Impression Share Prediction: An Offline Evaluation Task for Ranking Systems

Mohsen Malmir
mohsenm@meta.com
Meta Platforms, Inc.
USA

Houssam Nassif
houssamn@meta.com
Meta Platforms, Inc.
USA

Danish Nasir Shaikh
danishshaikh556@meta.com
Meta Platforms, Inc.
USA

Taher Rahgooy
trahgooy@meta.com
Meta Platforms, Inc.
USA

Murat Ali Bayir
mbayir@meta.com
Meta Platforms, Inc.
USA

Abstract

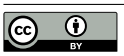
Offline evaluation is a major gateway before online evaluation of ranking models in A/B testing. Standard offline metrics measure predictive accuracy, but are only a surrogate for downstream utility: a model can improve them while redistributing impressions across objective buckets in ways that degrade downstream utility. No offline method surfaces these impression share shifts before online evaluation. We propose *impression share prediction* as an offline evaluation task: given a candidate ranking model, predict the distribution of impressions it would produce across objective buckets - impressions grouped by optimization goal (e.g., click, video view). The task is inherently counterfactual, since the candidate has never served live traffic. We propose a structural causal model of how model predictions and delivery capacity jointly determine impression allocation, and show the counterfactual effect is identified from observational data. Building on this, we develop a statistical learning framework that predicts impression shares from a candidate's early-interaction confidence signals and current system state, trained on historical data. On data from multiple ranking model families, a Random Forest reduces L1 error by 49% over a constant baseline for models seen during training. For held-out models, evaluated by time since first appearance, the first hour is the closest analog to true online evaluation and the hardest: the Random Forest falls below the baseline because the capacity state still reflects the prior model. An encoder-conditioned architecture that simulates a 2-hour rollout over recent auction dynamics recovers +22% L1 in this regime.

CCS Concepts

• **Information systems** → *Information retrieval*; • **Computing methodologies** → **Machine learning**.

Keywords

Offline evaluation, ranking, impression share, counterfactual prediction, auction dynamics, delivery capacity



RecSys '26, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2284-4/2026/09
https://doi.org/10.1145/3773078.3831809

ACM Reference Format:
Mohsen Malmir, Houssam Nassif, Danish Nasir Shaikh, Taher Rahgooy, and Murat Ali Bayir. 2026. Impression Share Prediction: An Offline Evaluation Task for Ranking Systems. In *20th ACM Conference on Recommender Systems (RecSys '26), September 27-October 02, 2026, Minneapolis, MN, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3773078.3831809

1 Introduction

Offline evaluation is a major gate before A/B testing in ranking systems [3, 7, 9, 23]. Impressions are organized by optimization goal - click, video view, and so on - forming a set of objective buckets, each generating distinct downstream utility when an impression leads to its target action. Standard offline metrics measure predictive accuracy on logged data [6, 13, 14, 28, 29] but are only a surrogate for downstream utility - the total outcome those impressions generate. A model that improves these metrics may still degrade downstream utility [15]: predictions feed into shared auctions where delivery capacity and pacing controllers jointly determine which impressions are served [1, 5, 26], and even a one-percentage-point shift from high-priority to lower-priority objective buckets can materially affect aggregate outcomes [30, 32]. No existing offline evaluation method provides visibility into these impression share shifts before online evaluation. We address this gap by proposing *impression share prediction*: given a candidate ranking model and the current system state, predict the distribution of impressions across objective buckets it would produce in an online environment. To our knowledge, no existing work addresses this prediction task directly.

One challenge is that impression share prediction is inherently counterfactual. The candidate model has never served traffic in the live system: it has not influenced delivery capacity, pacing controllers have not adapted to it, and its model confidence has not been shaped by data it generated. The impression shares we observe reflect the current model's imprint on the system, not what a new model would produce. We frame this as a counterfactual prediction task and propose a statistical learning framework that uses the candidate's model confidence features alongside current system state to estimate the impression shares it would induce.

