Are Neural Click Models Pointwise IPS Rankers?

PHILIPP HAGER, University of Amsterdam, The Netherlands

MAARTEN DE RIJKE, University of Amsterdam, The Netherlands

ONNO ZOETER, Booking.com, The Netherlands

Inverse-propensity scoring and neural click models are two popular methods for learning rankers from user clicks affected by position bias. Despite their prevalence, the two methodologies are rarely directly compared on equal footing. In this work, we focus on the pointwise learning setting to compare the theoretical differences of both approaches and present a thorough empirical comparison on the prevalent semi-synthetic evaluation setup in unbiased learning-to-rank. We show theoretically that neural click models, similarly to IPS rankers, optimize for the true document relevance when the position bias is known. However, our work also finds small but significant empirical differences between both approaches indicating that neural click models might be affected by position bias when learning from shared, sometimes conflicting, features instead of treating each document separately.

CCS Concepts: • **Information systems** → **Learning to rank**.

ACM Reference Format:

Philipp Hager, Maarten de Rijke, and Onno Zoeter. 2022. Are Neural Click Models Pointwise IPS Rankers?. In *CONSEQUENCES+REVEAL Workshop at RecSys '22, September 22–23, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Click modeling [12, 13, 15, 16, 32] and inverse-propensity scoring (IPS) [1, 20, 23, 28, 37] are two popular methods for learning rankers from biased user clicks. One well-known problem with learning from interaction data is that the position at which an item is displayed affects how likely a user is to see and interact with it [13, 22, 23, 36, 39]. IPS-based methods mitigate position bias by re-weighting clicks during training inversely to the probability of a user observing the clicked item [23, 38]. In contrast, click models are generative models that represent position bias and item relevance as latent parameters to directly predict the biased user behavior [12, 13, 15, 16, 32].

IPS approaches were introduced to improve over click models [23, 38] by: (i) requiring less observations of the same query-document pair by representing items using features instead of inferring a separate relevance parameter for each document [1, 23, 38, 39], (ii) decoupling bias and relevance estimations into separate steps since the joint parameter inference in click models can fail [2, 25, 39], and (iii) optimizing the order of documents through pairwise [20, 23] and listwise loss [27] functions instead of independent pointwise relevance estimations for each document [23, 39]. Since then, neural successors of click models have been introduced [7, 11, 17, 18, 41, 42] that can leverage feature inputs, similarly to IPS-based rankers. At the same time, the IPS community has introduced pointwise ranking losses [5, 33]. Are both approaches two sides of the same coin when it comes to pointwise learning-to-rank?

To address this question, we introduce both approaches and investigate their ability for unbiased relevance estimation. We perform an empirical comparison on the prevalent semi-synthetic benchmarking setup in unbiased learning-to-rank. And we investigate emerging differences, simulating circumstances under which the neural click model, in contrast to IPS, cannot obtain unbiased relevance estimates even when the true position bias is known.

2 METHODS

We assume the position-based model (PBM) [13, 32] of how user behavior is affected by position bias. Let y_d be the probability of a document being relevant and o_k the probability of observing a rank k; then clicks occur only on items that were observed and relevant: $c_{d,k} = o_k \cdot y_d$.

Neural click models mirror this user model in their architecture [7, 11, 17, 41]. We use a neural network to estimate document relevance from features x_d and estimate position bias using a single parameter $\hat{o}_k$ per rank. We use sigmoid activations and multiply the resulting probabilities: $\hat{c}_{d,k} = \sigma(\hat{o} \mid k) \cdot \sigma(g(\hat{y} \mid x_d))$. A common choice to fit neural click models is the binary cross-entropy loss between predicted clicks and the observed clicks in our dataset [17, 18, 41–43]:

$$\mathcal{L}_{\text{pbm}}(\hat{y}, \hat{o}) = -\sum_{(d,k)\in D} c_{d,k} \cdot \log(\hat{y}_d \cdot \hat{o}_k) + (1 - c_{d,k}) \cdot \log(1 - \hat{y}_d \cdot \hat{o}_k).$$

Instead of predicting clicks, IPS directly predicts the document relevance $\hat{y}_d$. Thus, our IPS model only uses the relevance network g: $\hat{y}_d = g(\hat{y} \mid x_d)$. Bekker et al. [5] introduce a pointwise IPS loss that minimizes the binary cross-entropy between predicted and true document relevance. Note how the PBM assumption is used to recover the unbiased document relevance by dividing clicks by the estimated position bias $\hat{o}_k$:

$$\mathcal{L}_{\text{ips}}(\hat{y}, \hat{o}) = -\sum_{(d,k)\in D} \frac{c_{d,k}}{\hat{o}_k} \cdot \log(\hat{y}_d) + (1 - \frac{c_{d,k}}{\hat{o}_k}) \cdot \log(1 - \hat{y}_d).$$

We note that both approaches are equivalent in the case in which we correctly assume that no position bias exists $o = \hat{o} = 1$, i.e., clicks indicate relevance, and in the case of falsely assuming no position bias exists: $\hat{o} = 1 \wedge o < 1$.

