi_q \mid q)$. It implies that the crux of IPS methods is a well-specified observation probability.

2.2 Causal View for IPS-Based Models

From a causal view, the spirit of counterfactual learning to rank is to answer an intervention question – for each query, would the document be clicked if we "forced" the user to observe it? Here, user's observation behavior at a specific position is a "treatment" and his click behavior is an "outcome". The problem is that there are observed and unobserved confounders, variables that affect both observation behavior and click behavior, which leads to spurious effects induced by an imbalance of the confounder distributions among different treatments. Thus, the success of causal predictions with observational data depends on whether we have properly accounted for all confounders [37]. In this way, the observation behavior is *identifiable* from click data, i.e. we can not modify the conditional distribution characterized by the observation behavior without disturbing the click data distribution.

2.2.1 *Position-Based Model (PBM).*

In order to simplify the generative model of observation behavior and widen the applicability of IPS-methods to larger slates [32], [29] assumes that the observation probability depends only on position p, but not on q or π_q, which is formulated as:

$$\Pr(o_p^q = 1 \mid q, \pi_q, p) = \Pr(o_p = 1 \mid p).$$

Such a model factors out q and π_q, and makes the estimation of observation probability simpler. Most of the existing works [4, 22, 44] follow this assumption of PBM since it has a sufficiently low variance due to its small extra parameter space. However, when there exist other confounders except for position, like q and π_q, the causal effect cannot be identified. For example, each query shares the same propensity score w.r.t. position, regardless of the uniqueness of each query context. As a consequence, the estimator is biased for the misspecified propensities of the PBM no matter how much data is collected. In section 4.2, We empirically found that under some circumstances, PBM-based CLTR performs even worse than that without using any debiasing methods.

2.2.2 *Contextual Position-Based Model (CPBM).*

To attain a better specified generative model for observation behavior, CPBM [16] takes one step further by assuming that the observation probability is determined not only by position p but the handcrafted selected query context features $f^{\text{sel}}(\boldsymbol{x}_q)$. Here, $\boldsymbol{x}_q$ signifies the original query context features and $f^{\text{sel}}(\cdot)$ signifies a handcrafted feature selection process. The query context features include the query itself and features describing the query (e.g., query length), the candidate set (e.g., size), and the user (e.g., age) [16]. Thus, the observation probability of CPBM can be defined as:

$$\Pr(o_p^q = 1 \mid q, \pi_q, p) = \Pr(o_p^q = 1 \mid p, f^{\text{sel}}(\boldsymbol{x}_q)).$$

Compared to PBM, CPBM introduces additional confounders to reduce query contextual bias and is capable of handling the circumstance where observation bias varies from query to query. Note that, instead of directly using the original query context features, CPBM manually selects a small subset of features. This is because while high-dimensional query context features may ameliorate the identifiability issue, they may pose new challenge on accurate propensity score estimation [7, 30].

2.3 Interactional Observation-Based Model (IOBM)

While PBM [29] and CPBM [16] have proven effective, they do not perform well particularly when there are interactions between observations and clicks since they assume that each observation is isolated from every other observation and click behaviors in the ranking list. In some scenarios, these interactions dominate the click decision. This is because a click (or an observation) action not only serves as an indicator of how attractive the document is, but also affects user's current motivation for whether to continue browsing. Therefore, we formulate observation probability at position p as

$$\Pr(o_p^q = 1 \mid p, \boldsymbol{x}_q, o_1, \dots, o_{p-1}, o_{p+1}, \dots, o_{|\pi_q|}, c_1, \dots, c_{p-1}, c_{p+1}, \dots, c_{|\pi_q|}). \tag{2}$$

The issue with directly estimating Eq. (2) is that the full network structure among observations/clicks random variables will incur a combinatorial explosion in the number of possible dependencies, resulting in high variance with learning process and making it impractical. To address this problem, some existing works [32, 41] first set the structure of the dependencies between observations/clicks manually, and then learn patterns of the observation interaction from the predefined set of rules. Nevertheless, a well-specified set of dependency rules is not available in prior. Causal estimates based on misspecified models are inherently suspect. Additionally, compared to PBM and CPBM, Eq. (2) imposes unobserved confounders $o_1, \dots, o_{p-1}, o_{p+1}, \dots, o_{|\pi_q|}$, which violates unconfoundedness assumption [35] and impedes causal estimate.

To tackle these two issues, inspired by [31, 42], we leverage observational data to find a proxy for the observed and unobserved confounders. In particular, [42] claimed that it is not required to recover all the information from confounders. Instead, it suffices to recover only the part of confounders that are relevant for the prediction of the propensity score. Therefore, if we can build a good predictive model for the treatment then we can plug the outputs into a causal effect estimate directly, without any demand to recover the true confounders. To achieve this goal, we adopt the embedding method to construct the proxy for confounders from observational data. This is because the embedding method offers an alternative to the fully specified generative model and decouples the complex network structure of the dependencies among observations/clicks random variables. Formally, the observation probability can be defined as

$$\Pr(o_p^q = 1 \mid \mathbf{Emb}_o(p, \boldsymbol{x}_q, c_1, \dots, c_{p-1}, c_{p+1}, \dots, c_{|\pi_q|})),$$

where $\mathbf{Emb}_o(\cdot)$ denotes the interactional observation embedding function. Each embedding explains the query context and local dependency structure of observation p.

3 MODEL IMPLEMENTATION

Up to this point, we have shown that to effectively apply CLTR in practice, we need to learn a low-dimensional interaction observation embedding that suffices for causal identification and enables efficient propensity estimation from click data. In this section, we first introduce the two key components of the proposed Interactional Observation-Based Model (IOBM): (1) **Individual Observation Embedding** learning representation of each observation by concatenating its click embedding and position embedding to capture the related information from specific user observation event, and (2) **Interactional Observation Embedding** learning the hidden relationships among each observation. Then, we present the optimization objective and training process of the proposed model. The overall model architecture is illustrated in Figure 2. To simplify writing, we omit the superscript q of each variable when a query q is given in the context of the paper.