We validate this approach on data from multiple ranking model families, predicting impression shares observed during A/B tests from a candidate's earliest-online signals as a proxy for online evaluation inputs. Our goal is not to replace reward-based offline

evaluation, but to complement it: impression share prediction surfaces allocation behavior that current offline evaluation does not reveal.

Our contributions are as follows:

(1) We propose impression share prediction as an offline evaluation task that gives practitioners visibility into a candidate model's allocation behavior before A/B testing.
(2) We propose a structural causal model that formalizes how model predictions, delivery capacity, and pacing jointly determine impression allocation, and show the counterfactual effect is identified from observational data.
(3) We develop a statistical learning framework and show that for models already seen in the system, it reduces L1 prediction error by 49% over a constant baseline.
(4) We identify a systematic failure in the first hour after a held-out model's first appearance in the system, and introduce an encoder-conditioned architecture that recovers this gap, achieving 22% L1 improvement precisely where the baseline fails.

2 Related Work

Standard offline evaluation of ranking models relies on predictive metrics computed over logged impressions: normalized entropy (NE), relative information gain (RIG), AUROC, and group-level AUROC (GAUC) [3, 4, 8, 9, 18, 33]. These measure a model's accuracy on data generated under a different ranking policy: they do not capture how the model would redistribute impressions across objective buckets in an online environment, or whether that redistribution would benefit or harm downstream utility. Impression share prediction is a complementary offline task that addresses this gap directly. A parallel line strengthens offline-metric fidelity through unbiased learning-to-rank [13, 21] and safe-deployment objectives [7], but still targets per-impression accuracy on a fixed exposure, not how exposure redistributes once a candidate is serving.

A related line studies the auction and pacing mechanisms that govern impression allocation in ranking systems [1, 2, 5, 26], including interference bias when experiments share delivery capacity [16, 17] - a structural feature we also exploit in our causal model. These characterize allocation as a system-level equilibrium property but do not predict the specific impression shares a new ranking model would induce.

Several works study offline-to-online gaps and the detection of impression or exposure shift [10, 11]. Zheng and Zhao [32] formalize algorithm adaptation bias, where partial-rollout A/B tests misstate full-deployment impact. Wang et al. [30] show a retrieval model can improve aggregate offline metrics while substantially reallocating impressions across objectives. Off-policy evaluation methods [6, 12, 14, 29], including auction-simulation benchmarks [27] and distributional or pessimistic estimators [24, 31], estimate counterfactual reward from logged data but assume stationarity and policy overlap and target reward rather than allocation across objective buckets. These assumptions break down when models interact through shared delivery capacity and pacing. None of these predict the impression shares a candidate model would produce before online evaluation; that is the task we address.

3 Method

3.1 Problem Formulation

We consider a ranking system where impressions are allocated across C objective buckets (e.g., click-, conversion-, view-optimized). Multiple ranking models operate simultaneously, each covering a subset of objective buckets. Models are evaluated through parallel A/B tests: each arm serves a different candidate model alongside shared incumbent models that collectively cover all C objective buckets. Each campaign has a finite *delivery capacity*, set externally at the campaign level and drawn down by a pacing system regardless of which arm serves the impression, so all arms draw from the same capacity pool. This shared pool is the structural coupling between arms that makes impression allocation a system-level outcome rather than a per-model property. We use historical A/B test data–observed hourly snapshots of impression shares across objective buckets, model confidence signals, and delivery capacity states–to train a statistical model that predicts the impression shares a new candidate model would produce before it enters the live system. Formally, given current system state and a candidate model m, we predict the normalized impression share vector $\hat{Y}^m = (\hat{Y}_1^m, \ldots, \hat{Y}_C^m) \in \Delta^{C-1}$, where $\Delta^{C-1} = \{\mathbf{v} \in \mathbb{R}^C_{\geq 0} : \sum_{c=1}^C v_c = 1\}$ is the $(C-1)$-dimensional probability simplex and $\hat{Y}_c^m$ is the fraction of impressions allocated to objective bucket c.

