Surrogate-Based Prevalence Measurement for Large-Scale A/B Testing

Zehao Xu
Pinterest, Inc.
Toronto, ON, Canada
zehaoxu@pinterest.com

Tony Paek
Pinterest, Inc.
New York, NY, USA
tpaek@pinterest.com

Kevin O'Sullivan
Pinterest, Inc.
New York, NY, USA
kosullivan@pinterest.com

Attila Dobi
Pinterest, Inc.
San Francisco, CA, USA
adobi@pinterest.com

Abstract

Online media platforms often track the share of impressions associated with specific content attributes, or *prevalence*, to evaluate trade-offs and set guardrails in A/B experiments. LLM-based labeling provides a high-fidelity reference measurement, but is cost-prohibitive to run per experiment, per arm, per segment, and per day on a platform with hundreds of concurrent experiments.

We describe a surrogate-based prevalence measurement system deployed in Pinterest's experimentation platform. The contribution is system-level rather than estimator-level: the system converts a recurring global LLM-labeled calibration stream into reusable bucket-level prevalences, which are then consumed by a per-experiment SQL metric and a delta-focused dashboard. In Pinterest's deployment, this calibration stream is provided by an existing daily batch LLM prevalence pipeline, so the surrogate incurs zero incremental labeling cost. More generally, teams without such infrastructure can instantiate the same pattern by running a recurring global calibration-labeling workflow, whose cost is amortized across downstream experiments rather than paid separately for each experiment, arm, segment, and day.

The system currently serves ~100 experiments and ~250 arms per day across four calibrated content categories, spanning Content Quality, Trust & Safety, and the company-wide Overall Holdout program. Relative to per-experiment LLM labeling, which in practice produces a single one-shot read per arm on only a small subset of experiments, the surrogate provides per-arm prevalence daily on over 20× as many concurrent arms under the same labeling budget, at near-log-query latency. On live experiments, the surrogate agrees with the LLM-based reference within 95% confidence intervals, including a null UI experiment where both methods correctly detect no effect. A day-level aggregation further improves sensitivity to small 2–5% relative shifts.

KDD '26, Jeju, Korea

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/XXXXXXX.XXXXXXX

Keywords

Prevalence Estimation; A/B Testing; Surrogate Outcomes; LLM as a Judge; Calibration; Production Measurement Systems

ACM Reference Format:
Zehao Xu, Tony Paek, Kevin O'Sullivan, and Attila Dobi. 2026. Surrogate-Based Prevalence Measurement for Large-Scale A/B Testing. In *Proceedings of The 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '26).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Modern recommender systems and media platforms must balance user engagement against the need to manage exposure to certain content attributes. Teams often summarize such exposure objectives in terms of *prevalence*: the fraction of impressions associated with a given target category. At the same time, product decisions are largely driven by large scale A/B experiments, where variants adjust ranking or filtering and are evaluated on both engagement and attribute specific exposure metrics.

One way to measure prevalence is to sample contents from traffic, label it with LLMs using expert validated prompts, and apply an estimation. In our setting, we use PPSWOR (probability proportional to size without replacement) sampling [8] and the Hansen–Hurwitz estimator [6] to obtain high quality and unbiased measurements.

However, directly using LLM-based prevalence as a default metric for every experiment is impractical. Running a separate LLM job per arm and per segment is expensive, and quickly becomes infeasible on a platform with hundreds of concurrent experiments. Moreover, experimenters often care about relatively small but meaningful changes in prevalence. A single LLM measurement per arm tends to emphasize the *absolute* level, so many small treatment-control deltas appear statistically non-significant. Running LLM labeling per experiment, per segment, and per day to address this would further amplify cost.

The contribution of this paper is not a new prevalence estimator but a deployed measurement system. We adopt an ML-score surrogate that turns a recurring global LLM-labeled calibration sample into reusable bucket-level prevalences: model scores are discretized into buckets, bucket-level prevalences are calibrated once, and per-experiment estimates are then computed from impression logs alone via a deterministic SQL metric and a delta-focused dashboard. In Pinterest's deployment, this calibration sample is supplied by an

existing daily LLM-based prevalence workflow for platform-level measurement, so the surrogate incurs zero incremental labeling cost.

The system has been deployed in Pinterest's experimentation platform since January 2026, currently serving ~100 experiments and ~250 arms per day across the Content Quality, Trust & Safety, and Overall Holdout programs. We state the methodology in Section 4, audit against the LLM-based reference on live A/B experiments in Section 5, describe system implementation in Section 6, and discuss design choices and the LLM cost outlook in Section 7.

2 Related Work

Our work sits at the intersection of five strands of prior work, and combines pieces from each into a deployed measurement system for online experimentation.

Surrogate outcomes. Surrogate outcome methods replace an expensive or slow-to-observe outcome with a cheaper proxy that is correlated with it [12]. Recent work has extended this idea to treatment effect estimation in the short run/long run setting [1]. We adopt this framing operationally: a fast model score serves as the surrogate, and an expensive LLM labeled measurement serves as the reference target the surrogate is calibrated against and periodically audited by.

Post-stratification and calibrated estimation. The underlying estimator is a post-stratification of model scores into discrete buckets, calibrated on an LLM labeled sample [7, 14]. Compared with direct Hansen–Hurwitz estimation [6] on a per experiment sample, post stratification reuses the calibration across experiments and segments, which is what makes the cost amortization in this paper possible.

Online experimentation and variance reduction. Large scale A/B testing has produced a rich literature on variance reduction for treatment effect estimation, of which CUPED [2] is the most widely deployed example. CUPED uses pre experiment covariates to reduce variance in the metric of interest. Our method addresses a different but complementary problem: the metric itself is too costly to *compute* per experiment, not just too noisy. The two ideas are composable; we leave a CUPED on $\Delta(d)$ extension to future work.