Saito et al. [33] show that $\mathcal{L}_{\text{ips}}(\hat{y})$ is unbiased if the position bias is correctly estimated, $\forall k \in K, \hat{o}_k = o_k$. The notion of an unbiased estimator is harder to apply to neural click models, since relevance is a parameter to be inferred. Recent work by Oosterhuis [25] shows that click models jointly estimating bias and relevance parameters are not consistent estimators of document relevance. This means that there are cases in which even an infinite amount of click data will not lead to the true document relevance estimate.

But what happens if click models do not have to jointly estimate bias and relevance parameters, but only item relevance? Since IPS approaches often assume access to a correctly estimated position bias [1, 23, 27, 37], we investigate this idealized setting for our click model and explore if initializing the model parameters $\hat{o}_k$ with the true position bias leads to an unbiased relevance estimate. For this, we take the partial derivative of $\mathcal{L}_{\text{pbm}}$ with regard to the estimated document relevance in our click model, its minima, and get in expectation: $\hat{y} = \frac{o \cdot y}{\hat{o}}$. We provide more details in Appendix A. So given the correct position bias, we find that the click model and IPS objective optimize for the unbiased document relevance, suggesting a similar performance in an idealized benchmark setup.

We encounter one notable difference between both loss functions concerning their magnitude and relationship with position bias. While IPS-based loss functions are known to suffer from high variance due to dividing clicks by potentially small probabilities [35, 40], the neural click model seems to suffer from the opposite problem since both $y_{d,k}$ and $\hat{y}_{d,k}$ (assuming our user model is correct) are multiplied by a potentially small examination probability. Thus, independent of document relevance, items at lower positions have a click probability closer to zero, impacting the magnitude of the loss (and gradient) as visualized in Appendix B. Note that while the magnitude differs, the minimum of the loss, as computed earlier in this section, is still correct. We will explore if this difference in loss magnitude might negatively impact items at lower positions in our experiments below.

3 EXPERIMENTAL SETUP

We perform a thorough empirical comparison of both approaches on the semi-synthetic click simulation setup that is prevalent in unbiased learning-to-rank [19, 21, 23, 26, 28, 29, 36, 37]. For the exact setup and model implementation, we defer to Appendix C. We use query-document pairs judged by human experts from three extensive LTR datasets: *Yahoo! Webscope* [9], *MSLR-WEB30k* [30], and *Istella-S* [14] (more in Appendix D) to generate clicks under the assumption of our PBM model. We also generate a fully synthetic dataset with 10,000 one-hot encoded documents with randomly sampled relevance. When reporting statistical significance, we use a two-tailed student's t-test [34] with significance level $\alpha = 0.0001$ and Bonferroni correction [6]. We compare five models in our experiments:

- **Pointwise IPS / PBM - Naive:** Naive version of (both) models that does not compensate for position bias.
- **Pointwise IPS - True Bias:** Pointwise IPS ranker with access to the true simulated position bias.
- **PBM - Estimated Bias:** PBM jointly inferring position bias and document relevance during training.
- **PBM - True Bias:** PBM initialized with the true position bias; the bias is fixed during training.
- **Production Ranker:** LambdaMART production ranker used to pre-rank queries during simulation.

4 RESULTS AND DISCUSSION

4.1 Is the neural click model empirically equivalent to the pointwise IPS ranker?

To answer this question, we first turn to the three classic LTR datasets. We train all models with up to 100M clicks and display the results in Figure 1, full tabular results are available in Appendix E. *PBM - Estimated Bias*, jointly estimating position bias and relevance, performs significantly better than the naive baseline on two of the three LTR datasets (except *Istella-S*). While being less stable than other models in this line-up, it still achieves a high performance without knowledge of the simulated bias. Next, we see that providing the *PBM - True Bias* model access to the correct position bias stabilizes and improves performance significantly over the naive baseline on all datasets. While having a lower variance, the improvements over *PBM - Estimated Bias* are not significant on any of the datasets.