3.1 Individual Observation Embedding

Given a displayed ranking list π_q of a query q, each observation o is linked to a position $p \in \{1, 2, \dots, |\pi_q|\}$ and a click $c \in \{0, 1\}$. Consider the inherent complexity of observation interaction, existing works [4, 22, 29] that represent both the click and position as scalars may be insufficient for the task. To make them more

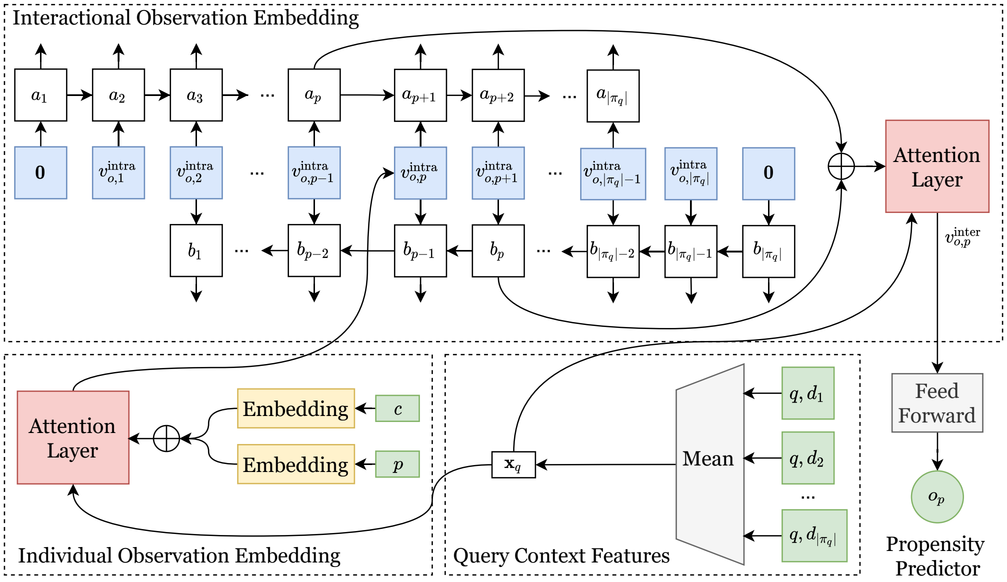


Figure 2: Framework of the proposed IOBM

expressive, We first embed position p and click c:

$$\boldsymbol{v}_p = \mathbf{Emb}_p(p), \boldsymbol{v}_c = \mathbf{Emb}_c(c), \tag{3}$$

where $\mathbf{Emb}_p \in \mathbb{R}^{|\pi_q|\times l_p}, \mathbf{Emb}_c \in \mathbb{R}^{2\times l_c}$. l_p and l_c denote the embedding size of the position and click, respectively. Note that documents placed at the top position have a high probability to be clicked. The skewed distribution of click data render the learning of bottom position scalars difficult. Thus, another benefit of using embedding is that projecting the position and click into a common embedding space could alleviate this problem by learning all positions in a collaborative way.

We concatenate the two embeddings together to represent user's observation as $[\boldsymbol{v}_p \oplus \boldsymbol{v}_c]$. While it can be used as user's observation embedding, it is oversimplified to tackle the scenario where the observation bias varies from query to query [16]. A straightforward solution is to incorporate query context features $\boldsymbol{x}_q \in \mathbb{R}^{l_s}$, where l_s denotes the feature size, and formulate user's observation as a query-dependent function. Since the original query context contains too much noisy and irrelevant information, [16] considered to select important features in a handcrafted way. However, we argue that the handcrafted features can not be naturally applied to different scenarios. Instead, we implement the query context features $\boldsymbol{x}_q$ as the average of query and document features. Besides, from the view of observation embedding, not all dimensions of the observation embeddings are equally important in terms of different query context. The importance of each feature should be dynamic and dependent on the query context. Therefore, we apply an **intra-observation attention** mechanism to filter out irrelevant noises and alleviate the error propagation. Formally, the final observation embedding $\boldsymbol{v}^{\text{intra}}_o$ can be formulated as:

$$\boldsymbol{v}^{\text{intra}}_o = f^{\text{att}}_{\boldsymbol{x}_q}([\boldsymbol{v}_p \oplus \boldsymbol{v}_c]; W^{\text{intra}}, b^{\text{intra}}), \tag{4}$$

where the j-th axis value of the function $f^{\text{att}}_{\boldsymbol{x}_q}(\cdot)$ is defined as:

$$\begin{aligned} & f^{\text{att}}_{\boldsymbol{x}_q}(\boldsymbol{x}; W^{\text{intra}}, b^{\text{intra}})_{(j)} \\ = & \frac{l_{\boldsymbol{x}} \cdot \exp(\tanh_{(j)}(W^{\text{intra}}[\boldsymbol{x}_q \oplus x] + b^{\text{intra}}))}{\sum_{i=1}^{l_{\boldsymbol{x}}} \exp(\tanh_{(i)}(W^{\text{intra}}[\boldsymbol{x}_q \oplus x] + b^{\text{intra}}))} \cdot \boldsymbol{x}_{(j)}, \end{aligned} \tag{5}$$

where $\boldsymbol{x}_{(j)}$ denotes the j-th axis of the vector $\boldsymbol{x}$, $l_{\boldsymbol{x}}$ denotes the size of input vector $\boldsymbol{x}$, $W^{\text{intra}} \in \mathbb{R}^{(l_p+l_c)\times(l_p+l_c+l_s)}$ and $b^{\text{intra}} \in \mathbb{R}^{(l_p+l_c)}$ are learnable weights for the intra-observation attention. Note that to align the scale of each dimension of output of $f^{\text{att}}_{\boldsymbol{x}_q}(\cdot)$ with that of input, we multiple the attention score in left term in Eq. (5) by a constant $l_{\boldsymbol{x}}$.

3.2 Interactional Observation Embedding

The intra-observation attention smooths over position and click information as it compresses each observation into a single embedding. However, each individual observation usually depends on other observations in the displayed ranking list π_q and their joint effect [15, 19, 20]. Such interactions between observations can be especially dominant when user's attention is relatively limited to the entire displayed ranking list π_q. However, directly modeling the full network structure between observations would incur a combinatorially large number of potential dependencies. As such, we propose to learn the interactional observation embedding from training data, which allows us to capture more complex observation interaction patterns than the handcrafted ones.