Score calibration. Mapping classifier scores to calibrated probabilities is a long studied problem [11, 15]. Bucketed ("histogram") calibration is the simplest such mapping and is convenient for our setting because it is computed once offline and consumed via a simple table lookup at SQL query time. We treat the choice of bucket scheme as a deployment decision rather than a contribution; alternatives are discussed in Section 7.1.

LLM-based labeling. Prompted large language models, optionally with expert reviewed prompts and calibration are increasingly used as scalable judges in practice [9, 10, 16]. At Pinterest, an LLM labeling pipeline of this form is in production as the reference quality measurement system for content attribute prevalence.

3 Prevalence Estimation

We consider the problem of estimating the prevalence of a category k on a large scale media platform. Let

$$\mathcal{K} = \{\text{Food-Recipe, Lawn and Garden, Gen-AI Generated, ...}\}$$

denote the set of content attributes considered in this study, and fix a particular $k \in \mathcal{K}$.

Our target quantity is the prevalence of category k within S, which can denote the full population or a specific subset such as an experiment arm (control, treatment), a user demographic (country, age), an app surface, or intersections of these.

The engineering details of our production prevalence pipeline are described in Farooq et al. [5]. In this section, we briefly recap the core estimator and introduce notation that we will use throughout the rest of the paper, in particular when we describe the deployed ML score surrogate method and its calibration.

3.1 Notation

We consider a large population of content items, indexed by $i = 1, \ldots, N$. For each item i, we observe:

- I_i: the total number of impressions of item i over a given time window.
- $Z_{i,k} \in \{0, 1\}$: a label indicating whether item i is truly in category k (1) or not (0).

For a given segment S, let $\mathcal{D}(S) \subseteq \{1, \ldots, N\}$ denote the set of items that receive impressions from S, and let $I_i(S)$ be the number of impressions of item i from S over the time window of interest. The total impressions of item i satisfy $I_i = \sum_S I_i(S)$. The prevalence of category k in segment S can be written as:

$$\begin{aligned} \mathcal{P}_k(S) &= \mathbb{P}(Z_k = 1 \mid \text{impressions} \in S) \\ &= \frac{\sum_{i \in \mathcal{D}(S)} Z_{i,k} I_i(S)}{\sum_{i \in \mathcal{D}(S)} I_i(S)}, \end{aligned} \tag{1}$$

When S is the entire population, $\mathcal{D}(S) = \{1, \ldots, N\}$ and $I_i(S) = I_i$, recovering the global prevalence definition.

3.2 Sampling Design and Hansen–Hurwitz Estimator

Directly labeling *all* items in $\mathcal{D}(S)$ (often billions of items across many segments) is infeasible, so we estimate these quantities from a sample. We adopt a PPSWOR design, in which items are sampled with probabilities proportional to a chosen size measure. Compared with simple random sampling, PPSWOR allows us to oversample items that are more informative (e.g., high impression items), achieving similar statistical power with a much smaller sample size.

For category k and segment S, each item $i \in \mathcal{D}(S)$ is assigned a sampling weight $w_{i,k}(S) \propto f\big(I_i(S)\big)$, and is selected into the sample with probability $p_{sample,i,k}(S) = w_{i,k}(S)/\sum_{j \in \mathcal{D}(S)} w_{j,k}(S)$. Here f is a function of impressions; it can be $I_i(S)$ itself, or a combination, such as $I_i(S)$ multiplied by other factors (e.g., model score). We discuss concrete choices of f in more detail in Section 6.1.

Suppose we draw a sample of size n from $\mathcal{D}(S)$, with sampled indices $\{i_1, \ldots, i_n\}$. The Hansen–Hurwitz estimator for the total target category impressions in S and the prevalence in segment S are estimated as

$$\widehat{T}_k(S) = \frac{1}{n} \sum_{t=1}^{n} \frac{Z_{i_t,k} I_{i_t}(S)}{p_{\text{sample},i_t,k}(S)}, \quad \widehat{\mathcal{P}}_k(S) = \frac{\widehat{T}_k(S)}{\sum_{j \in \mathcal{D}(S)} I_j(S)}. \tag{2}$$

In practice, both $\sum_{j \in \mathcal{D}(S)} I_j(S)$ and $\sum_{j \in \mathcal{D}(S)} w_{j,k}(S)$ are computed directly from the underlying logs for segment S.

3.3 Weighted Reservoir Sampling

To efficiently draw a PPSWOR sample at scale, we use *weighted random sampling with a reservoir* [4]. Each item i is assigned an index

$$U_i^{1/w_{i,k}(S)}, \tag{3}$$

where $U_i \sim \text{Uniform}(0,1)$ and $w_{i,k}(S)$ is the sampling weight. The n items with the largest index are selected into the sample. This procedure implements PPSWOR in a single streaming pass, without the need to know N in advance.

3.4 LLM Labeling

A central design choice in our prevalence pipeline is the source of "ground truth" labels for whether an item belongs to a category $k \in \mathcal{K}$. Beyond the widely used LLM judges discussed in Section 2, an alternative is human expert annotation (specialists or trained raters), which provides very high quality labels but does not scale: obtaining tens of thousands of fresh labels per category would be much more costly [13].

For each sampled item i, the LLM produces a binary label $Z_{i,k} \in \{0,1\}$, where $Z_{i,k} = 1$ indicates that the item is judged to belong to category k, and $Z_{i,k} = 0$ otherwise. Substituting these labels into the Hansen–Hurwitz estimator and the prevalence formula (2) yields an unbiased estimate of category prevalence.

We use this LLM-based estimator both as the offline calibration target for the ML score surrogate (Section 6.1) and as an ongoing reference against which the deployed surrogate is audited on live experiments (Section 5).