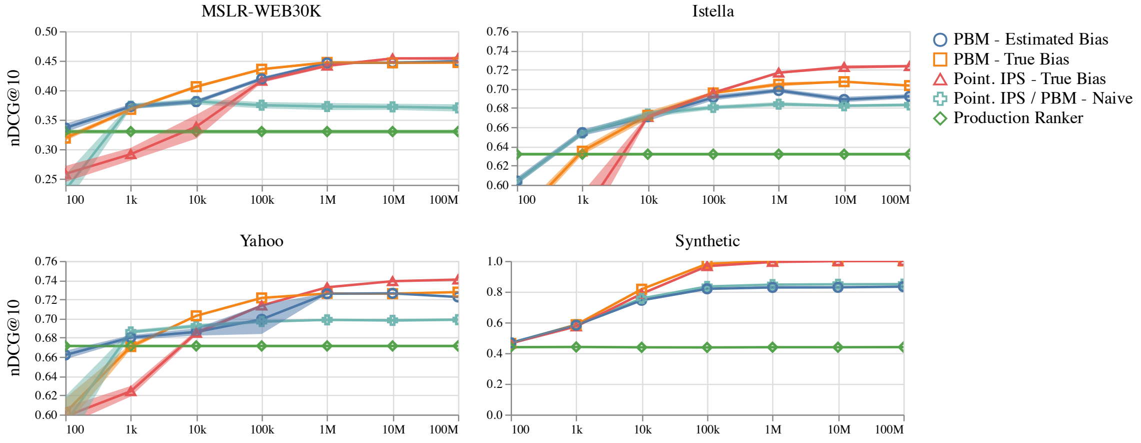

Fig. 1. Test performance on three LTR datasets and one fully synthetic dataset after training on up to 100M simulated queries. All results are averaged over 10 independent runs, and we display a bootstrapped 95% confidence interval.

Pointwise IPS performs lower than the neural click models for the first 100k clicks but ends up outperforming the click model significantly on two of the three LTR datasets (*Istella-S* and *Yahoo! Webscope*). These differences under idealized conditions between pointwise IPS and the click model are small, but significant. In this setup, the neural click model performs worse than the pointwise IPS model, even with access to the true position bias.

4.2 Is the neural click model biased?

In Section 2, we find that click models should be able to recover the true document relevance when the true bias is known. Given that this is not the case on the LTR datasets, we revisit the role of position bias and the magnitude of the loss functions discussed earlier. Our first hypothesis as to what might be happening concerns model tuning. We verify manually that items at lower positions indeed have smaller gradient updates, affecting the choice of learning rate and number of epochs. While this is certainly a concern when using SGD, our extensive hyperparameter tuning and use of adaptive learning rate optimizers should mitigate this issue. Instead, we hypothesize that higher ranked items might overtake the gradient of lower ranked items, given their higher potential for loss reduction. This case might occur when encountering two documents with similar features but different relevance. The item at the higher position could bias the expected relevance towards its direction. This is exactly what we find in a toy scenario in Figure 2.

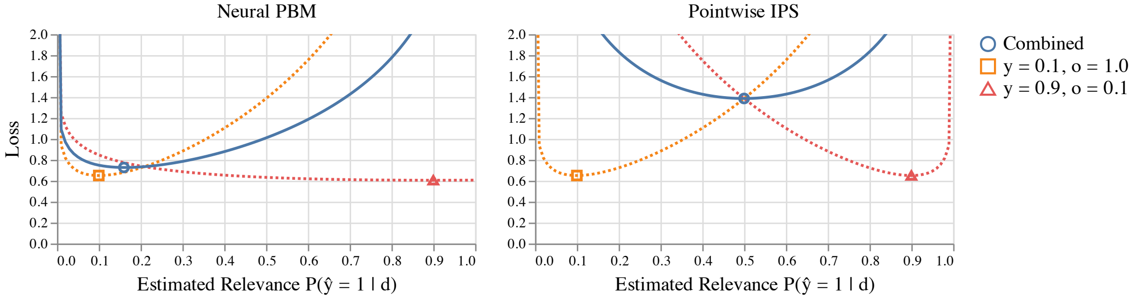

Fig. 2. Visualizing the loss and expected document relevance of two documents. Note how the expected relevance is correctly estimated when computing the loss for each item separately (dotted lines). When computing the combined loss, IPS converges to the average relevance of both documents, while the neural click model biases towards the item with the higher examination probability.