Specifically, we instantiate the learning of interactional observation embedding using the bidirectional LSTM (Bi-LSTM) model [18]. This is because users may not obey the ranking order to view documents. Documents locating at the lower position will still affect ones locating at the higher position due to the users' potential looking back actions. The Bi-LSTM model consists of two components, forward LSTM and backward LSTM. Forward LSTM goes over the observation embeddings in the ranking list except the last position in the top down order, i.e., $\boldsymbol{v}_{o,1}, \boldsymbol{v}_{o,2}, ..., \boldsymbol{v}_{o,|\pi_q|-1}$, to produce embeddings for the observation in the next position, i.e., $\boldsymbol{a}_2, \boldsymbol{a}_3$, ..., $\boldsymbol{a}_{|\pi_q|}$. Backward LSTM goes over the observation embeddings in the ranking list except the first position in the reverse order, i.e., $\boldsymbol{v}_{o,|\pi_q|}, \boldsymbol{v}_{o,|\pi_q|-1}, ..., \boldsymbol{v}_{o,2}$ and produce embeddings for the previous observation, i.e., $\boldsymbol{b}_{|\pi_q|-1}, \boldsymbol{b}_{|\pi_q|-2}, ..., \boldsymbol{b}_1$. Since there is no precedent and succeeding observation embedding to produce $\boldsymbol{a}_1$ and $\boldsymbol{b}_n$, respectively, without loss of generality, we assign $\boldsymbol{a}_1 = \text{Forward}(\mathbf{0})$, and $\boldsymbol{b}_n = \text{Backward}(\mathbf{0})$. Therefore, the interactional observation embedding could be formulated as:

$$\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{|\pi_q|} = \text{Forward}(\mathbf{0}, \boldsymbol{v}_{o,1}, \boldsymbol{v}_{o,2}, \ldots, \boldsymbol{v}_{o,|\pi_q|-1})$$
$$\boldsymbol{b}_{|\pi_q|}, \boldsymbol{b}_{|\pi_q|-1}, \ldots, \boldsymbol{b}_1 = \text{Backward}(\mathbf{0}, \boldsymbol{v}_{o,|\pi_q|}, \boldsymbol{v}_{o,|\pi_q|-1}, \ldots, \boldsymbol{v}_{o,2}) \tag{6}$$

In this way, the predicted embedding $\boldsymbol{a}_p$ (or $\boldsymbol{b}_p$) could adaptively capture complex patterns of all previous (or subsequent) observation interaction. Then, we concatenate the two embeddings $\boldsymbol{a}_p$ and $\boldsymbol{b}_p$ together to represent user's interactional observation embedding at position p as $[\boldsymbol{a}_p \oplus \boldsymbol{b}_p]$. Note that it do not contain any information about $\boldsymbol{v}_{o,p}$, so no click information will leak. On the other side, since $[\boldsymbol{a}_p \oplus \boldsymbol{b}_p]$ contains all positions information except position p, which is a complementary of position p and has the same effect as extracting information from position p directly.

Similarly, the patterns of observation interaction often vary from query to query. The inversed order predicted embedding $\boldsymbol{b}_p$ should be mostly irrelevant when a user tend to examine the ranking list in the top-down order. Besides, the importance of each feature should be dynamic and dependent on the query context. Thus, we apply an **inter-observation attention** mechanism to reduce the negative effect of irrelevant information. Formally, the final interactional observational embedding can be defined as:

$$\boldsymbol{v}_{o,p}^{\text{inter}} = f_{\boldsymbol{x}_q}^{\text{att}}([\boldsymbol{a}_p \oplus \boldsymbol{b}_p]; W^{\text{inter}}, b^{\text{inter}}), \tag{7}$$

where $W^{\text{inter}} \in \mathbb{R}^{2l_o \times (2l_o + l_s)}$ and $b^{\text{inter}} \in \mathbb{R}^{2l_o}$ are learnable weights (Denote the size of both a_p and b_p as l_o).

3.3 Propensity Predictor and Objective Function

Given the interactional observational embedding $\boldsymbol{v}_{o,p}^{\text{inter}}$ for each position p, we can leverage a simple feed forward layer to define the propensity predictor as

$$\hat{o}_p = \sigma(W \cdot \boldsymbol{v}_{o,p}^{\text{inter}} + b), \tag{8}$$

where $\sigma(\cdot)$ denotes the *sigmoid* activation function. $W \in \mathbb{R}^{2l_o}$ and $b \in \mathbb{R}$ denotes trainable weights.

Subsequently, we leverage a pointwise-based loss to train our IOBM model. Given the observation ground truth o_p, the objective function can be written as follows:

$$\mathcal{L}_{\text{IOBM}}(\hat{o}_p, o_p; \theta) = -o_p \log \hat{o}_p - (1 - o_p) \log(1 - \hat{o}_p) + \lambda ||\theta||_2^2, \tag{9}$$

where λ is the hyper-parameter controlling the L2 regularization, and θ represents the kernel weights denoted by W in Eq. (4), Eq. (7), Eq. (8) and kernel matrices in LSTM cells.

In practice, the ground truth o_p isn't available. [29] proposed methods to estimate it by randomization of search results. Some researchers [3, 16] further developed methods for estimating it from click data offline. More recently, jointly estimating observation and training a ranker from click data becomes popular [4, 22, 24], since they do not require a separate experiment to estimate click bias. In our empirical study, we focus on jointly optimization-based methods, like DLA [4] and Regression-EM [44] due to space limitation. It is worth noting that the proposed IOBM is a plug-and-play model, which can be seamlessly integrated into any IPS-based CLTR frameworks. The overview of the algorithm for integrating IOBM into CLTR frameworks is summarized in Algorithm 1.