4 Methodology: ML Score Bucket Surrogate Prevalence

Estimating per experiment, per arm prevalence is, in principle, a well studied problem: given a sample of labeled items, the Hansen–Hurwitz estimator produces an unbiased prevalence estimate, and using model scores as auxiliary information to reduce variance is a standard technique [7, 14]. The challenge addressed by our system is therefore not whether prevalence can be estimated for any one experiment, but how to produce *thousands* of such estimates per day across hundreds of concurrent experiments and segments, with calibration uncertainty correctly propagated and at near log query latency – without incurring per experiment LLM labeling cost. This is a system level problem rather than an estimator level one. We address this by calibrating model-score buckets once and reusing that calibration across downstream experiment cohorts.

4.1 Bucket Level Prevalence

Let $m_{i,k} \in [0,1]$ be the model score for category k on item i; higher values indicate that item i is more likely to be true in category k. We discretize the model scores into B buckets:

$$\mathcal{B} = \{b_1, \ldots, b_B\},$$

where each bucket b_j corresponds to an interval

$$b_1 = [u_0, u_1),\ b_2 = [u_1, u_2),\ \ldots,\ b_B = [u_{B-1}, u_B],$$

with $0 = u_0 < u_1 < \cdots < u_{B-1} < u_B = 1$.

For a given segment S and category k, we define the *bucket level prevalence* as

$$\mathcal{P}_{k,b}(S) \;=\; \mathbb{P}(Z_k = 1 \mid m_k \in b,\ \text{impression} \in S)\,, \tag{4}$$

i.e., the probability that impressions from segment S are truly in category k, conditional on model score m_k falling in bucket b.

4.2 Prevalence Estimation and Variance Propagation

Let $c_{k,b}(S)$ denote the share of impressions from S whose model scores for category k fall into bucket b:

$$\begin{aligned} c_{k,b}(S) &= \mathbb{P}(m_k \in b \mid \text{impression} \in S) \\ &= \frac{\text{Impressions in } S \text{ with } m_k \in b}{\text{Total Impressions in } S}, \end{aligned} \tag{5}$$

so that $\sum_{b \in \mathcal{B}} c_{k,b}(S) = 1$.

Using the law of total probability, we estimate the prevalence of category k in segment S as:

$$\widehat{\mathcal{P}}_k(S) \;=\; \sum_{b \in \mathcal{B}} c_{k,b}(S) \cdot \widehat{\mathcal{P}}_{k,b}(S). \tag{6}$$

In practice, we compute the calibration once on a large "global" segment (i.e., all traffic) and reuse the resulting bucket level estimates $\widehat{\mathcal{P}}_{k,b}$ across all segments S, yielding the segment level approximation:

$$\widehat{\mathcal{P}}_k(S) \;\approx\; \sum_{b \in \mathcal{B}} c_{k,b}(S) \cdot \widehat{\mathcal{P}}_{k,b}. \tag{7}$$

We defer the rationale for this design choice and a comparison against the segment specific alternative to Section 7.2. Intuitively, (7) is a bucketed approximation to Equation (1): we treat the bucket level prevalences as a learned "score → likelihood" mapping, and then reweight by the score bucket distribution of impressions in segment S.

Also, we can assume the bucket level estimators are independent across b (reasonable given disjoint buckets), then applying standard variance propagation to (7) yields:

$$\text{Var}\Big(\widehat{\mathcal{P}}_k(S)\Big) \;\approx\; \sum_{b \in \mathcal{B}} c_{k,b}(S)^2\, \text{Var}\Big(\widehat{\mathcal{P}}_{k,b}\Big)\,. \tag{8}$$

5 Production Audits Against the LLM Reference

Before describing the implementation (Section 6), we first report production audits that establish the surrogate's agreement with the LLM-based reference. Because the deployed surrogate produces a metric that product teams act on, we run periodic audits against the LLM-based reference estimator on live A/B experiments, and report the results back to metric owners. Three such audits: a large content filtering intervention, a UI only refresh with no expected prevalence shift, and a subcategory targeting experiment—are summarized below as case studies, assessing how well the surrogate recovers both (i) the absolute prevalence levels in each arm, and (ii) the between-group deltas.

5.1 Experiment A: Target Category Filtering

Our first case study covers a production A/B experiment that applies additional filtering for two target categories, k_1 and k_2, on a major recommendation surface. The product team's goal is to

evaluate how aggressive filtering of these categories affects user experience and engagement; the surrogate provides their primary read on category exposure throughout the experiment window. The experiment includes three arms:

- **Control:** baseline production settings.
- **Treatment$_1$:** baseline settings plus additional filtering for category k_1, removing items with $m_{i,k_1} \geq 0.70$
- **Treatment$_2$:** baseline settings plus additional filtering for category k_2, removing items with $m_{i,k_2} \geq 0.10$.

For each arm, we estimate the prevalences by:

(1) **LLM-based estimator (reference):** impression weighted sampling, and LLM labeling for each arm.
(2) **ML score surrogate estimator:** using the calibrated bucket prevalences $\widehat{\mathcal{P}}_{k_\ell,b}$ for each category k_ℓ and bucket b, together with the arm specific impression shares $c_{k_\ell,b}(\text{Control})$ and $c_{k_\ell,b}(\text{Treatment}_\ell)$, aggregated according to Equation (7). In this experiment we use $B = 10$ equally spaced buckets.

Figure 1 shows the impression share distributions $c_{k_\ell,b}(\text{Control})$ (salmon) and $c_{k_\ell,b}(\text{Treatment}_\ell)$ (cyan) over score buckets across the entire experiment window for both categories, under the given thresholds, respectively. Both treatments shift impression mass away from higher-score buckets targeted by the filtering thresholds.

