Counterfactual Ranking Evaluation with Flexible Click Models

Alexander Buchholz
buchhola@amazon.com
Amazon Music
Berlin, Germany

Ben London
blondon@amazon.com
Amazon Music
Seattle, WA, USA

Giuseppe Di Benedetto
bgiusep@amazon.com
Amazon Music
Berlin, Germany

Jan Malte Lichtenberg
jlichten@amazon.com
Amazon Music
Berlin, Germany

Yannik Stein
syannik@amazon.com
Amazon Music
Berlin, Germany

Thorsten Joachims
thorstj@amazon.com
Amazon Music
Ithaca, NY, USA

ABSTRACT

Evaluating a new ranking policy using data logged by a previously deployed policy requires a counterfactual (off-policy) estimator that corrects for presentation and selection biases. Some estimators (e.g., the *position-based model*) perform this correction by making strong assumptions about user behavior, which can lead to high bias if the assumptions are not met. Other estimators (e.g., the *item-position model*) rely on randomization to avoid these assumptions, but they often suffer from high variance. In this paper, we develop a new counterfactual estimator, called *Interpol*, that provides a tunable trade-off in the assumptions it makes, thus providing a novel ability to optimize the bias-variance trade-off. We analyze the bias of our estimator, both theoretically and empirically, and show that it achieves lower error than both the position-based model and the item-position model, on both synthetic and real datasets. This improvement in accuracy not only benefits offline evaluation of ranking policies, we also find that *Interpol* improves learning of new ranking policies when used as the training objective for learning-to-rank.

CCS CONCEPTS

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

KEYWORDS

Off-policy evaluation, Learning-to-rank, Position bias, Position-based model, Item-position model

ACM Reference Format:
Alexander Buchholz, Ben London, Giuseppe Di Benedetto, Jan Malte Lichtenberg, Yannik Stein, and Thorsten Joachims. 2024. Counterfactual Ranking Evaluation with Flexible Click Models. 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/10.1145/3626772.3657810

1 INTRODUCTION

The practice of evaluating new ranking policies using data logged by a previously deployed policy is critical to improving search and recommendation systems. A key advantage of such offline evaluation over online A/B tests is improved experimental velocity and reduced impact on the user experience. In particular, offline evaluation can screen out bad ranking policies so that users are exposed only to the most promising new policies. Furthermore, accurate offline evaluation enables offline learning of new ranking policies, including model-selection processes (e.g., hyperparameter tuning, feature selection) that would be difficult to perform online.

The core component of offline evaluation is counterfactual (a.k.a. *off-policy*) estimation. It corrects for the difference between the *logging* (or *behavior*) policy that produced the rankings when the data was logged and the *target* policy that we want to evaluate offline. Since the rankings of the logging and target policies are typically different, the counterfactual estimator needs to address the following counterfactual question: how would the new target policy have performed if it had been used instead of the logging policy? This question is somewhat easier to answer in the standard contextual bandit setting, where we already have a healthy repertoire of accurate and practically effective counterfactual estimators [17, 29, 30, 48, 52]. For the problem of ranking, however, we still lack equally effective estimators that come with strong theoretical guarantees for both learning and evaluation.

The key challenge in designing counterfactual estimators for ranking lies in the combinatorial nature of rankings. In particular, the large number of possible rankings would lead to unacceptable variance if we naively applied contextual-bandit estimators [17, 29, 30, 48, 52]. To reduce variance, counterfactual estimators are often designed with modeling assumptions about how the users' interactions (e.g., clicks) decompose across positions in the ranking, and correct for the associated presentation biases. Unfortunately, no single model is right for every problem. Existing estimators either make unrealistically strong modeling assumptions (e.g., the *position-based model* (PBM) [12, 26, 50]) which lead to biased estimates, or they do not provide sufficient variance reduction (e.g. the *item-position model* (IPM) [31]). In many real-world situations, this leaves practitioners with no good choice of estimator to achieve a reasonable bias-variance trade-off, as modeling assumptions are fixed and cannot be parameterized easily.

To fill this gap, we introduce a new estimator, called *Interpol*, which provides flexible control over a rich space of modeling assumptions so as to better balance bias and variance under realistic conditions. In particular, *Interpol* is the first estimator that allows practitioners to adjust its modeling assumptions to detailed properties of the user interface (e.g., screen size, pagination) in which

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657810

ranking is applied. The estimator subsumes the PBM and IPM estimators as special cases, and we propose two variants that apply to both the full and limited visibility settings. The key modeling flexibility lies in the definition of *windows* in which a local version of the PBM is (approximately) correct. Different window systems model different user behavior; for example, if scrolling happens in batches. Our approach thereby allows to tailor the estimator to the specifics of the user interface. Different window systems characterize the user experience and how trustworthy we deem position bias correction in these windows. We show that the choice of windows and their size provides a bias-variance trade-off that can deliver substantial improvements in estimation accuracy. Furthermore, we show that *Interpol* is always unbiased when the position bias curve is correctly specified; and since it can have lower variance than both PBM and IPM, it can have lower overall error.

In addition to characterizing *Interpol*'s theoretical properties, we evaluate its empirical performance on both synthetic and benchmark datasets, as well as on a real-world estimation problem from a major media streaming service. Our experiments confirm that *Interpol* can indeed provide improved estimation accuracy, compared to the PBM and IPM estimators—especially when the position bias curve is misspecified. We also show how to incorporate *Interpol* into the training objective of a learning-to-rank algorithm. We thereby provide a novel training procedure for policies in the span of the IPM and PBM at both extremes. This leads to improved offline learning of new ranking policies.

2 RELATED WORK

Our work addresses off-policy evaluation of ranking policies [26, 45, 50]. The primary challenge in off-policy evaluation is presentation bias, such as positional examination bias or trust bias (see, e.g., [3, 24, 25]). To counteract these biases and to keep variance under control, an estimator must make certain assumptions about how users interact with the system, commonly referred to as a *click model* [11, 31]. We extend, and compare to, two popular click models and their corresponding estimators: the position-based model [12, 26, 50] and the item-position model [31]. Our focus is on these two models using inverse propensity weighting, in contrast to reward regression, see for example [34, 41]. Other popular click models are based on, for example, cascade behavior and probabilistic extensions, which we do not study here. See, for example, [7, 15, 18, 19, 49].

Another consideration is whether policies rank all available items (i.e., the *full visibility* setting) or output only the top-k most promising items (i.e., the *limited visibility* setting). Estimators for the latter, more challenging setting have been proposed in [38]. One of our proposed estimators builds on this work.

The bias-variance trade-off inherent to importance weighting estimators [22, 39] motivates our work. Other ways to mitigate variance include importance weight truncation (a.k.a. *clipping*) [23] and *control variates*, such as self-normalization [33, 44], or *doubly-robust* methods [14]—the latter of which have recently been extended to the ranking setting [27, 37, 41]. Our key ideas are complementary to all of these methods, and these techniques could also be composed with our proposed estimators. The key novelty of our idea is the interpolation between modelling assumptions (i.e., the PBM and IPM) with its resulting bias-variance trade-off compared to the aforementioned approaches.

While off-policy evaluation assumes that a target policy has been given, *off-policy learning* tries to find a target policy that maximizes reward. In this work, we focus on the *policy gradient* [8, 9, 35, 36, 42, 53, 54] approach to off-policy learning, which directly optimizes a counterfactual reward estimator using a differentiable class of policies. Studies have shown that policy learning with more accurate counterfactual estimators leads to better policies [2, 21, 28]. Existing applications using counterfactual evaluators [54] have been built on top of the PBM estimator. We are therefore motivated to try the off-policy policy gradient with *Interpol*, resulting in better performing models than using the PBM or IPM as underlying reward estimators.

