# Generalized Position-Based Model: Rethinking Position Weights in Ranking Off-Policy Evaluation

Norman Knyazev*
Radboud University
Nijmegen, Netherlands
norman.knyazev@ru.nl

Vito Bellini
Amazon Music
Berlin, Germany
vitob@amazon.de

Huseyin Yurtseven
Amazon Music
Berlin, Germany
hyurts@amazon.de

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

## Abstract

Off-policy evaluation (OPE) estimates the performance of new recommendation policies using logged data, thus enabling fast, safe and inexpensive iteration prior to costly A/B tests. To evaluate ranking policies, existing OPE estimators all make structural assumptions about user behavior, leading to a spectrum of trade-offs between bias and variance. The recently proposed INTERPOL estimator navigates these trade-offs through a *window system* that defines how clicks at different positions are combined. However, this approach has two key limitations: the window configuration must be specified *a priori*, which can impede practical use, and all positions within a window are weighted uniformly, regardless of their relative utility, potentially limiting accuracy. To address these gaps, we introduce Generalized PBM (GPBM), an estimator that learns position-specific weights by minimizing an approximate upper bound on the estimation error. GPBM retains the unbiasedness guarantees of INTERPOL while significantly simplifying its hyperparameter selection. Our experiments demonstrate that GPBM provides more accurate and robust estimates across a wide range of position bias misestimation levels, logging policies, and data sizes.

## CCS Concepts

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

## Keywords

Off-policy evaluation, Ranking, Bias, Variance

**ACM Reference Format:**
Norman Knyazev, Vito Bellini, Huseyin Yurtseven, and Ben London. 2026. Generalized Position-Based Model: Rethinking Position Weights in Ranking Off-Policy Evaluation. In *20th ACM Conference on Recommender Systems (RecSys '26), September 27-October 02, 2026, Minneapolis, MN, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3773078.3831731

## 1 Introduction

A key requirement for industrial recommender systems is the ability to evaluate new policies prior to A/B testing, yet online evaluation is costly and carries the risk of exposing users to suboptimal experiences [27]. Off-policy evaluation (OPE) addresses this by estimating how a new *target* policy would perform using only historical interactions collected under an existing *logging* policy [7, 46]. This enables rapid iteration and selection of candidate policies before committing resources to online experiments [44]. Like all estimation, OPE must navigate a trade-off between *bias*—how accurately the estimate captures the statistic of interest—and *variance* (the estimator's variability caused by randomness). For ranking applications, the bias is influenced by positional effects: items at higher positions generally receive more interactions than those at lower positions, irrespective of their relevance to the user [22].

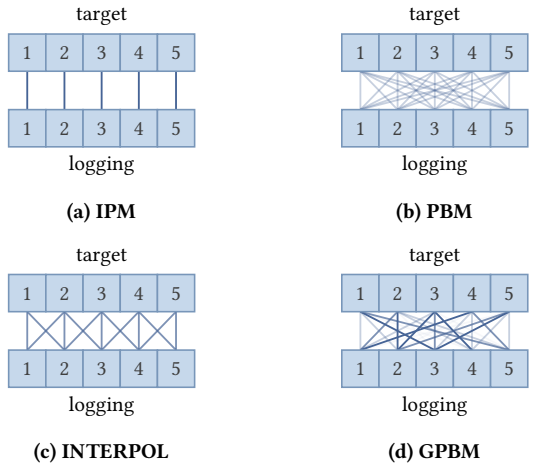



**Figure 1: Comparison of ranking estimators. Connection opacity represents the strength of the contribution of clicks across positions in the logged data towards predicting the clicks across each position under the target policy.**

Many OPE estimators counteract bias using *inverse propensity scoring (IPS)*, an importance weighting technique which corrects the distribution of interactions based on their relative observation probability under the logging and target policies [7]. Yet applying IPS to ranking applications is challenging: as ranking length increases, these probabilities become vanishingly small, leading to high-variance estimates that are sensitive to noise [25]. To reduce variance, ranking OPE estimators typically assume that user behavior follows a particular *click model* [11]. The two most widely adopted are the item-position model (IPM) [28], which assumes clicks at different positions are mutually independent, and the position-based model (PBM) [37], which further assumes that click probabilities decompose into item-specific relevance and position-specific examination components. These additional assumptions allow PBM to achieve lower-variance estimates, but may introduce significant bias when misspecified; conversely, IPM's weaker

*Work done while interning at Amazon Music.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
*RecSys '26, Minneapolis, MN, USA*
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2284-4/2026/09
https://doi.org/10.1145/3773078.3831731

assumptions result in lower bias at the cost of higher variance. The recently proposed INTERPOL estimator [8] seeks a middle ground by applying PBM only within *windows* around each position, achieving lower mean squared error (MSE) than either IPM or PBM with appropriately defined windows. The flexibility of INTERPOL's window formalism actually generalizes the IPM and PBM, with these click models sitting at opposite extremes of a spectrum of bias-variance trade-offs.

In spite of its flexibility, the window formalism has one key limitation: it treats all positions within a window as being equally important. We argue that this uniform weighting can lead to suboptimal performance when the true positional dependencies are dataset-specific or do not match the chosen window system. We address these limitations by proposing Generalized PBM (GPBM), a new ranking OPE estimator that learns position-specific weights from data, rather than assuming fixed patterns. GPBM subsumes IPM, PBM, and INTERPOL as special cases while enabling more flexible modeling of positional dependencies. Our data-driven weight learning method relies on highly interpretable and robust hyperparameters, and our experiments demonstrate consistent improvements over existing methods without significant computational overhead.

We release all experimental details, along with the implementation source code and supplemental appendices, to ensure the reproducibility of our findings.[1]

## 2 Related Work

OPE is a general framework to estimate the performance of a new policy using logged data from users interacting with a different policy [7, 44, 46]. Central to most OPE estimators is a form of importance weighting known as IPS [19, 23], which re-weights logged rewards by a ratio of observation probabilities induced by the target and logging policies. This enables unbiased estimation of the target policy's expected utility (a.k.a. *reward*) without actually deploying it. Since it is relatively fast and inexpensive, and does not expose users to potentially suboptimal experiences, OPE is often used as a precursor to the more costly A/B testing [13, 16, 50].

The challenge with IPS is that it often suffers from high variance, particularly when the logging policy assigns low probabilities to actions favored by the target policy, or when logged data is limited [3, 12, 37, 48]. There are several common methods to reduce variance. Clipping (a.k.a. truncation) [3, 29] bounds IPS weights to a fixed range, thus reducing the effect of excessively high weights, at the cost of introducing bias. Self-normalized IPS (SNIPS) [30, 48] leverages multiplicative control variates to reduce variance, which introduces finite-sample bias, but is asymptotically unbiased. Similarly, doubly robust (DR) methods [12, 25, 36] use additive control variates from a learned reward model, which reduces variance when the model is accurate, but potentially exacerbates it otherwise.

As there are many variance reduction techniques, each with their own bias-variance trade-off, recent work has focused on estimator selection and ensembling. SLOPE [47] is a procedure for selecting the lowest-variance estimator that is plausibly unbiased, relying on an assumption that base estimators can be ordered by monotonically decreasing variance and increasing bias (such as clipped IPS). PAS-IF [49] is another estimator selection procedure that sub-samples data to estimate MSE, avoiding any monotonicity assumptions. Ensembling methods combine multiple base estimators to form a single estimate with better bias-variance properties. BLUE [21] learns an optimal linear combination of estimators—which, under an unbiasedness assumption and with sufficient data, is guaranteed to perform at least as well as the best individual estimator. Similarly, OPERA [33] learns an optimal linear combination of estimators using bootstrapping to estimate base estimator MSE. All of these variance reduction and estimator selection techniques are complementary to our work and may be combined with it to further improve performance.

The variance problem is especially acute in ranking settings, where the probability of an entire ranking (i.e., an ordered selection of multiple items) can be extremely small, leading to arbitrarily large importance weights [25]. The above mitigations are usually insufficient, so they are typically combined with structural assumptions about user behavior—so-called *click models* [11]. A relatively mild assumption is that clicks at different positions within the same ranking are mutually independent—known as the IPM [28]—which yields an estimator with slight variance reduction (compared to "naive" IPS), but still relatively high variance. PBM [37] builds on IPM by further assuming that click probabilities factor into a product of item-specific relevance and position-specific examination probability (a.k.a. *position bias*). This further reduces variance but may introduce bias when the decomposition assumption is violated or the position bias curve is misspecified. Position bias estimation is a known challenge, as intervention-based methods may excessively harm the user experience [1, 14, 23], while offline methods often fail to provide sufficiently accurate estimates [2, 4, 10, 20, 42].

IPM and PBM can be viewed as opposite ends of a spectrum of bias-variance trade-offs. The INTERPOL estimator [8] seeks to find a middle ground between these extremes by combining IPM and PBM weighting. When appropriately configured, INTERPOL can achieve lower MSE than either click model, even when the position bias curve is misspecified. However, INTERPOL has two key limitations: it introduces hyperparameters that strongly dictate how the click models are combined, and it combines them using a uniform weighting that ignores inter-positional dependencies. (We discuss this in greater detail in Section 3.) Our estimator, GPBM, addresses these limitations by learning position-specific weights from data, only relying on a single hyperparameter vector that, as we later demonstrate, does not require careful tuning. Like INTERPOL, GPBM generalizes the IPM and PBM, and also recovers INTERPOL using binary weights.

## 3 Background: Off-Policy Evaluation

Let $\pi$ denote a policy that (given some contextual information) induces a ranking (i.e., ordering), $y$, on a given set of items $d \in \mathcal{D}$. When $y$ is shown to a user, it receives feedback, commonly referred to as *reward*. Without too much loss of generality, we will assume that reward is measured by clicks, denoted by the binary variable $c(y, k) \in \{0, 1\}$ indicating whether the user clicked on the item ranked at position $k \in \{1, \dots, K\}$. The central goal of OPE is to estimate the expected per-ranking reward, $\Delta(\pi) = \mathbb{E}_\pi\left[\sum_{k=1}^{K} c(y, k)\right]$, using rankings and rewards that were logged

[1] https://github.com/amazon-science/gpbm

under a previously deployed policy, $\pi_0$, commonly referred to as the *logging policy*.

Importantly, the logged rewards (clicks) depend on the choice and order of items shown to the user by the logging policy. Furthermore, due to *position bias*, items shown in certain (usually, higher) positions are likely to receive more clicks, irrespective of their relevance. To correct these effects, IPS-based estimators *re-weight* rewards so as to simulate the distribution of rankings under the target policy, $\pi$, and any position biases that apply to where $\pi$ ranks items. The generic form of an IPS-based estimator is given by:

$$\hat{\Delta}(\pi) \;=\; \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} \underbrace{c(y^{(i)},k)}_{\text{click?}} \cdot \underbrace{w(y_k^{(i)},k)}_{\text{weight}}, \tag{1}$$

where $y^{(i)}$ denotes the $i$'th logged ranking, $y_k^{(i)}$ denotes the item ranked at position $k$, and $c(y^{(i)},k)$ indicates whether the item at position $k$ was clicked. The choice of weights, $w$, reflects our assumptions about user behavior—i.e., the click model. Two common click models are IPM and PBM [28, 37], whose respective estimators are defined by the following importance weights:

$$w_{\text{IPM}}(d,k) = \frac{\pi(d,k)}{\pi_0(d,k)}, \quad w_{\text{PBM}}(d,k) = \frac{\sum_{k'=1}^{K}\pi(d,k')\cdot\hat{p}_{k'}}{\sum_{k'=1}^{K}\pi_0(d,k')\cdot\hat{p}_{k'}}, \tag{2}$$

where we have substituted $d = y_k^{(i)}$ to denote the ranked item.