Table 1 summarizes the prevalence estimates $\widehat{\mathcal{P}}_{k_\ell}(\text{Control})$ and $\widehat{\mathcal{P}}_{k_\ell}(\text{Treatment}_\ell)$.

- For k_1, the treatment reduces prevalence from approximately 3.6% to 2.9% (around 19.4% reduction) under the LLM-based estimator and from 3.8% to 2.8% (26.3% reduction) under the ML score surrogate, corresponding to treatment-control deltas of 0.70% and 1.03%, respectively.
- For k_2, the treatment reduces prevalence from about 1.3% to 1.0% (LLM; 23% drop) and from 1.4% to 0.9% (Surrogate; 35.7% drop), with deltas of 0.31% and 0.49%.

In all cases, the surrogate's arm level estimates lie within the 95% confidence intervals of the LLM-based reference, and the inferred treatment effects agree in sign and magnitude.

Table 1: Comparison of LLM-based vs. ML score surrogate prevalence in Experiment A for categories k_1 and k_2. The ranges denote 95% confidence intervals

Metric	LLM-based.	ML score surrogate.
$\widehat{\mathcal{P}}_{k_1}(\text{Control})$	3.61% [3.40%, 3.82%]	3.81% [3.67%, 3.95%]
$\widehat{\mathcal{P}}_{k_1}(\text{Treatment}_1)$	2.91% [2.72%, 3.10%]	2.78% [2.64%, 2.92%]
$\Delta_{k_1,\ \text{Treat}_1-\text{Ctrl}}$	−0.70%	−1.03%
$p$value	0.00 (**Stats-Sig**)	0.00 (**Stats-Sig**)
$\widehat{\mathcal{P}}_{k_2}(\text{Control})$	1.32% [1.20%, 1.43%]	1.36% [1.20%, 1.52%]
$\widehat{\mathcal{P}}_{k_2}(\text{Treatment}_2)$	1.01% [0.91%, 1.11%]	0.87% [0.70%, 1.04%]
$\Delta_{k_2,\ \text{Treat}_2-\text{Ctrl}}$	−0.31%	−0.49%
$p$value	0.00 (**Stats-Sig**)	0.00 (**Stats-Sig**)

5.2 Experiment B: UI Only Change with No Expected Prevalence Shift

Our second case study covers a user interface refresh where we do not expect any change in category k_1 and k_2 prevalence. The experiment updates the web overflow menu by modifying the hide/report icon, without altering the underlying functionality or ranking logic.

Table 2 summarizes the estimates over the whole experiment window. For k_1, the control and treatment arms have very similar prevalences: 3.44% vs. 3.31% under the LLM-based estimator and 3.39% vs. 3.40% under the ML score surrogate. For k_2, the prevalences are likewise close: 1.32% vs. 1.40% (LLM) and 1.39% vs. 1.35% (surrogate), with treatment-control deltas near zero in all cases.

The surrogate estimates lie within the 95% confidence intervals of the LLM-based reference, and none of the inferred treatment effects are statistically significant.

Table 2: Comparison of LLM-based vs. ML score surrogate prevalence for categories k_1 and k_2 in Experiment B.

Metric	LLM-based	ML score surrogate
$\widehat{\mathcal{P}}_{k_1}(\text{Control})$	3.44% [3.23%, 3.64%]	3.39% [3.25%, 3.53%]
$\widehat{\mathcal{P}}_{k_1}(\text{Treatment})$	3.31% [3.11%, 3.51%]	3.40% [3.26%, 3.53%]
$\Delta_{k_1,\ \text{Treat}-\text{Ctrl}}$	−0.13%	0.01%
$p$value	0.38 (**No Stats-Sig**)	0.95 (**No Stats-Sig**)
$\widehat{\mathcal{P}}_{k_2}(\text{Control})$	1.32% [1.19%, 1.45%]	1.39% [1.22%, 1.56%]
$\widehat{\mathcal{P}}_{k_2}(\text{Treatment})$	1.40% [1.26%, 1.53%]	1.35% [1.18%, 1.52%]
$\Delta_{k_2,\ \text{Treat}-\text{Ctrl}}$	−0.07%	−0.04%
$p$value	0.40 (**No Stats-Sig**)	0.67 (**No Stats-Sig**)

5.3 Experiment C: Detecting a Small Production Shift Beyond the Reach of a Single LLM Measurement

Our third case study covers an experiment that targets a subcategory of category k_1. Because this subcategory accounts for only a small fraction of the overall k_1 impression share, any change in $\mathcal{P}_{k_1}$ is expected to be modest but systematic. This is the regime in which the deployed surrogate is most useful: the per arm calibration uncertainty of any single LLM measurement is wide compared with the expected effect, while the surrogate can be re-evaluated daily without any additional label cost.

Aggregated throughout the experiment window, the LLM-based estimator yields $\widehat{\mathcal{P}}_{k_1}(\text{control}) \approx 3.83\%$, and $\widehat{\mathcal{P}}_{k_1}(\text{treatment}) \approx 3.67\%$ with a two sided p-value ≈ 0.31 and overlapping 95% confidence intervals $[3.69\%, 3.96\%]$ and $[3.53\%, 3.80\%]$, respectively.