3 BACKGROUND

We are interested in estimating the expectation of a reward signal based on clicks (or any other user feedback) for a given target policy, π, using recorded interactions from a logging policy, π_0. The expected reward depends on the user's context, x, the order in which items were displayed (i.e., the ranking), Y, and the user's behavior when browsing the ranking (i.e., a click model). We denote the expected reward by

$$\Delta_\lambda(\pi) = \mathbb{E}_x \mathbb{E}_c \mathbb{E}_{Y\sim\pi(\cdot|x)} \left[\sum_{y\in Y} \lambda(y|Y) c(y|Y) \right],$$

where $c(y|Y) \in \{0, 1\}$ denotes if the item y in list Y was clicked and $\lambda(y|Y)$ is a weighting factor that lets us represent different linearly decomposable IR metrics (see also [2, 38]). For example, for $\lambda(y|Y) = \log_2(1 + Y[y])^{-1}$, where $Y[y]$ denotes the rank of item y in list Y, we obtain discounted cumulative gain (DCG); or, for $\lambda(y|Y) = Y[y]$, we obtain the average relevance position (ARP). Both the logging and target policies can be stochastic, and we use $\pi_0(\cdot|x)$ and $\pi(\cdot|x)$, respectively, to denote their conditional distributions over the space of rankings.

For simplicity of exposition[1] we will focus on the case where $\lambda(y|Y) = 1$, resulting in the expected number of total clicks,

$$\Delta(\pi) = \mathbb{E}_x \mathbb{E}_c \mathbb{E}_{Y\sim\pi(\cdot|x)} \left[\sum_{y\in Y} c(y|Y) \right]. \tag{1}$$

To estimate Equation (1), we use a dataset of n logged interactions,

$$\mathcal{D} = \left\{ x^i, Y_0^i, c(\cdot|Y_0^i), \pi_0 \right\}_{i=1}^n,$$

where rankings are produced by sampling $Y_0^i \sim \pi_0(\cdot|x^i)$. We use $Y[y]$ to denote the rank of item y in ranking Y. If y is not ranked at a visible position (e.g., below k in a top-k ranking), then $Y[y]$ outputs a null value (i.e., $Y[y] = \emptyset$).

Importance weighting estimators. The types of counterfactual estimators for $\Delta(\pi)$ that we consider employ some form of *importance weighting*. For a non-negative weighting function, $w(y|Y, Y_0) \geq 0$, we define a generic ranking estimator that weights clicks at the item level,

$$\hat{\Delta}(\pi|\mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{y\in Y^i} w(y|Y, Y_0) \times c(y|Y_0^i), \tag{2}$$

[1]The generalization of our work is straightforward but omitted for the sake of succinctness; see for example [2, 38].

which we instantiate with different weighting functions $w(y|Y, Y_0)$ subsequently. In general, this weight can be a function of the item y in question, the target and logged rankings Y and Y_0, and implicitly other logged variables. It corrects for selection biases and user behavior (e.g., position bias) under different click models. Following are two popular click models used in the literature.

Item-position model. One general click model is the *item-position model* (IPM) [31]. It allows clicks to jointly depend on the displayed item, its rank and the context. By assumption, clicks are independent of items displayed at other ranks. The expectation of a click on item y in list Y (in context x) is $\mathbb{E}_c[c(y|Y)|x] = \bar{c}(y, Y[y]|x)$. In this model, we assume that the logging policy π_0 is stochastic, and that we know the marginal probability (i.e., *propensity*) $\mathbb{P}(Y_0[y] = j \mid x, \pi_0)$ with which item y is ranked at position j. The assumption of independence between positions in the ranking implies that it suffices to estimate the expected clicks for each position in the ranking individually. We can therefore appeal to the standard *inverse propensity score* (IPS) estimator. The IPM estimator for rankings is a sum of IPS estimators. It is expressed by the generic estimator in Equation (2) with importance weight

$$w_{ip}(y|Y, Y_0) = \frac{I\{Y_0[y] = Y[y]\}}{\mathbb{P}(Y_0[y] = Y[y]|x, \pi_0)}, \tag{3}$$

where $I\{\cdot\}$ denotes the indicator function that returns 1 if its argument is true, and 0 otherwise.[2] When its modeling assumptions hold, the IPM estimator is unbiased. Unfortunately, it often suffers from variance. It only weights a click if the clicked item has matching rank under the logging and target policies, potentially dividing by small propensities. It does not correct explicitly for differences in positions, and thus uses less efficiently the available data.

Position-based model. Another popular click model [4, 12, 20, 51] is the *position-based model* (PBM). Like the IPM, the PBM assumes that clicks are independent of other positions in the ranking. However, the PBM makes an additional assumption that the probability of each click factorizes as the product of the item's expected relevance to the user (independent of where it is ranked), and the probability that the user looks at the position in which the item is ranked (independent of whether the item is relevant). This explicitly models the user's examination probability, which we refer to as a *position bias*. Using $\text{rel}(y)$ to denote item relevance, and $p_k = \mathbb{P}(o(y)|Y[y] = k)$ to denote the position bias (where $o(y)$ denotes the event that item y is observed), the expected click value is $\mathbb{E}_c[c(y|Y)|x] = \text{rel}(y|x) \times \mathbb{P}(o(y)|Y[y])$. The PBM estimator weights items according to the ratio of their examination probabilities under the target ranking Y and the logged ranking Y_0. This leads to the importance weight

$$w_{pbm}(y|Y, Y_0) = \frac{p_{Y[y]}}{p_{Y_0[y]}}, \tag{4}$$

in Equation (2). This weight reflects the intuition that clicks on items that have greater visibility under the target ranking should get a higher weight, and vice versa. As no information about the logging policy beyond the ranking itself is used, we will sometimes refer to this estimator as being *policy-oblivious*. Like the IPM estimator, the PBM estimator can be unbiased, but the conditions for unbiasedness are more involved. Its modeling assumptions must hold and it requires the true position biases to be known; further, in this version of the estimator, all relevant items must be observable under the logging policy [38] (i.e., the full visibility setting). In practice, the true position biases are estimated from data [4, 6, 16, 40, 51], which can introduce significant bias in the reward estimates. Yet, the PBM estimator tends to have lower variance than the IPM estimator. Informally speaking, the variance reduction comes from a more efficient use of all data (not just matching ranks) combined with stronger assumptions on the user behavior. Thus, the PBM estimator trades an increase in bias for a reduction in variance.

Policy-aware estimation in the limited visibility setting. If the number of slots k where items are displayed is smaller than the number of ranked items K (i.e., the limited visibility setting), the PBM estimator using Equation (4) suffers from a selection bias. Recent work by [38] mitigates this bias by extending the PBM estimator to explicitly account for visibility under the logging policy (e.g., top-k rankings). This leads to the following importance weight:

$$w_{pa}(y|Y, Y_0) = \frac{p_{Y[y]}}{\sum_{j=1}^{k} \mathbb{P}(Y_0[y] = j|x, \pi_0) p_j}. \tag{5}$$

The expression in the denominator equals the probability that an item is observed in one of the k visible positions under the logging policy, i.e., $\mathbb{P}(o(y)|x, \pi_0)$, and hence integrates out the position where an item was shown. The policy-aware PBM estimator can be unbiased in the limited visibility setting, provided the PBM assumptions hold and the true position biases are known. However, like the policy-oblivious PBM estimator, the policy-aware version still suffers from bias when the position biases are misspecified, so it represents a similar bias-variance trade-off.

4 *INTERPOL* ESTIMATOR

In the following, we define a spectrum of modeling assumptions (on which the IPM and PBM are extreme cases) that will allow us to control the bias-variance trade-off of off-policy evaluation in these two click models. This leads us to developing a new class of estimators that parameterizes the modeling assumptions, with the IPM and PBM as special cases. This new estimator, called *Interpol*, is based on the idea that a position bias correction of PBM can be *locally* accurate, even if the full PBM may produce unacceptable bias. For example, the PBM may be accurate enough to predict the clickthrough rate of an item y at position 1 in the target ranking if the logging policy puts y in position 2, but not if the logging policy ranked y in position 20. In the latter case, even small inaccuracies greatly distort $w_{pbm}(y|Y, Y_0)$ and thus lead to unacceptable bias.

4.1 Window systems

Based on the above reasoning, we introduce the notion of a *window system* to restrict where the PBM is used. Informally, a window system defines regions of the ranking that are considered "safe" for applying position bias correction. For a given item at a particular rank under the target policy, we check if its rank under the logging policy falls within the window; if so, we apply position bias correction. Otherwise the assigned weight is set to zero, as would be the case under the IPM.

Definition 4.1 (Window System). A *window system* $\mathcal{W}$ assigns every visible position j in the target ranking a non-empty set of

[2] We could alternatively use the probability under the target policy instead of the indicator function, which is equivalent in expectation for any stochastic target policy.

