

Impression Share Prediction: An Offline Evaluation Task for Ranking Systems

Mohsen Malmir Houssam Nassif Danish Nasir Shaikh Taher Rahgooy Murat Ali Bayir
Meta Platforms, Inc. | ACM RecSys 2026, Minneapolis

Standard offline ranking metrics cannot reveal how a candidate model will redistribute impressions across objective buckets. We propose *impression share prediction* as an offline evaluation task: given a candidate ranking model, predict the distribution of impressions it would produce — before online evaluation.

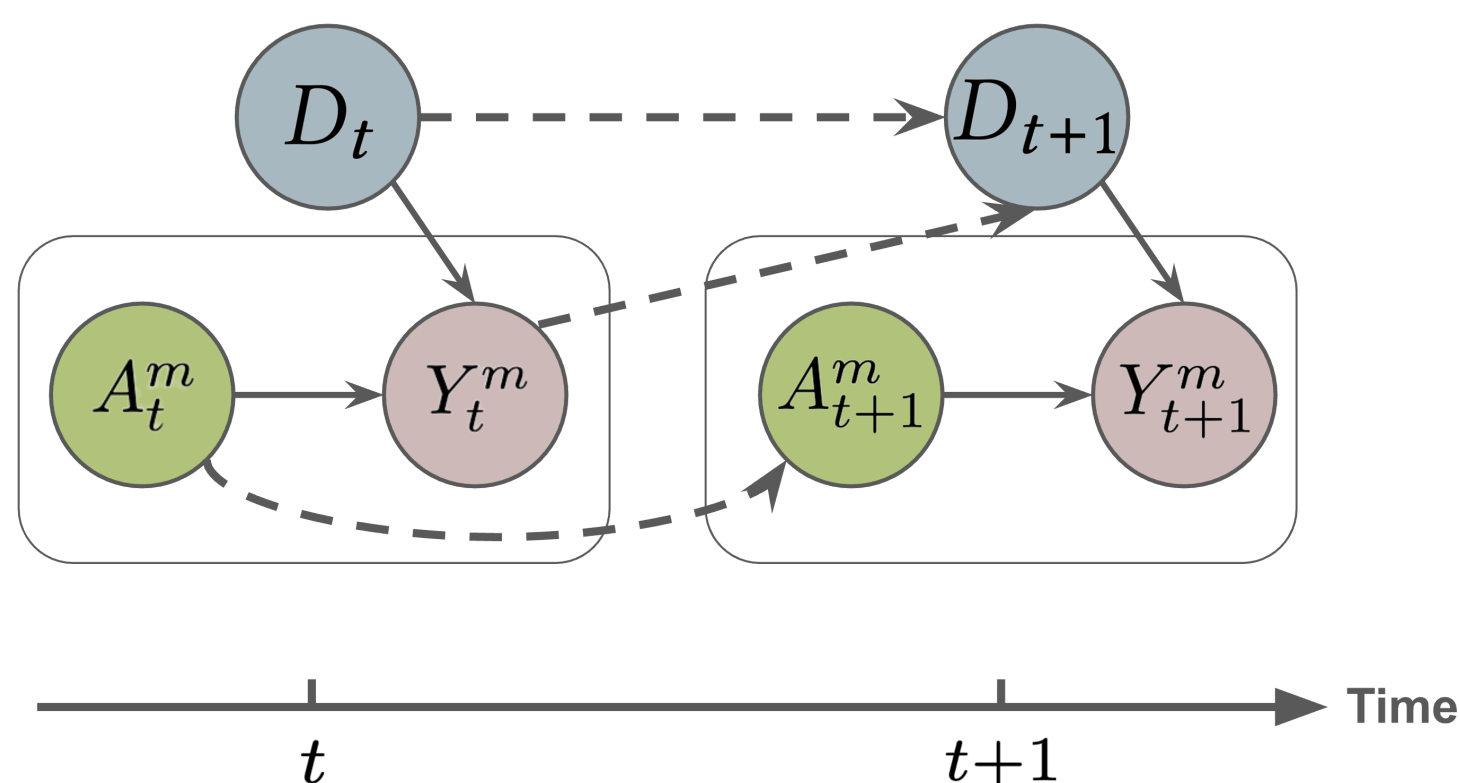
1. Problem / Motivation

- ▶ Offline metrics such as **AUROC**, **NE**, **RIG**, and **GAUC** evaluate predictive accuracy on logged impressions.
- ▶ Ranking models operate inside **shared auctions with pacing and delivery constraints**; changing a model can therefore redistribute which objective buckets receive impressions.
- ▶ Those allocation shifts can change downstream utility **even when aggregate offline metrics improve**.