IPM assumes that the probability a user clicks on an item at a certain position is independent of all other items and positions in the ranking [28]. Accordingly, $\hat{\Delta}_{\text{IPM}}$ uses clicks on item $d$, which $\pi_0$ ranked at position $k$, to predict the CTR for the same item and position under $\pi$. Since clicks are independent, the importance weight $w_{\text{IPM}}(d,k)$ (Equation 2, left) is a ratio of *marginal* probabilities (a.k.a. *propensities*), denoted $\pi(d,k)$ and $\pi_0(d,k)$, that each policy ranks $d$ at position $k$. As the IPM makes relatively mild assumptions, its estimator tends to have low *bias*. However, significant differences between $\pi_0$ and $\pi$—particularly when $\pi(d,k) \gg \pi_0(d,k)$ for some $d$ and $k$, e.g. when $\pi_0$ is highly deterministic—may result in high-magnitude weights. This can in turn lead to high estimator *variance* and thus unreliable estimates [43], especially when data is limited.

PBM, on the other hand, additionally assumes that the click probability can be factorized into a product of relevance to the user, denoted $\theta_d \in [0,1]$, and the probability of the user examining position $k$ (independent of content), which is captured by the (true) position bias curve $p = (p_1,\ldots,p_K)$ [37]. The PBM estimator aggregates clicks for an item $d$ across all positions proportional to the marginal placement probabilities $\pi(d,k')$ and the position bias $p_{k'}$, for each position $k' = 1,\ldots,K$. Its corresponding importance weight, $w_{\text{PBM}}(d,k)$ (defined in Equation 2, right), has lower magnitude than $w_{\text{IPM}}(d,k)$. While this allows PBM to further reduce estimator variance, in practice we only have access to an estimated position bias curve $\hat{p} = (\hat{p}_1,\ldots,\hat{p}_K)$, which may differ from $p$ [1, 2, 4]. As a consequence, PBM may be systematically over- or under-estimating $\Delta(\pi)$, introducing more bias in its reward estimates.

To provide a more favorable bias-variance trade-off in terms of MSE, the recently proposed INTERPOL estimator [8] applies PBM only among the *logged* positions $k$ contained in a window $W(j)$ defined for each *target* rank $j$; outside this window, the IPM is used. This yields importance weights:

$$w_{\text{INT}}(d,k) = \overbrace{\sum_{j=1}^{K}}^{\text{target positions } j} \pi(d,j) \cdot \frac{\overbrace{\mathbb{1}[k\in W(j)]}^{\substack{\text{logged position } k\\ \text{within window of } j?}}\cdot\,\hat{p}_j}{\sum_{k'\in W(j)}\pi_0(d,k')\cdot\hat{p}_{k'}}. \tag{3}$$

Buchholz et al. focused on a *banded* window system, $W_s(j) = \{j-s,\ldots,j+s\}$, parameterized by window size $s$. They empirically demonstrated that INTERPOL with an optimal choice of $s$ yielded lower MSE compared to both IPM and PBM (which can also be recovered by setting $s=0$ and $s=K-1$, respectively). Figure 1(a-c) illustrates how IPM, PBM and INTERPOL with $s=1$ combine clicks across positions. Furthermore, the authors show that if the true position bias curve $p$ is known, then INTERPOL provides an unbiased estimate of $\Delta(\pi)$ for any window size. However, the unbiasedness guarantee relies on knowing the true position bias curve, and MSE only improves for the right choice of windows—conditions that are difficult to satisfy in practice. Buchholz et al. [8] suggest that SLOPE could potentially be used to tune INTERPOL's hyperparameters, e.g. by picking $s$ for the banded window system. Yet they also note that INTERPOL invalidates SLOPE's key monotonicity assumption, which may in turn invalidate SLOPE's applicability. Furthermore, INTERPOL's structural assumptions are at odds with how it combines clicks: for many window systems, such as the banded one, there is an implicit assumption that (relative) errors in position bias estimation are smaller when closer to the target position; yet all within-window positions are treated equally, regardless of distance. Consequently, estimates for target position $j$ can be significantly biased if $\hat{p}_k$ is misestimated for any within-window position $k$.

## 4 Generalized Position-Based Model

Motivated by the above discussion, we aim to develop an estimator that functions comparably to INTERPOL, but without rigid structural constraints that must be determined *a priori*. Ideally, the estimator should automatically decide how to combine clicks from different positions, adapting these decisions to the available data.

We begin by observing that INTERPOL's window membership indicator, $\mathbb{1}[k\in W(j)]$, can be viewed as a binary-valued weight determining how interactions across logged positions $k$ are combined to estimate the reward for target position $j$. Crucially, we note that one could replace the indicator with a continuous-valued weight. We therefore replace it with a weighting function, $F:\mathbb{Z}^+\times\mathbb{Z}^+\to\mathbb{R}_{\geq 0}$, where $F(j,k)$ regulates the influence of logging position $k$ on the reward estimate for target position $j$. Observe that this formulation recovers INTERPOL when the weights are restricted to $\{0,1\}$, so the resulting estimator is a strict generalization of INTERPOL, and therefore also IPM and PBM. The importance weights for this estimator, which we term GPBM, have the following form:

$$w_{\text{GPBM}}(d,k) = \sum_{j=1}^{K}\frac{F(j,k)\cdot\pi(d,j)\cdot\hat{p}_j}{\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}}. \tag{4}$$

The weighting function $F$ can also be seen as a $K\times K$ matrix, whose entries $F_{j,k}$ determine the specific ranking estimator (Figure 1d).

Analogous to Buchholz et al.'s proof for INTERPOL, we can show that, under similar assumptions of (i) correct position bias and (ii)

*common support under* $F$, where for every $(d, j)$ with $\pi(d, j) > 0$, there exists a position $k$ s.t. $\pi_0(d, k) > 0$ and $F(j, k) \cdot \hat{p}_k > 0$, GPBM yields an unbiased estimate of $\Delta(\pi)$ (see Appendix A for details). Consequently, $F(j, k)$ can be determined (almost) arbitrarily for every $(j, k)$ combination, leaving us free to choose the values that lead to low estimation error on our data.

## 4.1 Determining $F$

We now describe our approach to optimizing the entries of $F$ in a data-driven manner. Our aim is to minimize the MSE of the resulting estimate, given the dataset, position bias estimate and choice of target policy. However, since the true MSE value is not directly available (as it depends on the unknown expected reward), we instead propose to select $F$ by minimizing an approximate upper bound on the MSE.

Let the shorthand $\hat{\Delta}(\pi) := \hat{\Delta}(\pi; \hat{p}, D_N)$ denote the GPBM estimate of the expected reward for policy $\pi$, using an estimate $\hat{p}$ of the true position bias $p$, and logged dataset $D_N$ of $N$ rankings, their associated rewards and logging propensities. Using standard bias-variance decomposition [15], we have that

$$\mathrm{MSE}\big(\hat{\Delta}(\pi)\big) = \mathrm{Var}\big(\hat{\Delta}(\pi)\big) + \mathrm{Bias}\big(\hat{\Delta}(\pi)\big)^2, \tag{5}$$

$$\text{where } \mathrm{Bias}\big(\hat{\Delta}(\pi)\big) = \mathbb{E}[\hat{\Delta}(\pi)] - \Delta(\pi), \tag{6}$$

letting us approximate the bias and variance components separately.

To approximate the variance of GPBM, we follow the bootstrap-based approach of OPERA [33]. The key idea is to estimate the variance of our full-sample estimate by measuring its deviation from many, much smaller bootstrap sub-samples, with rescaling to account for the differences in sample size. We produce $B$ bootstrap estimates, $\hat{\Delta}(\pi; \hat{p}, D_M)$, each using $M \ll N$ sampled records, then calculate the bootstrap variance around the full-sample estimate, $\hat{\Delta}(\pi; \hat{p}, D_N)$. We then approximate the variance of the full-sample estimate around its (potentially biased) population mean, $\mathbb{E}[\hat{\Delta}(\pi)]$, rescaling the $M$-sample variance by $M/N$ as:

$$\mathrm{Var}\big(\hat{\Delta}(\pi)\big) = \mathbb{E}_{D_N}\Big[\big(\hat{\Delta}(\pi; \hat{p}, D_N) - \mathbb{E}[\hat{\Delta}(\pi)]\big)^2\Big] \tag{7}$$

$$\approx MN^{-1}B^{-1}\sum\nolimits_{b=1}^{B}\Big(\hat{\Delta}(\pi; \hat{p}, D_M^{(b)}) - \hat{\Delta}(\pi; \hat{p}, D_N)\Big)^2 \tag{8}$$

$$= \widehat{\mathrm{Var}}\big(\hat{\Delta}(\pi)\big) \tag{9}$$

From Equation 8, it is clear that the variance estimate decreases with the data size $N$, or when the bootstrapped estimates $\hat{\Delta}(\pi; \hat{p}, D_M^{(b)})$ do not significantly deviate from the full-data estimate $\hat{\Delta}(D_N)$. Such deviations are likely when the target policy departs significantly from the logging policy, causing high-magnitude importance weights. Importantly, using very small bootstrap samples of size $M \ll N$ has been shown to reduce the impact of such outliers on estimation accuracy [5, 18]. While OPERA uses this bootstrapping procedure to approximate the MSE of its base estimators, it may be unable to account for bias introduced by position bias misspecification, which does not vanish asymptotically. We account for this by incorporating an explicit bias component into our optimization objective.