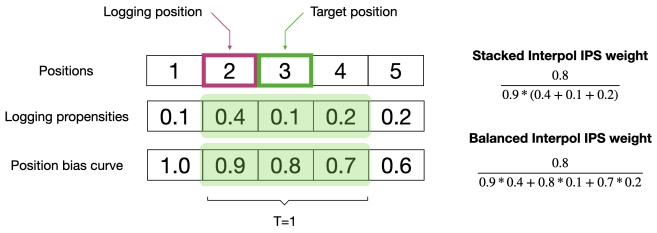

Figure 1: Illustration of the importance-weight computation for the stacked and balanced versions of *Interpol*, for the banded window system with $T = 1$ (green region). The target and logging policies place an item at position 3 and 2 respectively. For stacked *Interpol*, we compute the probability that the logging policy placed the item inside the window corresponding to the rank assigned by the target policy. For balanced *Interpol*, we compute the probability that the item was seen by the user in the window under the logging policy.

associated visible positions $\mathcal{W}(j)$ in the logged ranking in which position bias correction can be applied.

We could alternatively define windows that differentiate on the context x and the full target ranking Y, but for simplicity of notation we do not consider this explicitly.

Note that the window system $\forall j : \mathcal{W}_{IPM}(j) = \{j\}$ recovers the IPM (i.e., only consider cases where the rank under π_0 and π match). Similarly, the window system $\forall j : \mathcal{W}_{PBM}(j) = \{1, 2, \dots\}$ recovers the PBM (i.e., always use position bias correction). More generally, however, the choice of window system provides a means of tailoring the assumptions of the estimator to the properties of the user interface and its effect on user behavior, as the following examples illustrate.

- **Paging Windows:** In some interfaces, the user reads the ranking in pages of 4. The PBM may be reasonably accurate on each page, but not between pages. The window system $\mathcal{W}(j) = \{1, 2, 3, 4\}$ for $j \in \{1, 2, 3, 4\}$, $\mathcal{W}(j) = \{5, 6, 7, 8\}$ for $j \in \{5, 6, 7, 8\}$, etc. models this.
- **Scrolling Windows:** In some interfaces (e.g., mobile apps), the top 4 positions can be seen without scrolling, such that the PBM is more accurate there. This motivates using windows $\mathcal{W}(j) = \{1, 2, 3, 4\}$ for $j \in \{1, 2, 3, 4\}$, and then the IPM windows in positions $j \geq 5$ with $\mathcal{W}(j) = \{j\}$.
- **Banded Windows:** The window system $\forall j : \mathcal{W}_T(j) = \{j - T, \dots, j+T\}$ applies the PBM to the local window of radius T around each position j in the target ranking. This captures a continuous scrolling UI, wherein the PBM is accurate within a sliding window.

For the top-k setting, only the first k positions are visible, so each window needs to be suitably restricted to positions that are visible under the logging policy.

In our experiments, we focus on the Banded Window System $\mathcal{W}_T$, but we do not argue that any of the window systems listed above is superior to the others. Our key point in providing these examples lies in demonstrating the flexibility of our framework for exploiting the properties of a given application.

In the following subsections, we develop two versions of *Interpol*. One is related to the policy-oblivious approach [38], the other to the policy-aware approach [38]. After introducing both *Interpol* variants, we show that both versions are unbiased if user behavior is correctly modeled by a position-based model under the given window system $\mathcal{W}$. Furthermore, we bound the bias of *Interpol* in case the position bias curve is misspecified.

4.2 Stacked *Interpol*

The first variant of *Interpol* removes the selection bias of the logging policy in a first step, and the position bias in a second step. This leads to a *stacked* importance weight, where the first weight corrects for the probability $\mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0)$ of the logging policy π_0 hitting the window $\mathcal{W}(Y[y])$ corresponding to the target position $Y[y]$, and the second weight corrects for the position mismatch inside the window using the PBM:

$$w_{\text{stack}}(y|Y, Y_0) = \frac{I\{Y_0[y] \in \mathcal{W}(Y[y])\}}{\mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0)} \frac{p_{Y[y]}}{p_{Y_0[y]}}. \quad (6)$$

The term $I\{Y_0[y] \in \mathcal{W}(Y[y])\}$ sets the weight to zero if the logging policy places y outside the target window.

For illustration, consider the example in Figure 1. Item y is ranked at position 3 by the target policy, and at position 2 by the logging policy. The green region indicates the Banded Window $\mathcal{W}_T(3) = \{2, 3, 4\}$ around position 3 with radius $T = 1$. The first weight removes the selection bias by dividing by the probability of the logging policy hitting this window, i.e. $1/(0.4+0.1+0.2)$. The second weight $0.8/0.9$ corrects the position bias of position 3 vs. 2 per the PBM.

Observe that $w_{\text{stack}}(y|Y, Y_0)$ recovers the PBM and the IPM as extreme cases. For the IPM window system $\forall j : \mathcal{W}_{IPM}(j) = \{j\}$, the ratio $p_{Y[y]}/p_{Y_0[y]}$ is always 1, and the first part of $w_{\text{stack}}(y|Y, Y_0)$ is identical to the IPM. Similarly, for the PBM window system $\forall j : \mathcal{W}_{PBM}(j) = \{1, 2, \dots\}$, the denominator $\mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0)$ is equal to 1 under full visibility, since the logging policy always puts any item y somewhere in the ranking. What remains is the PBM weight, $p_{Y[y]}/p_{Y_0[y]}$.

For other window systems, the first term of $w_{\text{stack}}(y|Y, Y_0)$ generalizes the IPM estimator by considering matches inside the window. *Interpol* thus divides by the probability of hitting the window,

$$\mathbb{P}(I\{Y_0[y] \in \mathcal{W}(Y[y])\}|\pi_0, x) = \sum_{j \in \mathcal{W}(Y[y])} \mathbb{P}(Y_0[y] = j|\pi_0, x),$$

instead of the probability of an exact match. The larger the window, the bigger the probability that our *Interpol* estimator exploits an observation. This can lead to variance reduction compared to the IPM. Furthermore, we show that the following relaxed support condition suffices for *Interpol* to be unbiased.

Definition 4.2 (Full Window Support). The logging policy π_0 has *full window support* for target policy π in window system $\mathcal{W}$ if, for all contexts x, items y, and positions j with $\mathbb{P}(Y[y] = j|x, \pi) > 0$ in the target ranking Y, it follows that $\mathbb{P}(Y_0[y] \in \mathcal{W}(j)|x, \pi_0) > 0$.

Informally speaking, this means that for all positions where the target policy could place an item, we require the logging policy to provide a range of positions that intersects with the target positions. Full window support is strictly weaker than the support condition required by the IPM (i.e., positivity), where every position in the

ranking must have full support under the logging policy. We now show that the stacked *Interpol* estimator $\hat{\Delta}^{\text{stack}}_{\mathcal{W}}(\pi|\mathcal{D})$ (Equation (2) with the weight in Equation (6)) is unbiased for any window system.

