# Unbiased Learning-to-Rank Needs Unconfounded Propensity Estimation

Dan Luo
Lehigh University
Bethlehem, PA, USA
dal417@lehigh.edu

Lixin Zou†
Wuhan University
Wuhan, China
zoulixin@whu.edu.cn

Qingyao Ai
Tsinghua University
Beijing, China
aiqy@tsinghua.edu.cn

Zhiyu Chen*
Amazon.com, Inc.
Seattle, WA, USA
zhiyuche@amazon.com

Chenliang Li
Wuhan University
Wuhan, China
cllee@whu.edu.cn

Dawei Yin
Baidu Inc.
Beijing, China
yindawei@acm.org

Brian D. Davison
Lehigh University
Bethlehem, PA, USA
davison@cse.lehigh.edu

## ABSTRACT

The logs of the use of a search engine provide sufficient data to train a better ranker. However, it is well known that such implicit feedback reflects biases, and in particular a presentation bias that favors higher-ranked results. Unbiased Learning-to-Rank (ULTR) methods attempt to optimize performance by jointly modeling this bias along with the ranker so that the bias can be removed. Such methods have been shown to provide theoretical soundness, and promise superior performance and low deployment costs. However, existing ULTR methods don't recognize that query-document relevance is a confounder – it affects both the likelihood of a result being clicked because of relevance and the likelihood of the result being ranked high by the base ranker. Moreover, the performance guarantees of existing ULTR methods assume the use of a weak ranker – one that does a poor job of ranking documents based on relevance to a query. In practice, of course, commercial search engines use highly tuned rankers, and desire to improve upon them using the implicit judgments in search logs. This results in a significant correlation between position and relevance, which leads existing ULTR methods to overestimate click propensities in highly ranked results, reducing ULTR's effectiveness. This paper is the first to demonstrate the problem of propensity overestimation by ULTR algorithms, based on a causal analysis. We develop a new learning objective based on a backdoor adjustment. In addition, we introduce the Logging-Policy-aware Propensity (LPP) model that can jointly learn LPP and a more accurate ranker. We extensively test our approach on two public benchmark tasks and show that our proposal is effective, practical and significantly outperforms the state of the art. Our code is available at https://github.com/rowedenny/UPE.

## CCS CONCEPTS

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

† Corresponding author.
* The work was performed at Lehigh University prior to joining Amazon.

This work is licensed under a Creative Commons Attribution International 4.0 License.

*SIGIR '24, July 14–18, 2024, Washington, DC, USA*
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657772

## KEYWORDS

Unbiased Learning to Rank, Propensity Overestimation, Causal Intervention, Backdoor Adjustment

**ACM Reference Format:**
Dan Luo, Lixin Zou†, Qingyao Ai, Zhiyu Chen*, Chenliang Li, Dawei Yin, and Brian D. Davison. 2024. Unbiased Learning-to-Rank Needs Unconfounded Propensity Estimation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657772

## 1 INTRODUCTION

Unbiased Learning to Rank (ULTR), *i.e.,* leveraging implicit user feedback for optimizing learning-to-rank systems, has been studied for decades in information retrieval [20]. Usually, directly optimizing rankers with click data will suffer from the intrinsic noise and bias in user interactions [28, 57]. In particular, the position bias [13] that occurs because users are more likely to examine documents at higher ranks is known to distort ranking optimization if not handled properly [22, 23]. Among existing solutions, ULTR algorithms that jointly estimate click bias and construct unbiased rankers, namely AutoULTR algorithms, have drawn attention [2, 18, 43] because they do not need to conduct online interventions to estimate user bias (*i.e.,* the propensity models) and can be deployed on existing systems without hurting user experiences.

Recent AutoULTR methods investigate various biases, including position bias [1, 8, 18, 32, 39], contextual position bias [7, 9, 45, 54], trust bias [38, 39], exploitation bias [48], and bias against non-clicked but relevant results [40]. Despite their success, a key issue ignored is the quality of logging policies, *i.e.,* the previously deployed base ranker. In particular, AutoLTR efficacy is typically validated under a weak simulated logging policy [2, 8, 24, 38]. A weak logging policy means that user clicks are collected in a system in which the ranker can barely rank documents correctly according to their relevance to the query, *i.e.,* the relevance is independent of the ranking position. In contrast, most industrial ranking systems collect the clicks under a decent logging policy, *i.e.,* a more relevant document is more likely to be ranked in top positions, and resulting in more examinations. However, neglecting the influence of logging policies damages the learning of rankers through existing AutoULTR methods. Intuitively, when the logged positions for documents come from a perfect ranking model, the positions

sufficiently explain the clicks. The resulting ranking model, on the other hand, can be completely random and thus useless [53]. This outcome is undesirable in practical applications. Therefore, to realize the value of implicit feedback for learning to rank, there is a significant demand for an ULTR algorithm that can effectively learn an unbiased ranker under both weak and strong policies.

To this end, in this work, we investigate ULTR through the lens of causality and identify the *propensity overestimation* problem, in which the estimated examination probabilities of top positions will be much higher than their actual values under strong logging policies. Furthermore, by analyzing the causal relations in ULTR, we identify the query-document relevance feature confounder and derive the propensity's composition in existing ULTR methods: (1) the causal effect between position and examination, which is desired for propensity models; and (2) the confounding effect caused by relevance features, *i.e.,* the overestimated part. To eliminate the confounding effect, a straightforward solution is to adopt a backdoor adjustment [31], which, in ULTR, requires building a propensity model that takes both position and relevance into account. However, optimizing this propensity model is non-trivial because separating ranking and propensity models in AutoULTR algorithms is infeasible when they share a common input (*i.e.,* relevance features) and target (*i.e.,* user clicks) [46] (please refer to Section 5.5.2 for detailed analysis).

To learn an unconfounded propensity estimation (UPE) model in ULTR, we propose a Logging-Policy-aware Propensity Model. Based on the derivation of the propensity's compositions, we design a distinct two-step optimization strategy: **1) logging-policy-aware confounding effect learning** captures the confounding effect by the relevance feature confounder, and thereby separates the effects of ranking and propensity model; **2) joint propensity learning** obtains the propensity model desired, by fixing the confounding effect part and solely optimizing the position-related part. Thereafter, we are able to conduct unconfounded propensity inference via a backdoor adjustment and actualize unconfounded AutoULTR.

The contributions can be summarized as follows: **1)** We identify the propensity overestimation phenomenon. Based on our causal analysis, we ascribe this problem to the confounding effect between query-document relevance and position. **2)** We propose a novel Logging-Policy-aware Propensity Model and its distinct two-step optimization strategy: logging-policy-aware confounding effect learning and joint propensity learning, which solves the difficulty of backdoor adjustment in ULTR. **3)** We conduct extensive experiments on two benchmarks on online and offline simulations to demonstrate the superiority of our proposal.

## 2 PRELIMINARIES

### 2.1 Problem Formulation

Let $\mathcal{D}$ be the universal set of documents, and $\mathcal{Q}$ be the universal set of queries. For a user-issued query $q \in \mathcal{Q}$, $\pi_q$ is the ranked list retrieved for query $q$, $d_k \in \pi_q$ is the document presented at position $k$, and $\mathbf{x}_k \in \mathcal{X}$ and $r_k \in \{0, 1\}$ are the feature vector and binary relevance of the query document pair $(q, d_k)$, respectively. The goal of learning to rank is to find a mapping function $f$ from a query document feature vector $\mathbf{x}_k$ to its relevance $r_k$. In most cases, we are only concerned with the position of relevant documents ($r_d = 1$) in retrieval evaluations (*e.g.,* MAP, nDCG [19], ERR [6]), so we can formulate the ideal local ranking loss $\mathcal{L}_{ideal}$ as:

$$\mathcal{L}_{ideal}(f, q|\pi_q) = \sum_{d_k \in \pi_q, r_k=1} \Delta(f(\mathbf{x}_k), r_k|\pi_q), \tag{1}$$

where $\Delta$ is a function that computes the individual loss on each document. An alternative to relevance $r_k$ is implicit feedback from users, such as clicks. If we conduct learning to rank by replacing the relevance label $r_k$ with click label $c_k$ in Eq. 1, then the empirical local ranking loss is derived as follows,

$$\mathcal{L}_{naive}(f, q|\pi_q) = \sum_{d_k \in \pi_q, c_k=1} \Delta(f(\mathbf{x}_k), c_k|\pi_q), \tag{2}$$

where $c_k$ is a binary variable indicating whether the document at position $k$ is clicked in the ranked list $\pi_q$. However, this naive loss function is biased, due to factors such as position bias [24]. To address this issue, unbiased learning-to-rank aims to train a ranking model $f$ with the biased user clicks collected but immune to the bias in the implicit feedback.