We test our hypothesis that the click model's gradient updates are biased towards items with higher examination probability with three small experiments. First, we should see an equivalent performance of both approaches in a setting in which documents share no features since the gradient magnitude should not matter in this setting. For this, we turn to the one-hot encoded synthetic dataset and find that both approaches are able to recover the true document relevance (see Figure 1). Second, gradually forcing documents to share features by introducing random feature collisions into our synthetic dataset should lead to a stronger drop in performance for the click model. We show in Appendix F that the click model deteriorates faster than IPS when introducing feature collisions. A last interesting consequence is that this problem should get worse with an increase in (known) position bias. Simulating an increasing position bias and supplying the examination probabilities to both approaches on *Istella-S* shows that IPS can recover from high position bias, while the click model increasingly deteriorates in performance (Appendix G).

Concluding, we show theoretically that the neural click model, similarly to a pointwise IPS ranker, optimizes for the true document relevance when the position bias is known. However, we find small but significant empirical differences between both approaches in an idealized LTR benchmark setting. Our findings indicate that neural click models might be affected by position bias when generalizing over shared, sometimes conflicting, features instead of treating each document separately. We end by emphasizing that these findings are specific to our setup, and we make no claims about other neural click model architectures [17, 39, 41, 42], which we leave as future work. The code for this work is available at: https://github.com/philipphager/ultr-cm-vs-ips

ACKNOWLEDGMENTS

We thank our reviewers for valuable feedback. This research was supported by the Mercury Machine Learning Lab, a collaboration between TU Delft, the University of Amsterdam, and Booking.com. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

[2] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating Position Bias without Intrusive Interventions. In *International Conference on Web Search and Data Mining (WSDM)*.

[3] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W. Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

[4] Qingyao Ai, Tao Yang, Huazheng Wang, and Jiaxin Mao. 2021. Unbiased Learning to Rank: Online or Offline? *ACM Transactions on Information Systems (TOIS)* 39, 2, Article 21 (2021).

[5] Jessa Bekker, Pieter Robberechts, and Jesse Davis. 2019. Beyond the Selected Completely at Random Assumption for Learning from Positive and Unlabeled Data. In *Machine Learning and Knowledge Discovery in Databases: European Conference (ECML PKDD)*.

[6] Carlo Bonferroni. 1936. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R. Istituto Superiore di Scienze Economiche e Commericiali di Firenze* 8 (1936), 3–62.

[7] Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. 2016. A Neural Click Model for Web Search. In *International Conference on World Wide Web (WWW)*.

[8] Christopher J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview.* Technical Report MSR-TR-2010-82. Microsoft.

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

[10] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected Reciprocal Rank for Graded Relevance. In *International Conference on Information and Knowledge Management (CIKM)*.

[11] Wenjie Chu, Shen Li, Chao Chen, Longfei Xu, Hengbin Cui, and Kaikui Liu. 2021. A General Framework for Debiasing in CTR Prediction. (2021). https://doi.org/10.48550/arXiv.2112.02767 arXiv:2112.02767

[12] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search.* Morgan & Claypool. https://doi.org/10.2200/S00654ED1V01Y201507ICR043

[13] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-Bias Models. In *International Conference on Web Search and Data Mining (WSDM)*.

[14] Domenico Dato, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2016. Fast Ranking with Additive Ensembles of Oblivious and Non-Oblivious Regression Trees. *ACM Transactions on Information Systems (TOIS)* 35, 2, Article 15 (2016).

[15] Georges E. Dupret and Benjamin Piwowarski. 2008. A User Browsing Model to Predict Search Engine Click Data from Past Observations.. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

[16] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click Chain Model in Web Search. In *International Conference on World Wide Web (WWW)*.

[17] Huifeng Guo, Jinkai Yu, Qing Liu, Ruiming Tang, and Yuzhou Zhang. 2019. PAL: A Position-Bias Aware Learning Framework for CTR Prediction in Live Recommender Systems. In *ACM Conference on Recommender Systems (RecSys)*.

[18] Malay Haldar, Prashant Ramanathan, Tyler Sax, Mustafa Abdool, Lanbo Zhang, Aamir Mansawala, Shulin Yang, Bradley Turnbull, and Junshuo Liao. 2020. Improving Deep Learning for Airbnb Search. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

[19] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. 2013. Reusing Historical Interaction Data for Faster Online Learning to Rank for IR. In *International Conference on Web Search and Data Mining (WSDM)*.