Following MacKinnon and Smith [32], we approximate the bias of the estimate $\hat{\Delta}(\pi; p, D_N) =: \hat{\Delta}(p)$, which uses the true position bias curve $p$, by a first-order Taylor approximation around the bias of $\hat{\Delta}(\pi; \hat{p}, D_N) =: \hat{\Delta}(\hat{p})$, which uses the estimated curve $\hat{p}$. Since the estimator is unbiased when using $p$ (see Appendix A), we have:

$$0 = \mathrm{Bias}\big(\hat{\Delta}(p)\big) \approx \mathrm{Bias}\big(\hat{\Delta}(\hat{p})\big) + \nabla_{\hat{p}}\mathrm{Bias}\big(\hat{\Delta}(\hat{p})\big)^\top(p - \hat{p}), \tag{10}$$

where $\nabla_{\hat{p}}\mathrm{Bias}\big(\hat{\Delta}(\hat{p})\big)$ is the gradient of the bias w.r.t $\hat{p}$. Rearranging the terms and combining them with Equation 6 yields:

$$\big|\mathrm{Bias}\big(\hat{\Delta}(\hat{p})\big)\big| \approx \big|\nabla_{\hat{p}}\mathrm{Bias}\big(\hat{\Delta}(\hat{p})\big)^\top(\hat{p} - p)\big| \tag{11}$$

$$= \big|\nabla_{\hat{p}}\mathbb{E}[\hat{\Delta}(\hat{p})]^\top(\hat{p} - p) - \nabla_{\hat{p}}\Delta(\pi)^\top(\hat{p} - p)\big| \tag{12}$$

$$= \big|\nabla_{\hat{p}}\mathbb{E}[\hat{\Delta}(\hat{p})]^\top(\hat{p} - p)\big|. \tag{13}$$

The last identity follows from the fact that the true policy value, $\Delta(\pi)$, does not depend on $\hat{p}$; hence, the gradient, $\nabla_{\hat{p}}$, is zero. We can then upper-bound Equation 13 using the triangle inequality, and approximate the resulting bound via bootstrapping:

$$(13) \le \big|\nabla_{\hat{p}}\mathbb{E}[\hat{\Delta}(\hat{p})]\big|^\top\big|\hat{p} - p\big| \tag{14}$$

$$\approx \big|B^{-1}\sum\nolimits_{b=1}^{B}\nabla_{\hat{p}}\hat{\Delta}(\pi; \hat{p}, D_M^{(b)})\big|^\top\epsilon \tag{15}$$

$$= \widehat{\mathrm{Bias}}\big(\hat{\Delta}(\pi)\big)\,. \tag{16}$$

In the second line, $\epsilon = (|\hat{p}_1 - p_1|, \ldots, |\hat{p}_K - p_K|)$ denotes an upper bound on the position bias curve misestimation at each position, and functions as GPBM's sole hyperparameter. When $\epsilon_k$ is large, it indicates potentially high estimation error—so assigning high weight, $F(j, k)$, to such positions risks introducing bias, even if the actual difference between $\hat{p}_k$ and $p_k$ is small. We assume that an estimate of $\epsilon$ is available, even if potentially imprecise, e.g. based on prior knowledge or position bias estimation uncertainty [41, 42]. (In Section 6.3 we investigate the estimator's sensitivity to $\epsilon$.) Equation 15 also involves computing the gradient of the estimator, $\hat{\Delta}(\pi; \hat{p}, D_M^{(b)})$, with respect to $\hat{p}$, for each bootstrap fold $b$. This is easily derived manually (Appendix C) or using automatic differentiation [6, 38]. For computational efficiency, we also estimate this gradient using $M \le N$ rankings per bootstrap fold.

Substituting our approximation of bias, $\widehat{\mathrm{Bias}}$, and variance, $\widehat{\mathrm{Var}}$, into Equation 5 yields our fully differentiable approximate upper bound on the MSE:

$$\mathrm{MSE}\big(\hat{\Delta}(\pi)\big) \lessapprox \widehat{\mathrm{Var}}\big(\hat{\Delta}(\pi)\big) + \widehat{\mathrm{Bias}}\big(\hat{\Delta}(\pi)\big)^2\,. \tag{17}$$

Intuitively, $\widehat{\mathrm{Var}}$ is low when the entries of $F$ are more uniform, but that might come at a cost of increased bias. On the other hand, adapting $F$ to the data, to account for position bias estimation error ($\epsilon$), should reduce bias, while potentially increasing variance. Thus, we find the inevitable trade-off between bias and variance.

We use Equation 17 as an objective to optimize the entries of $F$, which we solve using standard stochastic gradient methods [24]. At every optimization step we: i) independently generate $B$ bootstrap samples $D_M^{(1)}, \ldots, D_M^{(B)}$ with propensities and clicks for $M$ rankings; ii) estimate the target policy value for each bootstrap sample, $D_M^{(b)}$, as well as the full data sample, $D_N$; iii) calculate the gradient of bootstrapped estimates w.r.t. $\hat{p}$; iv) use this gradient to estimate the approximate bias bound (Equation 15); v) calculate the spread of bootstrapped estimates around the full data estimate, rescaling it by $M/N$ (Equation 8); vi) compute the gradient of the combined expression (Equation 17) w.r.t. GPBM weights $F$; and finally, vii) take a step with the optimizer. The resulting approach is highly

efficient, and as we will see, able to adapt the learned estimator to the available data and our knowledge of the position bias.

## 5 Experimental Setup

We structure our evaluation around four research questions. We first evaluate whether GPBM yields more accurate off-policy estimates across settings that, as discussed in Section 2, critically affect ranking OPE: dataset size, logging policy stochasticity and position bias misestimation:

**RQ1** Does GPBM provide consistently lower MSE than baselines across different ranking settings?

Furthermore, as user behavior may not follow the PBM in practice:

**RQ2** Does GPBM also perform better when user behavior follows the IPM?

Moreover, GPBM's performance depends on $\epsilon$, the assumed upper bound on per-position deviation between the true and estimated bias curves:

**RQ3** How does under- or over-estimating $\epsilon$ affect GPBM's MSE?

Finally, to understand the mechanism behind GPBM's performance:

**RQ4** How does GPBM's $F$ matrix combine weights from different positions across settings?

### 5.1 Data

Our experiments use two public datasets: *Yahoo! Webscope* [9] and *MSLR-WEB30k* datasets [40]. Both datasets consist of user queries, their corresponding retrieved items, and expert relevance labels. Yahoo contains 29,921 queries with an average of 24 items per query, while MSLR contains 30,000 queries with an average of 125 items per query. Each query-item pair $(q, d)$ is represented by a set of query-item features, with a 5-level relevance label, $r_d \in [0, 4]$. Following common practice [34] we rescale $r_d$ to the range $[0, 1]$ and denote this by $\theta_d$.

To simulate user feedback, we extend the methodology of Buchholz et al. [8]. Specifically, we use the logging policy to re-rank (up to) the first 50 items of each query. For each query, we independently sample $n \in \{1, 5, 25\}$ rankings from the logging policy (described in Section 5.2), yielding $N$ total logged rankings across all queries. We then generate rewards for each ranking using the relevance labels and a given click model. In most of our experiments, we use the PBM, in which users examine each position $k \leq 10$ based on a decreasing position bias curve, $p_k = 1/\log_2(k+1)$, and click an inspected item $d$ proportional to its relevance $\theta_d$.

### 5.2 Policies

Both logging and target are Plackett-Luce ranking policies [31, 39] trained on the respective training sets of each fold. To simulate policy improvement, the stronger target policy has access to all features, while the weaker logging policy is trained without the top third most relevance-correlated features [26]. Each policy ranks items based on item scores produced by a neural network with two hidden layers of size 32, sigmoid activations, and a final layer $f(x) = 10 \cdot \tanh(x)$ constraining the model's logits to the range $[-10, 10]$ to avoid the prevalence of effectively deterministic rankings. Both policies are trained to minimize PL-Rank-3 NDCG loss [35], with 100 sampled rankings, via 20 epochs of Adam [24], with both learning rate and $L_2$ regularization set to 0.01.

Plackett-Luce policies generate rankings by repeatedly sampling items from a softmax distribution without replacement. The softmax is parameterized by an inverse temperature, $\tau^{-1}$, which controls how uniform (low $\tau^{-1}$) or peaked (high $\tau^{-1}$) the distribution is. In our experiments, we take $\tau^{-1} \in [0.2, 0.4, 1]$. We clip the propensities of the (Plackett-Luce) logging policy [26] to $10^{-7}$ so as to avoid extremely high importance weights when inverted.

### 5.3 Baselines and Evaluation

We compare GPBM to the following baselines: IPM, PBM, INTERPOL with window size selected via SLOPE, and their respective self-normalized versions SN-IPM, SN-PBM and SN-INT. Additionally, we examine the performance of ensemble methods BLUE and OPERA combining (self-normalized) IPM and PBM.[2] To establish an upper bound on INTERPOL's performance, we also include an oracle version, INT* (and its self-normalized version, SN-INT*), whose window size is chosen retroactively to minimize MSE.

Except for IPM, all of the above estimators rely on an estimate of the position bias curve, which in practice may be incorrect. To explore the effect of an incorrectly estimated curve, we simulate it following the approach of Buchholz et al. [8], defining the estimated position bias curve $\hat{p}$ as an exponential transformation of the true curve $p$, s.t. $\hat{p} = p^\alpha$, where $\alpha \in [0.6, 0.8, 1, 1.2, 1.4]$, as well as $\hat{p} = \max(p - 0.1, 0)$. We also evaluate the robustness of all models under a more complex *zigzag* setting $\hat{p} = p \pm z$, where the position bias curve is over-estimated at odd positions and under-estimated at even positions (see Appendix B for more details).

We optimize GPBM's $F$ matrix via the procedure outlined in Section 4.1, using Adam with learning rate 0.01, and sampling $B = 50$ bootstrap folds of size $M = \sqrt{N}$. Since the upper bound on position bias misestimation, $\epsilon_k$, may be over- or under-estimated, we evaluate a range of values, $\epsilon_k = \beta_k \cdot |\hat{p}_k - p_k|$. In our experiments, $\beta_k = \beta$ is either the same for all positions s.t. $\beta \in [0, 0.25, 0.5, 1, 1.5, 2, 3]$, reflecting a consistently optimistic or pessimistic bound on the position bias estimation error, or scales linearly across ranks between 0.5 and 1.5 via $\beta_k = 0.5 + (k-1)/(K-1)$ or $\beta_k = 1.5 - (k-1)/(K-1)$, where the error bound is underestimated for some positions and overestimated for others. For most experiments, we fix $\beta_k = \beta = 1.5$, reflecting a pessimistic upper-bound of the error.

We measure estimator performance using the MSE between the estimated policy value and its true (on-policy) average reward—which we compute as a product of relevance $\theta_d$, position bias $p_k$ (or $p_{q,k,d}$ for IPM user behavior in Subsection 6.2) and the target propensity $\pi(d, k)$. Unless otherwise noted, we report the average MSE over 5 folds, with 50 independent repeats of the data generating process for each fold. Error bars indicate the lowest and highest fold MSE. Due to space constraints, we report a selection of the above results, with full results available in Appendix D. Agent-assisted coding using Claude Opus 4.5 was used to produce experimental code, which we then extensively reviewed, adapted and validated.

[2] We observed that using only these base estimators led to consistently higher performance compared to using all estimators.