The deployed surrogate, in contrast, computes a daily prevalence per arm and the corresponding daily effect

$$\Delta(d) = \widehat{\mathcal{P}}_k(\text{treatment}; d) - \widehat{\mathcal{P}}_k(\text{control}; d), \tag{9}$$

under a fixed calibration snapshot. Aggregating these day level deltas using a sign test [3] (Section 6.2.2) gives a mean delta of approximately −0.16%, a 4.1% relative reduction on a baseline (control) prevalence of about 3.8% (using ML surrogate), with daily values that are consistently negative across the experiment window

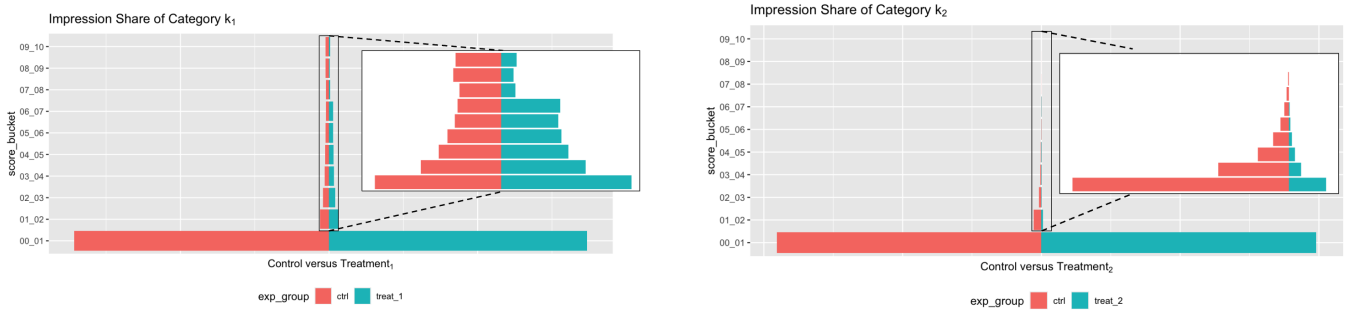


Figure 1: Bucket level impression share shifts for categories k_1 and k_2 in Experiment A. The treatment arm reduces exposure primarily by shifting impressions out of higher score buckets.

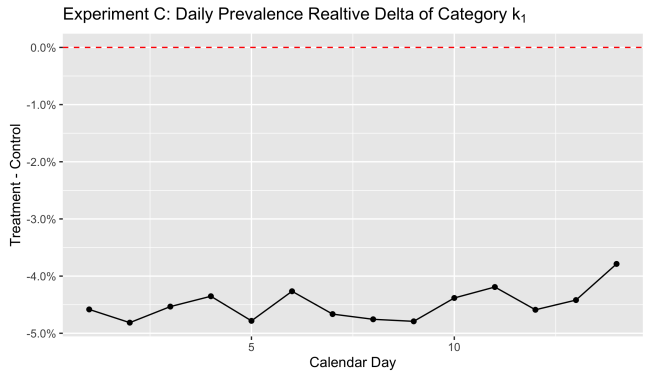


Figure 2: Relative prevalence reduction vs. calendar day: $\Delta(d)/\widehat{\mathcal{P}}_{k_1}(\text{control}; d)$ for category k_1 in Experiment C.

(Figure 2). The corresponding day level sign-test p-value ≈ 0.00, indicating a statistically significant shift at the experiment level.

As a sanity check and to guard against false positives, we also apply the same day level aggregation to Experiment B, where no prevalence shift is expected. This yields $p\text{-value}_{k_1} \approx 0.75$ and $p\text{-value}_{k_2} \approx 0.25$, aligned with the null hypothesis.

This case study illustrates a production relevant sensitivity gap: a small but systematic around -4% relative reduction is invisible to a single per arm LLM measurement at the labeling budget we can afford, but is recovered by the deployed surrogate via day level aggregation.

6 Implementation

We now describe how the surrogate is deployed in production: an offline calibration pipeline that periodically refreshes $\{\widehat{\mathcal{P}}_{k,b}\}$ from the always-on LLM pipeline's label stream (Section 6.1), a daily online integration with the experiment platform that materializes per experiment prevalences and renders the dashboard (Section 6.2), and the operational cadence under which both run (Section 6.3).

6.1 Offline Calibration Pipeline

The calibration step estimates the bucket-level prevalences $\{\widehat{\mathcal{P}}_{k,b}\}_{b\in\mathcal{B}}$ and their variances $\{\mathrm{Var}(\widehat{\mathcal{P}}_{k,b})\}_{b\in\mathcal{B}}$ that the online metric consumes. In Pinterest's deployment, these quantities are computed from the labeled stream already produced by the daily batch LLM prevalence workflow, which maintains a global PPSWOR sample via weighted reservoir sampling $w_{i,k} \propto f(I_{i,k})$. Reusing this stream gives the surrogate zero incremental labeling cost.

Given the sampled and LLM-labeled stream, we estimate each bucket-level prevalence by applying the Hansen–Hurwitz estimator with the bucket itself as the segment, $S = b$:

$$\widehat{\mathcal{P}}_{k,b} \;=\; \widehat{\mathcal{P}}_k(S = b). \tag{10}$$

Concretely, let $I_i(b)$ be the number of impressions of item i arising from bucket b in the calibration window, and let $\mathbb{1}_{i\in b}$ be the indicator that item i receives at least one impression from b. We estimate the joint probability that an impression is both in category k and in bucket b as

$$\widehat{P}\big(Z_{i,k} = 1 \cap \text{impression} \in b\big) \;=\; \frac{1}{n}\sum_{t=1}^{n} \frac{Z_{i_t,k}\,\mathbb{1}_{i_t\in b}\,I_{i_t}(b)}{p_{\text{sample},i_t,k}}. \tag{11}$$

Then, the marginal probability that a random impression falls in bucket b is computed directly from logs: $P(\text{impression} \in b) = \frac{\sum_{i\in\mathcal{D}(b)} I_i(b)}{\sum_{i\in\mathcal{D}} I_i}$. By the definition of conditional probability, the estimated prevalence of category k in bucket b is then

$$\widehat{\mathcal{P}}_k(S = b) \;=\; \frac{\widehat{P}\big(Z_{i,k} = 1 \cap \text{impression} \in b\big)}{P(\text{impression} \in b)}. \tag{12}$$

A practical concern in this shared sampling design is that model scores are highly skewed toward low values. If the always-on pipeline used impressions alone as weights, i.e., $w_{i,k} \propto I_i$, the resulting score distribution in the sample would be dominated by the lowest bucket: in our data, as shown in Figure 3 (a), roughly 80% of sampled impressions fall into the bucket $[0, 0.1)$, while buckets with larger scores (e.g., $m_{i,k} > 0.3$) each contain only about 2% of the mass. With a sample of 10,000 items, fewer than 200 would lie in each higher bucket, limiting statistical power for estimating $\mathcal{P}_{k,b}$ at the upper end of the score range.

