ACTIS: Adaptive Cascade Tuning for Semantic Filters with Statistical Guarantees

Neil Marchant
University of Melbourne
nmarchant@unimelb.edu.au

Renata Borovica-Gajic
University of Melbourne
rborovica@unimelb.edu.au

Benjamin Rubinstein
University of Melbourne
brubinstein@unimelb.edu.au

Olga Ohrimenko
University of Melbourne
oohrimenko@unimelb.edu.au

Abstract

Emerging semantic database systems use Large Language Models (LLMs) to evaluate natural language predicates during query processing. However, invoking an expensive frontier model for every tuple is prohibitively costly and induces high latency. Model cascades mitigate these limitations by using a cheap proxy to score all tuples, escalating only uncertain predictions to an expensive LLM oracle. To be deployed reliably, cascades are tuned on a sample of oracle-labeled tuples to satisfy precision and recall targets with high probability. Our comprehensive audit reveals that several existing tuning methods frequently violate these guarantees: state-of-the-art techniques are certified unsound on up to 90.5% of benchmarks and can miss accuracy targets by more than 70 percentage points.

To improve efficiency and ensure soundness guarantees for semantic databases, we design ACTIS, an adaptive cascade tuning framework. ACTIS acquires oracle labels incrementally using anytime-valid tests with finite-sample safeguards, avoiding the over- or under-sampling pitfalls of a fixed pilot sample. A decoupled fixed-sequence search explores fine-grained threshold grids without conservative multiple-testing corrections, while a variance-minimizing importance sampling distribution reduces labels spent on uninformative negatives for selective queries. A value-of-information stopping rule halts tuning when additional oracle labeling costs exceed expected savings across the remaining workload. Across 42 benchmarks with cloud and local proxies, ACTIS matches or outperforms the only sound baseline on 39 of 42 settings, reducing oracle calls by up to 5.1×.

Artifact Availability. The source code and data have been made available at https://github.com/ngmarchant/actis.

1 Introduction

Predicate evaluation is a foundational primitive of database management systems. In traditional query engines, declarative queries apply attribute-based predicates over structured columns (e.g., `AGE` $\geq$ 18 $\wedge$ `STATE` = 'CA') to filter tuples early in query execution plans. With the rise of foundation models and the rapid influx of unstructured data—such as text documents, customer communications, and multi-modal records—a new class of *semantic data processing systems* has emerged. Systems such as LOTUS [33], Palimpzest [28], and DocETL [40], alongside enterprise engines like Google BigQuery [9] and Snowflake Cortex AISQL [27], extend relational engines with declarative *semantic operators* [6, 8, 24, 25, 34, 42, 53]. In particular, the *semantic filter* evaluates natural-language conditions over tuple attributes expressed via prompt templates—for example:

SemFilter("Is {`summary`} an accurate synopsis of {`full_text`}?"),

where `summary` and `full_text` are tuple attributes.

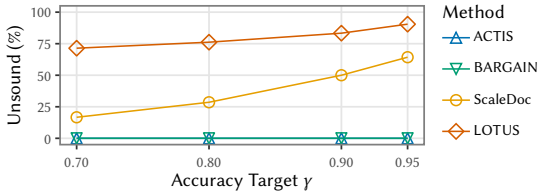


Figure 1: Percentage of 42 benchmarks failing statistical accuracy guarantees at $\delta = 0.05$ versus accuracy target γ. LOTUS [33] and ScaleDoc [52] fail on up to 90.5% and 64.3% of benchmarks, respectively, while ACTIS and BARGAIN [39] achieve zero unsound runs.

The most direct way to execute a semantic filter is to submit every tuple to an expensive frontier Large Language Model (LLM), which acts as an *oracle*. Although AI model providers offer batch APIs that discount high-throughput requests [32], evaluating an oracle LLM across millions of records remains prohibitively expensive and introduces high query latency. Filtering tuples early drastically reduces intermediate data volumes for downstream join, aggregation, and synthesis operators; consequently, optimizing semantic filters is critical to overall semantic query execution [33].

To reduce execution costs, recent work leverages *model cascades* [21, 33, 38, 44, 50, 52]. In a two-threshold cascade, a fast, lightweight proxy model—such as a small language model or an on-the-fly classifier trained over vector embeddings [5]—scores each tuple with a confidence value in $[0, 1]$. Tuples with proxy confidence exceeding an upper threshold $\hat{\tau}$ are predicted directly as `True` without calling the oracle; tuples falling below a lower threshold $\check{\tau}$ are predicted directly as `False`; only ambiguous intermediate tuples are escalated to the oracle. To be deployed reliably within a semantic data processing system, such cascades must guarantee output fidelity. Operationally, this means satisfying user-specified precision and recall targets $\gamma_P, \gamma_R \in (0, 1)$ with respect to the oracle with probability at least $1 - \delta$, where the probability is over the randomness in the cascade's tuning procedure.

Several pioneering cascade frameworks—notably LOTUS [33], ScaleDoc [52], and SUPG-IT [26]—claim to provide such statistical guarantees. However, our comprehensive empirical audit reveals a concerning reality: *the accuracy guarantees claimed by existing semantic filter cascades are frequently unsound in practice.* As illustrated in Fig. 1, under a standard error budget $\delta = 0.05$ with accuracy target $\gamma = 95\%$, LOTUS's guarantees are certified unsound—meaning its empirical failure rate statistically exceeds the error budget δ—on 90.5% of benchmarks, while ScaleDoc's guarantees are certified unsound on 64.3% of benchmarks. Moreover, the extent of these violations is severe: LOTUS falls short of the 95% target by more than 76 percentage points in the worst case, and ScaleDoc falls short by 86 percentage points.

In database query processing, an unsound accuracy guarantee is arguably worse than having no guarantee at all. Guarantees act as an SLA: users rely on them to ensure that approximate filtering will neither drop critical matching records (violating recall) nor pollute downstream queries with irrelevant data (violating precision). When an algorithm violates this contract, its cost savings are misleading, since any filter can reduce cost by silently dropping relevant tuples.

Why do existing tuning algorithms fail to uphold this contract? Our analysis reveals that these failures stem from compounding theoretical oversights and implementation gaps (App. C). Subtle flaws—ranging from uncorrected multiple hypothesis testing across grids of thresholds to fragile approximations in skewed data regimes—bias cascades toward overly aggressive thresholds whose true failure rates substantially exceed the nominal error budget δ.

While single-threshold cascade techniques like BARGAIN [50] establish finite-sample validity, the published literature lacks sound algorithms for two-threshold cascades. A recent, unpublished extension by BARGAIN's authors, *BARGAIN-PR* [39],[1] extends their framework to handle two-threshold cascades under uniform sampling, but suffers from sample inefficiency on imbalanced workloads—such as scanning thousands of records just to observe rare positives. Moreover, it lacks cost-aware stopping criteria that balance expenditure for tuning against expenditure for the remaining workload.

To resolve this conflict between efficiency and correctness, we formulate cascade tuning as an *online, cost-aware query optimization* problem. We present **ACTIS** (**A**daptive **C**ascade **T**uning via **I**mportance **S**ampling), a framework for tuning two-threshold model cascades that ensures statistical soundness while minimizing the total number of oracle model invocations. ACTIS introduces four key technical contributions:

(1) Decoupled Fixed-Sequence Testing. A naive search over a 2D grid of lower and upper thresholds $\check{G} \times \hat{G}$ requires severe Bonferroni corrections, requiring impractically large oracle samples for tuning. By proving that precision and recall margins exhibit monotonicity, we decouple the 2D search into two sequential 1D stages. Fixed-sequence testing [30] evaluates candidate thresholds in monotonic order with no multiplicative penalty over the upper threshold grid, enabling dense search without statistical power loss.

(2) Anytime-Valid Testing with Safeguards. Rather than relying on a predetermined pilot sample, ACTIS evaluates candidate thresholds sequentially using anytime-valid test processes [13, 35, 45]. To prevent failure modes of asymptotic Gaussian mixture tests on small or skewed samples, ACTIS integrates three finite-sample safeguards: a safe prior configuration, an empirical variance accumulation check, and a binomial check that prevents premature stopping on early success streaks.

(3) Variance-Minimizing Importance Sampling. Uniform sampling wastes oracle budget by scanning uninformative true negatives when target positives are rare. We derive the closed-form instrumental distribution that directly minimizes the total asymptotic variance of the empirical estimators for true positive rate, false positive rate, and positive prevalence, concentrating oracle calls on informative samples for tuning.

[1]For brevity we refer to the two-threshold BARGAIN-PR extension simply as "BARGAIN" throughout our evaluation.

(4) Adaptive Stopping. Balancing tuning cost against expected savings across the remaining workload requires knowing when additional oracle labels cease to be cost-effective. Drawing on the Value of Information framework [12], ACTIS constructs optimistic threshold bounds and halts tuning when the expected reduction in oracle invocations across the remaining workload falls below the marginal acquisition cost of the next batch.

We evaluate ACTIS across 42 benchmark scenarios spanning 23 datasets and queries derived from prior literature [21, 50, 52], evaluating both frontier cloud models (GPT-5.6) and local open-weight models (Llama-3.2-1B). Across all 42 benchmarks, ACTIS strictly adheres to the user-specified error budget δ, with zero certified unsound runs. Compared to BARGAIN—the only baseline that maintains statistical validity—ACTIS matches or reduces oracle call rates on 39 of 42 benchmarks, reducing oracle invocations by up to 5.1× on challenging workloads. Furthermore, our ablation studies show that variance-optimal importance sampling delivers its greatest benefits when proxy models are well-calibrated and query predicates are highly selective. Through these findings, we call on the database community to prioritize auditing of statistical guarantees, ensuring that cost optimizations in semantic query processing remain dependable in production.

2 Preliminaries

In this section, we formalize the problem of tuning model cascades for semantic filters and introduce the foundational principles of anytime-valid hypothesis testing.