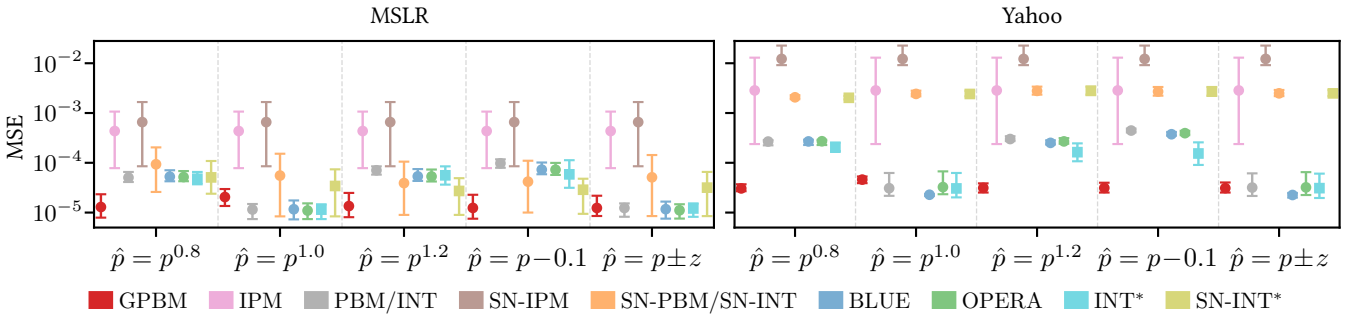



**Figure 2: Impact of position bias misestimation, with fixed inverse temperature $\tau = 0.6$ and $n = 5$ sampled rankings per query.**

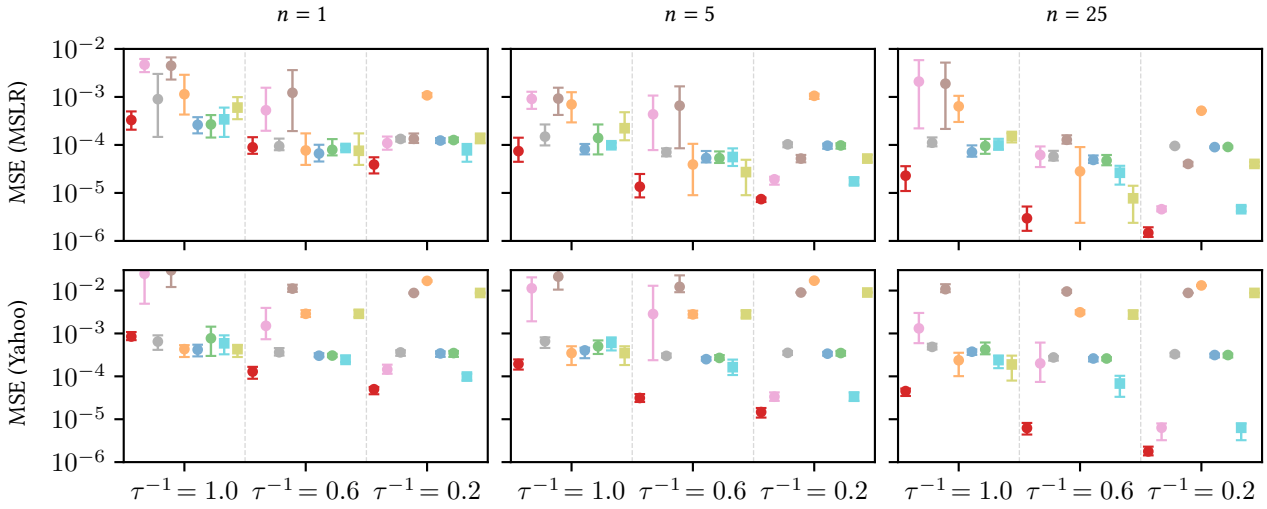



**Figure 3: Impact of logging policy stochasticity (via inverse temperature $\tau^{-1}$) and dataset size (via number of sampled rankings per query $n$). Misestimated position bias fixed at $\hat{p} = p^{1.2}$. Models as in Figure 2.**

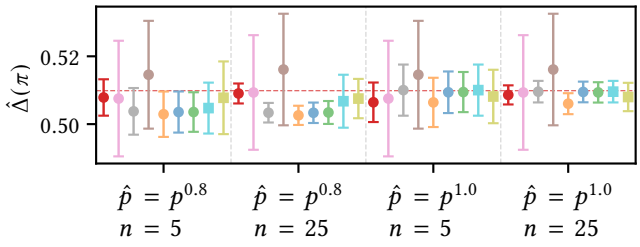



**Figure 4: Impact of position bias misestimation ($\hat{p}$) and dataset size (number of sampled rankings per query $n$) on estimator predictions on MSLR. Logging policy stochasticity fixed at $\tau^{-1} = 0.6$. Dotted red line denotes the true value of the target policy $\Delta(\pi)$. Error bars denote two standard deviations from the mean over 50 repeats. Models as in Figure 2.**

## 6 Results

### 6.1 Overall Performance (RQ1)

We first investigate how GPBM compares to the baselines across several dimensions that affect bias and variance: position bias misestimation, logging policy randomness and dataset size. Figure 2 plots the average MSE of all estimator across 5 folds, reflecting their bias-variance trade-off, for different misestimated position bias curves $\hat{p}$, with the logging policy's temperature and dataset size set to a "medium" value of $\tau^{-1} = 0.6$ and $n = 5$ respectively.

We find that the performance of the baselines varies across both position bias misestimation and datasets. Unsurprisingly, the effect of position bias misestimation is particularly strong for PBM, which achieves the lowest MSE on both datasets under the correctly estimated curve $\hat{p} = p$, yet progressively deteriorates under severe misestimation $\hat{p} = p^{\alpha}$, $\alpha \neq 0$, or when all positions are underestimated equally ($\hat{p} = p - 0.1$). Similarly, other estimators, such as BLUE and OPERA, perform well on some bias-dataset combinations but not others. Interestingly, oracle INTERPOL and its self-normalized version, which preemptively know the optimal window size, collectively obtain the lowest MSE across all settings. However, in practice, this optimal value is not known, and SLOPE appears to always select the largest window size, making (SN-)INT equivalent to (SN-)PBM. As such, how to choose the best ranking estimator and its parameters remains unclear.

Meanwhile, GPBM consistently demonstrates strong performance across most (mis-)estimated curves on both datasets. In most cases,

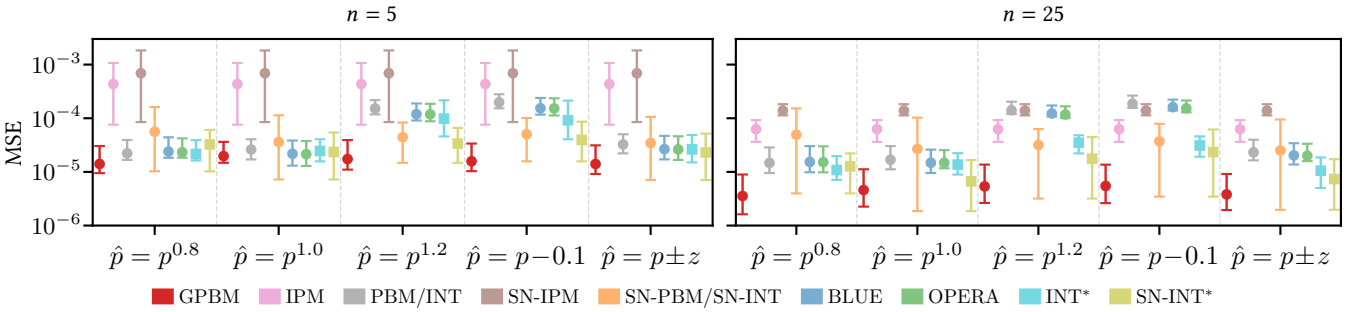



**Figure 5: Impact of $\hat{p}$ misestimation and data size (via $n$) on MSE under complex user behavior on MSLR.**

GPBM achieves substantially lower MSE than even the oracle INTERPOL (INT*), with over 6-fold improvement using overestimated curve $\hat{p} = p^{0.8}$ on Yahoo. Regardless of whether the position bias is correctly specified ($\hat{p} = p^{1.0}$) or highly irregular ($\hat{p} = p \pm z$), GPBM's performance remains comparable with the strongest baselines.

To understand estimator sensitivity to variance-inducing factors, Figure 3 plots MSE with respect to logging policy stochasticity (controlled via inverse temperature $\tau^{-1}$) and dataset size (controlled via the number of sampled rankings per query). Throughout, position bias misestimation is fixed to $\hat{p} = p^{1.2}$.

As with bias, here we also observe varied performance amongst the baselines. For instance, IPM is among the weakest baselines under the most deterministic logging policy ($\tau^{-1} = 1$), yet is consistently close to the optimal INTERPOL performance under more exploratory logging policies (lower $\tau^{-1}$) and with more sampled rankings. Other estimators, such as PBM, SN-IPM and the ensemble methods, show only minor MSE improvements with larger data sizes, suggesting they may be affected more by bias than variance.

In contrast, we again find that GPBM consistently performs well across all settings, benefiting from both more stochastic logging policies and more data. In particular, it achieves the lowest MSE for $n \geq 5$, under all logging policies. Notably, GPBM's performance is weakest in the most variance-inducing setting, $n = 1$ with $\tau = 1$. We hypothesize that this is due to the difficulty in optimizing $F$ under low effective sample size [45]. Nevertheless, when there is sufficient data and randomization, GPBM achieves consistent large improvements over all baselines.

Finally, Figure 4 shows the policy estimates of all models for the 50 repeats across a subset of settings on the first fold of MSLR. We can see that GPBM improves over the baselines in terms of both bias *and* variance when $\hat{p} = p^{0.8}$ (Figure 4, left). Furthermore, unlike most estimators, increasing the amount of data (via $n$, Figure 4, center-left), appears not only to bring its individual predictions closer to $\Delta(\pi)$, but also improve its mean estimate, suggesting GPBM's ability to focus on bias reduction as the variance is naturally reduced with more data. Interestingly, when the true position bias curve is provided (Figure 4, center-right), GPBM actually exhibits higher bias. We hypothesize that this is caused by overfitting the $\widehat{\text{Var}}$, as $\epsilon$ and thus the bias component of $\widehat{\text{MSE}}$ in this setting are set to 0. Consistent with this explanation, increasing the dataset size reduces the observed bias (Figure 4, right), which may come from reduced overfitting when optimizing $\widehat{\text{Var}}$.

Overall, we see that *GPBM proves robust to position bias misestimation, logging policy stochasticity, and dataset size, consistently outperforming all ranking OPE baselines across most settings* (**RQ1**).

## 6.2 Complex User Behavior (RQ2)

In practice, user click behavior may follow more complex patterns than the PBM, prompting our second research question: *what if the PBM assumptions do not hold?* To answer this, we construct a click model that adheres to the IPM (i.e., clicks do not depend on other positions' content) but not the PBM. Each query-item pair, $(q, d)$, is given its own position bias curve, $p_{q,d,k} = p_k + (p_k - p^{1.4}) \cdot (2 \cdot f(q,d) - 1)$, where $f(q,d)$ maps $(q,d)$ to a quantile between 0 and 1. To ensure non-uniform spread of curves across items, $f$ sorts items based on the scores assigned to them by a separate policy trained only on items' title features, representing their visual salience in a ranking. As a consequence, the average position bias curve over all items is still $p$, yet $p$ can no longer be used to accurately model clicks for any item, and the PBM assumptions thus no longer hold. Figure 5 shows estimator performance on MSLR using varying amounts of data generated from the above process, as well as varying levels of misestimation of the item-average curve.

Here, the baseline models generally perform comparably to our earlier results for the PBM setting. However, for many of them—especially PBM/INT and INT*—performance varies across different estimates' curves. In particular, in contrast with the PBM setting, $\hat{p} = p^{0.8}$ generally yields the best performance, while other curves lead to substantial performance losses. This strong baseline performance using the over-estimated item-average curve $\hat{p} = p^{0.8}$ is not surprising, as item title features that govern item-specific curves $p_{q,d,k}$ are also correlated with relevance. It follows that items placed higher by the logging policy are also more likely to have slower-decaying curves, which may be most similar to $\hat{p} = p^{0.8}$.

In contrast, GPBM's performance in the IPM setting does not differ strongly from its earlier performance in the PBM setting. Unlike the baselines, it maintains strong performance across all curve estimates, demonstrating that *GPBM is also robust when user behavior deviates from the PBM assumption* (**RQ2**).