The always-on pipeline therefore uses the model score as auxiliary information [14], i.e., $w_{i,k} \propto I_i\, m_{i,k}$, which boosts the inclusion probability of higher score items. On the same underlying dataset,

shown in Figure 3 (b), this design yields a much more balanced bucket distribution: every bucket has at least ∼7% of the sample, and the (0.9, 1] buckets account for roughly 16% each. Although this weighting was originally chosen so that the LLM-based reference estimator has good precision in the high score tail, the surrogate inherits a sample design that is already well suited to bucket level calibration, with minimal operational overhead.

6.2 Online Integration

The online side of the system runs as a two stage pipeline integrated with Pinterest's experimentation platform: a daily per experiment prevalence computation job, followed by a dashboard layer that produces the experiment level read from the day level results.

6.2.1 Stage 1: daily prevalence per experiment. For each qualified experiment — currently those run by the Content Quality, Trust & Safety, and Overall Holdout programs — a scheduled job runs once per day per arm and:

(1) joins the experiment assignment table (which users are in which arm) with the daily impression logs to obtain per (arm, day) impressions,
(2) pulls model scores for the calibrated category k on those impressions,
(3) aggregates to obtain the impression shares $c_{k,b}(S; d)$ per bucket per arm per day,
(4) reads the latest calibration snapshot $\{\widehat{\mathcal{P}}_{k,b}\}_{b\in\mathcal{B}}$ and computes the arm level prevalence $\widehat{\mathcal{P}}_k(S; d)$ together with its analytic 95% confidence interval.

The impression and assignment join is resolved here, in the scheduled job, rather than at query time. Each (experiment, arm, day) row is therefore computed once and reused by every downstream consumer.

6.2.2 Stage 2: dashboard rendering. The experimentation platform dashboard consumes the prevalence table from *Stage 1* and produces the experiment level verdict using a delta focused inference layer.

The motivation is that the per arm analytic confidence intervals are often dominated by the bucket level calibration variance $\mathrm{Var}(\widehat{\mathcal{P}}_{k,b})$ rather than by the experiment level variation in the bucket mixes $c_{k,b}(S)$. With a sufficiently large calibration sample, this calibration variance could be made negligible, but in practice our calibration budget is constrained by LLM labeling cost. At realistic calibration sizes the per arm CIs for $\widehat{\mathcal{P}}_k(\text{control})$ and $\widehat{\mathcal{P}}_k(\text{treatment})$ are therefore best interpreted as answering: *"If we repeated the global LLM calibration many times, where would the absolute prevalence for this arm lie?"*; a useful question when the experimental shift is large (e.g., Experiment A in Section 5.1, where prevalence moves by 20% or more), but a poor question in the more common production setting where a 2%–5% relative change is already meaningful (e.g., Experiment C in Section 5.3).

In that regime, calibration uncertainty can dominate the per arm CI and cause small but systematic treatment effects to appear statistically non-significant. The dashboard therefore reports inference computed on the day level deltas $\Delta(d)$ (Equation 9). Concretely, on day D the dashboard collects the trailing window of daily deltas $\{\Delta(D-m), \ldots, \Delta(D-1)\}$ for the experiment arm pair (we use minimum $m = 10$ days in production) and reports:

- the mean delta $\bar{\Delta}$ over the window,
- an empirical 95% confidence interval $[\Delta_{0.025}, \Delta_{0.975}]$ from the quantiles of $\{\Delta(d)\}$, and
- a sign-test p-value [3] based on the fraction of days on which $\Delta(d)$ is positive versus negative.

If the sign-test p-value is below the chosen significance level the observed mean effect $\bar{\Delta}$ is unlikely to be explained by random variation under the null hypothesis of no daily effect (Figure 2); otherwise, we fail to reject the null. These statistics are surfaced to the experiment owner as the headline prevalence read for day D. For experiments shorter than the m-day minimum, only the day level prevalences and a per day delta are shown. Because *Stage 1* has already materialized the prevalence rows the dashboard depends on, this dashboard query is effectively a small read over precomputed numbers and returns at near log query latency.

6.3 Operations

The cost structure depends on whether a platform already operates a recurring LLM prevalence workflow. In Pinterest's deployment, the daily batch LLM labeling pipeline is already funded as part of the platform-level reference measurement system, so the surrogate adds only calibration computation and the per-experiment SQL metric, with zero incremental labeling cost. For teams without such a workflow, the same architecture requires a recurring global calibration-labeling job; the labeling cost is not zero, but it is paid once at the platform or category level and amortized across all enrolled experiments.

The relevant comparison is per-experiment LLM labeling. At roughly 60,000 samples per arm per day, labeling every cell would require on the order of 10^9 labels per category per year, which is operationally infeasible. In practice, the realistic alternative is a one-shot LLM measurement per arm for only a small fraction of experiments under the available labeling budget (Table 3).