### 2.2 AutoULTR Algorithms

AutoULTR jointly estimates click bias and constructs unbiased ranking models. According to the *examination hypothesis* [33] ($c_k = 1 \iff (e_k = 1 \text{ and } r_k = 1)$), the problem of learning a propensity model from click data (*i.e.,* the estimation of bias in clicks) can be treated as a dual problem of constructing an unbiased learning-to-rank model. In AutoULTR, an unbiased ranking system $f$ and a propensity model $g$ can be alternatively learned by optimizing the local ranking loss as:

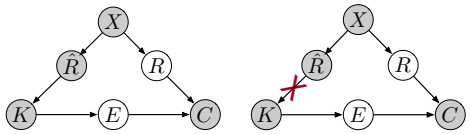
$$\mathcal{L}_{IPW}(f, q|\pi_q) = \sum_{d_k \in \pi_q, c_k=1} \frac{\Delta(f(\mathbf{x}_k), c_k|\pi_q)}{g(k)} \tag{3a}$$

$$\mathcal{L}_{IRW}(g, q|\pi_q) = \sum_{d_k \in \pi_q, c_k=1} \frac{\Delta(g(k), c_k|\pi_q)}{f(\mathbf{x}_k)}, \tag{3b}$$

where $g(k)$ approximates the propensity weights $P(E = 1|K = k)$. Following Equation 13 from Ai et al. [2], when examination is independent of relevance, the propensity weights tend toward the expected propensity weight. This convergence can be expressed as:

$$\frac{g(1)}{g(k)} = \frac{\mathbb{E}\left[e_1 \cdot r_1\right]}{\mathbb{E}\left[\frac{e_k \cdot r_k \cdot f_k}{f_k}\right]} = \frac{\mathbb{E}\left[\frac{r_1}{r_k}\right]}{\mathbb{E}\left[\frac{f_1}{f_k}\right]} \cdot \frac{\mathbb{E}\left[e_1\right]}{\mathbb{E}\left[e_k\right]}. \tag{4}$$

where $f_k$ is the abbreviation for $f(\mathbf{x}_k)$, which approximates the relevance for a document at position $k$; $\mathbb{E}\left[\frac{r_1}{r_k}\right]$ and $\mathbb{E}\left[\frac{f_1}{f_k}\right]$ are the real and estimated inverse relevance weights, and $\frac{\mathbb{E}[e_1]}{\mathbb{E}[e_k]}$ is the true inverse propensity weight we want to estimate. Intuitively, when the confounding effect of the logging policy induces a positive correlation between $e_k$ and $r_k$—that is, more relevant documents are observed more frequently—then the product $e_k \times r_k$ becomes amplified in cases where the document is more relevant than the average, i.e., the overestimation of the propensity weight. To theoretically prove it, in the subsequent section we will apply causal analysis to demonstrate that the conditional probability estimand $P(E|K)$ does not serve as an unbiased estimator of propensity.




(a) (b)

**Figure 1: (a) The causal graph $G$ for ULTR. (b) The causal graph with intervention $G'$ used in UPE. In particular, we apply a backdoor adjustment to cut the edge $\hat{R} \to K$ for unconfounded propensity learning. A gray node indicates that the variable is observable. Note that when $C$ is observed, there exists an association between $R$ and $E$; therefore $X$ acts as a confounder via $X \to \hat{R} \to K \to E$ and $X \to R \to C \leftarrow E$.**

# 3 CAUSAL ANALYSIS ON PROPENSITY OVERESTIMATION

## 3.1 Causal View of ULTR

To illustrate propensity overestimation in ULTR, we first scrutinize its causal relations and present a causal graph, as shown in Fig. 1a, which consists of six variables $\{X, R, \hat{R}, K, E, C\}$. Note that we use the capital letter (*e.g.,* $R$) and lowercase letter (*e.g.,* $r$) to represent a variable and its particular value, respectively.

- $X \to R$. We denote $X$ as the query-document feature representation and $R$ as the real query-document relevance. This edge shows that there exists a mapping from query-document representations to their relevance, which is goal of unbiased learning-to-rank.
- $X \to \hat{R}$. We denote $\hat{R}$ as the estimated relevance score by a logging policy. This edge shows that the feature representation determines the estimated relevance score, given the logging policy.
- $\hat{R} \to K$. We denote $K$ as the ranking position. As the logging policy presents a list of documents in descending order by the estimated relevance scores, without losing generality, we assume the estimated relevance score of the document decides its ranked position, and ignore the comparison with other documents.
- $K \to E$. We denote $E$ as the examination. According to the position-based bias assumption, the ranked position affects the probability that a document is examined by a user.
- $(R, E) \to C$. This edge follows the examination hypothesis [33] that a user would only click a document when it is observed by the user and considered relevant to the user's need.

According to the causal theory [31], relevance feature $X$ is a *confounder* as $K \leftarrow X \to C$, which leads to a spurious correlation in formulating the propensity model.

## 3.2 Analysis of Propensity Overestimation

According to the position bias assumption that examination only depends on position, existing AutoULTR methods naturally model the propensity estimator via the conditional probability $P(E|K = k)$ from implicit feedback. However, due to the confounding of relevance, the conditional probability $P(E|K = k)$ suffers from spurious correlation, which leads to propensity overestimation. Formally, given $K = k$, we derive the conditional probability $P(E|K = k)$, which corresponds to the propensity estimator in Eq. 3b when the algorithms converge, as:

$$\underbrace{P(E|K=k)}_{\text{AutoULTR propensity estimand}} = \sum_{\mathbf{x}} P(E, \mathbf{x}|k) \tag{5a}$$

$$= \sum_{\mathbf{x}} P(E|\mathbf{x}, k)P(\mathbf{x}|k), \tag{5b}$$

$$= P(E|\mathbf{x}_k, k)P(\mathbf{x}_k|k) \tag{5c}$$

$$\propto \underbrace{P(E|\mathbf{x}_k, k)P(\mathbf{x}_k)}_{\text{causal}} \cdot \underbrace{P(k|\mathbf{x}_k)}_{\text{confounding}}, \tag{5d}$$

where Eq. 5a is the definition of the law of total probability. Eq. 5b and Eq. 5d follow Bayes' rule. In Eq. 5c, $\mathbf{x}$ is known if rank position $K = k$ is given, thus $P(\mathbf{x}|k)$ is 1 if and only if $\mathbf{x}$ takes the $k$th document $\mathbf{x}_k$ in the rank list $\pi_q$; otherwise $P(\mathbf{x}|k) = 0$. In Eq. 5d, the proportion operation $\propto$ does not affect the effectiveness of the proposal. Please refer to Section 4.2.2 for more details.

From Eq. 5d, we can see the propensity estimand $P(E|K = k)$ in existing AutoULTR methods consists of two parts: (1) $P(E|\mathbf{x}_k, k)P(\mathbf{x}_k)$ that contributes to a *causal effect* between the position and examination, *i.e.,* the desired propensity, which will be illustrated in Eq. 6; (2) $P(k|\mathbf{x}_k)$ that contributes to a *confounding* effect. Remarkably, $P(k|\mathbf{x}_k)$ fundamentally changes the propensity estimand $P(E|K = k)$, especially when users' clicks are collected based on an industry-deployed (strong) logging policy. Based on Eq. 5d, we can see how logging policies affect the estimated propensity.

Theorem 3.1. *The conditional propensity estimator $P(E|K = k)$ converges to an overestimated model under a strong logging policy for top-ranking positions.*

Proof. Let $P^s(E|K = k)$ and $P^w(E|K = k)$ be the estimated propensity under strong and weak logging policies, respectively. Considering $k$ as a top-ranking position, a relevant document is more likely to appear at this position under a strong logging policy compared to a weak one, which is expressed as $P^s(k|\mathbf{x}_k) > P^w(k|\mathbf{x}_k)$. Consequently, it follows that $P^s(E|K = k) > P^w(E|K = k)$. In addition, let $P^r(E|K = k)$ and $P^*(E|K = k)$ be the estimated propensity when examination and relevance are independent, *i.e.,* one cannot infer the relevance from positions (random policy), and the real propensity (ground truth), respectively. According to the justification in Ai et al. [2], when examination and relevance are independent, $P^r(E|K = k)$ will converge to $P^*(E|K = k)$. Since a weak policy is better than a random policy, we have $P^w(k|\mathbf{x}_k) > P^r(k|\mathbf{x}_k)$. Thus, we can conclude that $P^s(E|K = k) > P^*(E|K = k)$, which indicates that the propensity estimator under a strong logging policy will converge to an *overestimated model* for top-ranking positions. □

**Relation with weak policies in existing ULTR methods**. The confounding effect is prevalent across all logging policies, but its effect is unintentionally concealed in the existing ULTR setting, where clicks are collected under a weak logging policy. *It means $P(k|\mathbf{x}_k)$ is nearly identical for all documents, and it implicitly satisfies the independence between examination and relevance.* Although this effect may appear minor under a weak logging policy, it should not be disregarded. Experimental results in Section 5.4 verify that our solution could obtain a better-performing ranking model even when the backdoor path leads to a minor confounding effect under a weak logging policy.