Proposition 4.3. *$\hat{\Delta}^{stack}_{\mathcal{W}}(\pi|\mathcal{D})$ is an unbiased estimator of $\Delta(\pi)$ for a window system $\mathcal{W}$ if the logging data is generated from a known logging policy π_0 with full window support for $\mathcal{W}$ (Definition 4.2), under the position-based model with known position bias curve $p > 0$.*

Proof. Let $\hat{\Delta}^{\text{stack}}_{\mathcal{W}}(\pi|x, Y_0, c)$ denote the estimator for a single observation (x, Y_0, c). To show that $\hat{\Delta}^{\text{stack}}_{\mathcal{W}}(\pi|\mathcal{D})$ is unbiased, it suffices to show that $\hat{\Delta}^{\text{stack}}_{\mathcal{W}}(\pi|x, Y_0, c)$ is unbiased, since it is straightforward to verify that $\mathbb{E}[\hat{\Delta}^{\text{stack}}_{\mathcal{W}}(\pi|\mathcal{D})] = \mathbb{E}[\hat{\Delta}^{\text{stack}}_{\mathcal{W}}(\pi|x, Y_0, c)]$. We evaluate

$$\mathbb{E}\left[\hat{\Delta}^{\text{stack}}_{\mathcal{W}}(\pi|x, Y_0, c)\right] \tag{7}$$

$$= \mathbb{E}_c\mathbb{E}_{\pi_0}\left[\sum_{y\in Y} \frac{I\{Y_0[y] \in \mathcal{W}(Y[y])\}}{\mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0)} \frac{p_{Y[y]}}{p_{Y_0[y]}} c(y|Y_0)\right] \tag{8}$$

$$= \mathbb{E}_{\pi_0}\left[\sum_{y\in Y} \frac{I\{Y_0[y] \in \mathcal{W}(Y[y])\}}{\mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0)} \frac{p_{Y[y]}}{p_{Y_0[y]}} \mathbb{E}_c c(y|Y_0)\right] \tag{9}$$

$$= \mathbb{E}_{\pi_0}\left[\sum_{y\in Y} \frac{I\{Y_0[y] \in \mathcal{W}(Y[y])\}}{\mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0)} p_{Y[y]}\text{rel}(y)\right] \tag{10}$$

$$= \sum_{y\in Y} \frac{\mathbb{E}_{\pi_0}\left[I\{Y_0[y] \in \mathcal{W}(Y[y])\}\right]}{\mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0)} p_{Y[y]}\text{rel}(y) \tag{11}$$

$$= \sum_{y\in Y} p_{Y[y]}\text{rel}(y) = \mathbb{E}_c\mathbb{E}_{\pi}\left[\sum_{y\in Y} c(y)\right]. \tag{12}$$

Line (8) is obtained by using the definition of (2). In line (9) we pull the expectation of the click model inside the sum, exploiting its linearity. Then we use the definition of a click under the PBM in line (10). Since $p_{Y_0[y]} > 0$ inside the window $\mathcal{W}(Y[y])$, the examination probabilities (i.e., position biases) under the logging policy cancel out. In line (11) we pull the expectation with respect to the logging policy inside the sum, then in line (12) we use $\mathbb{E}_{\pi_0}\left[I\{Y_0[y] \in \mathcal{W}(Y[y])\}\right] = \mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0)$ and simplify further. Note that the denominator $\mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0)$ is always positive due to full window support, and it is not a random variable w.r.t. Y_0 as we know the probability with which any item appears in a window around $Y[y]$. Finally, we apply the previous identities in reverse using the PBM under the target policy π. □

The conditions of the above proof can be further relaxed, as it only applies the PBM inside of each window $\mathcal{W}(j)$. Each window could use its own local position bias curve that is locally (but not necessarily globally) correct. Such local position bias estimates are naturally provided by the position bias estimators in [4].

Interestingly, this estimator is unbiased even in the limited visibility setting, providing an alternative to the estimator based on (5). If an item y is in Y but not visible in Y_0, then $Y_0[y] = \emptyset$ and the importance weight is zero, since $I\{Y_0[y] \in \mathcal{W}(Y[y])\} = 0$. When y is visible in Y_0, we use the probability of π_0 ranking y in the window to up-weight the position bias ratio. Thus, we only require that $\mathbb{P}(Y_0[y] \in \mathcal{W}(Y[y])|x, \pi_0) > 0$, which corresponds precisely to full window support.

4.3 Balanced *Interpol*

We now present a policy-aware variant of *Interpol* that uses the marginal of the two weighting components as its importance weight. In analogy to [1], we call this *balanced Interpol*, and we denote it by $\hat{\Delta}^{\text{bal}}_{\mathcal{W}}(\pi|\mathcal{D})$. The importance weight of balanced *Interpol*,

$$w_{\text{bal}}(y|Y, Y_0) = \frac{I\{Y_0[y] \in \mathcal{W}(Y[y])\} \times p_{Y[y]}}{\sum_{j\in\mathcal{W}(Y[y])} p_j \times \mathbb{P}(Y_0[y] = j|x, \pi_0)}, \tag{13}$$

computes the probability that an item is seen by the user in the window around its target rank under the logging policy. This is illustrated in Figure 1, where the examination probability is $0.9 \cdot 0.4 + 0.8 \cdot 0.1 + 0.7 \cdot 0.2 = 0.58$.

Like the stacked *Interpol* estimator, the balanced *Interpol* estimator coincides with the IPM estimator for the IPM window system. Similarly, for the PBM window system we recover the policy-aware PBM, and for the PBM window system restricted to the top-k positions $\forall j : \mathcal{W}(j) = \{1, \ldots, k\}$ we recover the top-k estimator from [38]. In between these extremes, the balanced estimator is a policy-aware PBM confined to a window around the target rank. We now show that the balanced *Interpol* estimator is unbiased.

Proposition 4.4. *$\hat{\Delta}^{bal}_{\mathcal{W}}(\pi|\mathcal{D})$ is an unbiased estimator of $\Delta(\pi)$ for a window system $\mathcal{W}$ if the logging data is generated from a known logging policy π_0 with full window support for $\mathcal{W}$, under the PBM with a known position bias curve $p > 0$.*

Proof of Proposition 4.4. Let $\hat{\Delta}^{\text{bal}}_{\mathcal{W}}(\pi|x, Y_0, c)$ again denote the estimator for a single observation, we show that $\hat{\Delta}^{\text{bal}}_{\mathcal{W}}(\pi|x, Y_0, c)$ is unbiased, since $\mathbb{E}[\hat{\Delta}^{\text{bal}}_{\mathcal{W}}(\pi|\mathcal{D})] = \mathbb{E}[\hat{\Delta}^{\text{bal}}_{\mathcal{W}}(\pi|x, Y_0, c)]$. Recalling the importance weight formula in (13), we note that the denominator is the probability that item y is observed in the window corresponding to its target rank, under the distribution induced by the user and the logging policy; and further, that this quantity is not a random variable w.r.t. $Y_0[y]$. Thus, for any window system $\mathcal{W}$ we have

$$\mathbb{E}_c\mathbb{E}_{\pi_0}\left[\sum_{y\in Y} \frac{I\{Y_0[y] \in \mathcal{W}(Y[y])\} p_{Y[y]} c(y \mid Y_0)}{\sum_{j\in\mathcal{W}(Y[y])} p_j \mathbb{P}(Y_0[y] = j|x, \pi_0)}\right] \tag{14}$$

$$= \mathbb{E}_{\pi_0}\left[\sum_{y\in Y} \frac{I\{Y_0[y] \in \mathcal{W}(Y[y])\} p_{Y[y]} \mathbb{E}_c[c(y \mid Y_0)]}{\sum_{j\in\mathcal{W}(Y[y])} p_j \mathbb{P}(Y_0[y] = j|x, \pi_0)}\right] \tag{15}$$

$$= \mathbb{E}_{\pi_0}\left[\sum_{y\in Y} \frac{I\{Y_0[y] \in \mathcal{W}(Y[y])\} p_{Y[y]} p_{Y_0[y]} \text{rel}(y)}{\sum_{j\in\mathcal{W}(Y[y])} p_j \mathbb{P}(Y_0[y] = j|x, \pi_0)}\right] \tag{16}$$

$$= \sum_{y\in Y} \frac{E_{\pi_0}[I\{Y_0[y] \in \mathcal{W}(Y[y])\} p_{Y_0[y]}] p_{Y[y]} \text{rel}(y)}{\sum_{j\in\mathcal{W}(Y[y])} p_j \mathbb{P}(Y_0[y] = j|x, \pi_0)} \tag{17}$$

$$= \sum_{y\in Y} p_{Y[y]}\text{rel}(y) = \mathbb{E}_c\mathbb{E}_{\pi}\left[\sum_{y\in Y} c(y)\right]. \tag{18}$$

Line (15) pulls the expectation of the click model inside the sum, exploiting the linearity of the expectation. Line (16) uses the definition of a click under the PBM. Then in Line (17) we pull

the expectation with respect to the logging policy inside the sum. Finally line (18) simplifies the expression using identity

$$E_{\pi_0}[I\{Y_0[y] \in \mathcal{W}(Y[y])\} p_{Y_0[y]}] = \sum_{j \in \mathcal{W}(Y[y])} p_j \mathbb{P}(Y_0[y] = j|x, \pi_0).$$

This leads to the expected number of clicks under the PBM using target policy π. In all steps we require that the denominator $\sum_{j \in \mathcal{W}(Y[y])} p_j \mathbb{P}(Y_0[y] = j|x, \pi_0) > 0$, which is assured by full window support and positivity of the position bias weights. □

4.4 Bias of *Interpol* for misspecified PBM

The two variants of our estimator are unbiased when using the true position bias curves. In practice, however, we often only have access to an inaccurately estimated $\hat{p}$. We now analyze the impact of this misspecification on the bias of our estimators.