2.1 Problem Formulation

Let $\mathcal{X}$ denote a domain of data records (e.g., text passages or document pairs). A workload consists of a finite table $\mathcal{D} \in \mathcal{X}^N$ containing N records. We denote by p the nominal uniform distribution over the workload table, such that $p(x) = 1/N$ for all $x \in \mathcal{D}$.

Oracle and Proxy Models. A semantic filter applies a predicate to each record in $\mathcal{D}$. Ideally, this predicate is evaluated by an accurate, highly capable model, which we refer to as the *oracle model* O. For each record $x \in \mathcal{D}$, evaluating the oracle yields a binary label $Y \sim \mathrm{O}(x) \in \{0, 1\}$. This formulation accommodates both deterministic oracles (such as greedy LLM decoding) and stochastic oracles (such as reasoning models, where $\mathrm{E}[Y|x] \in [0, 1]$). Evaluating the oracle incurs high latency and monetary cost. To alleviate this burden, a fast, lightweight *proxy model* $\mathrm{A}\colon \mathcal{X} \to [0, 1]$ assigns a confidence score $\mathrm{A}(x)$ to each record $x \in \mathcal{D}$.

Two-Threshold Model Cascade. To balance efficiency and fidelity, we approximate the oracle using a two-threshold model cascade $\mathrm{C}\colon \mathcal{X} \to \{0, 1\}$, parameterized by a lower threshold $\check{\tau} \in [0, 1]$ and an upper threshold $\hat{\tau} \in [\check{\tau}, 1] \cup \{\infty\}$:

$$\mathrm{C}(x; \check{\tau}, \hat{\tau}) = \begin{cases} 1, & \mathrm{A}(x) \geq \hat{\tau}, \\ 0, & \mathrm{A}(x) < \check{\tau}, \\ \mathrm{O}(x), & \check{\tau} \leq \mathrm{A}(x) < \hat{\tau}. \end{cases}$$

The cascade predicts `True` (1) for records whose proxy scores are $\geq \hat{\tau}$, predicts `False` (0) for records whose proxy scores are $< \check{\tau}$, and escalates intermediate records to the expensive oracle. The primary cost metric of the cascade is the *oracle call rate* $r(\check{\tau}, \hat{\tau})$, defined as

the fraction of the workload deferred to the oracle:

$$r(\check{\tau}, \hat{\tau}) = \frac{1}{N} \sum_{x \in \mathcal{D}} \mathbf{1}_{\{\check{\tau} \le \mathrm{A}(x) < \hat{\tau}\}}. \tag{1}$$

Workload Quality Metrics. Evaluating the quality of the cascade relative to the oracle requires tracking the positive prevalence, true positive rate, and false positive rate across the workload $\mathcal{D}$:

$$\pi = \mathop{\mathrm{E}}_{X \sim p, Y \sim \mathrm{O}(X)} [Y], \tag{2}$$

$$\mathrm{TPR}(\check{\tau}) = \mathop{\mathrm{E}}_{X \sim p, Y \sim \mathrm{O}(X)} \left[Y \mathbf{1}_{\{\mathrm{A}(X) \ge \check{\tau}\}} \right], \tag{3}$$

$$\mathrm{FPR}(\hat{\tau}) = \mathop{\mathrm{E}}_{X \sim p, Y \sim \mathrm{O}(X)} \left[(1 - Y) \mathbf{1}_{\{\mathrm{A}(X) \ge \hat{\tau}\}} \right]. \tag{4}$$

The cascade recall and precision over $\mathcal{D}$ are then given by:

$$\mathrm{Recall}(\check{\tau}) = \frac{\mathrm{TPR}(\check{\tau})}{\pi}, \tag{5}$$

$$\mathrm{Precision}(\check{\tau}, \hat{\tau}) = \frac{\mathrm{TPR}(\check{\tau})}{\mathrm{TPR}(\check{\tau}) + \mathrm{FPR}(\hat{\tau})}. \tag{6}$$

When the oracle is deterministic, expectations under $X \sim p, Y \sim \mathrm{O}(X)$ collapse to empirical averages over $\mathcal{D}$.

Cascade Tuning with Accuracy Guarantees. Given user-specified precision and recall targets $\gamma_P, \gamma_R \in (0, 1)$ and an error budget $\delta \in (0, 1)$, our objective is to find thresholds $(\check{\tau}^\star, \hat{\tau}^\star)$ that minimize the oracle call rate while guaranteeing that both targets are satisfied simultaneously with probability at least $1 - \delta$:

$$\begin{aligned} \min_{\check{\tau} \in [0,1], \hat{\tau} \in [\check{\tau}, 1] \cup \{\infty\}} \quad & r(\check{\tau}, \hat{\tau}) \\ \text{s.t.} \quad & \Pr\left[\mathrm{Precision}(\check{\tau}, \hat{\tau}) \ge \gamma_P \wedge \mathrm{Recall}(\check{\tau}) \ge \gamma_R \right] \ge 1 - \delta, \end{aligned} \tag{7}$$

where the probability is taken strictly with respect to the randomness in the cascade tuning algorithm, as described below.

Execution Model. Following prior cascade frameworks [21, 33, 50, 52], we assume proxy scores $\mathcal{S} = \{\mathrm{A}(x) : x \in \mathcal{D}\}$ are computed offline for all N records prior to tuning, and that oracle labels acquired during tuning are cached and reused when the tuned cascade is deployed over the remaining workload. Unlike static methods [33, 52] that draw a single predetermined pilot sample, adaptive methods ([39] and ACTIS) query the oracle sequentially in batches until statistical criteria are met.

2.2 Anytime-Valid Testing

To certify that candidate thresholds satisfy the precision and recall constraints in Eq. (7), the tuner must test statistical hypotheses over oracle-labeled samples. In an adaptive setting, batches of labeled data arrive sequentially. Evaluating standard fixed-sample confidence bounds (such as Hoeffding or empirical Bernstein bounds) repeatedly after each batch introduces the well-known *peeking problem*: repeatedly monitoring a fixed-sample test inflates the Type I error rate, causing the cumulative probability of a false rejection to approach 1 as sampling continues [16].

To evaluate hypotheses across sequential batches without error inflation, we leverage *anytime-valid testing* [35]. Let $Z_1, Z_2, \ldots$ be a sequence of observations with conditional expectations $\mu = \mathrm{E}[Z_t \mid Z_1, \ldots, Z_{t-1}]$. Suppose we wish to test the one-sided null hypothesis that μ is strictly below a target margin m, $H_0 : \mu < m$ versus $H_1 : \mu \ge m$. An anytime-valid test tracks a non-negative process $(M_t)_{t=0}^{\infty}$ starting at $M_0 = 1$ and rejects H_0 at the stopping time $T = \inf\{t \ge 1 : M_t \ge 1/\alpha\}$, guaranteeing Type I error control uniformly over all stopping times:

$$\Pr_{H_0}\left(\exists t \ge 1 : M_t \ge 1/\alpha \right) \le \alpha. \tag{8}$$

Below, we review two anytime-valid testing frameworks: non-asymptotic betting martingales and asymptotic mixture processes.

2.2.1 *Non-Asymptotic Testing via Betting Martingales.*

For a true anytime-valid test that guarantees validity across all $t \ge 1$, we can construct $(M_t)_{t=0}^{\infty}$ as a *test supermartingale* under H_0, satisfying $\mathrm{E}[M_t \mid Z_1, \ldots, Z_{t-1}] \le M_{t-1}$ for all $t \ge 1$ under H_0. By Ville's inequality [43], the maximum of a non-negative supermartingale satisfies $\Pr_{H_0}(\exists t \ge 1 : M_t \ge 1/\alpha) \le \mathrm{E}[M_0]/(1/\alpha) = \alpha$, ensuring Eq. (8) holds.

For observations bounded in the unit interval $Z_i \in [0, 1]$, Waudby-Smith and Ramdas [46] construct M_t via a betting game. Starting from capital $M_0 = 1$, wealth accumulates multiplicatively $M_t = \prod_{i=1}^{t} (1 + \lambda_i (Z_i - m))$, where m is the target margin, $\lambda_i \ge 0$ is a betting fraction determined by past observations $Z_1, \ldots, Z_{i-1}$:

$$\lambda_i = \min\left(\sqrt{\frac{2 \log(1/\alpha)}{\tilde{\sigma}_{i-1}^2 \, i \log(1 + i)}}, \; \frac{c}{m} \right). \tag{9}$$

Here $c \in (0, 1)$ ensures that the wealth multiplier remains strictly positive, and $\tilde{\sigma}_t^2$ is an online regularized sample variance estimator:

$$\tilde{\sigma}_t^2 = \frac{\frac{1}{4} + \sum_{i=1}^{t} (Z_i - \tilde{\mu}_i)^2}{t + 1} \quad \text{with} \quad \tilde{\mu}_t = \frac{\frac{1}{2} + \sum_{i=1}^{t} Z_i}{t + 1}.$$

This betting supermartingale guarantees non-asymptotic Type I error control $\le \alpha$ for any sample size $t \ge 1$.[2]

While betting martingales naturally accommodate observations bounded in $[0, 1]$, applying them to importance-weighted random variables requires rescaling the observations onto the unit interval using the maximum possible importance weight. Since importance weights must be reasonably large to improve sampling efficiency, the rescaled increments are compressed toward zero, forcing the bet size λ_i in Eq. (9) to become very small. Consequently, the betting martingale suffers from sluggish wealth accumulation and poor sample efficiency under importance sampling.

2.2.2 *Asymptotic Testing via Gaussian Mixtures.*

To achieve high sample efficiency under importance-weighted observations without suffering from conservative bounds, we adopt the asymptotic anytime-valid testing framework of Waudby-Smith et al. [45].