3.2 Causal Model

We formalize the system via a structural causal model (SCM) [22] over three time-varying variables:

- A_t^m: **Model state** (treatment) - the *model-determined* (intrinsic) prediction score distribution and calibration of model m, fixed by its architecture and training rather than by the traffic it serves; this is the object the identification argument requires, and it is what makes the missing edge $D_t \not\to A_t^m$ hold.
- Y_t^m: **Impression share** (outcome) - fraction of impressions allocated to each of the C objective buckets, $Y_t^m \in \Delta^{C-1}$.
- D_t: **Delivery capacity state** (shared covariate) - remaining delivery capacity across all objective buckets, shared across all A/B arms.

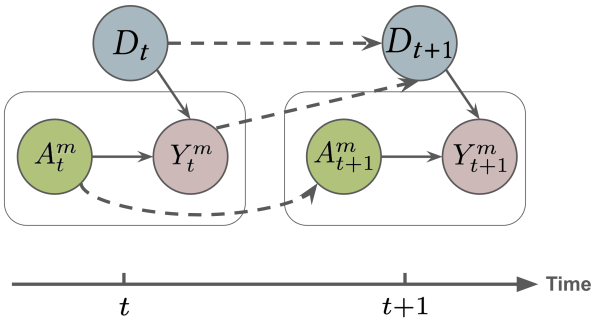


Figure 1: Causal DAG for impression share prediction. A_t^m: model state (treatment); Y_t^m: impression share (outcome); D_t: shared delivery capacity state. Solid edges: within-period effects; dashed edges: cross-period dependencies.

Figure 1 shows the DAG. The edges encode: (1) $A_t^m \to Y_t^m$: model confidence determines auction winners, restricted to the model's covered objective buckets V^m; (2) $D_t \to Y_t^m$: capacity depletion and pacing multipliers constrain eligibility; (3) $Y_t^m \to D_{t+1}$: impressions served deplete shared delivery capacity; (4) $D_t \to D_{t+1}$: remaining capacity carries forward; (5) $A_t^m \to A_{t+1}^m$: model architecture is fixed, producing autocorrelated states.

One structural property is the absence of an edge $D_t \not\to A_t^m$: a model's properties are determined by its architecture and training data, not by system conditions. This eliminates all backdoor paths from treatment to outcome, so the causal effect of an intervention on the model state is identified from observational data: $P(Y_t^m \mid do(A_t^m = a), D_t) = P(Y_t^m \mid A_t^m = a, D_t)$. Unlike settings with time-varying confounding [19, 25], no balanced representations, inverse propensity weighting, or domain-adversarial training are required.

3.3 Predictive Framework

We frame impression share prediction as estimating $P(Y_T^m \mid A_0^m, D_0)$ from historical A/B data, using the variable definitions of Section 3.2. Given a candidate model m, we observe its model state A_0^m and the delivery capacity state D_0 at evaluation time $t = 0$, and predict the impression share vector $Y_T^m \in \Delta^{C-1}$ aggregated over the forward window $[0, T]$ (we use $T = 24$h in our experiments). At this level of abstraction, A_0^m is summarized by **model features** describing the candidate's prediction behavior, and D_0 is summarized by **delivery capacity features** describing the system at $t = 0$; concrete feature definitions are deferred to Section 4.1.

We distinguish two prediction regimes, post-entry and pre-entry. In the *post-entry* setting the candidate has already been interacting with the live system, so A_0^m and D_0 are both directly observable from logs at any time $t = 0$, and we predict its share vector over the subsequent T hours. In the *pre-entry* setting the candidate has not yet served live traffic: it has not influenced delivery capacity through its consumption patterns, pacing controllers have not adapted to its behavior, and its confidence scores have not been shaped by data it generated [32]. The model state A_0^m is model-determined – ideally computed offline before online evaluation, and in practice estimated from the candidate's earliest online serving (Section 4.1). For the delivery capacity input we use the *current* system state D_0: we cannot know what the system will look like at the moment the candidate actually enters, so we ask what the candidate would produce under today's observed conditions:

$$P\big(Y_T^m \mid \underbrace{A_0^m}_{\text{model-determined}}, \underbrace{D_0}_{\text{current system state}}\big). \tag{1}$$