[20] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. 2019. Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm. In *The World Wide Web Conference (WWW)*.

[21] Rolf Jagerman, Harrie Oosterhuis, and Maarten de Rijke. 2019. To Model or to Intervene: A Comparison of Counterfactual and Online Learning to Rank from User Interactions. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

[22] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately Interpreting Clickthrough Data as Implicit Feedback. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

[23] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *International Conference on Web Search and Data Mining (WSDM)*.

[24] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *International Conference on Neural Information Processing Systems (NIPS)*.

[25] Harrie Oosterhuis. 2022. Reaching the End of Unbiasedness: Uncovering Implicit Limitations of Click-Based Learning to Rank. In *International Conference on the Theory of Information Retrieval (ICTIR)*.

[26] Harrie Oosterhuis and Maarten de Rijke. 2018. Differentiable Unbiased Online Learning to Rank. In *International Conference on Information and Knowledge Management (CIKM)*.

[27] Harrie Oosterhuis and Maarten de Rijke. 2020. Policy-Aware Unbiased Learning to Rank for Top-k Rankings. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

[28] Harrie Oosterhuis and Maarten de Rijke. 2021. Unifying Online and Counterfactual Learning to Rank: A Novel Counterfactual Estimator That Effectively Utilizes Online Interventions. In *International Conference on Web Search and Data Mining (WSDM)*.

[29] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for Selection Bias in Learning-to-Rank Systems. In *The Web Conference (WWW)*.

[30] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. (2013). https://doi.org/10.48550/arXiv.1306.2597 arXiv:1306.2597

[31] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2021. Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees?. In *International Conference on Learning Representations (ICLR)*.

[32] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting Clicks: Estimating the Click-through Rate for New Ads. In *International Conference on World Wide Web (WWW)*.

[33] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *International Conference on Web Search and Data Mining (WSDM)*.

[34] Student. 1908. The probable error of a mean. *Biometrika* (1908), 1–25.

[35] Adith Swaminathan and Thorsten Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *International Conference on Neural Information Processing Systems (NIPS)*.

[36] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2021. *Mixture-Based Correction for Position and Trust Bias in Counterfactual Learning to Rank.*

[37] Ali Vardasbi, Harrie Oosterhuis, and Maarten de Rijke. 2020. When Inverse Propensity Scoring Does Not Work: Affine Corrections for Unbiased Learning to Rank. In *International Conference on Information and Knowledge Management (CIKM)*.

[38] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

[39] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In *International Conference on Web Search and Data Mining (WSDM)*.

[40] Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudík. 2017. Optimal and Adaptive Off-Policy Evaluation in Contextual Bandits. In *International Conference on Machine Learning (ICML)*.

[41] Le Yan, Zhen Qin, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. Revisiting Two-Tower Models for Unbiased Learning to Rank. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

[42] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending What Video to Watch next: A Multitask Ranking System. In *ACM Conference on Recommender Systems (RecSys)*.

[43] Honglei Zhuang, Zhen Qin, Xuanhui Wang, Mike Bendersky, Xinyu Qian, Po Hu, and Chary Chen. 2021. Cross-Positional Attention for Debiasing Clicks. In *The Web Conference (WWW)*.

A EXPECTED MINIMUM OF THE CLICK MODEL LOSS

In the following, we calculate the partial derivative of the binary cross-entropy loss used by the neural click model with regard to the estimated document relevance $\hat{y}$:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{pbm}}}{\partial \hat{y}} &= -\left(c \cdot \frac{\partial}{\partial \hat{y}}\left[\log(\hat{o}\hat{y})\right] + (1-c) \cdot \frac{\partial}{\partial \hat{y}}\left[\log(1-\hat{o}\hat{y})\right]\right)\\ &= -\left(c \cdot \frac{\hat{o}}{\hat{o}\hat{y}} + (1-c) \cdot \frac{-\hat{o}}{1-\hat{o}\hat{y}}\right)\\ &= -\left(\frac{c}{\hat{y}} + \frac{-\hat{o}+\hat{o}c}{1-\hat{o}\hat{y}}\right)\\ &= -\frac{c-\hat{o}\hat{y}}{\hat{y}(1-\hat{o}\hat{y})}.\end{aligned}$$