Algorithm 1 Jointly optimization based CLTR frameworks integrated with IOBM

```
Require: $Q = \{(q, \pi_q, \boldsymbol{c}^q)\}$, an IPS-based framework $F$
Ensure: a ranking model denoted as $R$, an observation IOBM denoted as $C$
Initialize $R$ and $C$
repeat
    for $(q, \pi_q, \boldsymbol{c}_q) \in Q$ do
        for $p$ in $\{1, 2, \ldots, |\pi_q|\}$ do
            Calculate the propensity prediction $\hat{o}_p$, according to Eq. (3) - Eq. (8)
            Based on $F$, train $R$ with $\hat{o}_p$, then estimate the target propensity $o_p$
            Train $C$ with $\hat{o}_p, o_p$ according to Eq. (9)
        end for
    end for
until Convergence;
return $R, C$
```

4 EXPERIMENTS

In this section, we show our experimental setup and empirical results. We have published our code [1] based on Unbiased Learning To Rank Algorithms (ULTRA) framework [5, 6]. In general, We aim to answer the following research questions (RQs):

- **RQ1**: How does IOBM perform on the different user click patterns, compared to existing propensity models?
- **RQ2**: How do IOBM and PBM-IPS perform on different strengths of dependencies between click events in the same query session?
- **RQ3**: Whether IOBM can debias context-dependent examination bias efficiently?
- **RQ4**: Will IOBM still perform well when integrated into different CLTR frameworks and ranking models?

4.1 Experimental Settings

4.1.1 Dataset. We conducted a set of experiments on two benchmark LTR datasets.

- **Yahoo!**[2]. One of the most widely used benchmarks for ranking. It contains 29,921 queries with 710k documents, and each query-document pair has 700 features extracted from real-world search engines with 5-level relevance labels.
- **MSLR-WEB30K**[3]. It contains 31,531 queries with around 3,800k documents. Each query-document pair has 136 features extracted by human exports, with 5-level relevance labels as well.

[1] https://github.com/Keytoyze/Interactional-Observation-Based-Model

[2] https://webscope.sandbox.yahoo.com/

[3] https://www.microsoft.com/en-us/research/project/mslr/

We followed the given data split of training, validation, and testing of datasets. To generate an initial ranking list for each query, we followed the process described in [29] and trained a Ranking SVM model [26] using 1% of the training data with real relevance labels to sort the documents. Based on these initial ranking lists generated by the ranker, we simulated click data in the following ways.

4.1.2 Click Simulation. We considered three click generation models, PBM, UBM and BDCM, to simulate clicks in three scenarios: isolated scenario, cascade scenario and non-cascade interactional scenario, respectively.

PBM. In PBM, the clicks are sampled from examination hypothesis Eq. (1). Following the steps proposed by [12], the relevance probability is set to be:

$$\Pr(r_p^q = 1|q, p, o_p^q = 1) = \epsilon + (1 - \epsilon)\frac{2^y - 1}{2^{y_{\max}} - 1}, \tag{10}$$

where $y \in [0, y_{\max}]$ the relevance label of the document locating at p, given by the dataset. In both of two datasets, $y_{\max} = 4$. ϵ is the noise level describing how much probability a user may click on an irrelevant document. We had $\epsilon = 0.1$ as the default setting when not mentioned otherwise.

The examination probability is set to be:

$$\Pr_{PBM}(o_p^q = 1 \mid q, p) = \rho_p. \tag{11}$$

We adopted the presentation bias ρ_p estimated by Joachims et al. [39] through eye-tracking experiments.

UBM. The user browsing model (UBM) [15] is a well-known cascade-based click model. In UBM, a user is modeled as searching and clicking documents from top to bottom, and the examination probability is determined by the distance between the current position p and the last click position p_c. Similar to PBM, we sampled clicks according to Eq. (1) and Eq. (10), and the examination probability is:

$$\Pr_{UBM}(o_p^q = 1 \mid q, p, \mathcal{D}) = \rho(p, \mathcal{D}),, \tag{12}$$

where $\mathcal{D}$ is the position distance, $\mathcal{D} = p - p_c$. We adopted the value of $\rho(p, \mathcal{D})$ estimated by numerical experiments over 21 training sets in [15].

BDCM. To simulate a non-cascade interactional scenario and model the user behavior of looking back, we proposed a more complicated setting: a user first looks at documents in the top-down order, and then continue looking in the reversed order. Formally, we merged two click sequences generated by two cascade-based click models (DCM) [20] from opposite directions, which is called bidirectional cascade-based click model (BDCM). In DCM, a user examines the results from one end to the other end until she finds an relevant result, $\Pr(o_{j+1} = 1 \mid o_j = 1, c_j = 0) = 1$, where $j \in \{1, 2, \ldots, |\pi_q| - 1\}$ is the ordinal number. After each click, user has a chance of not satisfied depending on the current ordinal number, $\Pr(o_{j+1} = 1 \mid o_j = 1) = \lambda_j$.

BDCM consists of two DCM models $\Pr'_{DCM}$ and $\Pr''_{DCM}$ from opposite directions, which generates two click sequences c' and c'', respectively. The former is from top to bottom, and the later is from bottom to top:

$$\Pr'_{DCM}(o_p^q = 1 \mid q, p) = \prod_{1 \le i < p} (1 - c'_i(1 - \lambda_i)), \tag{13}$$

$$\Pr''_{DCM}(o_p^q = 1 \mid q, p) = \prod_{p < i \le |\pi_q|} (1 - c''_i(1 - \lambda_{|\pi_q| - i + 1})), \tag{14}$$

where the hyper parameters λ takes a reciprocal formula form in our settings: $\lambda_i = \frac{1}{i}$.

We sampled c' step by step according to Eq. (1), Eq. (10) and Eq. (13), then sampled c'' step by step according to Eq. (1), Eq. (10) and Eq. (14). The final output click sequence of BDCM c was generated by merging: $c_i = \min\{c'_i + c''_i, 1\}$.

4.1.3 Baselines. We implemented two groups of propensity models for comparing. The first group has no parameter to train.

- **Labeled-data**: This model uses the ground truth labels to train the ranker, to test the performance of the ranking model. Its performance can be considered as an upper bound for the ranker.
- **Click-data**: This model just uses the raw click data to train the ranker, without any correction.