Predicting Y_T^m exactly from only $t = 0$ inputs would otherwise require predicting the intermediate states (A_t^m, D_t, Y_t^m) for $0 < t < T$, which compounds errors across steps. We instead learn predictors that map (A_0^m, D_0) directly to Y_T^m, treating the intermediate dynamics implicitly through the historical training data. Since the true pre-entry outcome is unobservable, for empirical evaluation we use the candidate's *first-hour* observed impression shares – the window before the system has measurably adapted to it – as a surrogate for the pre-entry target.

4 Experiments

4.1 Dataset and Setup

The system runs multiple concurrent A/B tests, each evaluating multiple candidate ranking models in parallel. We capture minute-level snapshots of auction statistics and impression shares across C objective buckets from 150 candidate models spanning multiple model families, collected over roughly five weeks (~1.8M snapshots). We split the data temporally at a fixed cutoff into 23 training and 15 test days. Of the 103 models active in the test window, 20 are newly entered (never seen in training) and 83 were also present in the training period, letting us report performance separately on seen and unseen models. We evaluate under the two regimes introduced in Section 3.3 (post-entry and pre-entry). Predictions are compared against a *constant baseline* that emits the training-set mean share vector for every test instance. We report two metrics: **L1 distance** $\mathrm{L1}(\hat{Y}_t^m, Y_t^m) = \sum_c |\hat{Y}_{t,c}^m - Y_{t,c}^m|$, the primary error metric (lower = better), and **Spearman rank correlation** between predicted and observed share vectors, a secondary metric (higher = better). Per-method scores are reported as relative improvement over the constant baseline.

Feature instantiation. The model state A_0^m is realized as **model confidence features** (22-dim) – a 20-bin log-scale histogram of the candidate's prediction scores, plus mean and variance – together with a **coverage vector** $V^m \in \{0, 1\}^C$ encoding which objective buckets the candidate is eligible to score. Ideally these scores would come from an offline evaluation pass before the candidate serves any traffic – a full *pre-entry offline evaluation*; in this work we instead estimate them from a rolling window over the candidate's earliest online serving, so our setup is *early-entry forecasting*. For each objective bucket, the delivery capacity state D_0 is realized as **delivery capacity features** ($5 \times C$-dim): total capacity, consumed capacity, remaining capacity, pacing multiplier, and number of active campaigns.

4.2 Predictors

We evaluate two predictors mapping (A_0^m, D_0) to the forward share vector $\hat{Y}_T^m$. The first is a *Random Forest* regressor that predicts each share component independently and then projects onto the simplex (clip negatives, renormalize). It treats the inputs as a single snapshot at $t = 0$, with confidence features as the dominant signal and delivery capacity features contributing secondary predictive value. Two models with identical AUROC can produce very different score distributions and therefore very different impression shares.

The second is an *encoder-conditioned* architecture that augments the snapshot view with a short look-back over recent capacity dynamics. A PatchTST-style encoder [20] ingests a 2-hour capacity history at minute-level resolution, divides it into 15-minute patches (8 tokens), and processes them through a 2-layer Transformer (4 attention heads, $d_{\text{model}} = 64$, GELU) with mean pooling to produce a system state vector $\mathbf{h}_t \in \mathbb{R}^{64}$. A conditional MLP head maps $(\mathbf{h}_t, A_0^m)$ to $\hat{Y}_T^m$ via two hidden layers followed by softmax. The encoder absorbs the dynamics over the immediate past, while the conditional head treats the model state as a separate intervention input – enabling counterfactual queries (fix $\mathbf{h}_t$, swap the model state,

read off predicted shares). Trained end-to-end with KL divergence loss and AdamW.