## 6.3 Sensitivity to Misestimation Estimate (RQ3)

In order to determine the weight matrix $F$, GPBM relies on an estimate of $\epsilon_k$, the upper bound on the difference between the true position bias $p_k$ and its estimated value $\hat{p}_k$ at position $k$. Our third research question examines the effect of misestimating this

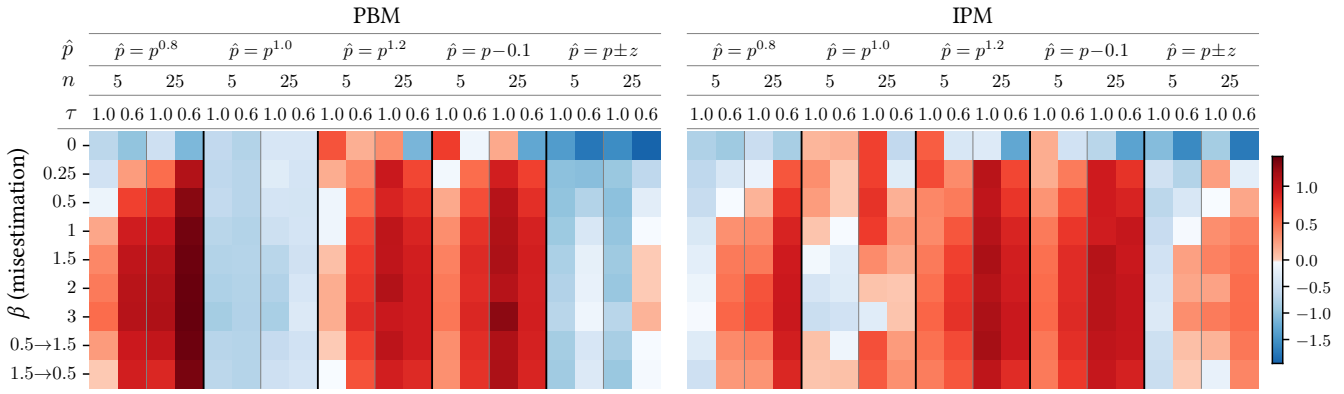



**Figure 6: Log-ratio (base 2) of the MSE of the best baseline and GPBM for different estimates of position bias misestimation bound $\epsilon$, controlled by $\beta$, under PBM and IPM user behavior across a selection of ranking settings on MSLR.**

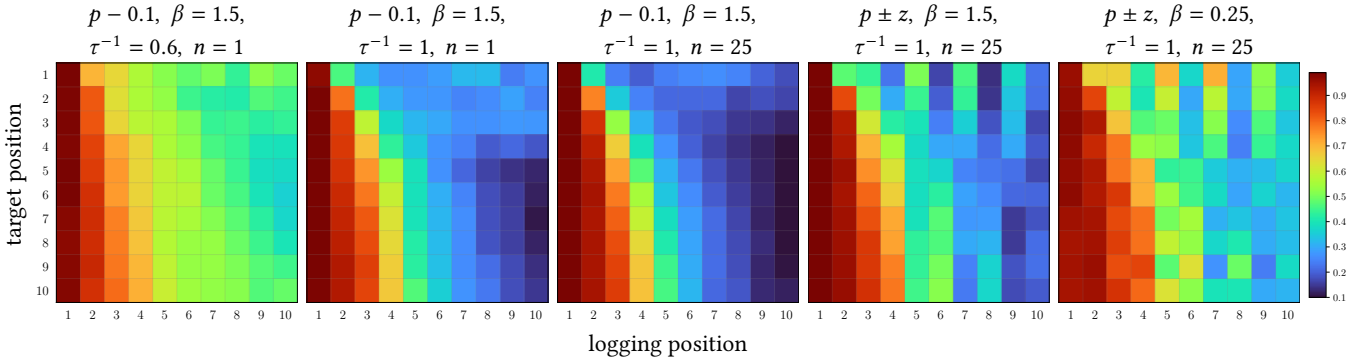



**Figure 7: Average learned $F$-matrices on MSLR for different misestimated position bias curves, error estimates, logging policies and amounts of data. Each row reflects the contributions of individual logging positions (columns) towards predicting a target position. Each row normalized by its highest weight and matrices then averaged over 50 repeats for a single fold.**

upper bound. We quantify the effect using the log-ratio (base 2) of the MSE values of the best non-oracle baseline and GPBM, where positive values indicate a relative improvement of GPBM over the baselines. Figure 6 shows the MSE log-ratio for earlier experiments using different values of $\epsilon_k = \beta_k \cdot |\hat{p}_k - p_k|$, where $\beta_k$ is either constant across the whole ranking or interpolates between two values. GPBM's performance is generally robust to the choice of $\epsilon$ (via $\beta$), irrespective of whether the position bias estimation error is either overestimated ($\forall \beta_k > 1$), underestimated ($\forall \beta_k < 1$) or both ($\exists k \neq k' : \beta_k > 1 \wedge \beta_{k'} < 1$). The fact that only significant over- or underestimation of the curve difference ($\forall \beta_k$ far from 1) degrades results suggests that *GPBM benefits from reasonably accurate*—but not exact—confidence estimates (**RQ3**).

## 6.4 Learned Adaptive Mechanism (RQ4)

Finally, to better understand the mechanism behind GPBM's consistent strong performance, we investigate the learned $F$ matrix, which governs how GPBM combines the IPM and PBM click models. Figure 7 visualizes the learned $F$-matrices for select combinations of position bias misestimation, estimates of this error (via $\beta$), logging policy stochasticity (via $\tau^{-1}$) and the available data amount (via $n$).

We find that with smaller amounts of data, for a consistently under-estimated curve and with medium-exploration logging policy (Figure 7, left), GPBM relies on clicks in early positions—and particularly the first position—to predict the click-through rate (CTR) across *all* positions of the target policy. Surprisingly, clicks at low positions are thus predicted on the basis of the top positions than the position itself. This effect is further amplified for a stronger logging policy (Figure 7, center-left) that more consistently places relevant items close to the top, and even more so when the dataset size is further increased (Figure 7, center). As here position bias misestimation estimate, $\epsilon_k$, is also the same across all positions, based on the above results, we hypothesize that GPBM prefers to rely on clicks in positions where it believes the effect of the position bias is weaker and clicks may thus be a less noisy indicator of relevance.

This also matches our zigzag position bias observations (Figure 7, center-right), where GPBM places more emphasis on odd positions, which it believes have the highest values of $p$. However, when it trusts the position bias estimate (Figure 7, right), position utilization becomes more equal, reflecting a different bias-variance trade-off. To answer **RQ4**, *GPBM appears to leverage positions where position bias is weaker, with this effect shaped by the logging policy, dataset size, and uncertainty in the position bias estimate.*

## 7 Discussion and Future Work

In this work, we introduced GPBM—a novel ranking OPE estimator that generalizes existing IPM, PBM and INTERPOL estimators. Rather than relying on fixed structural assumptions about user behavior, GPBM effectively learns these patterns using the available data and position bias estimate. Empirically, we demonstrated that GPBM achieves consistent improvements in MSE across a variety of settings. Broadly speaking, this suggests that positional dependencies in ranking are a structure that can be actively leveraged for more accurate policy evaluation.

As GPBM optimizes a pessimistic upper bound on the MSE, future work can extend GPBM by tightening this bound, e.g., by exploiting the correlations in the position bias estimation errors. A further extension could make the $F$ matrix contextual, so as to better accommodate different types of user behavior. Yet another improvement would be to combine GPBM with additive control variates, e.g., doubly-robust estimation [12, 25, 36] or optimal baseline corrections [17] for variance reduction. As our evaluation relies on simulated click data, which may not capture the nuances of real user behavior, an important next step will be to validate GPBM on real user interactions from an industrial system.

## Acknowledgments

We would like to thank Thomas Martynec, Sergei Bykov and Mazen Melouk for engaging discussions and their invaluable feedback.

## References

[1] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating position bias without intrusive interventions. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 474–482.

[2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 385–394.

[3] Imad Aouali, Victor-Emmanuel Brunel, David Rohde, and Anna Korba. 2023. Exponential Smoothing for Off-Policy Learning. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 984–1017.

[4] Grigor Aslanyan and Utkarsh Porwal. 2019. Position bias estimation for unbiased learning-to-rank in ecommerce search. In *International Symposium on String Processing and Information Retrieval*. Springer, 47–64.

[5] Krishna B Athreya. 1987. Bootstrap of the mean in the infinite variance case. *The annals of statistics* 15, 2 (1987), 724–731. doi:10.1214/aos/1176350371

[6] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. *JAX: Composable Transformations of Python+NumPy Programs*.

[7] Alexander Buchholz, Ben London, Giuseppe di Benedetto, and Thorsten Joachims. 2022. Off-Policy Evaluation for Learning-to-Rank via Interpolating the Item-Position Model and the Position-Based Model. *arXiv preprint arXiv:2210.09512* (2022).

[8] 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*. ACM, 1200–1210.

[9] Olivier Chapelle and Yi Chang. 2011. Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research* 14 (2011), 1–24.

[10] Mouxiang Chen, Chenghao Liu, Zemin Liu, Zhuo Li, and Jianling Sun. 2024. Identifiability matters: revealing the hidden recoverable condition in unbiased learning to rank. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna, Austria) *(ICML'24)*. JMLR.org.

[11] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool Publishers.

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

[13] Alex Egg. 2025. Off-policy Evaluation for Payments at Adyen. *arXiv preprint arXiv:2501.10470* (2025).

[14] Zhichong Fang, Aman Agarwal, and Thorsten Joachims. 2019. Intervention Harvesting for Context-Dependent Examination-Bias Estimation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 825–834.

[15] Stuart Geman, Elie Bienenstock, and René Doursat. 1992. Neural networks and the bias/variance dilemma. *Neural computation* 4, 1 (1992), 1–58.

[16] Ritam Guha and Nilavra Pathak. 2025. Off-Policy Evaluation and Counterfactual Methods in Dynamic Auction Environments. *arXiv preprint arXiv:2501.05278* (2025).

[17] Shashank Gupta, Olivier Jeunen, Harrie Oosterhuis, and Maarten de Rijke. 2024. Optimal Baseline Corrections for Off-Policy Contextual Bandits. *Proceedings of the 18th ACM Conference on Recommender Systems* (2024).

[18] Peter Hall. 1990. Using the bootstrap to estimate mean squared error and select smoothing parameter in nonparametric problems. *Journal of Multivariate Analysis* 32, 2 (1990), 177–203. doi:10.1016/0047-259X(90)90080-2

[19] Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association* 47, 260 (1952), 663–685.

[20] Shion Ishikawa, Yun Ching Liu, Young-Joo Chung, and Yu Hirate. 2024. Position Bias Estimation with Item Embedding for Sparse Dataset. In *Companion Proceedings of the ACM Web Conference 2024*. 895–898.

[21] Olivier Jeunen. 2025. Meta Off-Policy Estimation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems, RecSys 2025, Prague, Czech Republic, September 22-26, 2025*. ACM, 1228–1233. doi:10.1145/3705328.3759308

[22] 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, 4–11.

[23] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 781–789.

[24] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[25] 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.

[26] Norman Knyazev and Harrie Oosterhuis. 2026. Sample-Free Almost-Exact Estimation of Plackett-Luce Propensities for Off-Policy Ranking. In *Advances in Information Retrieval*. Springer Nature Switzerland, 163–177.

[27] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M Henne. 2009. Controlled experiments on the web: survey and practical guide. *Data mining and knowledge discovery* 18, 1 (2009), 140–181.

[28] Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, S 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*. ACM, 1685–1694.