The second group is existing IPS propensity models, to be trained along with a ranking model and provide a propensity estimation for it. For each query, only the top N documents were considered to be displayed. In our case, we set $N = 10$.

- **PBM-IPS**: This model has N parameters to describe the propensity scores depend on position.
- **UBM-IPS**: This model has $\frac{N(N+1)}{2}$ parameters, to feed each value of the user browsing function $\rho(p, \mathcal{D})$.
- **DCM-IPS**: This model is proposed by [41], and uses a DCM click model to estimate the observation probability.
- **CPBM-IPS**: Proposed by [16], this model uses an MLP to map a context feature vector to position-based propensity scores. We implemented this baseline with a hidden layer with size 256 and *elu* activation.

It's worth noting that PBM-IPS is the most suitable model for PBM setting, and UBM-IPS is the most suitable model for UBM setting. By comparing with them, we can find whether IOBM could mine the user's click pattern correctly.

4.1.4 Training and Evaluation. To make fair comparisons, we fixed the unbiased LTR framework to DLA [4], and integrated different propensity models into them to test their performances. We trained these methods with a batch size of 256. For the ranker side, following [41], we fixed the ranker to a DNN, with the loss being softmax cross-entropy. We used three layers with sizes $\{512, 256, 128\}$ and *elu* activation, and the last two layers use dropout with a dropping probability of 0.1. We used SGD to train the ranker, with a learning rate of 0.03 for Yahoo! and 0.3 for MSLR-WEB30K. For the propensity model side, we also used SGD to train the parameters, with a learning rate selected from $\{0.3, 0.03\}$. To avoid exploding variance, we used a propensity clipping constant of 100 based on [38]. For IOBM, we select the hyper parameter λ from $\{0.1, 0.01, 0.001, 0.0001, 0\}$, and set $l_p = l_c = 4$, $l_o = 8$.

We used NDCG@1, NDCG@3, NDCG@5, and NDCG@10 as the main performance metrics. We have also computed the MRR metric, with binarizing the relevance by clipping the label above 1. Each model was trained for 15K epochs, and we adopted the hyperparameters with the best results based on NDCG@10 tested

Table 1: Comparison of different IPS method under two datasets and three click generation models. Significant performance improvements (t-test with p-value < 0.05) with PBM-IPS and the best baseline are denoted as + and † respectively.

Click Model	Propensity Model	Yahoo!					MSLR-WEB30K				
		MRR	NDCG@k				MRR	NDCG@k			
			$k = 1$	$k = 3$	$k = 5$	$k = 10$		$k = 1$	$k = 3$	$k = 5$	$k = 10$
Labeled Data (Upper Bound)		0.9343	0.6826	0.6842	0.7054	0.7541	0.8338	0.3873	0.3828	0.3888	0.4113
PBM	IOBM	**0.9262**	**0.6746**	**0.6783**	**0.7000**	**0.7484**	0.8276	**0.3691**$^+$	**0.3672**$^+$	**0.3765**$^+$	**0.4005**$^+$
	PBM-IPS	0.9257	0.6742	0.6779	0.6990	0.7479	**0.8288**	0.3493	0.3576	0.3693	0.3953
	CPBM-IPS	0.9225	0.6674	0.6717	0.6938	0.7434	0.7764	0.3371	0.3310	0.3394	0.3616
	UBM-IPS	0.9256	0.6726	0.6773	0.6995	0.7481	0.8285	0.3643$^+$	0.3653$^+$	0.3754$^+$	0.3996$^+$
	DCM-IPS	0.9216	0.6577	0.6652	0.6884	0.7394	0.8222	0.3308	0.3428	0.3568	0.3858
	Click Data	0.9217	0.6564	0.6643	0.6875	0.7386	0.8201	0.3257	0.3370	0.3512	0.3800
UBM	IOBM	**0.9264**$^+$	0.6755$^+$	**0.6802**$^+$	**0.7011**$^+$	**0.7495**$^+$	0.8237$^+$	**0.3818**†	**0.3738**†	**0.3809**†	**0.4038**†
	PBM-IPS	0.9238	0.6704	0.6737	0.6957	0.7447	0.7917	0.3581	0.3517	0.3588	0.3803
	CPBM-IPS	0.9238	0.6704	0.6730	0.6956	0.7450	0.7824	0.3525	0.3431	0.3512	0.3717
	UBM-IPS	0.9255$^+$	**0.6759**$^+$	0.6797$^+$	0.7006$^+$	0.7493$^+$	0.8234$^+$	0.3783$^+$	0.3722$^+$	0.3797$^+$	0.4023$^+$
	DCM-IPS	0.9243	0.6699	0.6772$^+$	0.6987$^+$	0.7471$^+$	0.8322$^+$	0.3666$^+$	0.3687$^+$	0.3783$^+$	0.4023$^+$
	Click Data	0.9243	0.6707	0.6769$^+$	0.6988$^+$	0.7476$^+$	**0.8326**$^+$	0.3640$^+$	0.3674$^+$	0.3777$^+$	0.4019$^+$
BDCM	IOBM	**0.9265**$^+$	**0.6750**$^+$	**0.6783**$^+$	**0.6991**$^+$	**0.7480**$^+$	**0.8082**$^+$	**0.3696**$^+$	**0.3635**†	**0.3700**†	**0.3912**†
	PBM-IPS	0.9246	0.6707	0.6732	0.6955	0.7448	0.7916	0.3585	0.3505	0.3579	0.3786
	CPBM-IPS	0.9240	0.6702	0.6734	0.6952	0.7444	0.7899	0.3562	0.3474	0.3548	0.3757
	UBM-IPS	0.9252	0.6733$^+$	0.6753$^+$	0.6968	0.7462$^+$	0.8082$^+$	0.3681$^+$	0.3611$^+$	0.3678$^+$	0.3891$^+$
	DCM-IPS	0.9205	0.6591	0.6629	0.6852	0.7365	0.6837	0.2587	0.2566	0.2648	0.2875
	Click Data	0.9259	0.6750$^+$	0.6778$^+$	0.6987$^+$	0.7475$^+$	0.8063$^+$	0.3680$^+$	0.3613$^+$	0.3677$^+$	0.3890$^+$

on the validation set. We run each experiment over 6 runs and reported the average results testing on the test set.