4.3 Evaluation on Seen Models

We first evaluate on seen models: every test model also appears in training (the post-entry regime of Section 3.3, where the model has been interacting with the system and we predict 24 hours into the future). This experiment serves two purposes: (i) establish how predictable the task is when the model is in-distribution, and (ii) ablate the contribution of each feature group. We compare three Random Forest configurations differing in feature set against the constant baseline; Table 1 reports relative improvements over the baseline.

Confidence and coverage features alone improve L1 by 42.9%, accounting for the majority of the gain. Delivery capacity features alone improve L1 by a modest 25.3%. The full feature set improves L1 by 48.6%. The candidate model's score distribution - not the system state - is therefore the primary predictor of impression allocation when the model has been seen during training.

Spearman rank correlation is near-saturated across all configurations: the rank ordering of objective buckets by impression share is largely stable across model versions, since system structure (delivery capacity, campaign mix) determines which buckets are higher priority. L1 is the more consequential metric, since small share changes shift downstream utility (buckets differ in conversion rate).

Feature Set	L1 imp.	Spearman imp.
Constant Baseline	0	0
Capacity Only	+25.3%	+0.17%
Confidence + Coverage	+42.9%	+0.64%
Capacity + Confidence + Coverage	+48.6%	+0.66%

Table 1: Seen-models evaluation: relative improvement over the constant baseline (24h evaluation window). Higher = better. L1 is the primary metric.

4.4 Evaluation on Held-Out Models

This experiment has two ingredients. First, the test models are *held out*: they never appear in training, so the predictor has no in-distribution signal for them. Second, because we use online data, the input we feed – the model state A_0^m and especially the system state D_0 – itself depends on how long the model has been interacting with the live system. We therefore further break down evaluation by *time since the model first appeared.* The same held-out model yields very different inputs depending on whether D_0 is captured one hour after entry or five days after: longer interaction means D_0 has progressively absorbed the model's own footprint on capacity consumption and pacing, so even though the model is unseen by the predictor, the input has become much more predictable. The true offline-to-online gap lives in the **first hour**: the features extracted there represent the offline setup – the model has not yet measurably impacted the system, and D_0 still reflects the prior regime.

Table 2 reports L1 improvement over the constant baseline for the two predictors – the snapshot Random Forest and the encoder-conditioned PatchTST Transformer (Section 4.2) – binned by time since first appearance. One distinction is that the Transformer's encoder simulates a short rollout over the most recent 2 hours of auction dynamics, whereas the RF uses only the instantaneous snapshot at $t = 0$ – no rollout, neither backward over the recent past nor forward over the 24-hour prediction window. At 0–1h the RF is −20.5% (worse than the constant baseline) because D_0 still encodes the prior model's equilibrium and the rolling confidence window is only partially filled; it then gradually improves, recovering to +11.5% by 1–2h and converging to +37.9% by 7d+. The Transformer's 2-hour capacity rollout closes the first-hour gap directly: at 0–1h it achieves +22.1% – a **42.6 percentage-point swing** against the RF – and it leads through 12h. Beyond 2 days the RF catches up and slightly leads: by then the system has fully adapted to the new model and instantaneous features suffice.

Time since entry	RF %imp	Transformer %imp
0–1h	−20.5%	+22.1%
1–2h	+11.5%	+33.2%
2–4h	+27.2%	+27.5%
4–8h	+29.2%	+37.1%
8–12h	+29.6%	+37.1%
12–24h	+24.8%	+34.1%
1–2d	+30.8%	+32.3%
2–4d	+26.2%	+25.7%
4–7d	+36.4%	+34.6%
7d+	+37.9%	+33.8%

Table 2: L1 improvement over the constant baseline on held-out (unseen) models, by time since first appearance. The 0–1h row corresponds to the pre-entry surrogate. Positive = better than baseline. Rank correlation is saturated (Spearman > 0.98) across all bins and is omitted.