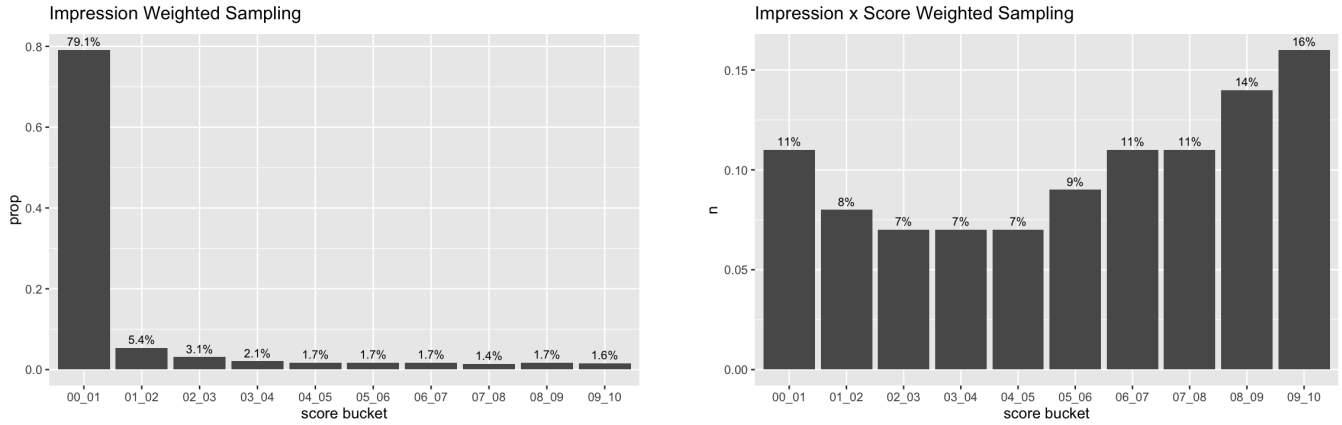
Table 3: Relative Operational Comparison: LLM vs. Surrogate

Metric	LLM-based	ML score surrogate
# of Measured Exp	Baseline	> 20×
Extra Labeling Cost	Baseline	$0
Frequency	One Run Per Exp	Daily Run Per Arm
Latency	10 to 24 Hours	<10s

Even on those covered arms, a single LLM measurement carries its own sampling uncertainty: small but systematic shifts in the 2–5% relative range, the regime that dominates the production caseload, are typically statistically indistinguishable from noise within a one shot confidence interval. The deployed surrogate closes both gaps: it provides per arm prevalence on every enrolled arm, and by re-evaluating daily under a fixed calibration snapshot, it accumulates day level deltas that recovers small effects that any single LLM measurement would miss.

Audits against the LLM-based reference are run on two cadences.

- Daily platform level prevalence is audited every day, since the always-on LLM pipeline produces a daily platform measurement that the surrogate can be compared against without additional labeling cost.



(a) Impression weighted sampling ($w_{i,k} \propto I_i$).

(b) Impression × score sampling ($w_{i,k} \propto I_i m_{i,k}$).

Figure 3: Comparison of score bucket distributions under two sampling schemes. Left: sampling with weights proportional to impressions alone produces a very low score heavy sample, leaving few examples in high score buckets. Right: sampling with weights proportional to impression × model score yields a more balanced distribution across buckets, improving the precision of bucket level prevalence estimates.

- Per arm audits on enrolled A/B experiments are run periodically, at program onboarding and on a recurring cadence, since each per arm audit requires fresh LLM labeling of arm specific samples.

Across approximately 300 such audits in total, the surrogate's 95% confidence interval contains the LLM-based reference point estimate in 92% of evaluations, and the surrogate's CI overlaps the LLM-based reference's CI in 96% of evaluations.

7 Discussion and Future Work

7.1 Choice of Score Buckets

The deployed system supports both fixed and adaptive score bucketization. For some calibrated categories, we use a simple default of $B = 10$ equal-width buckets over $[0, 1]$, combined with the impression×score weighted sampling described in Section 6.1. This default is easy to implement and stable across daily refreshes, but it can be unreliable for skewed score distributions and rare positives: many samples may pile up near zero while higher-score buckets receive too few labels or too few positives to support stable bucket-level prevalence estimates.

For such categories, we use adaptive binning in production. The method starts from equal-frequency score micro-bins, then merges neighboring micro-bins until each retained bucket satisfies minimum sample and positive-label thresholds. If too many buckets remain, it merges the adjacent pair with the smallest information loss, favoring merges between bins with similar empirical positive rates or small sample sizes. The final boundaries are rounded and cleaned to cover $[0, 1]$ and remain stable at scoring time.

We treat bucketization as a deployment choice rather than an estimator-level contribution. Equal-width buckets are sufficient when stable; adaptive buckets are used when rare positives or score skew make fixed-width calibration unreliable. More broadly, the surrogate can represent content or context features beyond raw model score, and the appropriate bucketization strategy depends on the application, calibration budget, and operational stability requirements.

7.2 Choice of Calibration Granularity

The deployed system currently applies a single global calibration $\widehat{\mathcal{P}}_{k,b}$ to every segment S, even though the framework of Section 4 also supports segment-specific calibrations $\widehat{\mathcal{P}}_{k,b}(S)$. This choice works well when, for a fixed category k and bucket b, the bucket-level prevalence $\mathcal{P}_{k,b}(S)$ is stable across the segment family of interest. In that regime, segment differences are mainly captured by the impression-share distribution $c_{k,b}(S)$, while the shared calibration pools labels across segments and reduces noise. We observe this pattern for many reads, such as surface-level and country-level. Figure 4 illustrates the country case: the LLM-based country estimates and the surrogate agree on absolute level, while the surrogate produces smoother per-country trajectories because its country-specific component comes from full impression logs and its label-derived component is pooled globally.