4.2 Click Settings Study (RQ1)

Table 1 summarizes the results about the performance of different propensity models under different settings. Particularly, we have the following findings:

- In the UBM settings, our model and UBM-IPS perform significantly better than PBM-IPS. This proves the necessity of introducing interactions in propensity estimation. In the BDCM settings where the document observation will be affected by all of the other documents, our model performs the best compared to other baseline IPS methods. These results prove that our model can find the correct user patterns hidden in the data, and perform similarly to (even better than) the best-fitted IPS model, if it exists.
- The propensity models that best fit the click generation models perform well in debiasing click data. However, a misused propensity model will estimate a wrong inversed propensity score, which leads to poor performance and even worse than the raw click data. Our model is robust regard of any click setting and always performs at the top level.
- Using raw click data from UBM and BDCM to train the ranker directly could get better performance, compared to PBM. One explanation is that documents listing at the bottom have a higher opportunity to be observed in these two click models, which dramatically reduces the click bias.
- In PBM settings, our model and UBM-IPS still perform better than PBM-IPS, especially on the MSLR-WEB30K dataset. One possible reason is that considering click signals of other documents can make the model stronger. It is beneficial even when the documents in the same session are independent.
- Human labeled data achieves the best performance and any IPS-based algorithm can still not beat it. There is still room for improvement in CLTR.

4.3 Dependency Study (RQ2)

To answer RQ2, we propose another hyper parameter $\sigma \in [0, 1]$ to control the click dependency strength in the click generation model. The observation probability of this model is defined to be

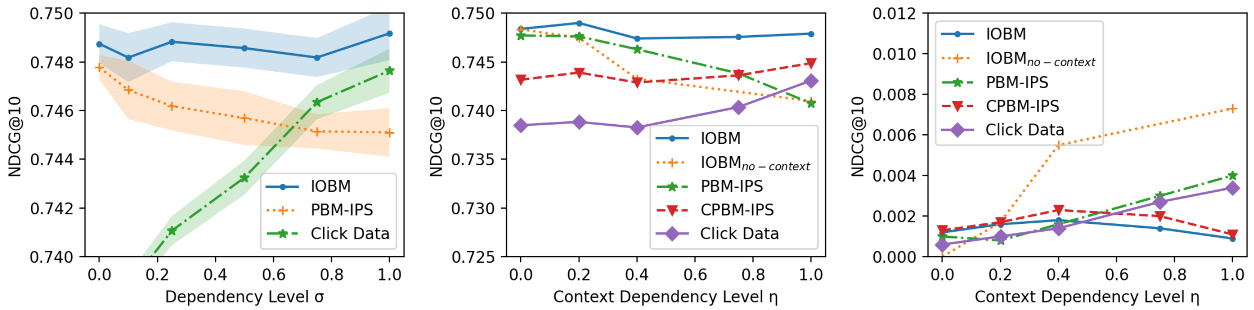
$$\Pr(o_q^p = 1 | q, p, p_c, \sigma) = \sigma\rho(p, \mathcal{D}) + (1 - \sigma)\rho(p), \qquad (15)$$

where $\rho(p, \mathcal{D})$ and $\rho(k)$ are the hyper parameters in UBM (Eq. (12)) and PBM (Eq. (11)) click generation models, respectively. Similarly, we sampled the clicks according to Eq. (1), Eq. (10) and Eq. (15). Note that the higher the value of σ, the strongest dependency is held in click sequence generated. $\sigma = 0$ and $\sigma = 1$ mean that the model degrades into PBM and UBM, respectively.

The performance across different σ is shown in Figure 3(a). We can notice that our model achieves the best performance since it properly handles interaction. The PBM-IPS only works when there is low dependence in the clicks, while its advantage over the click data diminishes with increased dependency. For a large dependency, PBM-IPS performs even worse than the raw click data. This proves once again that assuming the click independent cannot debias click data correctly, and our model could keep a stable well performance in all kinds of click settings.

4.4 Query Context-Dependency Study (RQ3)

We conducted a query context-dependency study in PBM settings to answer RQ3. In order to control the context-dependency, we adopted the methodology proposed by [16, 40] to modify the PBM observation probability. We set

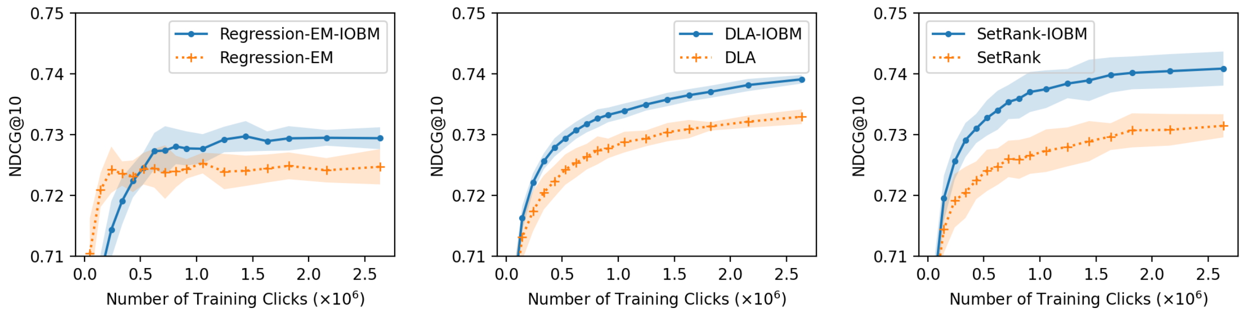


(a) Performance across Dependency Level

(b) Performance across Context-Dependency Level

(c) Standard Deviation of (b)

Figure 3: (a) Performance IOBM against PBM-IPS across different levels of dependency strength. The variance is displayed with the shadow areas. (b)(c) Performances and their standard deviations of different propensity models with different levels of context-dependency.