5 Conclusion

We introduced impression share prediction as an offline evaluation task for ranking systems, complementing reward-based metrics by surfacing how a candidate model would redistribute impressions across objective buckets before online evaluation. We formalized the task via a structural causal model that shows the effect is identified from observational data, requiring neither inverse propensity weighting nor balanced representations, and framed prediction as estimating $P(Y_T^m \mid A_0^m, D_0)$ over a 24-hour forward window. Two studies on data examine (i) feature importance under in-distribution conditions, and (ii) prediction accuracy on held-out models, broken down by how long the model has been interacting with the system. The first hour is where the offline-to-online gap actually lives: a snapshot Random Forest falls below the constant baseline (−20.5% L1), while a PatchTST encoder that simulates a 2-hour rollout over recent auction dynamics recovers +22% L1 improvement precisely in this regime.

Two directions follow. First, extending the predictor to a full rollout from $t = 0$ to T requires a Bayesian or sequential model that propagates uncertainty across (A_t^m, D_t, Y_t^m) and handles compounding error – the present work avoids this by direct prediction. Second, the most accurate setup would compute the candidate's confidence features from offline evaluation; in this work we use the rolling online-window confidence as a stand-in, and any discrepancy between offline-evaluation confidence and online rolling confidence could impact results in the pre-entry regime.

References

[1] Santiago R. Balseiro, Kshipra Bhawalkar, Zhe Feng, Haihao Lu, Vahab Mirrokni, Balasubramanian Sivan, and Di Wang. 2024. A Field Guide for Pacing Budget and ROS Constraints. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. Preprint: arXiv:2302.08530.

[2] Santiago R. Balseiro and Yonatan Gur. 2019. Learning in Repeated Auctions with Budgets: Regret Minimization and Equilibrium. *Management Science* 65, 9, 3952–3968. https://ideas.repec.org/r/inm/ormnsc/v65y2019i9p3952-3968.html

[3] Pablo Castells and Alistair Moffat. 2022. Offline Recommender System Evaluation: Challenges and New Directions. *AI Magazine* 43, 2 (2022), 225–238. doi:10.1002/aaai.12051

[4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS)*. doi:10.1145/2988450.2988454

[5] Vincent Conitzer, Christian Kroer, Debmalya Panigrahi, Okke Schrijvers, Eric Sodomka, Nicolas E. Stier-Moses, and Chris Wilkens. 2022. Pacing Equilibrium in First-Price Auction Markets. *Management Science* 68, 12 (2022), 8515–8535. Preprint: arXiv:1811.07166.

[6] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*. https://icml.cc/2011/papers/554_icmlpaper.pdf

[7] Shashank Gupta, Harrie Oosterhuis, and Maarten de Rijke. 2023. Safe Deployment for Counterfactual Learning to Rank with Exposure-Based Risk Minimization. In *Proceedings of SIGIR*. doi:10.1145/3539618.3591760

[8] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atber, Ralf Herbrich, Stuart Bowers, and Joaquin Quiñonero Candela. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD)*. doi:10.1145/2648584.2648589

[9] Balázs Hidasi, Domonkos Tikk, and Alexandros Karatzoglou. 2023. Widespread Flaws in Offline Evaluation of Recommender Systems. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*.

[10] Olivier Jeunen. 2023. A Probabilistic Position Bias Model for Short-Video Recommendation Feeds. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. 675–681. doi:10.1145/3604915.3608777

[11] Olivier Jeunen and Bart Goethals. 2021. Top-K Contextual Bandits with Equity of Exposure. In *Proceedings of the 15th ACM Conference on Recommender Systems (RecSys '21)*. 310–320. doi:10.1145/3460231.3474248

[12] Olivier Jeunen and Aleksei Ustimenko. 2024. Δ-OPE: Off-Policy Estimation with Pairs of Policies. In *Proceedings of the 18th ACM Conference on Recommender Systems (RecSys '24)*. 878–883. doi:10.1145/3640457.3688162

[13] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining (WSDM)*. doi:10.1145/3018661.3018699

[14] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*. doi:10.1145/1935826.1935878

[15] Zhaoqi Li, Houssam Nassif, and Alex Luedtke. 2027. Estimation of subsidiary performance metrics under optimal policies. *Statistica Sinica* 37, 3 (2027).