Proposition 4.5. *Let p be the correct position bias curve and let $\hat{p}$ be the misspecified version used in our estimators. Define the shorthand $\mathbb{P}_j = \mathbb{P}(Y_0[y] = j|x, \pi_0)$ and let $Bias^{\square} := \mathbb{E}[\hat{\Delta}^{\square}_{\mathcal{W}}(\pi|x, \pi_0, Y_0)] - \Delta(\pi)$ for $\square \in \{stack, bal\}$ denote the respective biases of the* Interpol *estimators defined by Equations* (6) *and* (13)*. Then,*

$$Bias^{\square} = \sum_{y \in Y} \left[\hat{p}_{Y[y]} \left(A^{\square}_{\mathcal{W}}(y) - \frac{p_{Y[y]}}{\hat{p}_{Y[y]}} \right) \right] \times rel(y). \tag{19}$$

where

$$A^{\square}_{\mathcal{W}}(y) = \begin{cases} \frac{\sum_{j \in \mathcal{W}(Y[y])} \mathbb{P}_j \frac{p_j}{\hat{p}_j}}{\sum_{j \in \mathcal{W}(Y[y])} \mathbb{P}_j}, & \text{for stacked Interpol } (\square = stack), \\ \frac{\sum_{j \in \mathcal{W}(Y[y])} \mathbb{P}_j p_j}{\sum_{j \in \mathcal{W}(Y[y])} \mathbb{P}_j \hat{p}_j}, & \text{for balanced Interpol } (\square = bal). \end{cases} \tag{20}$$

Proof sketch. The proof uses the same computations as in Propositions 4.3 and 4.4 applied to the estimator that uses $\hat{p}$ instead of p. We use the assumptions of our click model $\mathbb{E}[c(y|Y_0)] = \text{rel}(y) p_{Y_0[y]}$, simplify the resulting expressions accordingly, then rearrange the terms. Finally, we subtract the true reward to obtain an expression for the bias. □

This result provides several insights. First, the bias of stacked *Interpol* depends on a weighted average of the ratio $p_j/\hat{p}_j$ for all positions j inside window $\mathcal{W}(Y[y])$, with weights proportional to $\mathbb{P}_j$ that are normalized to sum to 1. Meanwhile, the bias of balanced *Interpol* depends on a ratio of windowed examination probabilities, where the numerator uses the true and the denominator the misspecified position bias. Without additional assumptions it is not possible to state a general relationship between the two variants.

Second, the bias could be positive or negative. A strong condition that would allow a prediction of the sign, like $\forall y : A^{\square}_{\mathcal{W}}(y) \geq p_{Y[y]}/\hat{p}_{Y[y]}$, is unlikely to hold in practice, as in all windows the position bias ratios would be systematically exaggerated.

Third, the bias is not necessarily monotonically decreasing in the size of the windows. Take the Banded Window system $\mathcal{W}_T$ with decreasing width T as an example. We compare $A^{\square}_{\mathcal{W}_T}(y)$ with $A^{\square}_{\mathcal{W}_{T-1}}(y)$. To derive a relationship between the two, we need explicit assumptions on the misspecification and the propensities of π_0. Furthermore, it is easy to construct examples that contradict monotonicity by considering items at the top or bottom of the list.

Whereas for PBM policy-aware/oblivious is an important distinction (oblivious is biased in the limited-visibility setting but does not need propensities), for *Interpol* this distinction is less important because both versions require propensities and both are unbiased. In practice, we find them to perform similar with no clear winner between the two.

5 *INTERPOL* FOR OFF-POLICY LEARNING

We now illustrate how *Interpol* applies to off-policy learning-to-rank. Our reasoning is that an improved estimator leads to a more reliable training objective. Optimizing this training objective should lead to learning a better policy. We optimize a parametric policy π_θ, where $\theta \in \Theta \subset \mathbb{R}^d$ denotes the model parameters. For example, θ can be the weights of a neural network that scores each item, and items are ranked in descending order of score. We want to find a policy with maximum expected reward,

$$\theta^\star \in \arg\max_{\theta \in \Theta} \Delta(\pi_\theta), \tag{21}$$

as defined in Equation (1). To approximate the optimization problem in (21) using a data set $\mathcal{D}$ from the logging policy π_0, we follow [2, 26] and optimize a counterfactual risk estimator. More specifically, given a logged ranking Y^i_0 and a target ranking Y^i, let $\tilde{\Delta}(Y^i, Y^i_0) = \sum_{y \in Y^i} w(y|Y^i_0, Y^i) \times c(y|Y^i_0)$ denote the importance weighted reward, where $w(y|Y^i_0, Y^i)$ is the importance weight of either version of *Interpol*. A counterfactual risk minimization objective for policy learning is given by

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{Y^i \sim \pi_\theta(\cdot|x^i)} \left[\tilde{\Delta}(Y^i, Y^i_0) \right] := \arg\max_{\theta \in \Theta} \hat{\Delta}(\pi_\theta|\mathcal{D}).$$

Gradient-based optimization of $\hat{\Delta}(\pi_\theta|\mathcal{D})$ is difficult, because the gradient $\nabla_\theta \mathbb{E}_{Y^i \sim \pi_\theta(\cdot|x^i)} \left[\tilde{\Delta}(Y^i, Y^i_0) \right]$ is not available in closed form. However, following the policy gradient approach used in the PG-Rank algorithm [42, 54], we use the "log-derivative trick" [53] to rewrite the gradient as

$$\nabla_\theta \hat{\Delta}(\pi_\theta|\mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{Y^i \sim \pi_\theta(\cdot|x^i)} \left[\nabla_\theta \log \pi_\theta(\cdot|x^i) \tilde{\Delta}(Y^i, Y^i_0) \right].$$

The above expectation over target rankings is approximated via Monte-Carlo sampling by drawing $m = 1, \ldots, M$ samples $Y^{i,m} \sim \pi_\theta(\cdot|x^i)$. This yields an approximate gradient

$$\nabla_\theta \hat{\Delta}(\pi_\theta|\mathcal{D}) \approx \frac{1}{n} \sum_{i=1}^{n} \frac{1}{M} \sum_{m=1}^{M} \nabla_\theta \log \pi_\theta(Y^{i,m}|x^i) \tilde{\Delta}(Y^{i,m}, Y^i_0).$$

We then use common gradient-based optimization on $\hat{\Delta}(\pi_\theta|\mathcal{D})$. Additional variance reduction methods may improve training [9, 35, 36], but our experiments focus on the basic variant above.

6 EXPERIMENTS

We first evaluate *Interpol* in a real-world setting using data from the streaming media service Amazon Music, highlighting its practical usefulness and its performance in comparison to three baselines. *Interpol* provides non-trivial improvements in reward estimation, as measured by *mean squared error* (MSE), at industry scale.

We then study *Interpol* in a synthetic setting. In this controlled environment we study varying conditions; i.e., how position bias misspecification, data set size and logging policy randomization impact the estimation. This allows us to isolate *Interpol*'s behavior

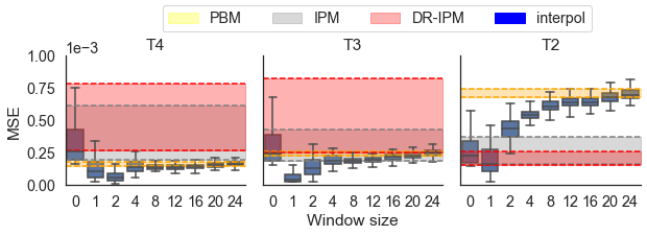

Figure 2: Experimental results on real world data from Amazon Music. MSE (lower is better) of balanced *Interpol* (blue boxplots) as a function of window size and three different target policies T_2, T_3, and T_4. MSE confidence intervals (90%) of three baselines (PBM, IPM and DR-IPM) are shown as horizontal bands with their respective color.