Goal: predict the impression share a candidate model would produce — identified under the causal model (block 2), forecast in practice from its earliest online serving.

2. Causal Model & Prediction Framework

We model impression allocation as a dynamic system in which the **candidate model state** and the **shared delivery capacity state** jointly determine the observed impression-share vector.



A_t^m : candidate model state D_t : shared delivery capacity Y_t^m : impression share over C objective buckets

Objective buckets group impressions by campaign optimization goal (click, video view, conversion, ...).

Key structural property: there is no $D_t \rightarrow A_t^m$ edge. The candidate model is determined independently of the current delivery capacity D_t , so the causal effect is identified from observational data:

$$P(Y_t^m \mid do(A_t^m = a), D_t) = P(Y_t^m \mid A_t^m = a, D_t).$$

Prediction target:

$$(A_0^m, D_0) \rightarrow \hat{Y}_T^m, \quad T = 24 \text{ h}$$

Candidate model state + shared delivery capacity $\rightarrow$ forward impression share.

3. Predictors

Random Forest

Input: **Confidence** + **Coverage** (together the model state A_0^m) and **Capacity** (the delivery capacity state D_0), all read at $t=0$; feature groups defined in 4.

$$(A_0^m, D_0) \xrightarrow{\text{Random Forest regressor}} \hat{Y}_T^m$$

What it does: predicts each share component independently, then clips negative values and renormalizes onto the simplex.

Output: $\hat{Y}_T^m$, the 24 h forward impression-share vector.

Encoder-Conditioned

Input: the same **Confidence** + **Coverage** (A_0^m), but **Capacity** as a 2 h minute-level history rather than a $t=0$ snapshot.

$$D_{-2\text{h}:0} \xrightarrow{\text{PatchTST-style encoder}} \mathbf{h}_t, \quad (\mathbf{h}_t, A_0^m) \xrightarrow{\text{conditional MLP head}} \hat{Y}_T^m$$

What it does: the encoder summarizes recent delivery capacity dynamics into $\mathbf{h}_t$; the conditional MLP combines $\mathbf{h}_t$ with A_0^m and produces the share vector through softmax.

Output: $\hat{Y}_T^m$, the same 24 h forward impression-share vector.

4. Setup & Seen-Model Ablation

Data. 150 candidate models spanning multiple model families, $\sim 1.8\text{M}$ minute-level snapshots over ~ 5 weeks of production data. Temporal split: **23 training / 15 test days**; of the 103 models active in test, **83 are seen** in training and **20 are newly entered**. Seen models are evaluated **post-entry** — already interacting with the live system, so A_0^m and D_0 are directly observable. Held-out models are binned by time since first entry — **0–1 h** is the closest proxy to the **pre-entry** regime.

Features. **Confidence** (A_0^m , 22-dim) — 20-bin log-scale histogram of the candidate’s prediction scores, plus mean and variance. **Coverage** ($V^m \in \{0, 1\}^C$) — which objective buckets the candidate is eligible to score. **Capacity** (D_0 , $5 \times C$) — total, consumed and remaining capacity, pacing multiplier, and number of active campaigns, per objective bucket.