[29] Jan Malte Lichtenberg, Alexander Buchholz, Giuseppe Di Benedetto, Matteo Ruffini, and Ben London. 2023. Double clipping: Less-biased variance reduction in off-policy evaluation. (2023). https://www.amazon.science/publications/double-clipping-less-biased-variance-reduction-in-off-policy-evaluation

[30] Ben London, Alexander Buchholz, Giuseppe Di Benedetto, Jan Malte Lichtenberg, Yannik Stein, and Thorsten Joachims. 2023. Self-normalized off-policy estimators for ranking. (2023). https://www.amazon.science/publications/self-normalized-off-policy-estimators-for-ranking

[31] R Duncan Luce. 2012. *Individual Choice Behavior: A Theoretical Analysis*. Courier Corporation.

[32] James G. MacKinnon and Anthony A. Smith. 1998. Approximate bias correction in econometrics. *Journal of Econometrics* 85, 2 (1998), 205–230. doi:10.1016/S0304-4076(97)00099-7

[33] Allen Nie, Yash Chandak, Christina J Yuan, Anirudhan Badrinath, Yannis Flet-Berliac, and Emma Brunskill. 2024. Opera: Automatic Offline Policy Evaluation with Re-weighted Aggregates of Multiple Estimators. *Advances in Neural Information Processing Systems* 37 (2024), 103652–103680.

[34] 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* (Virtual Event, Canada) *(SIGIR '21)*. ACM, 1023â€“1032.

[35] Harrie Oosterhuis. 2022. Learning-to-Rank at the Speed of Sampling: Plackett-Luce Gradient Estimation with Minimal Computational Complexity. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2266–2271.

[36] 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).

[37] Harrie Oosterhuis and Maarten de Rijke. 2020. Policy-Aware Unbiased Learning to Rank for Top-k Rankings. *Proceedings of the 43rd International ACM SIGIR*

*Conference on Research and Development in Information Retrieval* (2020), 489–498.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in neural information processing systems*. 8026–8037.

[39] Robin L Plackett. 1975. The Analysis of Permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 24, 2 (1975), 193–202.

[40] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *arXiv preprint arXiv:1306.2597* (2013).

[41] Oscar Rolando Ramirez Milian and Harrie Oosterhuis. 2026. An Epistemic Position-Based Click Model: From Interactions to Epistemic Distributions of Relevance and Bias. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1531–1542.

[42] Matteo Ruffini, Vito Bellini, Alexander Buchholz, Giuseppe Di Benedetto, and Yannik Stein. 2022. Modeling Position Bias Ranking for Streaming Media Services. In *Companion of The Web Conference 2022, Virtual Event / Lyon, France, April 25 - 29, 2022*. ACM, 72–76.

[43] Yuta Saito and Thorsten Joachims. 2022. Off-Policy Evaluation for Large Action Spaces via Embeddings. In *International Conference on Machine Learning*. PMLR, 19089–19122.

[44] Yuta Saito, Aihara Shunsuke, Matsutani Megumi, and Narita Yusuke. 2020. Open Bandit Dataset and Pipeline: Towards Realistic and Reproducible Off-Policy Evaluation. *arXiv preprint arXiv:2008.07146* (2020).

[45] Matthew Schlegel, Wesley Chung, Daniel Graves, Jian Qian, and Martha White. 2019. Importance resampling for off-policy prediction. *Advances in Neural Information Processing Systems* 32 (2019).

[46] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations As Treatments: Debiasing Learning and Evaluation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*. 1670–1679.

[47] Yi Su, Pavithra Srinath, and Akshay Krishnamurthy. 2020. Adaptive Estimator Selection for Off-Policy Evaluation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 9196–9205.

[48] Adith Swaminathan and Thorsten Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *Advances in Neural Information Processing Systems*, Vol. 28. 3231–3239.

[49] Takuma Udagawa, Haruka Kiyohara, Yusuke Narita, Yuta Saito, and Kei Tateno. 2023. Policy-Adaptive Estimator Selection for Off-Policy Evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 8 (2023), 10025–10033.

[50] Chelsea Weaver, Arvind Balasubramanian, Juan Borgnino, and Ben London. 2025. Efficient Off-Policy Evaluation of Content Blending in Station-Based Music Experiences. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*.

# A Unbiasedness of GPBM

Buchholz et al. [8] showed that, when the correct position bias curve is known, and the chosen window system satisfies a condition known as *common window support*[3], the INTERPOL estimator is unbiased. We now prove a similar claim for GPBM using similar assumptions. Since GPBM uses weights ($F$) instead of windows, we require an analog of common window support, adapted for the weight matrix, $F$.

*Definition A.1.* For a given target policy, $\pi$, and weight matrix, $F$, we say that a logging policy, $\pi_0$, has *common support with $\pi$ under $F$* if, for every item $d \in \mathcal{D}$ and target position $j \in \{1, \dots, K\}$ where $\pi(d, j) > 0$, exists a logging position $k \in \{1, \dots, K\}$ such that $(F(j,k) \cdot \hat{p}_k > 0) \wedge (\pi_0(d,k) > 0)$.

Definition A.1 can be interpreted as saying that, if $\pi$ ranks an item $d$ at position $j$, then $\pi_0$ must have nonzero probability of placing $d$ in at least one position $k$ where the product $F(j,k) \cdot \hat{p}_k > 0$. Note that, when the entries of $F$ are binary (and we recover INTERPOL), this condition is equivalent to common window support.

We are now ready to state our main result.

Proposition A.2. *Fix an i.i.d. dataset, $D_N$, consisting of rankings generated by a logging policy $\pi_0$, and rewards generated under the PBM with positive position bias $p > 0$. Fix a position bias estimate, $\hat{p}$, and assume that $\hat{p} = p$. Then, for any target policy $\pi$, and weight matrix $F$, such that $\pi_0$ has common support with $\pi$ under $F$ (Definition A.1), we have that GPBM is an unbiased estimate of the expected reward of $\pi$.*

Proof. Similar to Buchholz et al.'s proof, it suffices to analyze the expectation of a single record in the dataset, due to independence and linearity of expectation.

$$\mathbb{E}\left[\sum_{d\in\mathcal{D}}\sum_{k=1}^{K}\sum_{j=1}^{K}\frac{F(j,k)\cdot\pi(d,j)\cdot\hat{p}_j}{\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}}\mathbb{1}[y_k=d]c(y,k)\right] \tag{18}$$

$$=\sum_{d\in\mathcal{D}}\sum_{k=1}^{K}\sum_{j=1}^{K}\frac{F(j,k)\cdot\pi(d,j)\cdot\hat{p}_j}{\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}}\pi_0(d,k)\mathbb{E}[c(y,k)] \tag{19}$$

$$=\sum_{d\in\mathcal{D}}\sum_{k=1}^{K}\sum_{j=1}^{K}\frac{F(j,k)\cdot\pi(d,j)\cdot\hat{p}_j}{\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}}\pi_0(d,k)\theta_d p_k \tag{20}$$

$$=\sum_{d\in\mathcal{D}}\theta_d\sum_{j=1}^{K}\pi(d,j)\hat{p}_j\frac{\sum_{k=1}^{K}F(j,k)\cdot\pi_0(d,k)\cdot p_k}{\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}} \tag{21}$$

$$=\sum_{d\in\mathcal{D}}\theta_d\sum_{j=1}^{K}\pi(d,j)\hat{p}_j \tag{22}$$

$$=\Delta(\pi). \tag{23}$$

The assumptions of common support ensure that the denominator, $\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}$, is never zero. □

# B Additional Experimental Details

Following Buchholz et al. [8], we evaluate all models across a wide range of position bias misestimation patterns $\hat{p}$. However, in the above settings, all positions $\hat{p}_k$ are misestimated in the same direction (all positions either over- or underestimated), and the deviation between the two curves changes monotonically (grows for $\hat{p} \in \{p^{0.6}, p^{0.8}, p^{1.2}, p^{1.4}\}$ or is fixed for $\hat{p} \in \{p^{1.0}, \max(p-0.1, 0)\}$. We thus evaluate the models under a more complex misestimation pattern we term the *zigzag* setting $\hat{p} = p \pm z$. In this setting, the estimated curve $\hat{p}$ at position $k \in \{1, \dots, K\}$ is defined as:

$$\hat{p}_k = \min(1, p + (-1)^{(k+1)} \cdot 0.1 \cdot \sin(\frac{\pi \cdot k}{K+1})), \tag{24}$$

where $\pi \approx 3.14$ and sin is the sine function. Using this formulation, $\hat{p}$ is overestimated relative to $p$ at odd positions, and underestimated at even positions. Furthermore, the magnitude of the deviation between the two curves progressively increases from the first rank until the center of the ranking, followed by a symmetric decay for the second part of the ranking.

# C Gradient of $\hat{\Delta}_{\text{GPBM}}$ w.r.t. the Estimated Position Bias Curve

The gradient of the GPBM estimator $\hat{\Delta}_{\text{GPBM}}(\pi; \hat{p}, D_M^{(b)})$ using the estimated position bias curve $\hat{p}$ for a single bootstrap sample of ranking data $D_M^{(b)}$ w.r.t. $\hat{p}$ is:

$$\nabla_{\hat{p}}\hat{\Delta}_{\text{GPBM}}(\pi;\hat{p},D_M^{(b)}) = \nabla_{\hat{p}}\left[\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K}c(y^{(b,i)},k)w(y_k^{(b,i)},k)\right] \tag{25}$$

$$=\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K}c(y^{(b,i)},k)\nabla_{\hat{p}}w(y_k^{(b,i)},k), \tag{26}$$

where $y^{(b,i)}$ the $i$'th sampled ranking in the bootstrap sample $D_M^{(b)}$, $y_k^{(b,i)}$ is the item at the $k$'th position of that ranking and $\nabla_{\hat{p}}w(y_k^{(b,i)},k)$ is the vector of partial derivatives $[\frac{\partial w(y_k^{(b,i)},k)}{\partial\hat{p}_1},\dots,\frac{\partial w(y_k^{(b,K)},k)}{\partial\hat{p}_K}]$ of individual weights w.r.t. each value in the estimated position bias curve $\hat{p}$. For a given weight $w(d,k)$, this partial derivative w.r.t. $\hat{p}_m$, the estimated position bias value at position $m$, is then equal to:

$$\frac{\partial w(d,k)}{\partial\hat{p}_m} = \frac{\partial}{\partial\hat{p}_m}\left[\sum_{j=1}^{K}\frac{F(j,k)\cdot\pi(d,j)\cdot\hat{p}_j}{\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}}\right] \tag{27}$$

$$=\sum_{j=1}^{K}\frac{\partial}{\partial\hat{p}_m}\left[\frac{F(j,k)\cdot\pi(d,j)\cdot\hat{p}_j}{\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}}\right] \tag{28}$$

$$=\sum_{j=1}^{K}\frac{\frac{\partial}{\partial\hat{p}_m}\left[F(j,k)\cdot\pi(d,j)\cdot\hat{p}_j\right]}{\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}} - F(j,k) \tag{29}$$

$$\cdot\frac{\pi(d,j)\cdot\hat{p}_j\cdot\frac{\partial}{\partial\hat{p}_m}\left[\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}\right]}{\left(\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}\right)^2} \tag{30}$$

$$=\frac{F(m,k)\cdot\pi(d,m)}{\sum_{k'=1}^{K}F(m,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}} \tag{31}$$

$$-\sum_{j=1}^{K}\frac{F(j,k)\cdot\pi(d,j)\cdot\hat{p}_j\cdot F(j,m)\cdot\pi_0(d,m)}{\left(\sum_{k'=1}^{K}F(j,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}\right)^2} \tag{32}$$

$$=\frac{F(m,k)\cdot\pi(d,m)}{\sum_{k'=1}^{K}F(m,k')\cdot\pi_0(d,k')\cdot\hat{p}_{k'}} \tag{33}$$

[3] In their paper, Buchholz et al. call this "full" window support; but given the dependence on the target policy, we feel it is more appropriate to call this "common" window support.