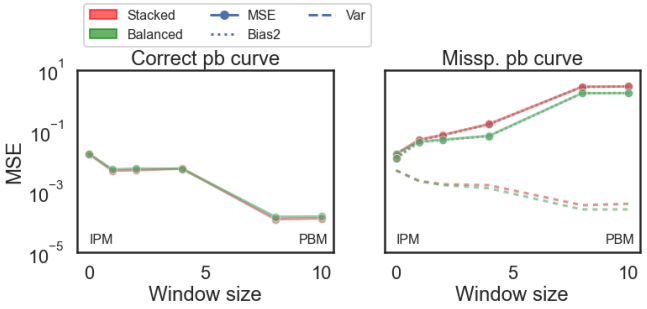

Figure 3: Synthetic data experiment. MSE, bias^2 and variance (lower is better) for the stacked and balanced version of *Interpol* for different window sizes, stay probability = 90%. Correct (left) and the misspecified (right) position bias curve with exponent $\alpha = 1.8$ (selected for illustrative purpose) under full visibility with $50,000$ observations.

along these dimensions. Finally, we highlight *Interpol*'s effectiveness for learning new ranking policies in an off-policy learning framework, applied to a common learning-to-rank benchmark.

Our experiments focus on the Banded window system, $\mathcal{W}_T$ for the *Interpol* framework. This scenario is parameterized by a single parameter, T. This corresponds to a sliding window user interface, in which the PBM is only accurate enough to compare positions that are visible simultaneously. We compare to two natural baselines: the IPM estimator (equivalent to *Interpol* with $T = 0$) and the PBM estimator (equivalent to T being the length of the ranked list). For our real world experiment we also compare *Interpol* to a doubly robust version of the IPM based on [13], where a reward regression is fit to each position. We refer to this method as DR-IPM. The reward regressor is a gradient boosted tree ensemble (lightGBM) that includes the displayed position as an input feature. Weight clipping, smoothing and control variates [5, 32, 33] are other methods to reduce variance in importance weighting estimators. Since these methods are complimentary to all estimators considered here, we abstain from their use to simplify and focus our experiments.

6.1 Offline evaluation on real-world data

Our real-world experiment is based on user interaction logs of a ranking task for the streaming media service Amazon Music. The task is to rank candidate sets of 25 items that were selected by a *first-stage retriever*. The data set consists of logs from four different ranking polices: T_1 is a stochastic policy that uses random swaps on top of a deterministic ranker; T_2 is a deterministic policy whose ranker is very similar to T_1's, but not identical; T_3 and T_4 are stochastic policies that sample actions according to a Plackett-Luce model, where T_3 has a less expressive feature representation than T_4. The data D_i corresponds to the policy T_i. Each data set contains around $800,000$ records. Only D_1 comes with exact propensities, since they can be computed for the random swapping algorithm. Thus, we designate T_1 as the logging policy and use D_1 to obtain off-policy estimates of the other policies' expected reward. Since we also have data from running T_2, T_3 and T_4 online, we can approximate their true expected rewards using the online estimates. We use this approximate *ground truth* to assess the accuracy of off-policy estimates. By evaluating policy $T_i : i \in \{2, 3, 4\}$ on data set D_1, we get a predicted reward, which we then compare to the observed average reward calculated from D_i. We measure the discrepancy between the predicted and observed average rewards by MSE. The position bias curve used in this experiment is estimated on D_1 using the approach given in [4]. We obtain confidence intervals around the mean by bootstrapping our estimates 100 times.

6.1.1 How does Interpol *perform in a real world experiment?* Figure 2 plots MSE (w.r.t. the approximate ground truth) as a function of window size. For T_3, a window size around 1 yields the lowest MSE for balanced *Interpol*. For T_4 a window size of 2 is best. For T_2 there seems to be only a small benefit from increasing the window size beyond 0. Note that the PBM performs particularly poorly on T_2, suggesting that the PBM assumptions are substantially violated. The doubly robust version of the IPM (DR-IPM) yields a variance reduction for T_2, but is otherwise harmful for T_3 and of little effect for T_4. We also evaluated the stacked version of *Interpol*, and it performs similarly to balanced *Interpol* (not shown here). Overall, we conclude that *Interpol* can provide substantial benefits over PBM, IPM, and DR-IPM in this real-world setting.

6.2 Offline evaluation on synthetic data

Our synthetic experiment is designed so that we know the true reward by construction, and we can thus reliably evaluate the accuracy of the estimators under a range of conditions. In particular, it is designed to allow control of the strength of the logging policy's randomization and the level of position bias misspecification. This setup lets us study the key properties of *Interpol* in response to various environmental conditions. We simulate a ranking application in which the reward function conforms to the PBM assumptions, with a position bias curve that we control. We generate 50,000 observations from a logging policy that ranks $k = 5$ different actions out of a total of 10 actions in the limited visibility setting and $k = 10$ in the full visibility setting. We set the true position bias curve to a decreasing function of rank: $p = 1 - j/10$, for rank $j = 0, \ldots, k$. To simulate misspecification of the PBM, we use biased curves defined as $\hat{p} = p^\alpha$ component-wise, where $\alpha \in [0.2, 0.4, \ldots, 1, \ldots, 2.0]$. Moving α away from 1 controls the misspecification of the position bias curve. Without loss of generality, there are four relevant items.

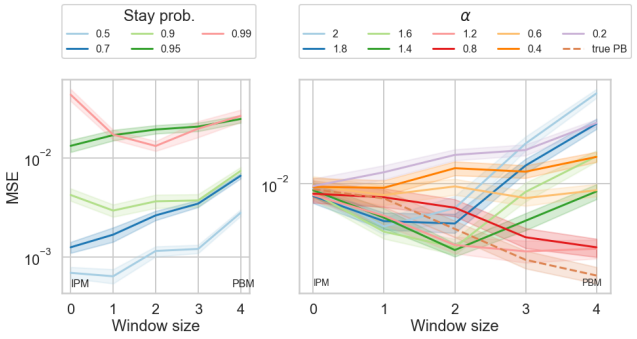

Figure 4: Synthetic data experiment. Left Figure: MSE (lower is better) over window sizes for different misspecification exponents α. Logging policy with stay probability 80% for balanced *Interpol* under limited visibility. Right Figure: MSE (lower is better) of balanced *Interpol* as a function of window size, levels of logging policy randomization and a misspecified position bias curve ($\alpha = 1.4$). Shaded areas correspond to 95% confidence intervals obtained from bootstrapping with 100 repetitions.

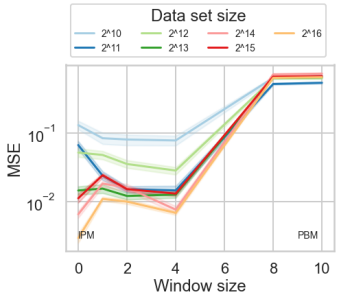

Figure 5: Synthetic data experiment, full visibility. MSE (lower is better) for data set sizes as a function of window size, stay probability = 80%, misspecification $\alpha = 1.4$. As the data size increases, balanced *Interpol* with window size 0 (IPM) performs best. For intermediate data sizes, window sizes between 0 and 10 perform best. Shaded areas are to 95% bootstrap confidence intervals (100 repetitions).

The logging policy π_0 orders two relevant items at the top and the other relevant items arbitrarily at the bottom of the list. Additionally, the logging policy swaps the ranked items randomly, where every item has a $q\%$ probability of staying in its original position and a $(100 - q\%)/9$ probability of being ranked in all other positions. We set these *stay probabilities* to $[50\%, 70\%, 90\%, 95\%, 99\%]$ to control the randomness of the logging policy. The target policy π deterministically ranks two relevant items in the first and fourth position and the two other relevant items outside the visible range, or at the bottom in the full visibility setting. The order of other items is arbitrary. Relevant items get a reward of 1 that is revealed according to the examination probability (i.e., the true position bias curve p). The true expected reward for the target policy can be computed analytically: it is 1.7 (resp. 2) in the limited (resp. full) visibility setting. First, we study the impact of misspecification and logging policy randomization on balanced *Interpol* under limited visibility. Then, we evaluate how varying data sizes impact *Interpol* and illustrate its bias-variance decomposition under full visibility.