Next, we find the ideal model which would minimize the loss by finding the roots of the function. We note that this function is convex and any extrema found will be a minimum:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{pbm}}}{\partial \hat{y}} &= 0\\ -\frac{c-\hat{o}\hat{y}}{\hat{y}(1-\hat{o}\hat{y})} &= 0\\ c-\hat{o}\hat{y} &= 0.\end{aligned}$$

Lastly, we see in expectation that:

$$\begin{aligned}\hat{y} &= \frac{\mathbb{E}_o\left[c\right]}{\hat{o}}\\ \hat{y} &= \frac{oy}{\hat{o}}.\end{aligned}$$

B HOW POSITION BIAS AFFECTS LOSS

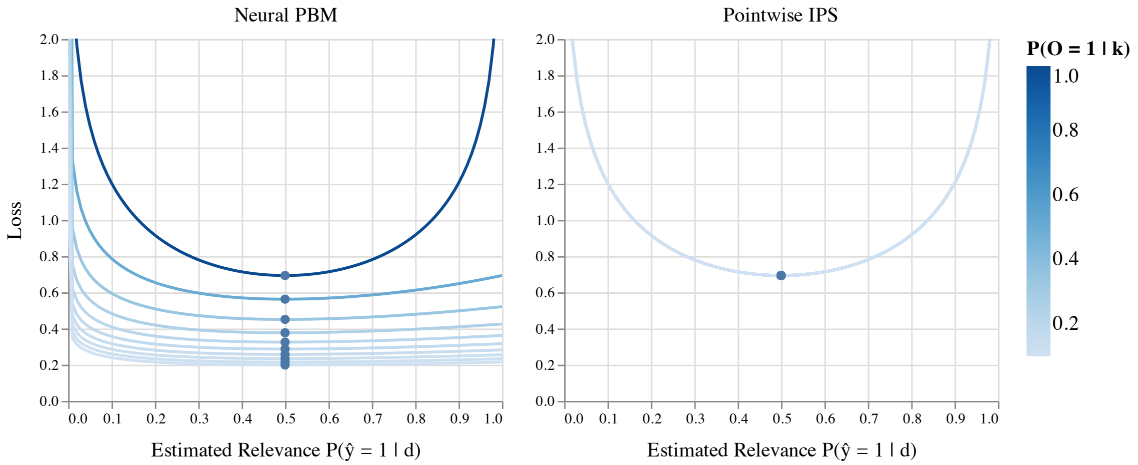

Fig. 3. Visualizing the loss of the neural click model and the pointwise IPS approach for a single document of relevance $y_d = 0.5$ under varying degrees of position bias in expectation of infinite clicks. We highlight the relevance prediction that minimizes the loss as a dot. Note how the magnitude of the loss on the click model gets smaller with increasing position bias, while the IPS loss always converges to the same distribution as the number of clicks approaches infinity.

C EXPERIMENTAL SETUP

C.1 Synthetic click datasets

We compare the neural click model and pointwise IPS model empirically using the semi-synthetic evaluation setup that is prevalent in the unbiased learning-to-rank community [19, 21, 23, 26, 28, 29, 36, 37]. Our simulation uses three large-scale public LTR datasets: *Yahoo! Webscope* (set 1) [9], *MSLR-WEB30k* (fold 1) [30], and *Istella-S* [14]. Each query-document pair is represented by a feature vector x_d and is accompanied by a score $s_d \in \{0, 1, 2, 3, 4\}$ indicating relevance as judged by a human annotator. We also generate a synthetic dataset with 10,000 documents, representing each with a one-hot encoded feature vector and assign each a relevance score s_d uniformly at random. Note that every document in the validation or test set appears once in the training dataset, thus achieving a perfect ranking score (e.g., $nDCG@10 = 1.0$) is possible on this dataset. Appendix D contains an overview of the dataset statistics. During preprocessing, we normalize document features on *MSLR-WEB30k* and *Istella-S* using $\text{log1p}(x_d) = \log_e(1 + |x_d|) \odot \text{sign}(x_d)$ as suggested by Qin et al. [31]. *Yahoo! Webscope* comes already normalized [9]. In addition, we use stratified sampling to limit the document set per query to a maximum number of documents (p90 in Table 1), improving computational speed while keeping the distribution of relevant documents almost identical.