# 4 METHODOLOGY

In this section, we detail our solution, which is referred to as Unconfounded Propensity Estimation (UPE). We first resort to the *backdoor adjustment* [31, 44], which enables the causal effect estimation without causal intervention. Then we propose the Logging-Policy-aware Propensity (LPP) model, and its novel learning strategy, namely logging-policy-aware confounding effect learning and joint propensity learning. Thereafter, we conduct the backdoor adjustment on the LPP model for unconfounded propensity inference.

## 4.1 Backdoor Adjustment

According to the theory of *backdoor adjustment* [31], our unconfounded propensity estimand is formulated as $P(E|do(K))$, where $do(K)$ can be intuitively seen as cutting the edge $\hat{R} \rightarrow K$ in $G$, as illustrated in Fig. 1b. We then derive the specific expression of the backdoor adjustment and demonstrate its difference from the conditional probability, *i.e.,* $P(E|K)$, as:

$$P(E|do(K=k)) = P_{G'}(E|do(K=k)) \tag{6a}$$

$$= \sum_{\mathbf{x}} P_{G'}(E|\mathbf{x}, do(K=k)) P_{G'}(\mathbf{x}|do(K=k)) \tag{6b}$$

$$= \sum_{\mathbf{x}} P_{G'}(E|\mathbf{x}, do(K=k)) P_{G'}(\mathbf{x}) \tag{6c}$$

$$= \sum_{\mathbf{x}} P(E|\mathbf{x}, k) P(\mathbf{x}), \tag{6d}$$

where $P_{G'}$ denotes the probability function evaluated on intervened causal graph $G'$ in Fig. 1b. In particular, Eq. 6a is because of the *backdoor criterion* [31] as the only backdoor path $K \leftarrow X \rightarrow C$ has been blocked by $do(K)$; Eq. 6b is obtained by Bayes' rule; since $K$ and $\mathbf{x}$ are independent in $G'$, $P_{G'}(\mathbf{x}) = P_{G'}(\mathbf{x}|do(K=k))$ in Eq. 6c; in Eq. 6d, $P(E|\mathbf{x}, do(K=k)) = P_{G'}(E|\mathbf{x}, do(K=k))$ because the causal relation/association $K \rightarrow E$ and $X \rightarrow E$ are not changed when cutting off $\hat{R} \rightarrow K$, and $P(\mathbf{x}) = P_{G'}(\mathbf{x})$ has the same prior on the two graphs. To this end, we have demonstrated that the causal component in Eq. 5d exactly contributes to the causal effect, which is what is needed for the desired unconfounded propensity.

As compared to Eq 5, our estimand $P(E|do(K=k))$ estimates the examination probability for position $k$ with consideration of every possible value of document $\mathbf{x}$ in the rank list $\pi_q$ subject to the prior $P(\mathbf{x})$ in Eq. 6d, rather than $P(\mathbf{x}_k|k)$ in Eq. 5d. Therefore, the documents with high relevance will *not* receive high examination probability purely because of a higher rank position $k$ via $P(\mathbf{x}_k|k)$, which addresses propensity overestimation.

Notably, on the basis of causal theory, our unconfounded estimand learns the causal effect between the position and examination; therefore, its unbiasedness can be guaranteed at the presence of the unconfounded propensity model and learned by an in-principle unbiased algorithm, *e.g.,* inverse propensity weighting (IPW).

Theoretically, the sample space of $\mathbf{X}$ is infinite, which makes the calculation of $P(E|do(K=k))$ in Eq. 6d intractable. Therefore, we further devise an approximation of backdoor adjustment by empirically averaging over the training samples as:

$$P(E|do(K=k)) = \frac{1}{|Q_b| \cdot |\pi_q|} \sum_{q \in Q_b} \sum_{d_k \in \pi_q} P(E|\mathbf{x}_k, k), \tag{7}$$

where $d_k$ is the document displayed in position $k$ with relevance feature $\mathbf{x}_k$, $Q_b$ is a batch of queries in the raw click data, $\pi_q$ is the ranked list for query $q$, $|Q_b|$ and $|\pi_q|$ denote the number of queries within the batch and number of documents in the ranked list, respectively. In the following section, we will introduce the instantiation of $P(E|\mathbf{x}_k, k)$.

By adopting the backdoor adjustment, it means the propensity models need to take the relevance and position as input, *i.e.,* $P(E|\mathbf{x}_k, k)$. At first glance, readers may confuse it with contextual position bias [9, 45, 54]. The key difference is that our work **does not change the position-bias assumption that examination only depends on the position**; whereas the contextual position bias assumes that examination depends on both position and query context. We build our propensity model with $P(E|\mathbf{x}_k, K)$ because the backdoor adjustment needs to stratify the confounder. In this way, it enables us to address propensity overestimation caused by the relevance confounder as illustrated in Eq. 6 and finalizes the unconfounded propensity estimand $P(E|do(K=k))$.

## 4.2 Logging-Policy-aware Propensity Model

To facilitate the backdoor adjustment in Eq. 7, we need to instantiate a propensity model as $P(E|\mathbf{x}_k, k)$. However, it is difficult to extract separate ranking models and propensity models when they share common input (*i.e.,* document) and target (*i.e.,* user clicks) . Please refer to Section 5.5.2 for details. Given $P(E|K=k)$ provided by existing ULTR methods, inspired by the derivation in Eq. 5 and Eq. 6d, our solution is to: 1) estimate the confounding effect as $P(k|\mathbf{x}_k)$; 2) remove its effect from the propensity estimand to obtain $P(E|X, K)$; 3) obtain unconfounded propensity $P(E|do(K=k))$ via backdoor adjustment for AutoULTR. Therefore, we propose a Logging-Policy-aware Propensity (LPP) model (as shown in Fig. 2), and its novel two-step optimization strategy: logging-policy-aware confounding effect learning and joint propensity learning. In particular, we discuss the design choice of LPP model as follows.

**Unobservable confounder variable**. Existing deconfounding methods [41, 49, 50, 52] usually represent the joint effect by a simple multiplication among each decoupled factor. Unfortunately, it is not feasible in ULTR because neither $P(E|K=k)$ nor $P(E|X=\mathbf{x})$ is observable. In LPP, we leverage a shared feed-forward network to capture the joint effect of relevance features $X$ and position $K$.

*4.2.1 Logging-policy-aware Confounding Effect Learning.* To estimate the confounding effect as $P(k|\mathbf{x}_k)$ in Eq. 5d, we propose logging-policy-aware confounding effect learning, which learns a mapping from raw document features to logging policy scores $\hat{R}$. Careful readers may notice that the target is logging policy scores $\hat{R}$ instead of rank positions $K$. We will compare the effectiveness of different fitting targets in Section 5.5.1.

Directly learning the mapping in a point-wise way, however, is problematic. Since the logging policies, *i.e.,* ranking models, are usually optimized pairwise or listwise, the logging policy scores may follow different distributions under different queries. This issue would restrict the expressive ability of neural encoders.