6.2.1 How does stacked Interpol *compare against balanced* Interpol*?* Figure 3 (left) highlights the behavior of stacked and balanced *Interpol* with the correct position bias. Recall that we use the Banded Window system $\mathcal{W}_T$, in which $T = 0$ corresponds to the IPM baseline and $T = 10$ corresponds to the PBM baseline. As expected, since the data is indeed generated by a PBM, and the estimators have access to the correct position bias, the MSE decreases as the window size increases and the lowest MSE is achieved for the largest window sizes (i.e., the PBM). Since we use the correct position biases, all estimators are unbiased; hence, the MSE is dominated by the variance.

Figure 3 (right) illustrates how both versions of *Interpol* behave with a misspecified position bias (exponent $\alpha = 1.8$), with a moderately stochastic logging policy (stay probability = 90%), under full visibility. As expected, variance decreases when we increase the window size T, whereas the squared bias increases.

Both Figures in 3 show very similar behavior between the stacked and balanced versions of *Interpol*. We therefore focus mainly on the balanced version in the rest of our experiments.

6.2.2 How does Interpol *behave under increasing position bias misspecification?* Figure 4 (right) shows how MSE changes when varying the severity of misspecification, focusing on the limited visibility setting for balanced *Interpol*. Even for small levels of misspecification, the optimal window size is almost always smaller than $T = 4$, i.e., the PBM. For all levels of misspecification but 0.8 we recover erroneous PBM estimators at the maximum window size. The stronger the misspecification, the more beneficial it is to use *Interpol* with small window sizes and regularise towards the IPM. Consequently, the estimator that offers the best bias-variance trade-off is *Interpol* with a properly chosen window size.

6.2.3 How does Interpol *depend on the randomization of the logging policy?* Figure 4 (left) illustrates the effect of the logging policy's randomization (via stay probability)—which implicitly affects the amount of usable data; and hence, the bias-variance trade-off. With more randomization (i.e., lower stay probability), we find that moving towards the IPM is beneficial, as we have a greater chance of finding exact matches between the logging and target policies. With less randomization (i.e., high stay probability), we are less likely to find exact matches; and with fewer matches, we incur a larger variance. Consequently, it is better to increase the window size and leverage the (biased) position bias curve in order to reduce variance.

6.2.4 How do different data set sizes impact Interpol*?* When increasing the amount of data, the variance of estimates decreases, leading to smaller MSE. This change in variance can result in other window sizes being optimal. We highlight this in Figure 5 for balanced *Interpol* under full visibility. Overall, larger sample sizes favor smaller window sizes. This is consistent with our previous discussion; having more data lets us stay closer to the IPM, as variance contributes less to MSE.

6.3 Off-policy learning

We now evaluate *Interpol* for off-policy learning and compare it to conventional learning-to-rank algorithms that rely on the PBM model [26, 54]. We use click data derived from the Yahoo! learning-to-rank data set (YLTR, [10]) and largely follow the setup of [26] to simulate clicks. Training and testing sets were generated according to the position bias model based on binarized relevances from their respective full-information data sets. The rankings were generated by a neural network logging policy with randomly generated but fixed policy weights. To facilitate the comparison of different window sizes for the *Interpol* learning algorithm, we only used action set sizes with at least 10 actions and trimmed larger action sets to use only the first 10 actions, resulting in a training set of 14,665 instances and a testing set of 5,156 instances. The rankings were of length $k = 10$ and we use the same linearly decreasing true position bias curve as in our synthetic experiments, $p = [1.0, 0.9, \ldots, 0.1]$.

6.3.1 Algorithms and Baselines. We use our modification of the PG-Rank learning algorithm described in Section 5 with the balanced *Interpol* importance weight, $w_T^{\text{bal}}(y|Y, Y_0)$, and banded windows. We call this method *PGR-Interpol*. For large enough window sizes (and using DCG to weight observed clicks), PGR-Interpol is equivalent to the purely position bias-based version of PG-Rank used in [54] and [26] and hence serves as baseline (PGR-PBM). The ranking model trained in our experiments is a feed-forward neural network with a single hidden layer of 200 neurons, and learning rate of 0.005. Models were trained using mini-batches of size 1000, with 30 Monte-Carlo samples to approximate each gradient and stochastic gradient ascent. We varied the window size, $T = [0, 1, 3, 5, 7, 9]$, to recover learning with the PGR-IPM baseline ($T = 0$) or the PGR-PBM baseline ($T = 9$), as well as all versions of PGR-Interpol between those extremes.

6.3.2 Experiment setup. We vary experiment conditions using the same mechanisms as in Section 6.2. Specifically, we vary the position bias misspecification exponent, $\alpha \in [0, 0.01, 0.1, 1, 2, 3, 4]$, and the stay probability of the logging policy in [10%, 90%, 99.9%]. After 30 training epochs over the training data set, we evaluate the model's performance on the test set, using the same reward function (based on the true PBM) used to create the training data.

6.3.3 When is it beneficial to use Interpol *instead of IPM or PBM?.* Figure 6 shows DCG on the test set as a function of *Interpol*'s window size for three different levels of logging policy randomization and across different misspecifcation levels. The PGR-*Interpol* estimator routinely outperformed the pure PGR-IPM or PGR-PBM variants, as the maximum DCG is observed for window sizes $0 < T < 9$. The effect is most pronounced when the position bias is strongly misspecified and under weakly randomized logging policies. Using the pure PBM estimator ($T = 9$) produced low DCG for position bias exponents close to 0. Using the pure IPM estimator ($T = 0$) produced worse results when increasing stay probability (moving from left to right panel), which increases variance.

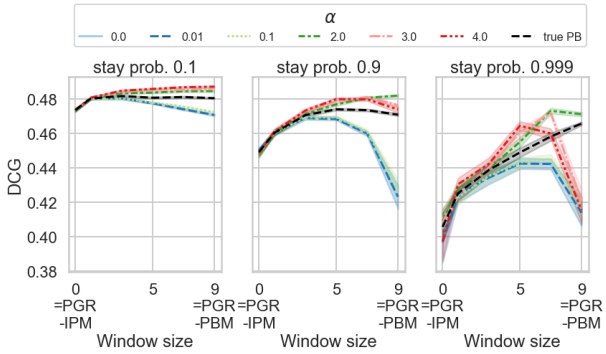

Figure 6: Off-policy learning results on YLTR dataset for PGR-Interpol and PGR-PBM and PGR-IPM baselines. DCG averaged across 100 random seeds (higher is better) for different window sizes, different levels of logging policy randomization (decreasing randomness from left to right), and different position bias misspecification factors (line color). 95% **confidence intervals are shaded.**

7 DISCUSSION

The size of the windows controls a bias-variance trade-off between the more general IPM and the potentially more biased PBM. For the Banded Window system $\mathcal{W}_T$, this is controlled by the parameter T. Choosing the window size is a model selection problem, as the true MSE is in general not accessible. To address model selection, recent work [43] proposes an adaptive method called *SLOPE*, which was explored for choosing the clipping constant. It was used for off-policy evaluation [47] and was further improved in [46]. Though *Interpol*'s window size parameter seems like it could be tuned using SLOPE, we note that SLOPE requires the bias to be monotonic in the free parameter. While we have seen in some experiments that the bias and variance are not necessarily monotonic in T, we tried this approach (not exposed) and found it to work well overall and hence can serve as a starting point for practical applications. However, a full assessment of this approach is still needed.

When *Interpol* is used for learning, the window size is a hyperparameter that can be tuned alongside other hyperparameters (such as learning rate, neural network structure, etc.) in the spirit of supervised learning problems, using common hyperparameter optimization techniques. The evaluation of the trained policy could again be done using the *SLOPE* procedure. From a practical perspective, window sizes of 1 and 2 work well, which could serve as a starting point for tuning policies.

8 CONCLUSION

We have introduced a novel counterfactual estimator for ranking evaluation, called *Interpol*, that spans a range of estimators between the IPM and PBM. *Interpol* has a favorable MSE, especially in the realistic situation when the modeling assumptions of the PBM are not fully satisfied. Furthermore, *Interpol* provides a rich modeling space to best match application requirements such as the visual layout of a ranking system. Our window system formalism allows flexibility in how much we trust the PBM and weakens the common assumption of full support. With *Interpol* we introduce a novel class of estimators that allows us to model a bias-variance trade-off explicitly, which leads to better evaluation and learning. We expect that further practical improvements can be achieved by combining our estimator with doubly-robust methods or self-normalization, which we conjecture will lead to further variance reduction.