Following Vardasbi et al. [36, 37], we train a LightGBM (version 3.3.2) [24] implementation of LambdaMART [8] ranker on 20 sampled train queries as our production ranker (100 trees, 31 leafs, and learning rate 0.1.). The intuition is to simulate initial rankings for the user model that are better than chance but leave room for further improvement. We generate clicks on our train and validation sets by repeatedly: (i) Sampling a query uniformly at random. (ii) Ranking the associated documents using our production ranker. (iii) Generate clicks using the PBM user model. Similar to [37], we generate validation clicks proportional to the train/validation split ratio provided by the datasets (see Table 1). We use the annotator scores s_d to compute a graded document relevance [3, 4, 10, 20] and add click noise $\epsilon = 0.1$: $y_d = \epsilon + (1 - \epsilon) \cdot \frac{2^{s_d} - 1}{2^4 - 1}$. For position bias, we adopt the prevalent definition by Joachims et al. [23]: $o_k = (\frac{1}{1+k})^\eta$ and set the strength of position bias by default to $\eta = 1$. Setting $\eta = 0$ corresponds to simulating no position bias.

Lastly, we apply an optimization step developed by Oosterhuis and de Rijke [27] and train on the average click-through-rate of each query-document pair instead of the actual raw click data. This step allows us to scale our simulation to millions of queries and multiple repetitions while keeping the computational load almost constant. Our experimental results hold without this trick.

C.2 Model implementation and training

We use neural networks to implement both the click model and IPS-based ranker. To estimate document relevance from features $g(\hat{y} \mid x_d)$, we use the same three layer feed-forward network architecture with $[512, 256, 128]$ neurons, ELU activations, and dropout 0.1 in the last two layers for both models. We pick the best-performing optimizer $\in$ {Adam, Adagrad, SGD} and learning rate $\in \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$ over five independent runs on the validation set for each model. In all experiments, we train our models on the synthetic click datasets up to 200 epochs and stop early after five epochs of no improvement of the validation loss. We do not clip propensities in the IPS model to avoid introducing bias into our comparison [1, 23]. We follow related work and report the final evaluation metrics on the original annotation scores of the test set [1, 23, 27]. We test differences for significance using a two-tailed student's t-test [34] and apply the Bonferroni correction [6] to account for multiple comparisons.

D DATASETS

Table 1. Overview of the LTR datasets used in this work.

Dataset	# Features	# Queries	% Train / Val / Test Split	# Documents per Query				
				min	mean	median	p90	max
Yahoo! Webscope	699	29,921	66.66 / 10 / 23.33	1	24	19	49	139
MSLR-WEB30K	136	31,531	60 / 20 / 20	1	120	109	201	1,251
Istella-S	220	33,018	58.3 / 19.9 / 21.8	3	103	120	147	182
Synthetic	10,000	1,200	33.3 / 33.3 / 33.3	25	25	25	25	25

E EXPERIMENTAL RESULTS

Table 2. Ranking performance on the full-information test set after 100M train queries as measured in nDCG and Average Relevant Position (ARP) [23]. We display statistical differences compared to the **PBM - True Bias** model computed with a two-sided student's t-test, marking significantly higher ▲ or lower performance ▼. We use a significance level of $\alpha = 0.0001$, declaring significance if $p < \alpha$, and use Bonferroni correction to adjust our significance level.

Dataset	**Model**	**nDCG@5 ↑**	**nDCG@10 ↑**	**ARP ↓**
Yahoo! Webscope	Production	0.613 (0.012) ▼	0.671 (0.009) ▼	10.439 (0.095) ▲
	Naive	0.647 (0.006) ▼	0.699 (0.004) ▼	10.199 (0.052) ▲
	PBM - Est. Bias	0.673 (0.005)	0.722 (0.003)	9.848 (0.055)
	PBM - True Bias	0.680 (0.004)	0.728 (0.003)	9.812 (0.035)
	IPS - True Bias	0.695 (0.001) ▲	0.741 (0.001) ▲	9.658 (0.011) ▼
MSLR-WEB30K	Production	0.301 (0.027) ▼	0.330 (0.024) ▼	49.223 (0.693) ▲
	Naive	0.348 (0.022) ▼	0.370 (0.020) ▼	48.386 (0.538) ▲
	PBM - Est. Bias	0.429 (0.010)	0.449 (0.008)	44.835 (0.274)
	PBM - True Bias	0.428 (0.006)	0.447 (0.006)	44.965 (0.230)
	IPS - True Bias	0.432 (0.011)	0.454 (0.010)	44.418 (0.227)
Istella-S	Production	0.566 (0.012) ▼	0.632 (0.010) ▼	10.659 (0.207) ▲
	Naive	0.616 (0.005) ▼	0.683 (0.005) ▼	9.191 (0.154) ▲
	PBM - Est. Bias	0.629 (0.008)	0.692 (0.007)	10.605 (1.193)
	PBM - True Bias	0.638 (0.003)	0.703 (0.004)	8.911 (0.212)
	IPS - True Bias	0.656 (0.005) ▲	0.724 (0.004) ▲	8.274 (0.141) ▼
Synthetic	Production	0.369 (0.005) ▼	0.439 (0.005) ▼	12.994 (0.038) ▲
	Naive	0.783 (0.005) ▼	0.849 (0.004) ▼	9.232 (0.026) ▲
	PBM - Est. Bias	0.772 (0.022) ▼	0.833 (0.019) ▼	9.335 (0.143) ▲
	PBM - True Bias	1.000 (0.000)	1.000 (0.000)	8.140 (0.004)
	IPS - True Bias	1.000 (0.000)	1.000 (0.000)	8.148 (0.003)