Therefore, we propose to learn the mapping in a list-wise way, which is invariant to different score distributions under different queries. Formally, given a query $q$ and its associated documents $\pi_q = [d_1, \cdots, d_N]$, each feature vector for query-document pair $\mathbf{x}_i$ is transformed into an $n$-dimensional latent feature representation:

$$\mathbf{m}_{d_i} = \text{Encoder}_D(q, d_i), \quad \text{where } \mathbf{m}_{d_i} \in \mathbb{R}^n. \tag{8}$$

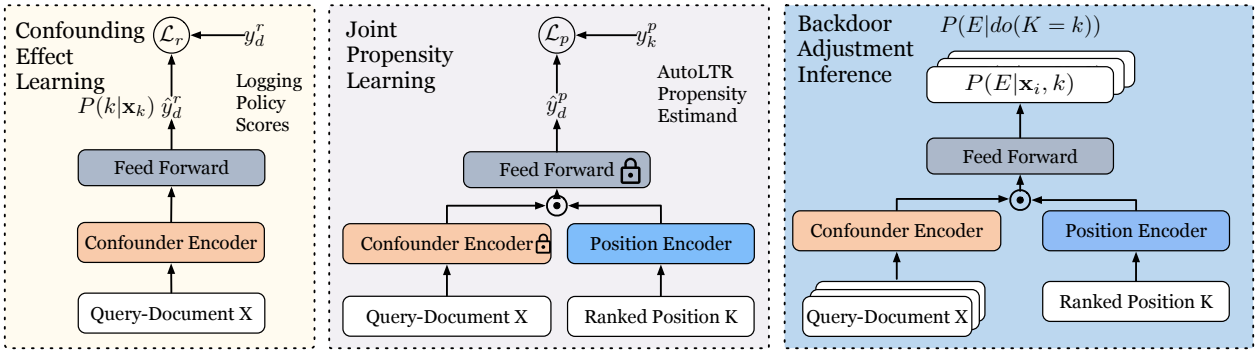



**Figure 2: The workflow of the proposed Logging-Policy-aware Propensity Model for UPE. Its optimization strategy consists of two steps: (1) confounding effect learning that estimates the confounding effect of relevance; and, (2) joint propensity learning that separates the propensity and ranking models with user clicks. Afterwards, we conduct backdoor adjustment inference over the LPP model to obtain the unconfounded propensity $P(E|do(K=k))$.**

$\text{Encoder}_D$ is a confounder encoder that projects the raw relevance feature vector $\mathbf{x}_i$ to latent representations $\mathbf{m}_{d_i}$. Afterward, the latent feature vector is passed into a feed-forward network, and generates the predictive scores as:

$$\hat{y}^r_{d_i} = \text{FFN}(\mathbf{m}_{d_i}), \quad \text{where } \hat{y}^r_{d_i} \in \mathbb{R}, \tag{9}$$

FFN is a point-wise feed-forward network that projects the latent representation to real-value scalar as prediction scores $P(K|\mathbf{x}_i)$.

Let $y^r_{d_i}$ be the logging policy score of document $d_i$ in query $q$ that is observed in the logging policy; we optimize in the list-wise paradigm through an attention rank loss function [1]:

$$\mathcal{L}_r(g_{\text{pt}}|\pi_q) = -\sum_{d_i \in \pi_q} \frac{\exp(y^r_{d_i})}{\sum_{z\in\pi_q}\exp(y^r_z)} \cdot \log \frac{\exp(\hat{y}^r_{d_i})}{\sum_{z\in\pi_q}\exp(\hat{y}^r_z)}, \tag{10}$$

where $g_{\text{pt}}$ denotes the parameters included in the confounder encoder $\text{Encoder}_D$ and feed-forward network FFN. In summary, the logging-policy-aware confounding effect learning captures confounding component $P(k|\mathbf{x}_k)$ in Eq. 5d.

*4.2.2 Joint Propensity Learning.* Given the confounded propensity estimand $P(E|K=k)$ by existing AutoULTR, we now present how to remove the confounding effect and obtain $P(E|X,K)$. In particular, we propose to learn the joint effect from position and relevance features to the propensity estimand from existing AutoULTR, but fixing the confounder encoder and feedforward network and solely tuning the position-related parameters, as marked locked in Fig. 2. This keeps the relevance confounding effect unchanged; therefore, the position encoder is able to correctly capture the influence of position over examination.

Formally, we design a position embedding function $\text{Encoder}_P$. It encodes the position of a document that is ranked in position $k$ to a vector with the same dimension as $\mathbf{m}_{d_i}$:

$$\mathbf{p}_k = \text{Encoder}_P(\text{rank}(d_k)), \quad \text{where } \mathbf{p}_k \in \mathbb{R}^n, \tag{11}$$

and $\text{rank}(d_k)$ is the ranked position for document $d_k$ generated by the logging policy. Afterward, we obtain predictions for *confounder* propensity scores $P(E|K=k)$ through the *frozen* FFN:

$$\hat{y}^p_k = \text{FFN}(\mathbf{m}_{d_k} + \mathbf{p}_k). \tag{12}$$

Let $y^p_k$ be the confounded propensity score by existing AutoULTR algorithms, such as Eq. 3b. We solely update the position embedding function $\text{Encoder}_P$ via the attention rank loss function, which is formally defined as,

$$\mathcal{L}_p(g_{\text{pos}}|\pi_q) = -\sum_{d_k \in \pi_q} \frac{\exp(y^p_k)}{\sum_{z\in\pi_q}\exp(y^p_z)} \cdot \log \frac{\exp(\hat{y}^p_k)}{\sum_{z\in\pi_q}\exp(\hat{y}^p_z)}, \tag{13}$$

where $g_{\text{pos}}$ denotes the parameters of position embedding $\text{Encoder}_P$. To this end, we obtain $P(E|X,K)$ for backdoor adjustment.

As shown in Eq. 10 and Eq. 13, the use of the softmax function assumes that the relevance probabilities and examination probabilities on different documents in $\pi_q$ will sum to 1, which is not true in practice. This, however, does not hurt the effectiveness of model training. In fact, the predicted values of $\hat{y}^p_k$ have a minor effect on the unconfounded propensity learning as long as their relative proportions are correct. For the same reason, we show normalized propensity against position 10, which reflects the relative proportion, throughout this paper. Such a technique has been widely applied in existing work, and its effectiveness has been extensively verified in prior work [2, 4, 27].

*4.2.3 Backdoor Adjustment for Unconfounded Propensity Inference.* Based on the aforementioned LPP model, we are ready to obtain the unconfounded propensity for each position. Formally, for a query-document pair with feature vector $x_k$ on position $k$ in the rank list, we first compute the propensity under relevance confounder:

$$P(E|\mathbf{x}_k, k) = \text{FFN}\big(\text{Encoder}_D(\mathbf{x}_k) + \text{Encoder}_P(k)\big). \tag{14}$$

Afterward, we leverage backdoor adjustment approximation in Eq. 7 to obtain unconfounded propensity $P(E|do(K=k))$, which learns the causal effect between positions and examinations from raw user clicks. Finally, we integrate it with an existing AutoULTR algorithm, such as Eq.3a, to obtain unbiased ranking models.

## 5 EXPERIMENTS

We conduct extensive experiments to demonstrate effectiveness. In particular, we analyze two learning paradigms that are proposed in Ai et al. [3]. The first, referred to as the deterministic online paradigm (*OnD*), is an online setting in which the displayed ranked list $\pi_q$ is created by the current logging policy, and the ranking model is updated based on $c_{\pi_q}$ collected online. The second, which is referred to as the offline paradigm (*Off*), is a classic setting where

we obtain a logging policy, and then both the displayed ranked list $\pi_q$ and the clicks on it $c_{\pi_q}$ are fixed and observed in advance.

## 5.1 Experimental Settings

*Datasets.* We conduct empirical studies on two of the largest publicly available LTR datasets:

- **Yahoo! LETOR**[1] comes from the Learn to Rank Challenge version 2.0 (Set 1), and is one of the largest benchmarks of unbiased learning to rank [2, 18]. It consists of 29,921 queries and 710K documents. Each query-document pair is represented by a 700-D feature vector and annotated with 5-level relevance labels [5].
- **Istella-S**[2] contains 33K queries and 3,408K documents (roughly 103 documents per query) sampled from a commercial Italian search engine. Each query-document pair is represented by 220 features and annotated with 5-level relevance judgments [26].

We follow the predefined data split of training, validation, and testing of all datasets. The Yahoo! set splits the queries arbitrarily and uses 19,944 for training, 2,994 for validation, and 6,983 for testing. The Istella-S dataset has been divided into train, validation, and test sets according to a $60\% - 20\% - 20\%$ scheme.

*Click Simulation.* We generate synthesized clicks with a two-step process as in Joachims et al. [24] and Ai et al. [2]. First, we generate the initial ranked list $\pi_q$ for each query $q$ based on learning paradigms, *i.e., OnD* and *Off*. Then, we simulate the user browsing process based on PBM [24] and sample clicks from the initial ranked list by utilizing the simulation model. The PBM models user browsing behavior based on the assumption that the bias of a document only depends on its position $P(e_{d_i}) = \rho_i^{\eta}$, where $\rho_i$ represents position bias at position $i$ and $\eta \in [0, +\infty]$ is a parameter controlling the degree of position bias. The position bias $\rho_i$ is obtained from an eye-tracking experiment in Joachims et al. [24] and the parameter $\eta$ is set as 1 by default. Following the methodology proposed by Chapelle et al. [6], we sample clicks with $\Pr(r_{d_i} = 1|\pi_q) = \epsilon + (1-\epsilon)\frac{2^y-1}{2^{y_{\max}}-1}$, where $y \in [0, y_{\max}]$ is the relevance label of the document $d_i$, and $y_{\max}$ is the maximum value of $y$, which is 4 on both datasets. $\epsilon$ is the noise level, which models click noise such that irrelevant documents (*i.e.,* $y = 0$) have non-zero probability to be perceived as relevant and clicked. We fix $\epsilon = 0.1$ as the default setting.