Adapting the conjugate Gaussian mixture processes of Robbins [36] and Howard et al. [13] to one-sided hypothesis testing, we track the cumulative excess drift $S_t = \sum_{i=1}^{t} (Z_i - m)$ and the empirical quadratic variation $V_t = \sum_{i=1}^{t} (Z_i - \bar{Z}_t)^2$, where $\bar{Z}_t = \frac{1}{t} \sum_{i=1}^{t} Z_i$. The mixture test process $(M_t)_{t=0}^{\infty}$ is defined with $M_0 = 1$ and

$$M_t = \begin{cases} \frac{1}{\sqrt{1 + V_t / v}} \exp\left(\frac{S_t^2}{2(v + V_t)} \right), & \text{if } S_t > 0, \\ 0, & \text{if } S_t \le 0, \end{cases} \tag{10}$$

where $v > 0$ is a prior variance parameter.[3] Standardizing increments using the empirical variance V_t rather than a known Gaussian

[2] Waudby-Smith and Ramdas [46] combine opposing bets with a hedging weight $\theta \in (0, 1)$ to construct two-sided confidence sequences. Since our focus is on one-sided testing of $H_0 : \mu < m$, we allocate full capital to the positive direction ($\theta = 1$).

[3] In sequential testing, v is typically tuned to the expected cumulative process variance over a planned sample horizon n_0 (i.e., $v \approx n_0 \sigma^2$) to optimize statistical power.

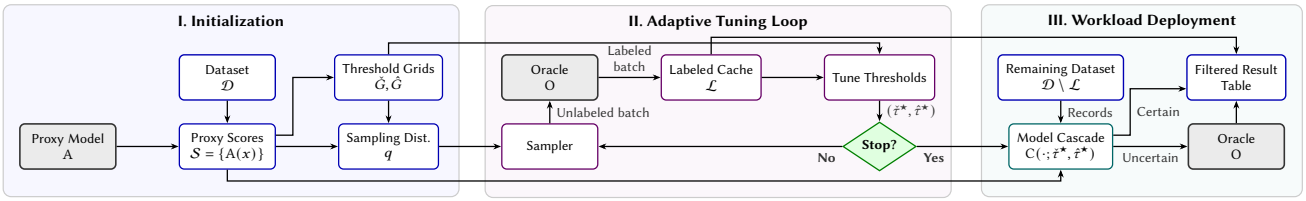


Figure 2: Overview of the three-stage ACTIS framework: (I) proxy scoring, threshold grid construction, and importance sampling initialization; (II) sequential batch acquisition, anytime-valid threshold tuning, and cost-aware stopping; and (III) cascade deployment over the remaining unlabeled workload.

variance or worst-case bound means that M_t is not in general a test supermartingale. Instead, its validity is established asymptotically via time-uniform central limit theorems [45]:

$$\limsup_{m_0 \to \infty} \Pr_{H_0}\left(\exists t \geq m_0 \colon M_t \geq \frac{1}{\alpha}\right) \leq \alpha, \tag{11}$$

where m_0 denotes the starting sample size for testing.

While Gaussian mixture processes provide the variance-adaptive power needed for importance sampling, they rely on asymptotic approximations. ACTIS therefore pairs them with finite-sample runtime safeguards (Sec. 3.3), to balance statistical soundness and cost efficiency, while also supporting betting martingales when strict finite-sample guarantees are prioritized.

3 ACTIS Framework

This section presents ACTIS—**A**daptive **C**ascade **T**uning via **I**mportance **S**ampling—our framework for tuning two-threshold model cascades under statistical accuracy guarantees. As illustrated in Fig. 2, ACTIS executes a semantic filter over a workload $\mathcal{D}$ in three stages:

Stage I: Offline Initialization. Before querying the oracle, the proxy model A evaluates all N records in $\mathcal{D}$ to produce proxy scores $\mathcal{S} = \{A(x) : x \in \mathcal{D}\}$. From $\mathcal{S}$, ACTIS constructs discrete lower and upper candidate threshold grids $\check{G}$ and $\hat{G}$ (Sec. 3.1), along with an instrumental sampling distribution q designed to minimize the asymptotic variance of the true positive rate, false positive rate and prevalence estimators across the grids (Sec. 3.6).

Stage II: Adaptive Tuning Loop (Alg. 1). Rather than committing to a fixed pilot sample size upfront, ACTIS acquires oracle labels incrementally in batches of size B sampled from $\mathcal{D}$ according to q (or uniformly) and appends them to a labeled cache $\mathcal{L}$. After each batch, ACTIS updates the cascade thresholds $(\check{\tau}^\star, \hat{\tau}^\star) \in \check{G} \times (\hat{G} \cup \{\infty\})$ to be as aggressive as possible while certifying the joint precision and recall targets at family-wise error rate δ. Individual candidate thresholds are verified using anytime-valid tests (Sec. 3.2)—instantiated via non-asymptotic betting martingales or asymptotic Gaussian mixtures with finite-sample safeguards (Sec. 3.3)—while directional monotonicity decouples the 2D grid search into two 1D fixed-sequence scans without conservative Bonferroni penalties (Sec. 3.4). After updating $(\check{\tau}^\star, \hat{\tau}^\star)$, a Value-of-Information stopping rule (Sec. 3.5) checks whether the expected reduction in oracle calls over the remaining dataset $\mathcal{D} \setminus \mathcal{L}$ justifies the cost of labeling another batch; if not, the tuning loop halts.

Stage III: Workload Deployment. Upon termination of the tuning loop, the tuned cascade $C(\cdot; \check{\tau}^\star, \hat{\tau}^\star)$ is deployed over the remaining unlabeled tuples $\mathcal{D} \setminus \mathcal{L}$. Tuples with proxy scores in $[\check{\tau}^\star, \hat{\tau}^\star)$ are escalated to the oracle O, while confident tuples ($A(x) < \check{\tau}^\star$ or $A(x) \geq \hat{\tau}^\star$) are filtered directly using the proxy predictions. Combined with the already-labeled cache $\mathcal{L}$, this produces the final filtered result table.

The remainder of this section details each component of ACTIS. All proofs are deferred to App. A.

3.1 Candidate Threshold Grids

Tuning a cascade over thresholds $\check{\tau} \in [0, 1]$ and $\hat{\tau} \in [\check{\tau}, 1] \cup \{\infty\}$ poses a severe multiple-testing challenge: a continuous search requires either loose uniform concentration bounds or massive samples to prevent overfitting. ACTIS restricts tuning to discrete grids of candidate thresholds: $\check{G} = (\check{\tau}_1 < \ldots < \check{\tau}_{\check{m}})$ for the lower candidates and $\hat{G} = (\hat{\tau}_1 < \cdots < \hat{\tau}_{\hat{m}})$ for the upper candidates. These grids are constructed from the empirical quantiles of the proxy scores. Specifically, for a target grid size M with $K = \min(M, N)$, we evaluate quantile points $p_i = 1 - (1 - \frac{i-1}{K-1})^\kappa$ for $i \in \{1, \ldots, K\}$, and select scores at ranks $k_i = 1 + \lfloor (N-1)p_i \rfloor$ as the threshold grid. The tail exponent $\kappa \geq 1$ controls candidate concentration.

Lower Candidate Grid ($\check{G}$). The lower threshold $\check{\tau}$ screens negatives to satisfy the recall target γ_R. Errors (false negatives) are typically distributed across the broad low- to mid-range of proxy scores. We therefore construct $\check{G}$ using a uniform quantile schedule ($\kappa = 1$) placing candidate thresholds at equal increments of empirical probability mass to ensure balanced resolution across the bulk of the score distribution.

Upper Candidate Grid ($\hat{G}$). The upper threshold $\hat{\tau}$ governs positive predictions to satisfy the precision target γ_P. Under realistic targets ($\gamma_P \to 1$), cascade precision is highly sensitive to small threshold shifts in the upper tail near 1. A uniform quantile schedule is ill-suited here, as it allocates most candidate thresholds to uninformative low scores and leaves negligible resolution in the upper tail. We therefore construct $\hat{G}$ using a power-law schedule with tail exponent $\kappa = \max(1, \log N / \log K)$, which concentrates candidate thresholds in the upper tail and adapts candidate resolution to workload size N without inflating the grid size $\hat{m}$.

Anchor Thresholds. To guarantee safe fallbacks for any accuracy target, we pair the lower anchor $\check{\tau}_1 = \min_{x \in \mathcal{D}} A(x) \in \check{G}$ with an upper sentinel $\hat{\tau}_{\hat{m}+1} \coloneqq \infty \notin \hat{G}$ (strictly exceeding $\max_{x \in \mathcal{D}} A(x)$). Since the cascade filters on $A(x) < \check{\tau}$ and $A(x) \geq \hat{\tau}$, choosing $\check{\tau}_1$ prevents false negatives ($\text{Recall}(\check{\tau}_1) = 1$) while choosing $\hat{\tau}_{\hat{m}+1}$ prevents false positives ($\text{Precision}(\check{\tau}, \hat{\tau}_{\hat{m}+1}) = 1$).

3.2 Verifying Individual Cascade Constraints

Before addressing the search over candidate threshold grids, we first consider testing whether individual precision/recall constraints are satisfied at fixed candidate thresholds $(\check{\tau}, \hat{\tau}) \in \check{G} \times \hat{G}$. Suppose that after drawing n samples, the tuner has collected an oracle-labeled set $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^{n}$, where each record x_i was sampled from instrumental distribution q over $\mathcal{D}$ with associated importance weight $w(x_i) = p(x_i)/q(x_i)$. Under uniform sampling, q is the nominal distribution p, yielding unit weights $w(x_i) \equiv 1$.

The precision/recall constraints in Eq. (7) are expressed as non-linear ratios of expectations. However, the anytime-valid testing frameworks introduced in Sec. 2.2 evaluate whether a population mean (i.e., a single expectation) exceeds a target margin. To apply these tests, we reformulate both the recall and precision constraints into equivalent linear expectations, derive their per-sample increments, and define testing subroutines for each constraint.