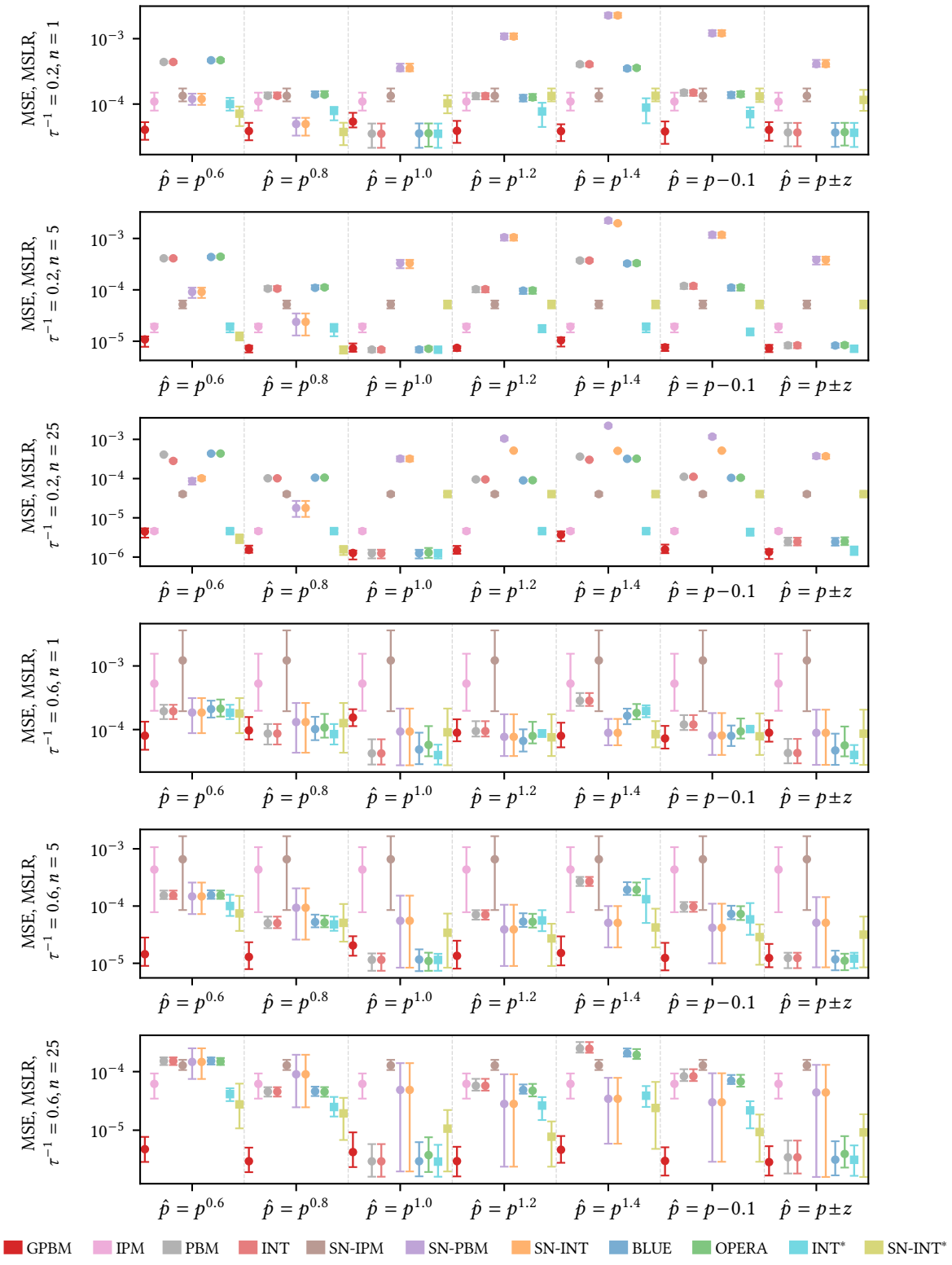
$$\cdot \left(1 - \frac{F(m,m) \cdot \pi_0(d,m) \cdot \hat{p}_m}{\sum_{k'=1}^{K} F(m,k') \cdot \pi_0(d,k') \cdot \hat{p}_{k'}}\right) \tag{34}$$

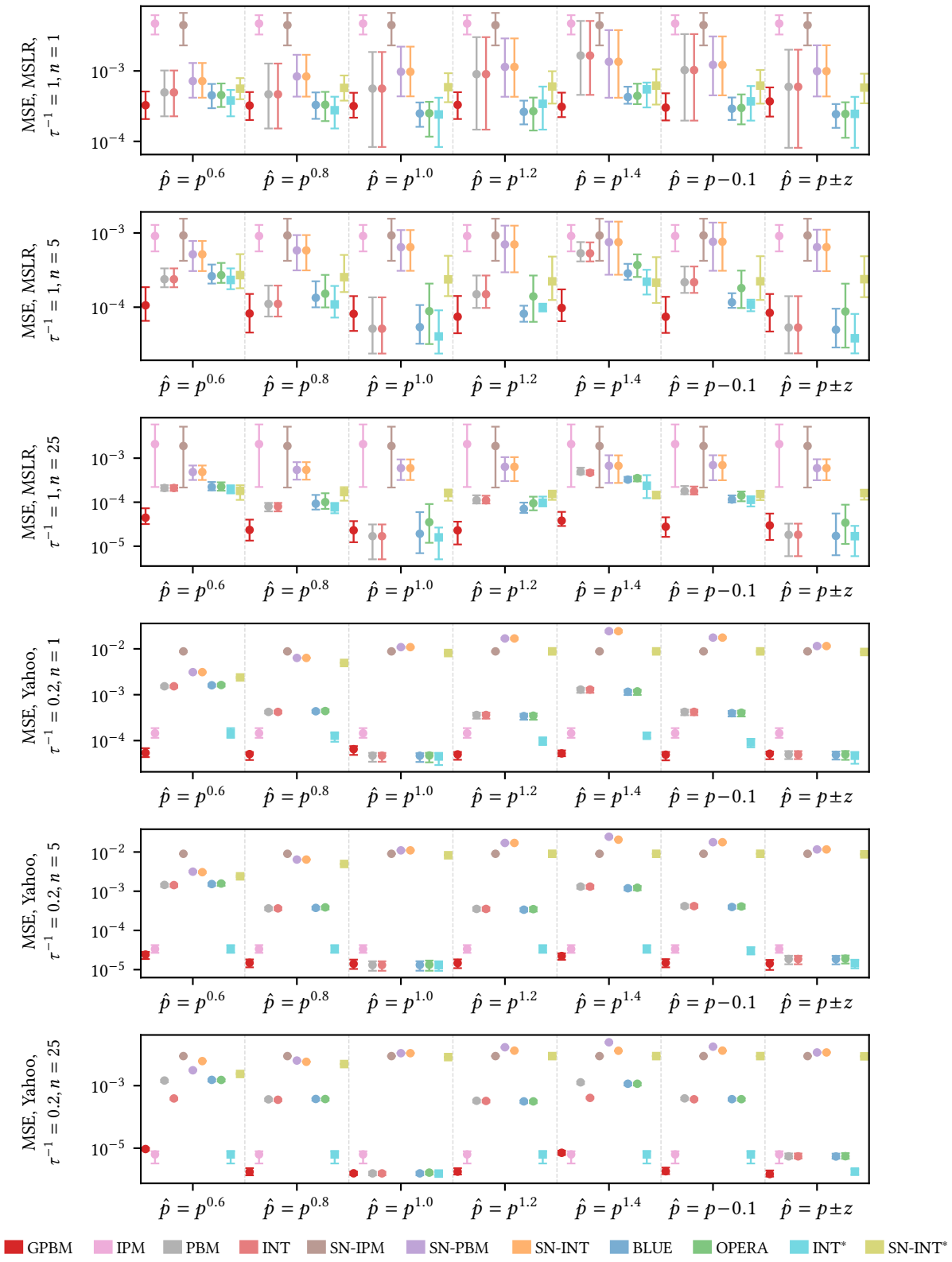
$$- \sum_{j \neq m} \frac{F(j,k) \cdot F(j,m) \cdot \pi(d,j) \cdot \pi_0(d,m) \cdot \hat{p}_j}{\left(\sum_{k'=1}^{K} F(j,k') \cdot \pi_0(d,k') \cdot \hat{p}_{k'}\right)^2}. \tag{35}$$

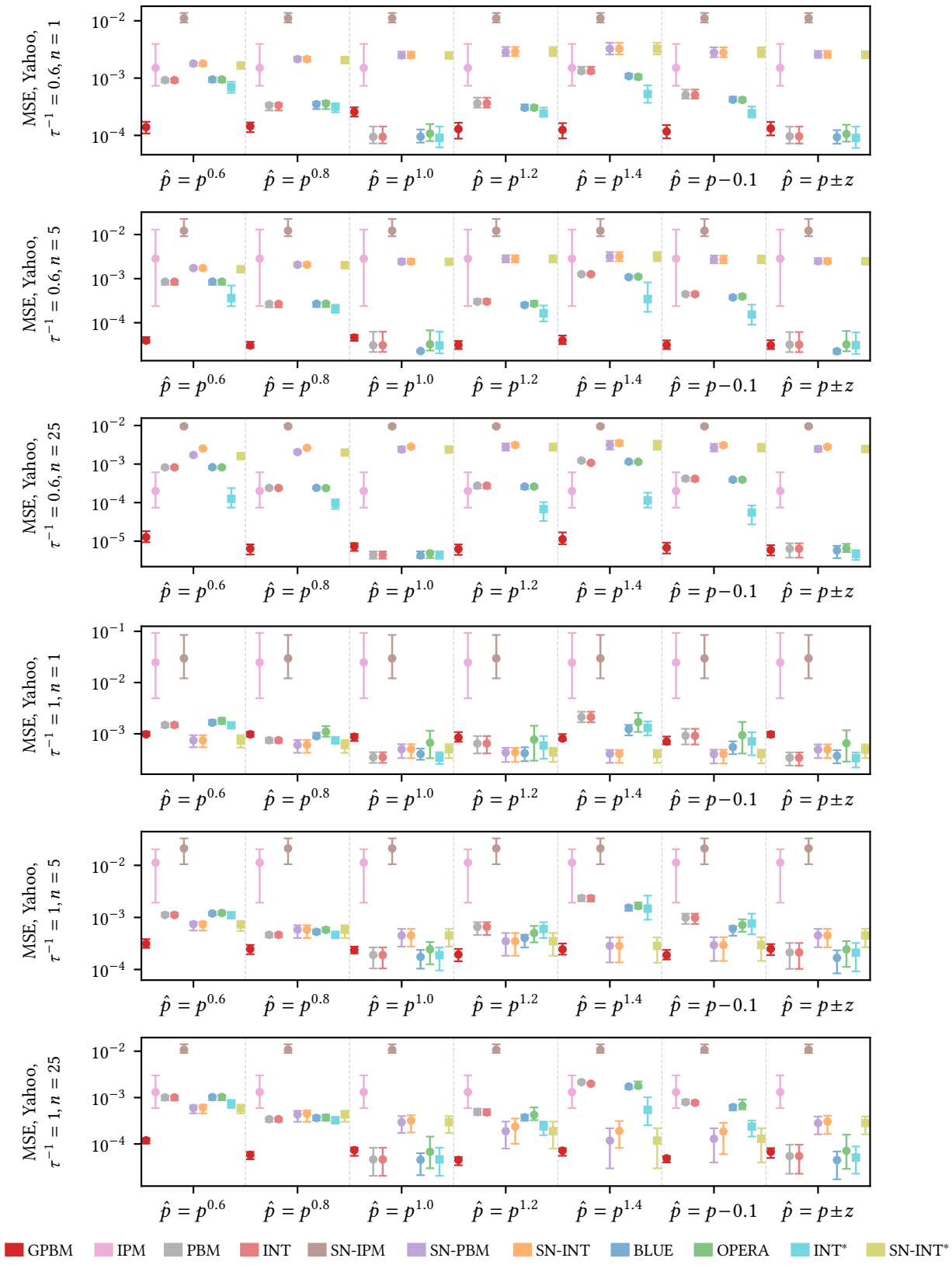
The full gradient $\nabla_{\hat{p}} \hat{\Delta}_{\text{GPBM}}(\pi; \hat{p}, D_M^{(b)})$ is then obtained by collecting the partial derivatives $\frac{\partial w(y_k^{(b,i)},k)}{\partial \hat{p}_m}$ for all position bias curve positions $m$ across all clicks in the bootstrap sample.