*Baselines.* To demonstrate the effectiveness of our proposed method, we compare it with baseline methods which are widely used in ULTR problems that address positional bias. **1) DLA**: The Dual Learning Algorithm [2] treats the problem of unbiased learning to rank and unbiased propensity estimation as a dual problem, such that they can be optimized simultaneously. This is the state-of-the art ULTR learning framework. **2) Vectorization**: Vectorization [8] expands the examination hypothesis to a vector-based one, which formulates the click probability as a dot product of two vector functions instead of two scalar functions. **3) REM**: The Regression EM model [43] uses an EM framework to estimate the propensity scores and ranking scores. **4) PairD**: The Pairwise Debiasing (PairD) Model [18] uses inverse propensity weighting for pairwise learning to rank. **5) IPW-Random**: Inverse Propensity Weighting [24, 42] uses result randomization to estimate the examination probabilities against positions and optimizes the ranking models accordingly. Its performance can be considered as an upper bound for learning-to-rank with implicit user feedback. **6) Naive**: This model just uses the raw click data to train the ranking model, without any correction. Its performance can be considered as a lower bound for the ranking model. **7) GradRev** and **Drop**: These two methods [53] mitigate the negative confounding effects by reducing the functionality of propensity learning.

[1]https://webscope.sandbox.yahoo.com/
[2]http://quickrank.isti.cnr.it/istella-dataset/

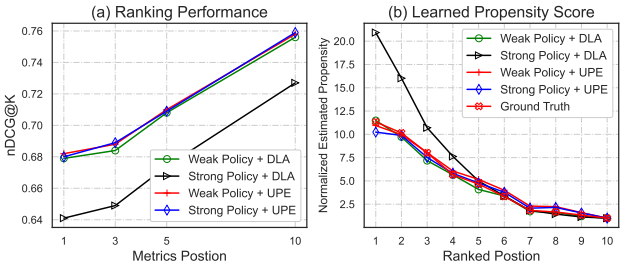



**Figure 3: Ranking performance and normalized estimated propensity ($\frac{\text{propensity@K}}{\text{propensity@10}}$) on different logging policies on Yahoo! LETOR.**

*Experimental Protocols.* We implement UPE and use the baselines in ULTRA [37] to conduct our experiments. In particular, UPE is integrated with DLA, as it is the state-of-the-art automatic ULTR algorithm. For each query, only the top $N = 10$ documents are assumed to be displayed to the users. For both datasets, all models are trained with synthetic clicks. Following the setting in [2], the click sessions for training are generated on the fly. We fix the batch size to 256 and train each model for 10K steps. We use the AdaGrad optimizer [16] and tune learning rates from 0.01 to 0.05 for each unbiased learning-to-rank algorithm on the validation dataset.

All reported results are produced using a neural network with three hidden layers with size [512, 256, 128] respectively, with the ELU [12] activation function and 0.1 dropout [36] as prior work [1, 8]. The confounder encoder in LPP model is configured with a single MLP layer. To evaluate all methods, we use the normalized Discounted Cumulative Gain (nDCG) [19] and the Expected Reciprocal Rank (ERR) [6]. For both metrics, we report the results at ranks 1, 3, 5, and 10 to show the performance of models on different positions. Following Ai et al. [2], statistical differences are computed based on the Fisher randomization test [35] with $p \leq 0.05$.

## 5.2 Justification for Propensity Overestimation

To justify the propensity overestimation problem, we demonstrate the normalized estimated propensities against ranking positions and ranking performance. Fig. 3(a) investigates the ranking performance of DLA [2], a state-of-the-art AutoULTR method, on weak and strong logging policies, respectively. We can see that DLA's ranking performance suffers from a significant drop on a strong logging policy. Furthermore, Fig. 3(b) shows the learned propensity score and we can indeed observe a propensity overestimation problem in which normalized estimated propensities for top positions are much larger than their actual values. In summary, Fig. 3 confirms the propensity overestimation problem, and it is detrimental to ranking models' optimization. Therefore, propensity overestimation needs to be addressed for industrial LTR systems, in which the logging policies are no longer weak.

**Table 1: Overall performance comparison between UPE and the baselines on Yahoo! and Istella-S datasets with deterministic online learning (*OnD*). "$*$" indicates statistically significant improvement over the best baseline without result randomization.**

| Methods | Yahoo! LETOR | | | | | | | | Istella-S | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NDCG@K | | | | ERR@K | | | | NDCG@K | | | | ERR@K | | | |
| | K = 1 | K = 3 | K = 5 | K = 10 | K = 1 | K = 3 | K = 5 | K = 10 | K = 1 | K = 3 | K = 5 | K = 10 | K = 1 | K = 3 | K = 5 | K = 10 |
| IPW-Random | 0.693 | 0.702 | 0.722 | 0.767 | 0.352 | 0.432 | 0.452 | 0.469 | 0.667 | 0.637 | 0.660 | 0.720 | 0.596 | 0.705 | 0.722 | 0.728 |
| UPE | **0.691**$^*$ | **0.703**$^*$ | **0.723**$^*$ | **0.768**$^*$ | **0.354**$^*$ | **0.432**$^*$ | **0.454**$^*$ | **0.469**$^*$ | **0.666**$^*$ | **0.633**$^*$ | **0.657**$^*$ | **0.718**$^*$ | **0.595**$^*$ | **0.704**$^*$ | **0.720**$^*$ | **0.727**$^*$ |
| GradRev [53] | 0.676 | 0.685 | 0.706 | 0.762 | 0.347 | 0.427 | 0.449 | 0.463 | 0.663 | 0.630 | 0.653 | 0.712 | 0.593 | 0.700 | 0.717 | 0.724 |
| Drop [53] | 0.675 | 0.683 | 0.708 | 0.759 | 0.346 | 0.428 | 0.450 | 0.464 | 0.663 | 0.631 | 0.654 | 0.714 | 0.593 | 0.701 | 0.718 | 0.725 |
| MULTR [27] | 0.672 | 0.680 | 0.702 | 0.755 | 0.345 | 0.423 | 0.448 | 0.462 | 0.662 | 0.629 | 0.653 | 0.712 | 0.592 | 0.701 | 0.717 | 0.725 |
| Vectorization [8] | 0.670 | 0.678 | 0.702 | 0.753 | 0.343 | 0.423 | 0.446 | 0.460 | 0.663 | 0.630 | 0.653 | 0.711 | 0.593 | 0.701 | 0.716 | 0.724 |
| DLA [2] | 0.671 | 0.677 | 0.701 | 0.750 | 0.345 | 0.423 | 0.445 | 0.460 | 0.663 | 0.629 | 0.653 | 0.712 | 0.592 | 0.701 | 0.717 | 0.724 |
| REM [43] | 0.674 | 0.678 | 0.699 | 0.747 | 0.349 | 0.425 | 0.446 | 0.462 | 0.642 | 0.611 | 0.631 | 0.690 | 0.574 | 0.684 | 0.702 | 0.709 |
| PairD [18] | 0.602 | 0.614 | 0.642 | 0.700 | 0.319 | 0.394 | 0.416 | 0.433 | 0.609 | 0.569 | 0.593 | 0.653 | 0.545 | 0.656 | 0.675 | 0.684 |
| Naive [24, 42] | 0.634 | 0.644 | 0.670 | 0.723 | 0.334 | 0.409 | 0.431 | 0.447 | 0.639 | 0.601 | 0.624 | 0.683 | 0.571 | 0.681 | 0.699 | 0.706 |

## 5.3 Dynamic LTR Simulation

Industrial ranking systems usually employ strong logging policies, and they undergo periodic updates, wherein sufficient implicit feedback are gathered over a few months, allowing the online logging policies to be updated with the latest ranking policy [47, 48]. To simulate this process, we perform deterministic online simulation (*OnD*) and update the logging policy after every 2.5K steps with the most recent ranking model. The experimental results are summarized in Table 1, and we have the following observations.