[16] Luofeng Liao and Christian Kroer. 2023. Statistical Inference and A/B Testing for First-Price Pacing Equilibria. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*. Preprint: arXiv:2301.02276.

[17] Luofeng Liao, Christian Kroer, Sergei Leonenkov, Okke Schrijvers, Liang Shi, Nicolas Stier-Moses, and Congshan Zhang. 2025. Interference Among First-Price Pacing Equilibria: A Bias and Variance Analysis. In *International Conference on Learning Representations (ICLR)*. https://proceedings.iclr.cc/paper_files/paper/2025/hash/b6ffbbacbe2e56f2ec9a0da907382b4a-Abstract-Conference.html

[18] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. doi:10.1145/3209978.3210104

[19] Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. 2022. Causal Transformer for Estimating Counterfactual Outcomes. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*.

[20] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022).

[21] Harrie Oosterhuis. 2022. Doubly-Robust Estimation for Correcting Position-Bias in Click Feedback for Unbiased Learning to Rank. In *Proceedings of SIGIR*. doi:10.48550/arXiv.2203.17118 arXiv:2203.17118.

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

[23] Mohamed A. Radwan, Quinn Lanners, Jiasheng Zhang, Serkan Karakulak, Houssam Nassif, and Murat Ali Bayir. 2024. Counterfactual Evaluation of Ads Ranking Models through Domain Adaptation. In *Proceedings of the Workshops at the 18th ACM Conference on Recommender Systems (RecSys 2024)*.

[24] Otmane Sakhi, Imad Aouali, Pierre Alquier, and Nicolas Chopin. 2024. Logarithmic Smoothing for Pessimistic Off-Policy Evaluation, Selection and Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://neurips.cc/virtual/2024/poster/92960

[25] Uri Shalit, Fredrik D. Johansson, and David Sontag. 2017. Estimating Individual Treatment Effect: Generalization Bounds and Algorithms. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

[26] Rotem Stram, Rani Abboud, Alex Shtoff, Oren Somekh, Ariel Raviv, and Yair Koren. 2024. Mystique: A Budget Pacing System for Performance Optimization in Online Advertising. In *Companion Proceedings of The Web Conference (WWW)*.

[27] Kefan Su et al. 2024. AuctionNet: A Novel Benchmark for Decision-Making in Large-Scale Ad Auctions. https://proceedings.neurips.cc/paper_files/paper/2024/hash/ab9b7c23edfea0011507f7e1eae82cd2-Abstract-Datasets_and_Benchmarks_Track.html

[28] Adith Swaminathan, Alekh Agarwal, Miroslav Dudík, John Langford, and Damien Jose. 2017. Off-policy Evaluation for Slate Recommendation. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/1605.04812

[29] Adith Swaminathan and Thorsten Joachims. 2015. Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization. *Journal of Machine Learning Research (JMLR)* 16, 52 (2015), 1731–1755. https://jmlr.org/papers/v16/swaminathan15a.html

[30] Yuan Wang, Peifeng Yin, Zhiqiang Tao, Hari Venkatesan, Jin Lai, Yi Fang, and PJ Xiao. 2023. An Empirical Study of Selection Bias in Pinterest Ads Retrieval. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*. doi:10.1145/3580305.3599771

[31] Runzhe Wu, Masatoshi Uehara, and Wen Sun. 2023. Distributional Offline Policy Evaluation with Predictive Error Guarantees. arXiv:2302.09456 https://arxiv.org/abs/2302.09456

[32] Chen Zheng and Zhenyu Zhao. 2025. Algorithm Adaptation Bias in Recommendation System Online Experiments. arXiv:2509.00199 https://arxiv.org/abs/2509.00199 Also appears as CONSEQUENCES'25 workshop, co-located with ACM RecSys 2025.

[33] Guorui Zhou, Xiaoqian Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yan Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.