3.2.1 Verifying the Recall Constraint. Recall depends only on the lower threshold $\check{\tau}$. Multiplying Eq. (5) by the prevalence π, the constraint $\text{Recall}(\check{\tau}) \geq \gamma_R$ is equivalent to a non-negative *recall margin*:

$$\Delta_R(\check{\tau}) \coloneqq \mathop{\mathrm{E}}_{X\sim q, Y\sim \mathrm{O}(X)} \left[f_R(X, Y; \check{\tau}) \right] \geq 0, \tag{12}$$

where the per-sample increment is defined as:

$$f_R(X, Y; \check{\tau}) \coloneqq w(X) Y \left(\mathbf{1}_{\{\mathrm{A}(X)\geq \check{\tau}\}} - \gamma_R \right). \tag{13}$$

To certify that candidate threshold $\check{\tau}$ meets the recall target, we test the one-sided null hypothesis that the recall margin is negative, $H_{0,R}(\check{\tau})\colon \Delta_R(\check{\tau}) < 0$ versus $H_{1,R}(\check{\tau})\colon \Delta_R(\check{\tau}) \geq 0$. Rejecting $H_{0,R}(\check{\tau})$ certifies $\text{Recall}(\check{\tau}) \geq \gamma_R$. For the asymptotic Gaussian mixture test this hypothesis is evaluated directly against the null margin $m_R = 0$.

3.2.2 Verifying the Precision Constraint. Precision depends on both $\check{\tau}$ (which governs true positives) and $\hat{\tau}$ (which governs false positives). Rearranging Eq. (6), the constraint $\text{Precision}(\check{\tau}, \hat{\tau}) \geq \gamma_P$ is equivalent to a non-negative *precision margin*:

$$\Delta_P(\check{\tau}, \hat{\tau}) \coloneqq \mathop{\mathrm{E}}_{X\sim q, Y\sim \mathrm{O}(X)} \left[f_P(X, Y; \check{\tau}, \hat{\tau}) \right] \geq 0, \tag{14}$$

where the per-sample increment is defined as:

$$f_P(X, Y; \check{\tau}, \hat{\tau}) \coloneqq w(X) \left((1-\gamma_P) Y \mathbf{1}_{\{\mathrm{A}(X)\geq \check{\tau}\}} - \gamma_P (1-Y) \mathbf{1}_{\{\mathrm{A}(X)\geq \hat{\tau}\}} \right). \tag{15}$$

To certify that threshold pair $(\check{\tau}, \hat{\tau})$ meets the precision target, we test the one-sided null hypothesis, $H_{0,P}(\check{\tau}, \hat{\tau})\colon \Delta_P(\check{\tau}, \hat{\tau}) < 0$ versus $H_{1,P}(\check{\tau}, \hat{\tau})\colon \Delta_P(\check{\tau}, \hat{\tau}) \geq 0$. Rejecting $H_{0,P}(\check{\tau}, \hat{\tau})$ certifies that $\text{Precision}(\check{\tau}, \hat{\tau}) \geq \gamma_P$. As with recall, the asymptotic test evaluates this hypothesis directly against the margin $m_P = 0$.

3.2.3 Testing Subroutine Interfaces. We encapsulate constraint verification for recall and precision in two boolean subroutines:

- RecallMet($\gamma_R, \mathcal{L}, \check{\tau}, \alpha$): Tests the null $H_{0,R}(\check{\tau})$ at significance level α using labeled sample $\mathcal{L}$, returning `True` if the null is rejected and `False` otherwise.
- PrecisionMet($\gamma_P, \mathcal{L}, \check{\tau}, \hat{\tau}, \alpha$): Tests the null $H_{0,P}(\check{\tau}, \hat{\tau})$ at significance level α using labeled sample $\mathcal{L}$, returning `True` if the null is rejected and `False` otherwise.

For the non-asymptotic betting test (Sec. 2.2.1), raw increments are mapped to $[0, 1]$ via an affine transformation (App. A.1). The subroutine evaluates wealth accumulation using default betting hyperparameters (App. A.2), returning `True` once the accumulated wealth satisfies $M_t \geq 1/\alpha$. For the asymptotic Gaussian mixture test (Sec. 2.2.2), increments are evaluated directly against margin $m = 0$. Its prior variance parameter is set to an estimated horizon variance v^{est} (App. A.2). Candidate thresholds must additionally satisfy the runtime safeguards detailed next.

3.3 Safeguards for Asymptotic Testing

The Gaussian mixture test introduced in Sec. 2.2.2 provides anytime-valid error control in the asymptotic limit ($m_0 \to \infty$) via a time-uniform central limit theorem (CLT) [45]. However, deploying asymptotic tests on finite, highly skewed workloads—particularly under strict precision/recall targets and importance sampling—can expose failure modes:

- *Premature Single-Sample Rejections*: In early samples ($t \approx 1$), empirical variance has not yet accumulated ($V_t = 0$). If the prior variance parameter v is too small, an isolated positive increment can trigger an immediate boundary crossing ($M_1 \geq 1/\alpha$) on the first sample.
- *Early Success Streaks*: Under strict targets, constraint violations (false negatives or false positives) are rare under the boundary null. In small samples containing no observed violations, early runs of consecutive successes generate positive cumulative drift with zero empirical variance ($V_t = 0$).

To address these vulnerabilities without sacrificing the sample efficiency of asymptotic testing, ACTIS introduces three complementary safeguards.

3.3.1 Safeguard 1: Single-Step Safety Floor. To ensure that a single sample cannot cause the wealth M_t to cross the rejection boundary $(1/\alpha)$, we clip the prior variance parameter v from below:

$$v \leftarrow \max\left(v, v^{\text{floor}}\right). \tag{16}$$

Let $b_R(\check{\tau}) \coloneqq (1-\gamma_R) w^{\geq}_{\max}(\check{\tau})$ and $b_P(\check{\tau}) \coloneqq (1-\gamma_P) w^{\geq}_{\max}(\check{\tau})$ denote the maximum possible recall and precision increments, where $w^{\geq}_{\max}(\cdot)$ is defined in Eq. (19), within App. A.1. For significance levels $\delta_R = \delta/2$ and $\alpha_P = \delta/(2\check{m})$, the safety floor is:

$$v_R^{\text{floor}}(\check{\tau}) \coloneqq \frac{2 b_R^2(\check{\tau})}{\log(1/\delta_R)}, \qquad v_P^{\text{floor}}(\check{\tau}) \coloneqq \frac{2 b_P^2(\check{\tau})}{\log(1/\alpha_P)}. \tag{17}$$

As proven in App. A.3, setting $v \geq v^{\text{floor}}$ guarantees that for any first observation ($V_1 = 0$), accumulated wealth satisfies $M_1 < 1/\alpha$.

3.3.2 Safeguard 2: Variance Accumulation Check. The variance-stabilization condition underpinning the time-uniform CLT [45] requires the empirical quadratic variation V_t to grow sufficiently large relative to the prior variance v. When V_t is negligible, the Gaussian mixture is dominated by the prior rather than empirical observations. Before rejecting $H_{0,R}$ or $H_{0,P}$, ACTIS requires the corresponding empirical quadratic variation to reach at least $\rho = 5\%$ of the estimated horizon variance: $V_t / v^{\text{est}} \geq \rho$.

3.3.3 Safeguard 3: Binomial Safety Check. To guard against premature false rejections on early success streaks, ACTIS requires the observed failure count to also pass an exact one-sided binomial test before rejecting $H_{0,R}$ or $H_{0,P}$. For each test, active records in $\mathcal{L}$ are partitioned into successes and failures:

- *Recall at* $\check{\tau}$: Active records are ground-truth positives ($Y = 1$). A success is a true positive ($\mathrm{A}(X) \geq \check{\tau}$), while a failure is a false negative ($\mathrm{A}(X) < \check{\tau}$).
- *Precision at* $(\check{\tau}, \hat{\tau})$: Active records are cascade positive predictions. A success is a true positive ($\mathrm{A}(X) \geq \check{\tau}, Y = 1$), while a failure is a false positive ($\mathrm{A}(X) \geq \hat{\tau}, Y = 0$).

Under uniform random sampling, the conditional failure probability of an active record drawn under the boundary null is $p_0 = 1 - \gamma$ (where $\gamma \in \{\gamma_R, \gamma_P\}$). Under importance sampling, the instrumental distribution q allocates non-uniform weight across proxy score regions. As derived in App. A.3, the probability of drawing a failure under q given that an active record is sampled shifts to:

$$p_0 \coloneqq \frac{(1-\gamma)\eta}{\gamma + (1-\gamma)\eta},$$

where $\eta \coloneqq \bar{q}_F / \bar{q}_S$ denotes the ratio of average instrumental densities across the failure and success regions.

Among the observed batches, suppose there are n_a active records and n_f failures. Safeguard 3 passes if and only if $F(n_\mathrm{f}; n_\mathrm{a}, p_0) \leq \alpha$, where $\alpha = \delta_R$ for recall and $\alpha = \alpha_P$ for precision, and $F(x; n, p)$ denotes the cumulative distribution function of a binomial random variable with n trials and per-trial failure probability p.

3.3.4 Integration into Test Subroutines. When evaluated via asymptotic testing, RecallMet and PrecisionMet return `True` if and only if the mixture test process crosses the nominal rejection threshold ($M_t \geq 1/\alpha$) and both Safeguards 2 and 3 are satisfied.

3.4 Two-Stage Tuning via Fixed-Sequence Testing

Given candidate threshold grids $\check{G}$ and $\hat{G}$, the goal of cascade tuning is to find a threshold pair $(\check{\tau}^\star, \hat{\tau}^\star) \in \check{G} \times (\hat{G} \cup \{\infty\})$ that minimizes the oracle call rate while guaranteeing the recall and precision targets are met with family-wise error rate (FWER) at most δ.