**UPE significantly outperforms all ULTR baseline methods without result randomization on both datasets**. This demonstrates the effectiveness of UPE for real-world LTR systems. To provide a more insightful understanding for the benefit of UPE, we also illustrate the learning curve of estimated propensity in Fig. 4. In particular, we present the normalized estimated propensity against position 1 on UPE, compared with DLA, the state-of-the-art automatic ULTR framework. We select position 1 for illustration because it suffers from propensity overestimation most (as seen in Fig. 3). The "ground truth" is computed by $\frac{\rho_1}{\rho_{10}}$, where $\rho_i$ is the position bias defined in Section 5.1. We can see that DLA suffers propensity overestimation, with its learning curves deviating from the ground truth in dynamic training scenarios. This underscores the limitations of existing ULTR methods in leveraging implicit feedback for the continuous improvement of LTR systems, due to propensity overestimation. In contrast, UPE's propensity estimation remains stable against periodic logging policy updates, with its curves closely matching the ground truth. As more user clicks are gathered, the ranking performance improves, underscoring UPE's efficacy in refining the training of the ranking model.

**UPE effectively mitigates the confounding effect by its two-step optimization, without manual hyper-parameter tuning**. GradRev and Drop [53] are proposed to address the confounding effect as well. However, their efficacy depends on the hyper-parameter tuning to reduce the functional strength of position in propensity learning. In other words, if the hyper-parameter tuning under-reduces or over-reduces the confounding effect in the propensity learning, the ranker could still become suboptimal. What is worse, when the ranking systems are periodically updated, the hyper-parameters cannot adaptively change. Thus, this reliance undermines the efficacy of GradRev and Drop. In contrast, UPE *learns to estimate* the confounding effect by its two-step optimization for any logging policy. This feature is particularly useful for ULTR in industrial ranking systems that are periodically updated, but is absent in prior work [53].

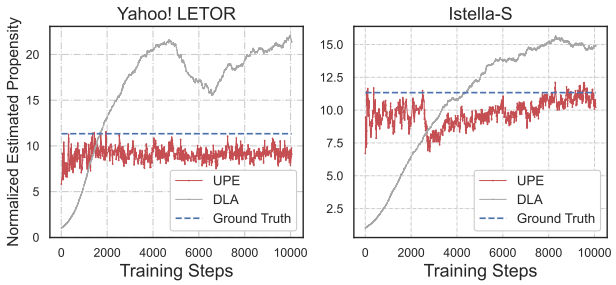



**Figure 4: The learning curve of normalized estimated propensity for position 1 with deterministic online learning (*OnD*).**

**The propensity overestimation is not caused by high variance of IPW**. It is well recognized that IPW suffers from high variance of propensity estimation. However, we can see the SOTA method, *i.e.,* MULTR [27], which leverages the counterfactual data augmentation and doubly robust approach to alleviate high variance, cannot address the propensity overestimation either. This is because the efficacy of MULTR is still built upon the assumption that examination and relevance are independent, which overlooks the confounding effect brought on by the logging polices.

**UPE can achieve similar performance to IPW-Random**. In comparison with IPW-Random, UPE has a giant benefit: it can accurately estimate the examination probability for each position, yet do so without result randomization, which would hurt users' search experience. Notably, it may seem counter-intuitive that the proposed UPE outperforms the ideal IPW-Random (theoretical upper bound), which utilizes result randomization to estimate the examination probability against positions. This unexpected result arises because, although result randomization leads to theoretical optimality in IPW-Random, it introduces large variance in practice. The "unexpected" outperformance observed in Tables 1 and 2 can mostly be attributed to these variances.

**Table 2: Overall performance comparison between UPE and the baselines on Yahoo! and Istella-S datasets with offline learning (*Off*). "$*$" indicates statistically significant improvement over the best baseline without result randomization.**

| Methods | Yahoo! LETOR | | | | | | | | Istella-S | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NDCG@K | | | | ERR@K | | | | NDCG@K | | | | ERR@K | | | |
| | K = 1 | K = 3 | K = 5 | K = 10 | K = 1 | K = 3 | K = 5 | K = 10 | K = 1 | K = 3 | K = 5 | K = 10 | K = 1 | K = 3 | K = 5 | K = 10 |
| IPW-Random | 0.684 | 0.693 | 0.715 | 0.763 | 0.352 | 0.430 | 0.451 | 0.467 | 0.676 | 0.640 | 0.664 | 0.723 | 0.604 | 0.710 | 0.727 | 0.733 |
| UPE | **0.687**$^*$ | **0.692**$^*$ | **0.715**$^*$ | **0.762**$^*$ | **0.351** | **0.429** | **0.451** | **0.466** | **0.669**$^*$ | **0.637**$^*$ | **0.659**$^*$ | **0.720**$^*$ | **0.598**$^*$ | **0.706**$^*$ | **0.722**$^*$ | **0.729**$^*$ |
| GradRev [53] | 0.682 | 0.689 | 0.711 | 0.759 | 0.350 | 0.428 | 0.449 | 0.465 | 0.663 | 0.630 | 0.651 | 0.712 | 0.594 | 0.701 | 0.719 | 0.723 |
| Drop [53] | 0.684 | 0.690 | 0.711 | 0.758 | 0.351 | 0.427 | 0.450 | 0.465 | 0.663 | 0.632 | 0.652 | 0.714 | 0.594 | 0.700 | 0.718 | 0.724 |
| MULTR [27] | 0.681 | 0.689 | 0.710 | 0.759 | 0.350 | 0.428 | 0.449 | 0.465 | 0.663 | 0.630 | 0.653 | 0.712 | 0.593 | 0.701 | 0.718 | 0.723 |
| Vectorization [8] | 0.681 | 0.689 | 0.711 | 0.759 | 0.350 | 0.428 | 0.449 | 0.465 | 0.663 | 0.630 | 0.652 | 0.714 | 0.593 | 0.699 | 0.718 | 0.722 |
| DLA [2] | 0.682 | 0.688 | 0.710 | 0.758 | 0.350 | 0.427 | 0.449 | 0.464 | 0.662 | 0.629 | 0.653 | 0.713 | 0.592 | 0.699 | 0.717 | 0.723 |
| REM [43] | 0.655 | 0.662 | 0.685 | 0.735 | 0.342 | 0.418 | 0.440 | 0.456 | 0.609 | 0.570 | 0.589 | 0.637 | 0.545 | 0.653 | 0.672 | 0.681 |
| PairD [18] | 0.659 | 0.664 | 0.688 | 0.739 | 0.337 | 0.415 | 0.438 | 0.453 | 0.609 | 0.590 | 0.621 | 0.687 | 0.543 | 0.664 | 0.684 | 0.692 |
| Naive [24, 42] | 0.662 | 0.666 | 0.688 | 0.739 | 0.338 | 0.416 | 0.438 | 0.454 | 0.630 | 0.607 | 0.634 | 0.699 | 0.562 | 0.679 | 0.697 | 0.705 |

## 5.4 Offline Simulation

To investigate the generalizability of UPE, we conduct experiments on the widely used ULTR setting, *i.e.,* offline learning paradigm (*Off*) as in Joachims et al. [24], Hu et al. [18] and Ai et al. [2]. Unlike the online paradigm, the ranked lists are generated by a Rank SVM model [21] that is trained with 1% of the training data with real relevance judgements, *i.e.,* a weak logging policy.

The experimental results are summarized in Table 2. We can see that UPE also outperforms all baseline methods without result randomization, and UPE achieves similar performance as IPW-Random on the offline paradigm *Off*. Notably, UPE outperforms the best baseline methods without result randomization in most metrics. This indicates that UPE can effectively address propensity overestimation even when there exists a minor confounding effect by relevance confounder $X$ on a weak logging policy, and more importantly this confounding effect is indeed non-negligible.

To validate the effectiveness of UPE in propensity estimation on both weak and strong logging policies, we also demonstrated the overall distribution of normalized estimated propensity against positions in Fig. 3. The results demonstrate the high generalizability of UPE, which can consistently obtain unconfounded propensity estimation on both strong and weak policies. Its ranking performance is also shown in Fig. 3, where UPE can consistently achieve SOTA ranking performance on both weak and strong policies.

## 5.5 Ablation Study

Our ablation studies investigate the influence of variant designs of LPP to the performance of unbiased learning to rank.