Global calibration is not universally optimal. For some segment families, such as top-level content verticals, the conditional prevalence within the same score bucket can differ materially across segments; for example, GenAI prevalence in an Art vertical may differ from that in Home Decor at the same model-score bucket. In such cases, a global calibration can introduce segment-specific bias, and calibration granularity should be chosen empirically. A future extension is a bucket-level heterogeneity check for each category k, segment family $\{S_1, \ldots, S_J\}$, and bucket b:

$$H_0 : \mathcal{P}_{k,b}(S_1) = \cdots = \mathcal{P}_{k,b}(S_J).$$

This could be implemented using Wald or χ^2 tests over the Hansen–Hurwitz bucket estimates, with multiple-testing correction across buckets. If heterogeneity appears in high-mass buckets and each

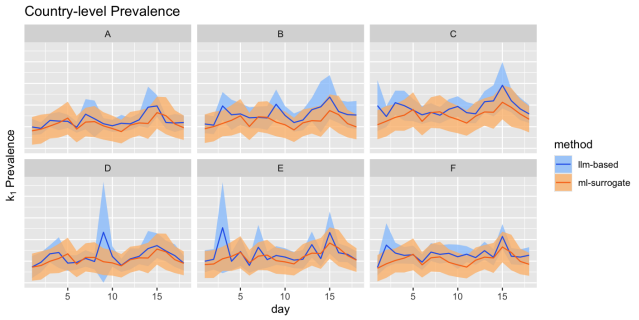


Figure 4: Daily k_1 prevalence for the six highest-engagement countries (A–F), comparing the LLM-based estimator and the ML-score surrogate over a 30-day window; ribbons denote 95% confidence intervals.

segment has enough labels and positives, segment-specific calibration may be preferable; otherwise, global calibration remains more stable.

We have not yet deployed automatic calibration-granularity selection. A natural middle ground is partial pooling, which shrinks segment-specific $\widehat{\mathcal{P}}_{k,b}(S)$ toward the global $\widehat{\mathcal{P}}_{k,b}$, capturing real segment-level differences when supported by data while avoiding high variance in sparse segment-bucket cells.

7.3 Outlook on LLM Cost Trends

A key motivation for the deployed system is that large scale LLM labeling is currently too expensive to serve as a default per experiment metric. In the longer term, as LLM inference becomes cheaper and more tightly integrated into serving stacks, it may become feasible to label much larger fractions of the corpus (or even all content) for multiple content attributes.

In that regime, the relative importance of the surrogate may decrease for absolute prevalence estimation. However, the contribution of this work is system level: amortizing one calibration across hundreds of concurrent experiments, materializing per experiment results in a single SQL metric, and separating absolute levels from deltas in the dashboard layer. We expect these ideas to remain useful even when LLM labeling is no longer the primary bottleneck — at that point, the contents of the calibration table can change, but the system pattern carries over.

Acknowledgments

We thank the following collaborators for their support and feedback throughout this project: Minli Zang, Benjamin Thompson, Xiaohan Yang, Wenjun Wang, Huan Yu, Faisal Farooq, Darren Reger, Andrey Gusev, Aravindh Manickavasagam, Qinglong Zeng, Jianjin Dong, Ziming Yin, Cindy Zhang, Gerardo Gonzalez, Ahmed Fayez, Yasmin ElBaily and Sari Wang

References

[1] Susan Athey, Raj Chetty, Guido W Imbens, and Hyunseung Kang. 2025. The Surrogate Index: Combining Short-Term Proxies to Estimate Long-Term Treatment Effects More Rapidly and Precisely. *The Review of Economic Studies* (2025).

[2] Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. 2013. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *Proceedings of the sixth ACM international conference on Web search and data mining*. 123–132.

[3] Wilfrid J Dixon and Alexander M Mood. 1946. The statistical sign test. *J. Amer. Statist. Assoc.* 41, 236 (1946), 557–566.

[4] Pavlos S Efraimidis and Paul G Spirakis. 2006. Weighted random sampling with a reservoir. *Information processing letters* 97, 5 (2006), 181–185.

[5] Faisal Farooq, Aravindh Manickavasagam, and Attila Dobi. 2025. How Pinterest Built a Real-Time Radar for Violative Content using AI. https://medium.com/pinterest-engineering/how-pinterest-built-a-real-time-radar-for-violative-content-using-ai-d5a108e02ac2. Pin Engineering Blog.

[6] Morris H Hansen and William N Hurwitz. 1943. On the theory of sampling from finite populations. *The Annals of Mathematical Statistics* 14, 4 (1943), 333–362.

[7] David Holt and TM Fred Smith. 1979. Post stratification. *Journal of the Royal Statistical Society Series A: Statistics in Society* 142, 1 (1979), 33–46.

[8] Daniel G. Horvitz and Donovan J. Thompson. 1952. A Generalization of Sampling Without Replacement from a Finite Universe. *J. Amer. Statist. Assoc.* 47, 260 (1952), 663–685. doi:10.1080/01621459.1952.10483446

[9] Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. 2023. MM-SafetyBench: A Benchmark for Safety Evaluation of Multimodal Large Language Models. *arXiv preprint* (2023). arXiv:2311.17600 https://arxiv.org/abs/2311.17600

[10] OpenAI. 2023. *Using GPT-4 for content moderation.* https://openai.com/blog/using-gpt-4-for-content-moderation

[11] John C. Platt. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*. 61–74.

[12] Ross L. Prentice. 1989. Surrogate Endpoints in Clinical Trials: Definition and Operational Criteria. *Statistics in Medicine* 8, 4 (1989), 431–440.

[13] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2020. Snorkel: rapid training data creation with weak supervision. *The VLDB Journal* 29 (2020), 709–730. doi:10.1007/s00778-019-00552-1

[14] Carl-Erik Särndal, Bengt Swensson, and Jan Wretman. 1992. *Model Assisted Survey Sampling*. Springer.

[15] Bianca Zadrozny and Charles Elkan. 2002. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. In *Proceedings of KDD*. 694–699.

[16] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv preprint* (2023). arXiv:2306.05685 https://arxiv.org/abs/2306.05685