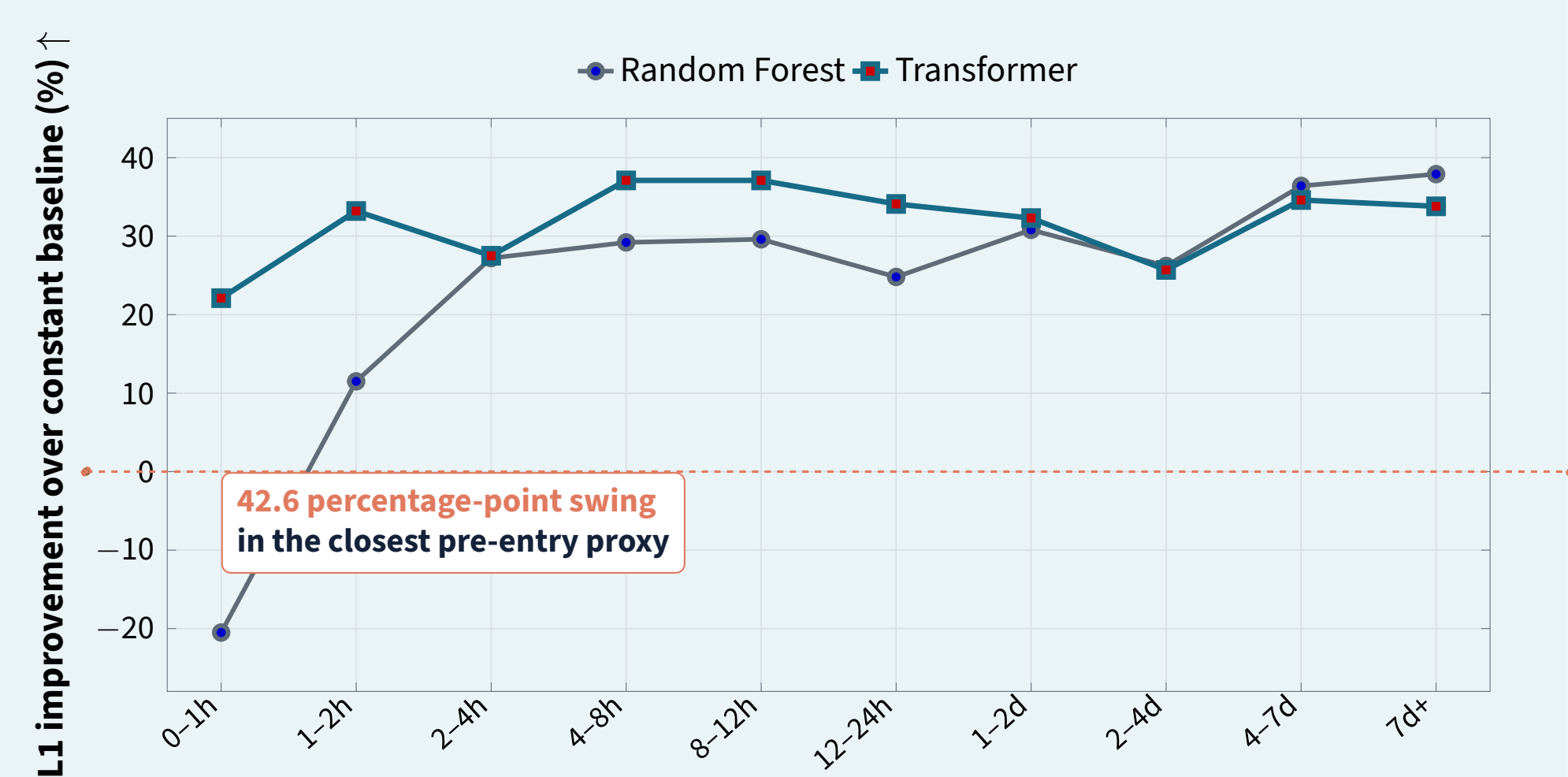
Metric. L1 distance $\downarrow$ (primary) and Spearman rank correlation $\uparrow$ (secondary), both reported as % improvement over a *constant baseline* emitting the training-set mean share vector.

SEEN MODELS (POST-ENTRY): FEATURE ABLATION

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%

5. Main Result: The First Hour Is the Hardest

Held-out models only, binned by **time since first entry**. Each line is one predictor’s % improvement in L1 over the constant baseline. The longer a candidate has served, the more D_0 has absorbed its own footprint on capacity and pacing, so the input gets easier — at **0–1 h** the model has not yet moved the system and D_0 still reflects the prior regime.



Spearman > 0.98 in all bins; omitted. The encoder’s advantage is largest precisely where the system has not yet adapted. A full *pre-entry offline evaluation* would compute A_0^m before the candidate serves any traffic; here it is estimated from a rolling window over its earliest online serving, so this setup is *early-entry forecasting*.

6. Takeaways

- ▶ **Missing signal:** Standard offline ranking metrics do not reveal how a candidate model will redistribute impressions across objective buckets.
- ▶ **Proposed metric:** We introduce **impression share prediction** as a complementary offline evaluation metric.
- ▶ **What it measures:** It estimates the distribution of impressions a candidate model is expected to generate across objective buckets before online evaluation.
- ▶ **Proof of concept:** Production experiments show impression shares are predictable — **+48.6%** L1 over the constant baseline on seen models, and **+22.1%** at **0–1 h** on held-out models.