*5.5.1 Different fitting targets $P(\hat{R}|\mathbf{x})$ and $P(K|\mathbf{x})$.* We compare different fitting targets for confounding effect learning in LPP optimization. In particular, to instantiate $P(K|\mathbf{x})$, we transform the ranked positions as relevance labels for ranking optimization in Eq. 10 as in Zhang et al. [51]:

$$\textbf{MRR-UPE}: \quad \text{MRR-LPP@}K = \frac{1}{K} \tag{15a}$$

$$\textbf{DCG-UPE}: \quad \text{DCG-LPP@}K = \frac{1}{\log_2(K+1)}. \tag{15b}$$

The experimental results are summarized in Table 3. We can see that UPE significantly outperforms the two variants, *i.e.,* MRR-UPE and DCG-UPE. This observation confirms that the logging policy scores are more informative than the ranked positions, because logging policy scores not only provide the order of the documents, *i.e.,* ranked positions, but also their relevance strengths.

**Table 3: The performance of different fitting targets in LLP optimization on Yahoo! LETOR with deterministic online learning (*OnD*). "$*$" indicates statistically significant improvement over the best baseline.**

| Methods | NDCG@K | | | | ERR@K | | | |
|---|---|---|---|---|---|---|---|---|
| | K = 1 | K = 3 | K = 5 | K = 10 | K = 1 | K = 3 | K = 5 | K = 10 |
| UPE | **0.691**$^*$ | **0.703**$^*$ | **0.723**$^*$ | **0.768**$^*$ | **0.354**$^*$ | **0.432**$^*$ | **0.454**$^*$ | **0.469**$^*$ |
| MRR-UPE | 0.688 | 0.694 | 0.715 | 0.762 | 0.352 | 0.429 | 0.451 | 0.466 |
| DCG-UPE | 0.686 | 0.691 | 0.714 | 0.760 | 0.350 | 0.428 | 0.451 | 0.465 |

*5.5.2 The two-step optimization strategy,* i.e., *logging-policy-aware confounding effect learning and joint propensity learning, is indispensable for LPP.* Readers may incorrectly believe that the learning of LPP can be achieved simply by jointly learning a ranking model $P(R|X)$ and a propensity model $P(E|X,K)$ that takes both the relevance feature and position into consideration. Please note that this is not feasible as we have argued that existing AutoULTR methods cannot extract separated ranking model $P(R|X)$ and propensity model $P(E|K,X)$ when they have shared input, *i.e.,* relevance feature, and output, *i.e.,* user click. In this section, we empirically justify this argument, and demonstrate the indispensability of our novel two-step optimization strategy.

We firstly show the logarithm of normalized estimated propensity by $\text{UPE}_N$ at positions 1, 2, 3 and 9 in Fig. 5(a). From step 3000, the estimated propensity by $\text{UPE}_N$ against position 1 has been much smaller than that against position 9, and those at other positions are almost identical to that at position 9. It indicates that $\text{UPE}_N$ fails to learn the examination probability for each position, which should have a larger value for a higher-ranked position. In summary, **jointly modeling the examination with the position and query-document pair naively cannot address the propensity overestimation problem.**

We present the ranking performance curve of $\text{UPE}_N$ in Fig. 5(b). We have two important observations. Firstly, there is a sudden performance decrease at around step 3000. We conjecture that it is because the propensity model has absorbed the majority of the relevance information, which fits the extreme case we have argued in the Section 1. Secondly, starting from step 3000, nDCG@10 of

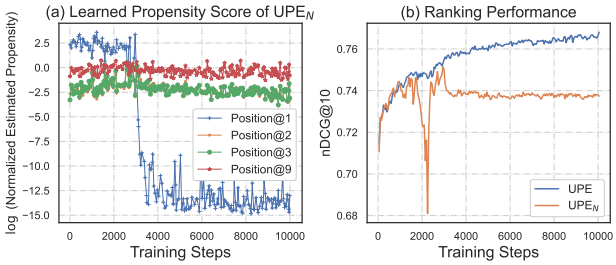



**Figure 5: The learning curve of normalized estimated propensity and ranking performance of UPE and UPE$_N$ on Yahoo! LETOR with deterministic online learning (*OnD*).**

UPE$_N$ does not increase. This means that collecting more click data does not improve the ranking model because the propensity model has failed in separating the impact of relevance and position on propensity. This phenomenon also verifies that an accurate propensity model is indeed necessary for optimizing ranking models.

Unlike UPE$_N$, the proposed logging-policy-aware confounding effect learning and joint propensity learning enable us to obtain an unconfounded propensity through the backdoor adjustment, which correctly estimates the causal effect between propensity and position, as shown in Fig. 4. Moreover, the learning curve of UPE in Fig. 5(b) shows that the ranking performance of our proposal consistently improves when more click data are collected. Therefore, the two-step optimization strategy is necessary for LPP optimization as it enables the separation of ranking and propensity models from raw clicks.

## 6 RELATED WORK

*Unbiased Learning to Rank.* To leverage implicit user feedback for optimizing LTR systems, there are two streams of research for ULTR. One relies on click modeling, which usually maximizes the likelihood of the observed data, models the examination probability and infers accurate relevance feedback from user clicks [6, 11, 13, 30]. However, click models usually require multiple appearances of the same query-document pair for reliable inference [29, 56]; thus they may fall short for tail queries. The other, derived from counterfactual learning, treats bias as a counterfactual factor and debiases clicks through inverse propensity weighting [24, 42]. Among them, AutoULTR methods, which jointly learn user bias models (*i.e.,* propensity models) with unbiased rankers, have received attention due to their superior performance and low deployment cost.

Based on AutoULTR, recent work has investigated various biases under weak policies, including position bias [1, 8, 18, 32, 39, 55], contextual position bias [7, 9, 45, 54], trust bias [38, 39], exploitation bias [48], bias against non-clicked but relevant results [40], and factorizability of the examination hypothesis [8]. Unfortunately, their effectiveness has only been justified under weak logging policies, and cannot be reproduced under a strong policy. However, industrial logging policies are usually strong policies, where the deployed ranking models already have a decent performance. According to our experiment, naively applying existing AutoULTR is not applicable under a strong policy, since it suffers from the propensity overestimation problem. In summary, our work is significantly different from existing AutoULTR methods in that our propensity model, *i.e.,* LPP, addresses propensity overestimation by estimating the relevance confounding caused by relevance features, and then removing it via a backdoor adjustment, which is applicable for both weak and strong policies.

For deconfounding under strong policies, prior work [14, 15] empirically identifies the policy distributional shift for debiasedness evaluation. Zhang et al. [53] recently proposed using observation dropout and gradient reversal to reduce the overestimation of bias tower. Our work diverges significantly by: **1)** providing a rigorous causal analysis for the confounding effect, and a strong theoretical guarantee of unbiasedness; **2)** demonstrating its consistent superiority across both strong and weak logging policies, notably for industrial logging policies, which are updated periodically, without need for hyper-parameter tuning.

*Deconfounding in IR.* Recently, causal-aware methods have thrived in information retrieval. In particular, some efforts have been made to address confounding problems in recommendation systems. Those methods adopt causal inference to analyze the root causes of bias problems [10, 17, 34, 41, 52] and apply backdoor adjustment [31] during the training or inference to address the bias problems. For example, Wang et al. [41] identify the distribution of historical interactions as a confounder for bias amplification. Zhang et al. [52] identify popularity as a confounder that affects both item exposures and user clicks. However, these methods require the confound and effect variables to be observable, while in ULTR the effect – query-document relevance – is unobservable.

There are also a few efforts that address confounding effects without the need to be observable [25, 44]. For example, Liu et al. [25] learn a biased embedding vector with independent biased and unbiased components in the training phase. In testing, only the unbiased component is used to deliver more accurate recommendations. Unlike those methods, extracting separate ranking and propensity models in unbiased learning to rank is difficult when they share a common input and target. In summary, these differences make existing deconfounding methods not applicable in ULTR.

## 7 CONCLUSION

We investigate unbiased learning to rank through the lens of causality and identify query-document relevance representation as a confounder, which leads to propensity overestimation. For unconfounded propensity overestimation, we propose a novel propensity model, *i.e.,* the Logging-Policy-aware Propensity Model, and its distinct two-step optimization strategy: (1) logging-policy-aware confounding effect learning and (2) joint propensity learning. Afterwards, we conduct backdoor adjustment for unconfounded propensity estimation, which serves for ULTR. Extensive experiments on two benchmarks with synthetic clicks with online and offline simulations justify the effectiveness of our proposal in addressing propensity overestimation and improving ranking performance.

A natural future direction would be to extend this work to pairwise learning to explore the feasibility of UPE across more ULTR frameworks. It also makes sense to apply our framework to other bias types, *e.g.,* addressing contextual-position bias or trust bias.

**Acknowledgements**. This research received funding support from the National Natural Science Foundation of China under Grant Numbers 62302345 and U23A20305, as well as from the Natural Science Foundation of Hubei Province, China, under Grant Numbers 2023AFB192 and 2023BAB160.