# D Full Results

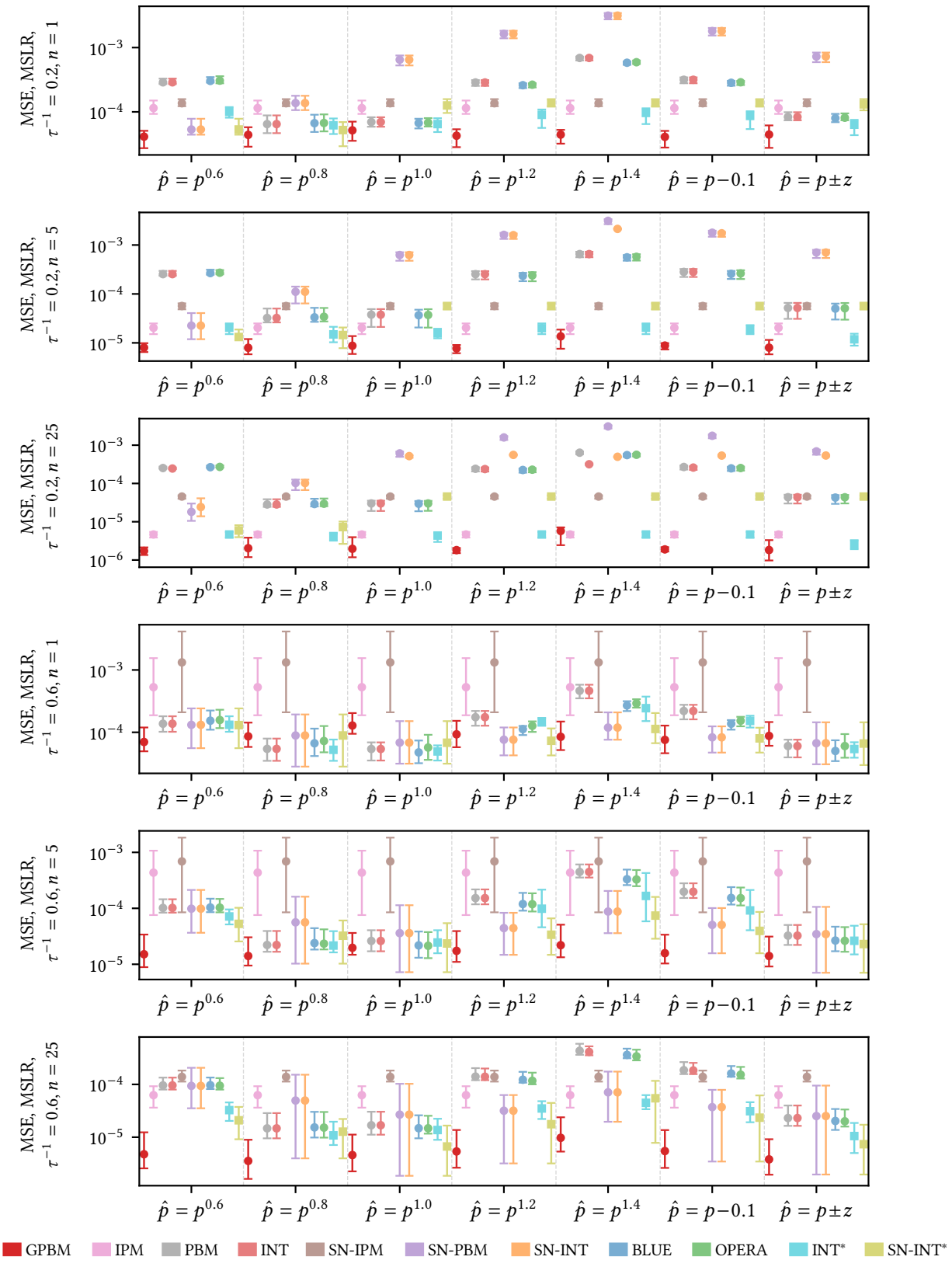
## D.1 PBM User Behavior (RQ1)

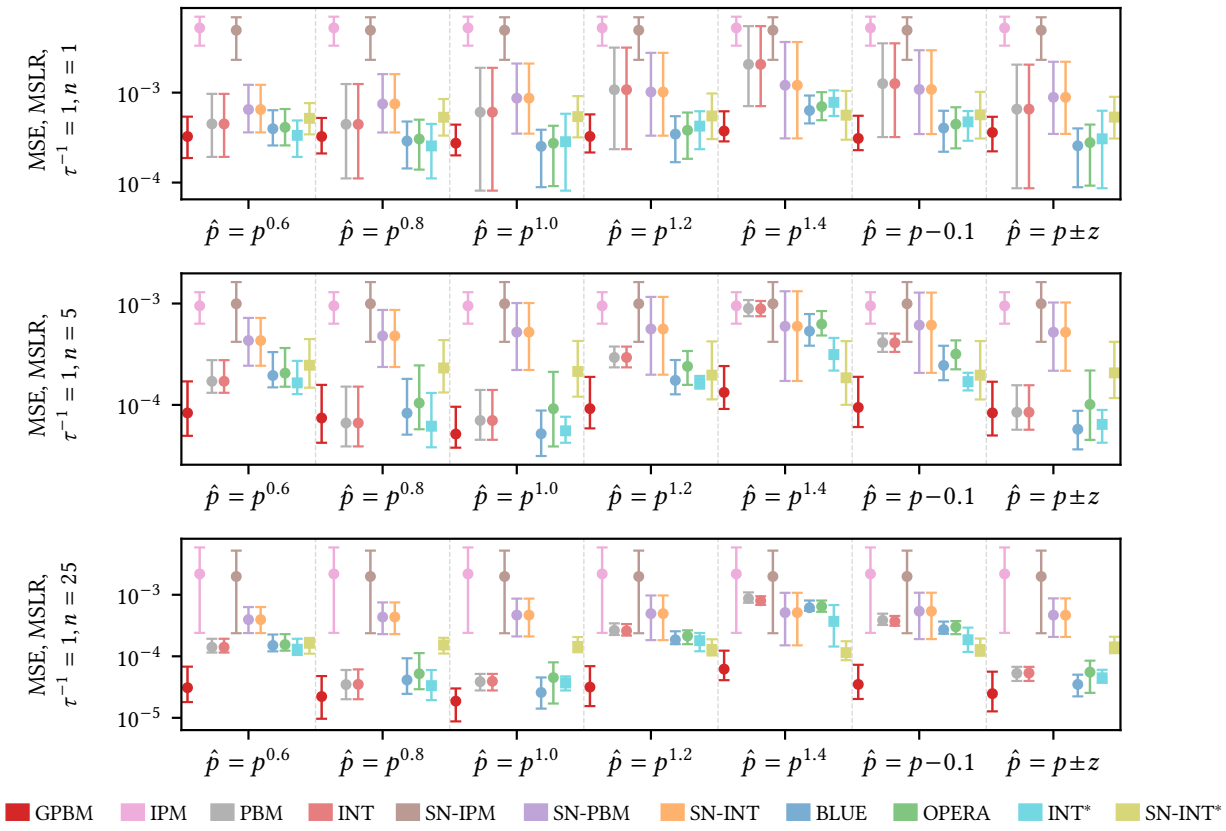



MSE, MSLR, $\tau^{-1} = 1, n = 1$
MSE, MSLR, $\tau^{-1} = 1, n = 5$
MSE, MSLR, $\tau^{-1} = 1, n = 25$
MSE, Yahoo, $\tau^{-1} = 0.2, n = 1$
MSE, Yahoo, $\tau^{-1} = 0.2, n = 5$
MSE, Yahoo, $\tau^{-1} = 0.2, n = 25$
$\hat{p} = p^{0.6}$
$\hat{p} = p^{0.8}$
$\hat{p} = p^{1.0}$
$\hat{p} = p^{1.2}$
$\hat{p} = p^{1.4}$
$\hat{p} = p - 0.1$
$\hat{p} = p \pm z$
GPBM
IPM
PBM
INT
SN-IPM
SN-PBM
SN-INT
BLUE
OPERA
INT*
SN-INT*


MSE, Yahoo, $\tau^{-1} = 0.6, n = 1$
MSE, Yahoo, $\tau^{-1} = 0.6, n = 5$
MSE, Yahoo, $\tau^{-1} = 0.6, n = 25$
MSE, Yahoo, $\tau^{-1} = 1, n = 1$
MSE, Yahoo, $\tau^{-1} = 1, n = 5$
MSE, Yahoo, $\tau^{-1} = 1, n = 25$
$\hat{p} = p^{0.6}$
$\hat{p} = p^{0.8}$
$\hat{p} = p^{1.0}$
$\hat{p} = p^{1.2}$
$\hat{p} = p^{1.4}$
$\hat{p} = p - 0.1$
$\hat{p} = p \pm z$
GPBM
IPM
PBM
INT
SN-IPM
SN-PBM
SN-INT
BLUE
OPERA
INT*
SN-INT*

## D.2 IPM User Behavior (RQ2)




MSE, MSLR, $\tau^{-1} = 1, n = 1$
MSE, MSLR, $\tau^{-1} = 1, n = 5$
MSE, MSLR, $\tau^{-1} = 1, n = 25$
$10^{-3}$
$10^{-4}$
$10^{-5}$
$\hat{p} = p^{0.6}$
$\hat{p} = p^{0.8}$
$\hat{p} = p^{1.0}$
$\hat{p} = p^{1.2}$
$\hat{p} = p^{1.4}$
$\hat{p} = p-0.1$
$\hat{p} = p \pm z$
GPBM
IPM
PBM
INT
SN-IPM
SN-PBM
SN-INT
BLUE
OPERA
INT*
SN-INT*