F SIMULATING FEATURE COLLISIONS

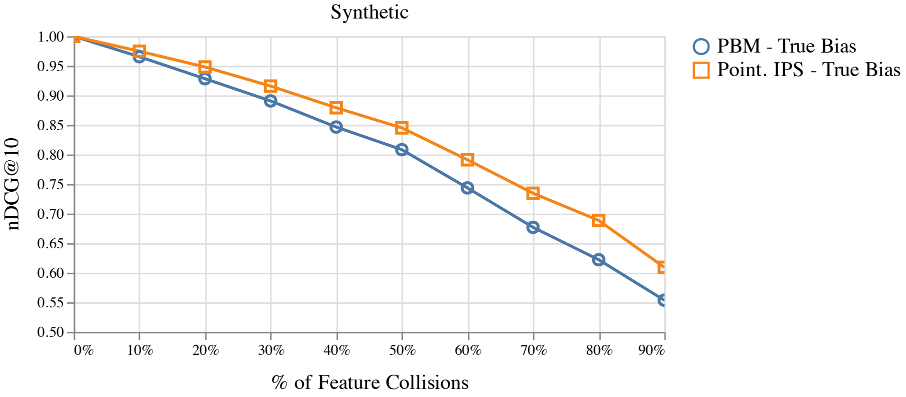

Fig. 4. Simulating collisions between document features on the synthetic dataset by projecting an increasing percentage of documents on feature vectors of other documents. Given that document relevance is randomly assigned in the synthetic dataset, we expect a gradual decrease in performance of both rankers. However, the click model degrades significantly faster in performance. In each step, we trained both models on 100M train clicks and display averaged results over 10 independent runs.

G MITIGATING POSITION BIAS

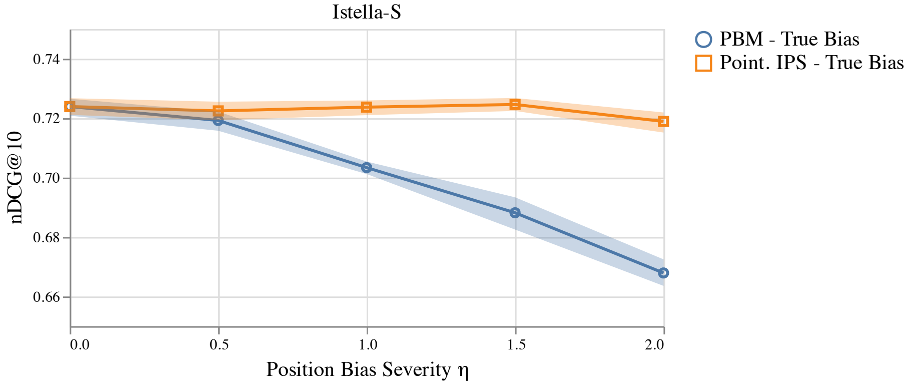

Fig. 5. Comparing the capacity of the neural click model and pointwise IPS model to mitigate varying degrees of simulated position bias on *Istella-S* ($\eta = 0$ implies no position bias). Note that both models have access to the currently simulated position bias at every step in this setup. IPS is able to consistently converge to a ranking performance comparable to the setting without any position bias. Despite access to the same propensities, the click model's performance degrades with an increase in position bias. We evaluate ranking performance on the full-information *Istella-S* test set after training on 100M queries over 10 independent runs and display a bootstrapped 95% confidence interval.