A naive approach would evaluate each threshold pair on the grid using the constraint verification subroutines RecallMet and PrecisionMet introduced in Sec. 3.2.3. However, testing all $\check{m} \cdot \hat{m}$ pairs without correcting for multiple hypothesis testing risks inflating the Type I error. Conversely, applying a Bonferroni correction over the grid would test each hypothesis at significance level $\delta/(\check{m} \cdot \hat{m})$. In practical configurations with $\check{m} = 50$ and $\hat{m} = 500$, dividing δ by 25,000 would be very conservative, requiring an impractically large sample $\mathcal{L}$ to reject any null hypothesis.

3.4.1 Fixed-Sequence Testing. To avoid the power loss of a 2D Bonferroni correction, we leverage the *fixed-sequence testing method* [30, 48]. Under this method, a family of null hypotheses $H_1, \ldots, H_M$ is evaluated in a pre-specified sequential order, terminating at the first non-rejection. Each individual hypothesis is tested at the full nominal level α, yet the family-wise error rate of the test remains bounded by α with no multiplicative penalty.

3.4.2 Monotonicity and Power. While fixed-sequence testing guarantees FWER control for any pre-specified sequence, the procedure halts at the first non-rejection. Consequently, an arbitrary hypothesis ordering risks premature termination on an early true null, missing valid candidate thresholds located further down the sequence. Fortunately, the precision and recall constraints satisfy a monotonicity property that aligns with sequential testing.

Proposition 3.1 (Monotonicity of Cascade Constraints). *For any precision/recall targets $\gamma_P, \gamma_R \in (0, 1)$:*

(i) *The recall margin $\Delta_R(\check{\tau})$ is monotonically non-increasing in $\check{\tau}$.*
(ii) *The precision margin $\Delta_P(\check{\tau}, \hat{\tau})$ is monotonically non-decreasing in $\hat{\tau}$.*

By Proposition 3.1, cascade constraint violations are nested. Scanning candidate thresholds monotonically (ascending for $\check{\tau}$ and descending for $\hat{\tau}$) ensures that all true null hypotheses form a contiguous suffix at the end of the sequence, so halting at the first non-rejection incurs no loss of statistical power.

3.4.3 Two-Stage Tuning Procedure. Although cascade tuning involves searching over the 2D grid $\check{G} \times \hat{G}$, the two constraints exhibit an asymmetry: the recall margin $\Delta_R(\check{\tau})$ depends exclusively on the lower threshold, whereas the precision margin $\Delta_P(\check{\tau}, \hat{\tau})$ depends on both thresholds. This asymmetry allows us to decouple the 2D search into two 1D stages: we first optimize the lower threshold $\check{\tau}$ against the recall constraint alone, and subsequently optimize the upper threshold $\hat{\tau}$ conditioned on the selected lower threshold.

To ensure that both constraints hold jointly with FWER at most δ, we split the error budget equally between the two stages via a union bound, allocating $\delta_R = \delta/2$ for recall and $\delta_P = \delta/2$ for precision. After each batch is added to $\mathcal{L}$, threshold updates proceed in two stages (lines 5–14 of Alg. 1):

(1) *Forward Scan for $\check{\tau}$.* Starting from safe anchor threshold $\check{\tau}_1$, we evaluate candidate lower thresholds in ascending order at level δ_R until the first non-rejection of the recall null $H_{0,R}$, selecting the largest passing threshold as $\check{\tau}^\star$.
(2) *Backward Scan for $\hat{\tau}$.* Conditioning on the data-dependent $\check{\tau}^\star$, we evaluate candidate upper thresholds in descending order starting from anchor threshold $\hat{\tau}_{\hat{m}+1} = \infty$ down to $\check{\tau}^\star$ at Bonferroni-adjusted level $\alpha_P = \delta_P/\check{m}$, stopping at the first non-rejection to select the smallest passing threshold $\hat{\tau}^\star$.

Proposition 3.2 (Family-Wise Error Rate Control). *Given anytime-valid test subroutines RecallMet and PrecisionMet, Alg. 1 satisfies the joint precision/recall guarantee in Eq. (7) with family-wise error rate (FWER) at most δ.*

By decoupling tuning in this way, fixed-sequence testing avoids any multiplicative penalty over the upper grid size $\hat{m}$. This allows ACTIS to use dense grids for the upper threshold (e.g., $\hat{m} = 500$) to maximize precision without sacrificing statistical power.

3.5 Adaptive Stopping via Marginal Value of Information

In an adaptive deployment, oracle labels are acquired sequentially in batches of size B sampled from the workload $\mathcal{D}$. Let $n = |\mathcal{L}|$ denote the number of labeled records collected so far across all batches. At each step, the tuner must decide whether to spend oracle calls labeling another batch of size B to tighten the thresholds, or stop tuning and deploy the cascade over the remaining workload.

Anytime Validity and Arbitrary Stopping. A fundamental advantage of the anytime-valid test processes introduced in Sec. 2.2 is

Algorithm 1 ACTIS Adaptive Cascade Tuning

Input: Dataset $\mathcal{D}$, proxy scores $\mathcal{S}$, grids $\check{G}, \hat{G}$, sampling distribution q, batch size B, accuracy targets γ_P, γ_R, error budget δ.
Output: Tuned thresholds $(\check{\tau}^\star, \hat{\tau}^\star)$, labeled cache $\mathcal{L}$

$\mathcal{L} \leftarrow \emptyset, \quad \delta_R \leftarrow \delta/2, \quad \delta_P \leftarrow \delta/2, \quad \alpha_P \leftarrow \delta_P/\check{m}$
repeat
  Sample batch $\mathcal{B}$ from $\mathcal{D}$ of size B via q (or uniformly)
  Query oracle on $\mathcal{B}$ and add to labeled cache $\mathcal{L}$
  $\check{k} \leftarrow 1$ ▷ Stage 1: Forward scan for $\check{\tau}^\star$
  while $\check{k} < \check{m}$ **and** RecallMet$(\gamma_R, \mathcal{L}, \check{\tau}_{\check{k}+1}, \delta_R)$ **do**
    $\check{k} \leftarrow \check{k} + 1$
  end while
  $\check{\tau}^\star \leftarrow \check{\tau}_{\check{k}}$
  $\hat{k} \leftarrow \hat{m} + 1$ ▷ Stage 2: Backward scan for $\hat{\tau}^\star$
  while $\hat{k} > 1$ **and** $\hat{\tau}_{\hat{k}-1} \geq \check{\tau}^\star$ **and** PrecisionMet$(\gamma_P, \mathcal{L}, \check{\tau}^\star, \hat{\tau}_{\hat{k}-1}, \alpha_P)$ **do**
    $\hat{k} \leftarrow \hat{k} - 1$
  end while
  $\hat{\tau}^\star \leftarrow \hat{\tau}_{\hat{k}}$
until ShouldStop$(N, \mathcal{L}, \check{\tau}^\star, \hat{\tau}^\star, B)$ (see Sec. 3.5)
return $(\check{\tau}^\star, \hat{\tau}^\star), \mathcal{L}$

that their statistical error guarantees hold under optional stopping. By Ville's inequality and Doob's optional stopping theorem [13], the Type I error bound δ holds at *any arbitrary, data-dependent stopping time*. Consequently, the tuner is free to stop whenever continuing is no longer cost-effective, without inflating the FWER or invalidating the cascade guarantees.

Optimistic Thresholds. To quantify how much the cascade could potentially improve with additional data, we invert the anytime tests to construct *optimistic thresholds*. Recall that Alg. 1 selects conservative thresholds by rejecting the null hypothesis that the quality margin is negative ($H_0 \colon \Delta < 0$), certifying that the true margin is non-negative (a lower confidence bound on quality).

Conversely, we define a candidate threshold as *plausible* if the data cannot reject the reverse null hypothesis that the quality margin is non-negative ($H_0 \colon \Delta \geq 0$). By testing this reverse hypothesis, we identify the most aggressive candidate thresholds that cannot yet be ruled out as invalid within the anytime confidence bounds. Specifically, the *optimistic lower threshold* $\check{\tau}^{\text{opt}}$ is the largest candidate in $\check{G}$ where the recall target remains plausible, while the *optimistic upper threshold* $\hat{\tau}^{\text{opt}}$ is the smallest candidate in $\hat{G} \cup \{\infty\}$ (with $\hat{\tau}^{\text{opt}} \geq \check{\tau}^{\text{opt}}$) where the precision target remains plausible. These optimistic thresholds are purely an internal heuristic for the stopping rule; they are not deployed or used to accept candidates, and therefore do not consume any FWER error budget.

Marginal Value of Information. Let $r^\star = r(\check{\tau}^\star, \hat{\tau}^\star)$ and $r^{\text{opt}} = r(\check{\tau}^{\text{opt}}, \hat{\tau}^{\text{opt}})$ denote the oracle call rates under the current conservative thresholds and optimistic thresholds, respectively. Since the plausible region contains the true optimal thresholds with high probability, the difference $\Delta r \coloneqq \max\left(0,\, r^\star - r^{\text{opt}}\right)$ represents the *uncertainty gap* [18]—the maximum reduction in oracle call rate that could be gained by collecting infinite additional samples.

Following the principle of the *Value of Information* [12], we compare the expected marginal savings from acquiring the next batch against its marginal acquisition cost:

- *Expected Marginal Savings*: Confidence sequence widths contract at rate $O(1/\sqrt{n})$. Adding a batch of size B to an existing sample of size n contracts the confidence bound width by a factor of $1 - \sqrt{n/(n+B)} \approx \frac{B}{2n}$. Across the $N - n$ remaining records in the workload $\mathcal{D}$, the expected reduction in oracle invocations is $\text{Savings}(B) \coloneqq \left(\frac{B}{2\max(1,n)}\right) \cdot \Delta r \cdot (N - n)$.
- *Marginal Acquisition Cost*: If tuning were terminated immediately and the cascade deployed, only a fraction $r^\star$ of the incoming batch would be escalated to the oracle. On the other hand, continuing tuning requires invoking the oracle on an additional B records, incurring a net marginal cost $\text{Cost}(B) \coloneqq B \cdot (1 - r^\star)$.