(a) Regression-EM

(b) DLA

(c) DLA with SetRank

Figure 4: Performances improvement by IOBM in different CLTR frameworks and ranking models across number of training clicks. Clicks generated by UBM. The variance is displayed with the shadow areas.

$$\Pr_{Context-PBM}(o_p^q = 1 | q, p, \boldsymbol{x}_q) = \rho_k^{\max\{\boldsymbol{w} \cdot \boldsymbol{x}_q + 1, 0\}}, \quad (16)$$

where $\boldsymbol{x}_q$ is the query context vector of query q. $\boldsymbol{w}$ is a ten-dimensional vector which is drawn uniformly from $[-\eta, \eta]$, where η is a hyper parameter controlling the strength of context-dependency. Following [40], we calculated $\boldsymbol{x}_q$ in the following ways: first, we trained ExtRa Trees [17] from queries in the total dataset, with normalized features and relevance. Then we selected and combined the top-10 important features as the document context features. Finally, we averaged all document context features in the same query as our query context feature x_q. We conducted an ablation study by replacing Query Context Features with a zero constant vector, to block any query contextual information input. This model is denoted as $\text{IOBM}_{no-context}$. We also trained CPBM-IPS with $\boldsymbol{x}_q$ as a baseline.

Figure 3(b) and 3(c) shows the average performance values and their standard deviations. We can see that when $\eta = 0$, there is no contextual bias in the click data, so IOBM, $\text{IOBM}_{no-context}$, and PBM-IPS models behavior similarly. With the increasing context-dependency, the performance of PBM-IPS and $\text{IOBM}_{no-context}$ decrease dramatically, and their standard deviations of NDCG@10 increase continuously, which indicates that a lack of context features input will lead to bad and volatile performance when position bias varying from query to query. In contrast, IOBM and CPBM-IPS always keep stable performance and low variance regardless of how severe contextual bias is. Besides, IOBM always performs significantly better than CPBM-IPS, which verifies the effectiveness of the proposed embedding method.

4.5 Generalizability Study (RQ4)

The previous study focused on the DLA framework with a DNN as the ranking model. To answer RQ4, we conducted more experiments on different unbiased LTR frameworks and different ranking models, where the click generation model is UBM. Here we tried three combinations:

- **Regression-EM**. Proposed by [44], the Regression-EM algorithm uses an EM framework to estimate propensity scores and ranking scores. We modified the M-step of position propensity updating, to train our IOBM model with the target that Regression-EM predicts, which is marked as Regression-EM-IOBM. A simple linear layer is used for relevance prediction.
- **DLA**. Proposed by [4], DLA views learning ranking models and learning IPS to be a dueling problem. We replace the raw position-dependent propensity model with our model, marked as DLA-IOBM.
- **DLA with SetRank**. The ranking model of the initial DLA framework is a DNN. We modified the ranking model to SetRank [34], which can capture the cross-document interactions in the ranking list. The IOBM version of DLA with SetRank is marked as SetRank-IOBM.

From Figure 4, we can see that our model significantly improves the performance of all of the frameworks and ranking models. With the increase in the number of training clicks, our model gets better performance. This shows the possibility of generalizing our model to the joint learning of propensities and unbiased ranker.

5 RELATED WORK

Here we review the related work from the two subareas: counterfactual learning to rank and interactional effect in learning to rank.

Counterfactual Learning to Rank. CLTR is an area introducing causal inferences to eliminate bias from LTR with implicit feedback. Previous works have mainly focused on position bias. [29] proposed the Inverse Propensity Scoring (IPS) method to reweigh the clicked documents based on the reciprocal of their observation propensities to achieve an unbiased estimate of the ranking objective. [29] adapted the Rank-SVM [25] to optimize IPS estimates. [1] provided a framework that can optimize any differentiable model w.r.t. an IPS estimation. The propensity scores are estimated by randomized interventions [29, 43], which unfortunately hurt the user's search experience. To avoid such interventions, [3, 16] further proposed to harvest intervention by exploiting historic click data with multiple different ranking functions. Despite their effectiveness, the strict assumption to construct interventional sets impedes their applicability. Recently, some researchers proposed to jointly estimate relevance and position bias, like Regression-EM [44], DLA [4] and Unbiased LambdaMART [22]. Note that our model could be seamlessly integrated into these IPS-based frameworks.

On the other side, researchers have attempted to develop models to deal with more types of click bias, based on the IPS framework. [2] modeled click noise as the position-dependent *trust bias* and proposed TrustPBM integrated with Regression EM framework and Beyes-IPS correction method. [16] proposed CPBM based on intervention harvest to address *contextual bias* where position bias varies from query to query. [33] studied the top-k ranking case where some items have no possibility to display, and introduced a policy-aware CLTR approach to addressing *selection bias*. [21] proposed a framework that allows multiple types of implicit feedback and incorporates related click models in a grid-based search scenario. [41] introduced cascade model-based click model into CLTR to address *cascade bias* where clicks obey a predefined model. In this work, we consider a novel *interactional bias*, since interactions among observations/clicks in the same ranking list are dominant in some domains.

Interactional Effect in Learning to Rank. In a ranking list, the relevance of a document would be influenced by other documents. This effects can be captured in the design of loss function, like pairwise [10] and list-wise [11]. Recently, taking a whole list of documents as input and directly ranking them together becomes popular [34, 36]. While these methods could capture cross-document relationships, the bias existing in click data cannot be eliminated. [24] proposed DRSR model to debias click data in a cascade scenario based on CLTR, which incorporates all the top-down contents in the list as context information and estimate the probability of user's conditional click rate. However, these methods only consider inter-document effects, regardless of the interactions between user's click.

Some researchers try to incorporate partial click data in the same query session to find user's behavior patterns. Using click models to predict click events is one of the methods to capture click interaction. Click models typically design a set of rules that describe user browsing behavior. For example, in cascade-based click models like UBM [15], DCM [20], DBN [13] and CCM [19], users are assumed to examine the results from top to bottom, where each next click depends on the previous click. These click models often have contradictory assumptions, and a well-specified model is not available in prior. Another kind of click model uses a universal adaptive neural model to learn the user click pattern from data [8, 9, 14]. However, click models commonly focus the most on predicting click events, rather than optimizing the overall ranking performance [4, 29]. The construction of click models usually separated from the learning to rank models, and the relevance estimation is an afterthought [29]. Besides, most click models need to be constructed offline and require each query-document pair to appear multiple times for reliable performance [4].