# REFERENCES


[1] Qingyao Ai, Keping Bi, Jiafeng Guo, and W. Bruce Croft. 2018. Learning a Deep Listwise Context Model for Ranking Refinement. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.*

[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, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.*

[3] Qingyao Ai, Tao Yang, Huazheng Wang, and Jiaxin Mao. 2021. Unbiased Learning to Rank: Online or Offline? *ACM Trans. Inf. Syst.* 39, 2 (2021), 21:1–21:29.

[4] Yinqiong Cai, Jiafeng Guo, Yixing Fan, Qingyao Ai, Ruqing Zhang, and Xueqi Cheng. 2022. Hard Negatives or False Negatives: Correcting Pooling Bias in Training Neural Ranking Models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022.*

[5] Olivier Chapelle and Yi Chang. 2011. Yahoo! Learning to Rank Challenge Overview. In *Proceedings of the Yahoo! Learning to Rank Challenge, held at ICML 2010, Haifa, Israel, June 25, 2010.*

[6] 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, CIKM 2009, Hong Kong, China, November 2-6, 2009.*

[7] Mouxiang Chen, Chenghao Liu, Zemin Liu, and Jianling Sun. 2022. LBD: Decouple Relevance and Observation for Individual-Level Unbiased Learning to Rank. In *Advances in Neural Information Processing Systems*, Vol. 35.

[8] Mouxiang Chen, Chenghao Liu, Zemin Liu, and Jianling Sun. 2022. Scalar is Not Enough: Vectorization-based Unbiased Learning to Rank. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 136–145.

[9] Mouxiang Chen, Chenghao Liu, Jianling Sun, and Steven C. H. Hoi. 2021. Adapting Interactional Observation Embedding for Counterfactual Learning to Rank. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021.*

[10] Konstantina Christakopoulou, Madeleine Traverse, Trevor Potter, Emma Marriott, Daniel Li, Chris Haulk, Ed H. Chi, and Minmin Chen. 2020. Deconfounding User Satisfaction Estimation from Response Rate Bias. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020.*

[11] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2016. Click Models for Web Search and their Applications to IR: WSDM 2016 Tutorial. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016.*

[12] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings.*

[13] Nick Craswell, Onno Zoeter, Michael J. Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008.*

[14] Romain Deffayet, Philipp Hager, Jean-Michel Renders, and Maarten de Rijke. 2023. An Offline Metric for the Debiasedness of Click Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023.* ACM, 558–568.

[15] Romain Deffayet, Jean-Michel Renders, and Maarten de Rijke. 2023. Evaluating the Robustness of Click Models to Policy Distributional Shift. *ACM Trans. Inf. Syst.* 41, 4 (2023), 84:1–84:28.

[16] John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* (2011).

[17] Priyanka Gupta, Ankit Sharma, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2021. CauSeR: Causal Session-based Recommendations for Handling Popularity Bias. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021.*

[18] 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, WWW 2019, San Francisco, CA, USA, May 13-17, 2019.*

[19] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* (2002).

[20] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada.* ACM, 133–142.

[21] Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006.*

[22] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005.*

[23] Thorsten Joachims, Laura A. 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 (2007), 7.

[24] 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, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017.*

[25] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2021. Mitigating Confounding Bias in Recommendation via Information Bottleneck. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021.*

[26] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Salvatore Trani. 2016. Post-Learning Optimization of Tree Ensembles for Efficient Ranking. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016.*

[27] Dan Luo, Lixin Zou, Qingyao Ai, Zhiyu Chen, Dawei Yin, and Brian D. Davison. 2023. Model-based Unbiased Learning to Rank. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023.* ACM, 895–903.

[28] Haitao Mao, Lixin Zou, Yujia Zheng, Jiliang Tang, Xiaokai Chu, Jiashu Zhao, and Dawei Yin. 2022. Whole Page Unbiased Learning to Rank. *arXiv preprint arXiv:2210.10718* (2022).

[29] Jiaxin Mao, Zhumin Chu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. Investigating the Reliability of Click Models. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2019, Santa Clara, CA, USA, October 2-5, 2019.*

[30] Jiaxin Mao, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Constructing Click Models for Mobile Search. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.*

[31] Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress* 19 (2000), 2.

[32] Yi Ren, Hongyan Tang, and Siwen Zhu. 2022. Unbiased Learning to Rank with Biased Continuous Feedback. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022.* ACM, 1716–1725.

[33] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007.*

[34] Masahiro Sato, Sho Takemori, Janmajay Singh, and Tomoko Ohkuma. 2020. Unbiased Learning for the Causal Effect of Recommendation. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020.*

[35] Mark D. Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007.*

[36] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958.

[37] Anh Tran, Tao Yang, and Qingyao Ai. 2021. ULTRA: An Unbiased Learning To Rank Algorithm Toolbox. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021.*

[38] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2021. Mixture-Based Correction for Position and Trust Bias in Counterfactual Learning to Rank. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021.* ACM, 1869–1878.

[39] Ali Vardasbi, Harrie Oosterhuis, and Maarten de Rijke. 2020. When Inverse Propensity Scoring does not Work: Affine Corrections for Unbiased Learning to Rank. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020.*

[40] Nan Wang, Zhen Qin, Xuanhui Wang, and Hongning Wang. 2021. Non-Clicks Mean Irrelevant? Propensity Ratio Scoring As a Correction. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021.* ACM, 481–489.

[41] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021.*

[42] 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, SIGIR 2016, Pisa, Italy, July 17-21, 2016.*

[43] 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, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018.*

[44] Yixin Wang, Dawen Liang, Laurent Charlin, and David M. Blei. 2020. Causal Inference for Recommender Systems. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020.*

[45] Le Yan, Zhen Qin, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. Revisiting Two-tower Models for Unbiased Learning to Rank. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022.*

[46] Tao Yang, Shikai Fang, Shibo Li, Yulan Wang, and Qingyao Ai. 2020. Analysis of Multivariate Scoring Functions for Automatic Unbiased Learning to Rank. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020.* 2277–2280.

[47] Tao Yang, Cuize Han, Chen Luo, Parth Gupta, Jeff M. Phillips, and Qingyao Ai. 2023. Mitigating Exploitation Bias in Learning to Rank with an Uncertainty-aware Empirical Bayes Approach. *CoRR* abs/2305.16606 (2023).

[48] Tao Yang, Chen Luo, Hanqing Lu, Parth Gupta, Bing Yin, and Qingyao Ai. 2022. Can Clicks Be Both Labels and Features?: Unbiased Behavior Feature Collection and Uncertainty-aware Learning to Rank. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022.* ACM, 6–17.

[49] Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua. 2021. Deconfounded Video Moment Retrieval with Causal Intervention. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021.*

[50] Ruohan Zhan, Changhua Pei, Qiang Su, Jianfeng Wen, Xueliang Wang, Guanyu Mu, Dong Zheng, Peng Jiang, and Kun Gai. 2022. Deconfounding Duration Bias in Watch-time Prediction for Video Recommendation. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022.* 4472–4481.

[51] Junqi Zhang, Jiaxin Mao, Yiqun Liu, Ruizhe Zhang, Min Zhang, Shaoping Ma, Jun Xu, and Qi Tian. 2019. Context-Aware Ranking by Constructing a Virtual Environment for Reinforcement Learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019.*

[52] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021.*

[53] Yunan Zhang, Le Yan, Zhen Qin, Honglei Zhuang, Jiaming Shen, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2023. Towards Disentangling Relevance and Bias in Unbiased Learning to Rank. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023.* ACM, 5618–5627.

[54] Honglei Zhuang, Zhen Qin, Xuanhui Wang, Michael Bendersky, Xinyu Qian, Po Hu, and Dan Chary Chen. 2021. Cross-Positional Attention for Debiasing Clicks. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021.*

[55] Lixin Zou, Changying Hao, Hengyi Cai, Shuaiqiang Wang, Suqi Cheng, Zhicong Cheng, Wenwen Ye, Simiu Gu, and Dawei Yin. 2022. Approximated doubly robust search relevance estimation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management.* 3756–3765.

[56] Lixin Zou, Haitao Mao, Xiaokai Chu, Jiliang Tang, Wenwen Ye, Shuaiqiang Wang, and Dawei Yin. 2022. A large scale search dataset for unbiased learning to rank. *Advances in Neural Information Processing Systems* 35 (2022), 1127–1139.

[57] Lixin Zou, Shengqiang Zhang, Hengyi Cai, Dehong Ma, Suqi Cheng, Shuaiqiang Wang, Daiting Shi, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model based ranking in Baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 4014–4022.