Stopping Condition. The ShouldStop subroutine in Alg. 1 halts tuning once expected marginal savings no longer exceed marginal acquisition cost, i.e., $\text{Savings}(B) \leq \text{Cost}(B)$, or once a maximum tuning budget of $n_{\max}$ samples (default $N/2$) is reached. To prevent premature stopping under severe class imbalance—where initial batches may contain zero positives—early stopping is deferred until at least $n_{\min}$ positive instances have been observed (App. A.6).

3.6 Variance-Minimizing Instrumental Distribution

In each step of the adaptive tuning loop, ACTIS acquires a batch of B labeled records. Under uniform sampling, records are drawn with equal probability $p(x) = 1/N$. When target positives are rare (e.g., prevalence $\pi \leq 0.05$), uniform sampling spends the vast majority of its oracle budget acquiring uninformative true negatives, starving the tuner of positive examples and leading to wide confidence sequences. To accelerate sequential testing, ACTIS uses importance sampling: it draws records from an instrumental distribution $q \in \Delta(\mathcal{D})$ biased toward informative records, and weights each observed increment by $w(x) = p(x)/q(x)$.

The statistical efficiency of importance sampling hinges on q. Rather than adopting ad-hoc heuristics, we derive q following the principle of asymptotic variance minimization across the candidate threshold grids $\check{G}$ and $\hat{G}$. Let $\check{m}(s) \coloneqq \sum_{\tau \in \check{G}} \mathbf{1}_{\{s \geq \tau\}}$ and $\hat{m}(s) \coloneqq \sum_{\tau \in \hat{G}} \mathbf{1}_{\{s \geq \tau\}}$ denote the number of candidate lower and upper thresholds that fall at or below proxy score s.

Proposition 3.3. *Under a calibrated proxy model ($\mathbb{E}[Y \mid \mathrm{A}(x) = s] = s$), the instrumental distribution minimizing the variance of the true positive rate, false positive rate, and prevalence estimators is $q^*(x) \propto \sqrt{c(\mathrm{A}(x))}$, where the score sensitivity function $c(s)$ is:*

$$c(s) = \lambda_{\text{TPR}}\, s \left(\frac{\check{m}(s)}{\check{m}}\right) + \lambda_{\text{FPR}}\,(1 - s)\left(\frac{\hat{m}(s)}{\hat{m}}\right) + \lambda_\pi\, s. \tag{18}$$

The component weights $\lambda_{\text{TPR}} = 1 + (\frac{1-\gamma_P}{\gamma_P})^2$, $\lambda_{\text{FPR}} = 1$, and $\lambda_\pi = (\frac{\gamma_R}{1-\gamma_R})^2$ standardize the recall and precision constraints by their slack scales $1 - \gamma_R$ and γ_P; they concentrate sampling priority on discovering true positives for recall ($\lambda_{\text{TPR}} \geq 1$) and detecting false positives for precision ($\lambda_{\text{FPR}} = 1$), while scaling up prevalence discovery for practical recall targets ($\gamma_R \to 1$). The joint variance objective, weight derivation, and proof of Proposition 3.3 are deferred to App. A.7.

In practice, proxy models are imperfect: LLM proxies like GPT-5.6-luna often output saturated scores of 0 and 1 (see Table 4), and proxies may be miscalibrated across the score domain. To defend against this, ACTIS employs two mechanisms:

Tail Tolerance. Since $\hat{m}(0)/\hat{m} \approx 0$, plugging $s = 0$ into Eq. (18) yields $c(0) \approx 0$, meaning the instrumental distribution $q^*(x)$ assigns negligible weight to 0-scored items. This creates a blind spot that ignores potential false negatives. To guard against saturated scores without excessive distortion, ACTIS evaluates $q^*(x)$ by replacing raw proxy score s with a clipped proxy score $\text{clip}(s,\ \nu\hat{\pi},\ 1-\nu(1-\hat{\pi}))$, where $\hat{\pi} \coloneqq \frac{1}{N}\sum_{x\in\mathcal{D}} \mathrm{A}(x)$ estimates the positive prevalence and $\nu \in (0, 1)$ is the tail tolerance parameter (we set $\nu = 0.1$ by default).

Defensive Mixture. To prevent importance weights from exploding under a poorly-calibrated proxy, ACTIS mixes q^* with the uniform distribution $q(x) \leftarrow \epsilon p(x) + (1-\epsilon)q^*(x)$ where $\epsilon \in (0, 1]$ is a defensive mixing parameter (we set $\epsilon = 0.4$ by default). This guarantees importance weights are bounded by $1/\epsilon$.

Since both defensive mechanisms depend only on proxy scores and candidate grids, $q(x)$ can be computed prior to cascade tuning without querying the oracle.

4 Empirical Evaluation

We empirically evaluate ACTIS across 42 benchmarks spanning 23 datasets/queries. Our evaluation addresses four research questions:

- RQ1. *Statistical Soundness Audit:* Do ACTIS and existing cascade tuning methods uphold their statistical accuracy guarantees across diverse workloads and error budgets? (Sec. 4.2)
- RQ2. *Cascade Cost-Effectiveness:* How effective is ACTIS at reducing expensive oracle invocations compared to statistically valid baselines? (Sec. 4.3)
- RQ3. *Ablations and Sensitivity Analysis:* How do the algorithmic components and hyperparameters of ACTIS influence its cost efficiency and tuning latency? (Sec. 4.4)
- RQ4. *Tuning Runtime and Overhead:* What is the computational latency of ACTIS's tuning phase, and is it practical for database query optimization? (Sec. 4.5)

4.1 Experimental Setup

4.1.1 Workloads and Benchmarks. We evaluate ACTIS across 23 *workloads* (each defined as a dataset–query pair) spanning computer vision, information extraction, legal contracts, review sentiment, and technical documents [21, 50, 52]. As summarized in Table 1, workload sizes range from N = 1,000 (Court) to N = 973,085 (Jackson), with positive prevalence π (pos. rate) spanning three orders of magnitude (from 0.1% to 86.5%).

We define a *benchmark* as a workload paired with a specific proxy model. For the 19 text-based workloads derived from Zeighami et al. [50] and Zhang et al. [52], we evaluate both cloud-hosted and local proxy models. The remaining 4 workloads from Kang et al. [21] were released with precomputed proxy scores and oracle labels without retaining the raw data, contributing one fixed benchmark each. In total, this yields 42 benchmarks. Evaluating these benchmarks across target pairs (γ, δ) produces our full set of experimental scenarios. A comprehensive per-benchmark breakdown is provided in App. B (Table 4).

Table 1: Summary of experimental benchmarks. Detailed per-benchmark statistics are reported in Table 4 in App. B.

Benchmark Suite	Benchmarks	Size (N)	Pos. rate	PR-AUC
Pre-LLM Vision & NLP [21] ImageNet, Jackson, Onto, TACRED	4	11–973k	0.1–29%	0.79–0.98
General Text Corpora [50] Court, Review, Screenplay, Wiki	8	1–10k	5.9–87%	0.12–0.97
Technical Documents [52] PubMed, BigPatent, GovReport	30	10k	0.6–79%	0.004–1.00
Total/Overall	42	1–973k	0.1–87%	0.004–1.00

4.1.2 Oracle and Proxy Models. For text benchmarks from Zeighami et al. [50] and Zhang et al. [52], we instantiate the oracle using GPT-5.6-terra deployed in Microsoft Foundry. We disable reasoning and set the sampling temperature to zero to ensure deterministic oracle ground truth. A minimal content filter is applied by Microsoft at the API level; records triggering the filter ($\leq$ 0.4% across all datasets) are excluded. For the four vision and NLP benchmarks from Kang et al. [21], we use the precomputed oracle labels and proxy scores released by the authors.

For the proxy model, we evaluate two deployment regimes:

- *Cloud Proxy (GPT-5.6-luna).* A lightweight model from the same family as the oracle (also deployed in Microsoft Foundry) with 10× lower API inference costs. Confidence scores are extracted from output logits for the tokens `True` and `False`.
- *Local Proxy (Llama-3.2-1B).* A small open-weight language model deployed locally using Ollama on a single NVIDIA A100 GPU (80GB VRAM) supporting batched inference.

4.1.3 Baseline Algorithms. We compare ACTIS against three state-of-the-art cascade tuning algorithms:

BARGAIN. An extension of BARGAIN [50] developed to tune two-threshold cascades under joint precision and recall targets. While not present in the BARGAIN paper, its code was released in the authors' open-source repository [39]. As it represents the only prior baseline that is statistically sound, we evaluate it as a primary baseline. To enable execution on large workloads, we forked the authors' repository to incorporate NumPy vectorization and configured its batch size to 100.

LOTUS [33]. The cascade tuning algorithm implemented in the LOTUS semantic query engine. We configure it with the default pilot sample size of 10%. To allow LOTUS to scale to our larger workloads, we forked the authors' repository and re-implemented its `learn_cascade_thresholds` routine using NumPy prefix sums and binary search, reducing its computational complexity from nested iterative loops to $O(N \log N)$.

ScaleDoc [52]. The tuning algorithm from ScaleDoc, evaluated using the authors' published implementation. Unlike ACTIS, BARGAIN, and LOTUS, which tune cascades to satisfy joint precision and recall targets, ScaleDoc guarantees that the cascade exceeds a user-specified F1 score target γ_{F1} with probability $1 - \delta$.

4.1.4 ACTIS Configuration. We evaluate three configurations:

- *ACTIS* (default): our primary variant, pairing importance sampling with asymptotic Gaussian mixture tests and runtime safeguards ($\epsilon = 0.4$, $\rho = 0.05$, $\nu = 0.1$);

- *ACTIS-U*: a non-asymptotic variant combining uniform random sampling with betting supermartingales, guaranteeing finite-sample validity; and
- *ACTIS (w/o safeguards)*: an ablation of the default variant omitting safeguards to isolate their impact on statistical soundness.