Based on CLTR, [41] recently proposed to remove the cascade bias with a predefined and simple click model, which is aligned with our spirit of eliminating interactional bias. However, in practice, a well-specified and predefined model of click behavior is not available in prior. Our proposed model can directly learn the complex interaction patterns with query context from click data.

6 CONCLUSION

In this work, we propose the Interactional Observation-Based Model (IOBM) to estimate the observation probability in a more general observation/clicks interaction settings. We first analyze two traditional IPS-based models, PBM and CPBM, in a causal view. Then we extend the assumption of the propensity model and introduce IOBM. Since there exist complex observed and unobserved confounders, we use the embedding as a substitute confounder to uncover the relevant information for the prediction of the observation propensity. We implement IOBM with two components, an Individual Observation Embedding learning the click and position information from a specific user observation event, and an Interactional Observation Embedding uncovering the hidden relationship among observations. Additionally, we utilize query context features and propose the intra-observation attention and inter-observation attention, to filter out irrelevant information and reduce contextual bias. The proposed IOBM is a plug-and-play model, which can be seamlessly integrated into any IPS-based CLTR frameworks. Extensive experiments on two LTR benchmark datasets demonstrate that our model consistently improves the performance of CLTR, in different click settings. In future work, it would be interesting to extend our basic idea of eliminating interactional bias into more CLTR frameworks, like non-IPS based CLTR [24, 40].

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A general framework for counterfactual learning-to-rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 5–14.

[2] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. 2019. Addressing trust bias for unbiased learning-to-rank. In *The World Wide Web Conference*. 4–14.

[3] 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.

[4] 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.

[5] Qingyao Ai, Jiaxin Mao, Yiqun Liu, and W. Bruce Croft. 2018. Unbiased Learning to Rank: Theory and Practice. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. ACM, New York, NY, USA, 2305–2306. https://doi.org/10.1145/3269206.3274274

[6] Qingyao Ai, Tao Yang, Huazheng Wang, and Jiaxin Mao. 2021. Unbiased Learning to Rank: Online or Offline? *ACM Trans. Inf. Syst.* 39, 2, Article 21 (Feb. 2021), 29 pages. https://doi.org/10.1145/3439861

[7] Susan Athey, Guido W Imbens, and Stefan Wager. 2016. Approximate residual balancing: De-biased inference of average treatment effects in high dimensions. *arXiv preprint arXiv:1604.07125* (2016).

[8] Alexey Borisov, Ilya Markov, Maarten De Rijke, and Pavel Serdyukov. 2016. A neural click model for web search. In *Proceedings of the 25th International Conference on World Wide Web*. 531–541.

[9] Alexey Borisov, Martijn Wardenaar, Ilya Markov, and Maarten de Rijke. 2018. A click sequence model for web search. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 45–54.

[10] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*. 89–96.

[11] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*. 129–136.

[12] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 621–630.

[13] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*. 1–10.

[14] 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.

[15] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 331–338.

[16] Zhichong Fang, Aman Agarwal, and Thorsten Joachims. 2019. Intervention harvesting for context-dependent examination-bias estimation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 825–834.

[17] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning* 63, 1 (2006), 3–42.

[18] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee, 6645–6649.

[19] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click chain model in web search. In *Proceedings of the 18th international conference on World wide web*. 11–20.

[20] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient multiple-click models in web search. In *Proceedings of the second acm international conference on web search and data mining*. 124–131.

[21] 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.

[22] 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.

[23] Rolf Jagerman and Maarten de Rijke. 2020. Accelerated Convergence for Counterfactual Learning to Rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 469–478.

[24] 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. *arXiv preprint arXiv:2004.14714* (2020).

[25] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 133–142.

[26] Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 217–226.

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

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

[29] 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*. 781–789.

[30] Kun Kuang, Peng Cui, Bo Li, Meng Jiang, and Shiqiang Yang. 2017. Estimating treatment effect in the wild via differentiated confounder balancing. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 265–274.

[31] Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. 2017. Causal effect inference with deep latent-variable models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6449–6459.

[32] James McInerney, Brian Brost, Praveen Chandar, Rishabh Mehrotra, and Benjamin Carterette. 2020. Counterfactual evaluation of slate recommendations with sequential reward interactions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1779–1788.

[33] Harrie Oosterhuis and Maarten de Rijke. 2020. Policy-aware unbiased learning to rank for top-k rankings. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 489–498.

[34] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. Setrank: Learning a permutation-invariant ranking model for information retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 499–508.

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

[36] Przemysław Pobrotyn, Tomasz Bartczak, Mikołaj Synowiec, Radosław Białobrzeski, and Jarosław Bojar. 2020. Context-aware learning to rank with self-attention. *arXiv preprint arXiv:2005.10084* (2020).

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

[38] Adith Swaminathan and Thorsten Joachims. 2015. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research* 16, 1 (2015), 1731–1755.

[39] Joachims Thorsten, Granka Laura, Pan Bing, Hembrooke Helene, and Gay Geri. 2005. Accurately Interpreting Clickthrough Data as Implicit. In *Proceedings of the 28th annual international ACM SIGIR conference*. 154–161.

[40] Mucun Tian, Chun Guo, Vito Ostuni, and Zhen Zhu. 2020. Counterfactual Learning to Rank using Heterogeneous Treatment Effect Estimation. *arXiv preprint arXiv:2007.09798* (2020).

[41] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2020. Cascade Model-based Propensity Estimation for Counterfactual Learning to Rank. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Jul 2020). https://doi.org/10.1145/3397271.3401299

[42] Victor Veitch, Yixin Wang, and David M Blei. 2019. Using embeddings to correct for unobserved confounding in networks. *arXiv preprint arXiv:1902.04114* (2019).

[43] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 115–124.

[44] 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 Eleventh ACM International Conference on Web Search and Data Mining*. 610–618.

[45] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th international conference on World wide web*. 1011–1018.