REFERENCES

[1] A. Agarwal, S. Basu, T. Schnabel, and T. Joachims. 2017. Effective Evaluation using Logged Bandit Feedback from Multiple Loggers. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

[2] 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.

[3] 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* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 4–14. https://doi.org/10.1145/3308558.3313697

[4] 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.

[5] Imad Aouali, Victor-Emmanuel Brunel, David Rohde, and Anna Korba. 2023. Exponential smoothing for off-policy learning. In *International Conference on Machine Learning*. PMLR, 984–1017.

[6] Giuseppe Di Benedetto, Alexander Buchholz, Ben London, Matej Jakimov, Yannik Stein, Jan Malte Lichtenberg, Vito Bellini, and Matteo Ruffini. 2023. Contextual position bias estimation using a single stochastic logging policy. In *RecSys 2023 Workshop on Learning and Evaluating Recommendations with Impressions (LERI 2023)*. https://www.amazon.science/publications/contextual-position-bias-estimation-using-a-single-stochastic-logging-policy

[7] 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.

[8] Sebastian Bruch, Shuguang Han, Michael Bendersky, and Marc Najork. 2020. A stochastic treatment of learning to rank scoring functions. In *Proceedings of the 13th international conference on web search and data mining*. 61–69.

[9] Alexander Buchholz, Jan Malte Lichtenberg, Giuseppe Di Benedetto, Yannik Stein, Vito Bellini, and Matteo Ruffini. 2022. Low-variance estimation in the Plackett-Luce model via quasi-Monte Carlo sampling. *arXiv preprint arXiv:2205.06024* (2022).

[10] Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In *Proceedings of the learning to rank challenge*. PMLR, 1–24.

[11] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. Click models for web search. *Synthesis lectures on information concepts, retrieval, and services* 7, 3 (2015), 1–115.

[12] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*. 87–94.

[13] Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. 2014. Doubly Robust Policy Evaluation and Optimization. *Statist. Sci.* 29, 4 (2014), 485–511.

[14] M. Dudik, J. Langford, and L. Lihong. 2011. Doubly Robust Policy Evaluation and Learning. In *International Conference on Machine Learning*.

[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 *International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[17] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. 2018. More Robust Doubly Robust Off-policy Evaluation. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 1447–1456. https://proceedings.mlr.press/v80/farajtabar18a.html

[18] 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.

[19] 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.

[20] Katja Hofmann, Anne Schuth, Alejandro Bellogin, and Maarten De Rijke. 2014. Effects of Position Bias on Click-Based Recommender Evaluation.. In *ECIR*, Vol. 14. Springer, 624–630.

[21] 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.

[22] Guido W Imbens and Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

[23] E. Ionides. 2008. Truncated Importance Sampling. *Journal of Computational and Graphical Statistics* 17, 2 (2008), 295–311.

[24] 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.

[25] 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.

[26] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (Cambridge, United Kingdom) *(WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 781–789. https://doi.org/10.1145/3018661.3018699

[27] Haruka Kiyohara, Yuta Saito, Tatsuya Matsuhiro, Yusuke Narita, Nobuyuki Shimizu, and Yasuo Yamamoto. 2022. Doubly robust off-policy evaluation for ranking policies under the cascade behavior model. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 487–497.

[28] Alexey Kurennoy, John Coleman, Ian Harris, Alice Lynch, Oisin Mac Fhearai, and Daphne Tsatsoulis. 2022. A General Framework for Pairwise Unbiased Learning to Rank. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*. 204–213.

[29] Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. 2012. An Unbiased Offline Evaluation of Contextual Bandit Algorithms with Generalized Linear Models. In *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2 (Proceedings of Machine Learning Research, Vol. 26)*, Dorota Glowacka, Louis Dorard, and John Shawe-Taylor (Eds.). PMLR, Bellevue, Washington, USA, 19–36. https://proceedings.mlr.press/v26/li12a.html

[30] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 297–306.

[31] Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, Shan Muthukrishnan, Vishwa Vinay, and Zheng Wen. 2018. Offline evaluation of ranking policies with click models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1685–1694.

[32] Jan Malte Lichtenberg, Alexander Buchholz, Giuseppe Di Benedetto, Matteo Ruffini, and Ben London. 2023. Double Clipping: Less-Biased Variance Reduction in Off-Policy Evaluation. *arXiv preprint arXiv:2309.01120* (2023).

[33] Ben London, Alexander Buchholz, Giuseppe Di Benedetto, Jan Malte Lichtenberg, Yannik Stein, and Thorsten Joachims. 2023. Self-normalized off-policy estimators for ranking. In *RecSys 2023 Workshop on Causality, Counterfactuals & Sequential Decision-Making (CONSEQUENCES)*. https://www.amazon.science/publications/self-normalized-off-policy-estimators-for-ranking

[34] 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*. 895–903.

[35] Harrie Oosterhuis. 2021. Computationally efficient optimization of plackett-luce ranking models for relevance and fairness. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1023–1032.

[36] Harrie Oosterhuis. 2022. Learning-to-Rank at the Speed of Sampling: Plackett-Luce Gradient Estimation With Minimal Computational Complexity. *arXiv preprint arXiv:2204.10872* (2022).

[37] Harrie Oosterhuis. 2023. Doubly robust estimation for correcting position bias in click feedback for unbiased learning to rank. *ACM Transactions on Information Systems* 41, 3 (2023), 1–33.

[38] Harrie Oosterhuis and Maarten de Rijke. 2020. *Policy-Aware Unbiased Learning to Rank for Top-k Rankings*. Association for Computing Machinery, New York, NY, USA, 489–498. https://doi.org/10.1145/3397271.3401102

[39] Reuven Y Rubinstein and Dirk P Kroese. 2016. *Simulation and the Monte Carlo method*. John Wiley & Sons.

[40] Matteo Ruffini, Vito Bellini, Alexander Buchholz, Giuseppe Di Benedetto, and Yannik Stein. 2022. Modeling Position Bias Ranking for Streaming Media Services. In *The Web Conference*.

[41] Yuta Saito. 2020. Doubly robust estimator for ranking metrics with post-click conversions. In *Fourteenth ACM Conference on Recommender Systems*. 92–100.

[42] Ashudeep Singh and Thorsten Joachims. 2019. Policy learning for fairness in ranking. *Advances in Neural Information Processing Systems* 32 (2019).

[43] Yi Su, Pavithra Srinath, and Akshay Krishnamurthy. 2020. Adaptive estimator selection for off-policy evaluation. In *International Conference on Machine Learning*. PMLR, 9196–9205.

[44] A. Swaminathan and T. Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *Neural Information Processing Systems*.

[45] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. 2017. Off-policy evaluation for slate recommendation. *Advances in Neural Information Processing Systems* 30 (2017).

[46] George Tucker and Jonathan Lee. 2021. Improved Estimator Selection for Off-Policy Evaluation. In *Workshop on Reinforcement Learning Theory at the 38th International Conference on Machine Learning*.

[47] Takuma Udagawa, Haruka Kiyohara, Yusuke Narita, Yuta Saito, and Kei Tateno. 2022. Policy-Adaptive Estimator Selection for Off-Policy Evaluation. *arXiv*

preprint arXiv:2211.13904 (2022).

[48] Nikos Vlassis, Aurelien Bibaut, Maria Dimakopoulou, and Tony Jebara. 2019. On the Design of Estimators for Bandit Off-Policy Evaluation. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 6468–6476. https://proceedings.mlr.press/v97/vlassis19a.html

[49] Hongning Wang, ChengXiang Zhai, Anlei Dong, and Yi Chang. 2013. Content-aware click modeling. In *Proceedings of the 22nd international conference on World Wide Web*. 1365–1376.

[50] 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.

[51] 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.

[52] Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudík. 2017. Optimal and Adaptive Off-policy Evaluation in Contextual Bandits. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 3589–3597. https://proceedings.mlr.press/v70/wang17a.html

[53] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.

[54] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. 2021. Policy-gradient training of fair and unbiased ranking functions. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1044–1053.