All configurations use grid sizes $\check{m} = 50$ and $\hat{m} = 500$, batch size $B = 100$, and $n_{\max} = N/2$ unless otherwise specified.

4.1.5 Simulation Protocol. To evaluate both average performance and tail failure behavior, we execute 1,000 independent trials for each experimental scenario. In each trial, the tuning algorithm queries a sample of records from $\mathcal{D}$ (one-shot for static baselines, sequentially in batches for adaptive methods) to tune thresholds $(\check{\tau}^\star, \hat{\tau}^\star)$. The cascade is then deployed over the remaining workload (not sampled during tuning), reusing oracle labels for records evaluated during tuning. We measure the resulting precision, recall, and oracle call rate across the entire workload $\mathcal{D}$ against ground-truth oracle labels. All experiments evaluate four accuracy targets $\gamma \in \{0.70, 0.80, 0.90, 0.95\}$ under three nominal error budgets $\delta \in \{0.01, 0.05, 0.10\}$. For LOTUS, BARGAIN, and ACTIS, γ specifies a symmetric precision and recall target ($\gamma_P = \gamma_R = \gamma$). Since ScaleDoc natively guarantees only an F1 target, we configure it according to the evaluation goal: (i) In RQ1 (auditing), we audit ScaleDoc under its native F1 guarantee by setting $\gamma_{F1} = \gamma$. (ii) In RQ2–RQ4 (efficiency), algorithms must achieve the same precision and recall targets. We therefore configure ScaleDoc with an F1 target of $\gamma_{F1} = \frac{2\gamma}{1+\gamma}$, which implies Precision $\geq \gamma$ and Recall $\geq \gamma$.

4.2 RQ1: Auditing Statistical Soundness

To assess whether a cascade tuning algorithm satisfies the theoretical guarantee defined in Eq. (7), we evaluate its empirical failure rate $\tilde{\delta}$—the fraction of 1,000 simulation trials in which the deployed cascade violates its native accuracy target (precision and recall target for ACTIS, BARGAIN, and LOTUS; F1 target for ScaleDoc). As tail probabilities are sensitive to finite-sample variance, we compute a 95% Clopper-Pearson binomial confidence interval $[\tilde{\delta}_{\text{LB}}, \tilde{\delta}_{\text{UB}}]$ for δ. An experimental scenario is certified *sound* if the upper bound satisfies the budget ($\tilde{\delta}_{\text{UB}} < \delta$), certified *unsound* if the lower bound exceeds the budget ($\tilde{\delta}_{\text{LB}} > \delta$), and *inconclusive* if the nominal budget falls within the interval ($\delta \in [\tilde{\delta}_{\text{LB}}, \tilde{\delta}_{\text{UB}}]$). Inconclusive outcomes indicate that 1,000 trials cannot statistically distinguish $\tilde{\delta}$ from δ, keeping both soundness and unsoundness certifications conservative.

Table 2 summarizes the audit across all 42 benchmarks aggregated over accuracy targets $\gamma \in \{0.70, 0.80, 0.90, 0.95\}$. A disaggregated breakdown across each individual (γ, δ) pair is reported in App. B (Table 5).

Unsoundness of Prior Tuning Algorithms. The audit reveals that LOTUS and ScaleDoc systematically fail to meet their guarantees:

- At nominal error budget $\delta = 0.05$, LOTUS is certified unsound on 80.4% of all scenarios. At the strict $\gamma = 0.95$ target, LOTUS fails on 90.5% of benchmarks (Table 5).
- ScaleDoc is certified unsound on 39.9% of scenarios at $\delta = 0.05$, and on 64.3% of scenarios under $\gamma = 0.95$, with empirical failure rates reaching 100% on multiple benchmarks.

Table 2: Statistical soundness across 42 benchmarks by error budget δ, reporting the percentage certified sound, inconclusive, or unsound (highlighted in red) via 95% CIs over 1,000 trials. $\tilde{\delta}_{\text{unsound}}$ and $\tilde{\delta}_{\max}$ denote the mean empirical failure rate over unsound cases and the maximum across all cases.

		Percentage Certified				
δ	Method	Sound	Inconcl.	Unsound	$\tilde{\delta}_{\text{unsound}}$	$\tilde{\delta}_{\max}$
0.01	ACTIS	99.4	0.6	0.0	—	0.004
	ACTIS (w/o safeguards)	94.0	2.4	3.6	0.113	0.217
	ACTIS-U	100.0	0.0	0.0	—	0.001
	BARGAIN	91.1	8.9	0.0	—	0.006
	ScaleDoc	44.0	3.0	53.0	0.329	1.000
	LOTUS	15.5	3.0	81.5	0.844	1.000
0.05	ACTIS	99.4	0.6	0.0	—	0.043
	ACTIS (w/o safeguards)	95.8	0.6	3.6	0.160	0.232
	ACTIS-U	100.0	0.0	0.0	—	0.002
	BARGAIN	100.0	0.0	0.0	—	0.030
	ScaleDoc	51.8	8.3	39.9	0.423	1.000
	LOTUS	18.5	1.2	80.4	0.872	1.000
0.10	ACTIS	100.0	0.0	0.0	—	0.065
	ACTIS (w/o safeguards)	96.4	0.6	3.0	0.206	0.283
	ACTIS-U	100.0	0.0	0.0	—	0.004
	BARGAIN	100.0	0.0	0.0	—	0.049
	ScaleDoc	62.5	3.6	33.9	0.481	1.000
	LOTUS	19.0	1.8	79.2	0.893	1.000

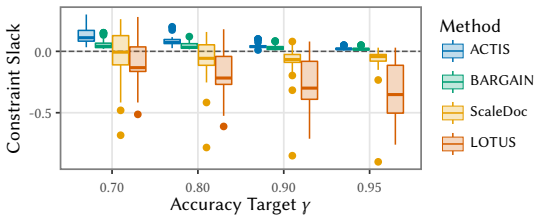


Figure 3: Distribution of accuracy constraint slack across 42 benchmarks at $\delta = 0.05$. Values falling below the dashed horizontal line represent violations of the accuracy target.

- Tighter error budgets exacerbate these failures: at $\delta = 0.01$, LOTUS and ScaleDoc are certified unsound on 81.5% and 53.0% of scenarios, respectively.

Severity of Target Violations. Fig. 3 plots the distribution of constraint slack across all 42 benchmarks with $\delta = 0.05$. For methods evaluated on joint precision and recall targets (ACTIS, BARGAIN, and LOTUS), slack is defined as $\min\{\text{Precision} - \gamma, \text{Recall} - \gamma\}$; for ScaleDoc, which is audited against its native F1 target, slack is defined as $F_1 - \gamma$. Points below zero represent violations where deployed cascades fail to achieve the specified accuracy target. At an accuracy target of $\gamma = 0.95$, LOTUS's median delivered accuracy (the minimum of precision and recall) is 0.60, falling to 0.19 in the worst case. ScaleDoc's median delivered F1 score just falls short of the target at 0.945; however, it drops catastrophically to 0.09 in the worst case. These failures demonstrate that prior methods achieve low oracle invocation costs by choosing overly aggressive thresholds (see App. C).

Validity of ACTIS and BARGAIN. By contrast, ACTIS—under both importance sampling and uniform sampling, with asymptotic or non-asymptotic testing—maintains statistical validity. Across all 42 benchmarks and all tested configurations of γ and δ, ACTIS is

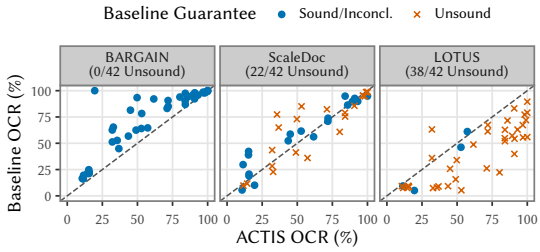


Figure 4: Pairwise comparison of oracle call rate (OCR) between baselines and ACTIS ($\gamma = 0.95, \delta = 0.05$). Crosses indicate workloads for which the baseline is certified unsound. ACTIS achieves a lower OCR on workloads for which the marker is above the dashed diagonal.

never certified unsound. Its empirical maximum failure rate $\tilde{\delta}_{\max}$ never exceeds the nominal budget δ. BARGAIN also achieves no certified unsound runs, establishing it as the only valid baseline against which cost efficiency can be fairly evaluated.

Crucially, the ablation of our safeguards (ACTIS w/o safeguards) underscores why they are necessary when deploying asymptotic mixture tests in finite samples. Without safeguards, the empirical failure rate violates the nominal budget on up to 3.6% of scenarios, with failure rates reaching up to 23.2% at $\delta = 0.05$. These failures validate the theoretical vulnerabilities identified in Sec. 3.3: on highly selective queries where positives are rare, the asymptotic test is susceptible to premature termination on early success streaks and zero-variance sample draws. By incorporating safeguards, ACTIS maintains nominal error control even in the finite sample regime.

4.3 RQ2: End-to-End Cascade Efficiency

Having established that LOTUS and ScaleDoc violate their guarantees, we evaluate the cost effectiveness of ACTIS against BARGAIN, the only statistically valid baseline. Recall that for this efficiency comparison, all methods are configured to guarantee the common precision and recall target $\gamma_P = \gamma_R = \gamma$ (setting ScaleDoc's F1 target to $\frac{2\gamma}{1+\gamma}$, as detailed in Sec. 4.1). Fig. 4 presents a pairwise head-to-head comparison of oracle call rates (OCR) across all 42 benchmarks for $\gamma = 0.95$ and $\delta = 0.05$. Table 3 provides a detailed snapshot across representative scenarios, with full results documented in App. B (Table 6). In these tables, *Ideal* denotes the minimum OCR that achieves the precision/recall target under perfect ground-truth knowledge. Valid methods (ACTIS and BARGAIN) incur a gap to this ideal because statistical certification inherently requires querying a minimum sample of oracle labels to guarantee the error budget δ, even for accurate proxies (e.g., ImageNet and PubMed Q0).

Comparison Against BARGAIN. At $\gamma = 0.95$ and $\delta = 0.05$, ACTIS matches or reduces oracle call rate (OCR) versus BARGAIN on 39 of 42 benchmarks (92.9%), and achieves the lowest OCR among all sound methods on 8 of the 11 benchmarks in Table 3. Even without importance sampling, our non-asymptotic uniform variant ACTIS-U matches or outperforms BARGAIN on 36 of 42 benchmarks (Table 6)—e.g., reducing OCR from 81.5% to 46.4% on Jackson and from 65.3% to 51.7% on Review Q0—demonstrating the gains from decoupled fixed-sequence testing and Value-of-Information stopping. Furthermore, comparing ACTIS against ACTIS-U and BARGAIN on selective workloads highlights how importance sampling overcomes uniform sampling bottlenecks:

- On ImageNet ($\gamma = 0.95$, 0.1% prevalence), both BARGAIN and ACTIS-U exhaust the workload (100% OCR) starving for rare positives under uniform sampling, whereas ACTIS satisfies the guarantee at 19.6% OCR (5.1× reduction).
- On Onto and TACRED (≈2.5% prevalence), ACTIS cuts OCR nearly in half compared to both ACTIS-U (32.1% vs. 61.3% and 54.4%) and BARGAIN (62.6% and 51.3%).

Impact of Proxy Model Quality. Comparing cloud-based GPT-5.6-luna against local Llama-3.2-1B reveals clear architectural trade-offs. As shown in Table 4, GPT-5.6-luna produces higher PR-AUC and lower expected calibration error (ECE). Consequently, cascades using GPT-5.6-luna achieve a lower OCR (e.g., 80.2% on Court Q0 vs. 96.8% for Llama-3.2-1B under $\gamma = 0.95$). ACTIS maintains validity under both proxies, letting operators trade off local compute against API spend.

4.4 RQ3: Ablation and Sensitivity Analysis

We conduct ablation and sensitivity studies to analyze the core algorithmic components of ACTIS and characterize the impact of hyperparameters on cascade efficiency and tuning latency.

Importance Sampling vs. Uniform Sampling. We evaluate the benefit of variance-optimal importance sampling by comparing default ACTIS against an identical configuration using uniform sampling (holding the safeguarded asymptotic testing engine fixed). At $\gamma = 0.95$ and $\delta = 0.05$, importance sampling achieves a lower oracle call rate on 39 of 42 benchmarks. As shown in Fig. 5, efficiency gains are governed by two factors: proxy discriminative power and query selectivity. When proxy PR-AUC is high (left panel), q^* accurately concentrates sampling mass on informative decision boundaries, tightening confidence sequences rapidly. Crucially, when target positives are rare ($\pi < 0.05$, right panel), uniform sampling wastes budget scanning uninformative true negatives, whereas q^* oversamples high-scoring tuples to verify recall, reducing total oracle calls by up to 80% relative to uniform sampling.

Batch Size Sensitivity. In sequential tuning, the batch size B controls the granularity of oracle label acquisition and stopping rule evaluation. Fig. 6 depicts the effect of scaling B from 100 to 10,000 on the million-row Jackson workload ($N = 973{,}085, \delta = 0.05$). As B increases, tuning latency drops substantially (left panel)—falling from 3.45 s to 0.55 s at $B = 5{,}000$ (6.3× speedup) and 0.45 s at $B = 10{,}000$ for $\gamma = 0.95$—because larger increments reduce the frequency of anytime-valid hypothesis tests. This speedup incurs negligible penalty in filtering efficiency (right panel): across all four accuracy targets, the oracle call rate increases by less than 0.38 percentage points (e.g., from 45.01% to 45.38% at $\gamma = 0.95$).

This stability holds because B remains small relative to the workload ($B/N \leq 1\%$). By contrast, on smaller tables like PubMed Q0 ($N = 10{,}000$), coarse batch sizes such as $B = 5{,}000$ force the tuner to label half the dataset in a single step, artificially inflating the oracle call rate. Consequently, configuring B as a small fraction of the table (e.g., $B \leq 0.01N$ or default $B = 100$) guarantees fine-grained stopping control while maintaining sub-second tuning latency across diverse workload scales.

Table 3: Oracle call rate (%) and tuner latency (ms) on representative benchmarks for precision/recall target $\gamma = 0.95$ and error budget $\delta = 0.05$. 'Ideal' denotes an omniscient tuner that selects the minimum-cost valid thresholds using oracle labels for the workload. Red* indicates certified unsoundness; boldface highlights the best result among statistically sound methods.

Benchmark			Oracle Call Rate (%)						Tuner Latency (ms)				
Workload	Proxy	N	Ideal	ACTIS	ACTIS-U	BARGAIN	ScaleDoc	LOTUS	ACTIS	ACTIS-U	BARGAIN	ScaleDoc	LOTUS
Onto	UFT	11k	6.2	**32.1**	61.3	62.6	28.4*	63.2*	17.2	42.6	**11.2**	4.4*	1.6*
ImageNet	ResNet-50	50k	0.0	19.6	100.0	100.0	10.1	**5.1**	48.1	191.6	123.2	7.1	**6.2**
Jackson	ResNet-50	973k	35.4	**45.0**	46.4	81.5	58.7	25.7*	3854.3	10110.6	3580.0	**109.6**	171.3*
TACRED	SpanBERT	23k	6.5	**32.1**	54.4	51.3	43.5*	35.9*	43.7	69.1	**22.2**	5.5*	3.0*
Court	GPT-5.6-luna	1k	44.5	**80.2**	88.1	94.0	60.8*	22.3*	3.6	5.6	**2.2**	3.3*	0.6*
Review	GPT-5.6-luna	10k	10.2	**32.8**	51.7	65.3	22.8*	7.4*	**10.7**	18.4	12.3	4.3*	1.6*
Screenplay	Llama-3.2-1B	3k	53.9	**84.1**	85.5	97.5	94.8	52.0*	44.2	87.2	4.1	**3.9**	0.9*
Wiki	GPT-5.6-luna	10k	37.0	71.1	**68.6**	83.1	82.3*	26.1*	69.5	129.1	**13.7**	3.4*	1.6*
PubMed Q0	GPT-5.6-luna	10k	0.0	**15.3**	15.8	24.8	42.3	9.4*	**3.9**	4.5	5.2	6.2	1.8*
BigPatent $Q0_{ext}$	Llama-3.2-1B	10k	60.2	85.9	**84.8**	94.0	86.4	71.6*	69.3	113.9	18.9	**5.4**	1.9*
GovReport Q10	GPT-5.6-luna	10k	12.5	**35.6**	43.1	52.7	77.4*	7.9*	13.3	17.9	**10.2**	4.6*	1.9*

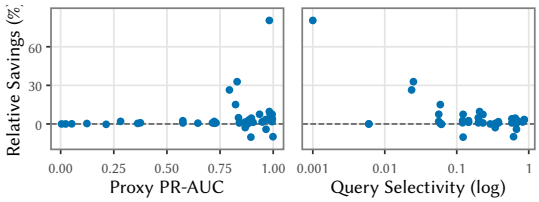


Figure 5: Relative reduction in oracle calls achieved by ACTIS with importance sampling versus uniform sampling across 42 benchmarks (with $\gamma = 0.95, \delta = 0.05$) as a function of proxy PR-AUC (left) and query positive rate π (right).

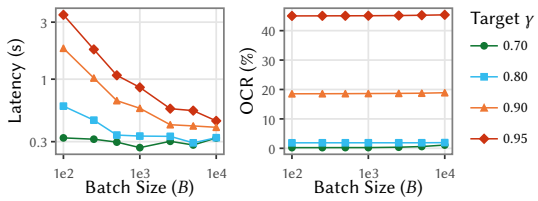


Figure 6: Sensitivity of ACTIS to batch size B on Jackson ($N = 973{,}085, \delta = 0.05$) across accuracy targets γ. At high accuracy, tuner latency (left) drops by up to 7× as B increases, while oracle call rate (right) remains virtually constant.

4.5 RQ4: Tuning Latency Overhead

In a semantic database engine, cascade tuning acts as an online query optimization step that trades a small upfront computational investment during query planning for large downstream reductions in oracle execution. Table 3 reports the in-memory CPU time consumed by the tuning algorithms ($\gamma = 0.95, \delta = 0.05$), isolating algorithmic execution from external oracle I/O and network latency.

Interactive Tuning Overhead. Across all 42 benchmarks at $\gamma = 0.95$, ACTIS requires a median CPU tuning time of only 27.3 ms, completing in under 90 ms on 41 of 42 benchmarks. This overhead is well within the budget of interactive query optimizers. While LOTUS and ScaleDoc exhibit faster runtimes (1.8 ms and 5.2 ms), their speed is an artifact of taking statistical shortcuts without multiple testing corrections or sequential validation (Sec. 4.2). Furthermore, the baseline runtimes reported in Table 3 reflect our optimized implementations; in their published upstream forms, both baselines exhibit substantial latency overheads. In a 10-trial benchmark on Review Q0 (N = 10k), upstream LOTUS requires 560 ms (300× slower than our vectorized version at 1.9 ms), while upstream BARGAIN requires 70 ms (5× slower than our vectorized version at 13.4 ms). On the million-row Jackson benchmark, both upstream implementations fail to scale, timing out (> 1 hour) on 80–100% of trials. By contrast, ACTIS natively scales to million-row workloads in 3.45 s (and 0.55 s with B = 5,000), comparable to our vectorized BARGAIN baseline (3.58 s). Among statistically sound methods, ACTIS is the only framework that scales to million-row tables out-of-the-box within interactive latency bounds.

Amortizing Tuning Costs. Directly measuring end-to-end latency across cloud LLM APIs is confounded by transient network jitter, throttling, and concurrency quotas. Since frontier LLMs like GPT-5.6-terra incur ≈0.9 s end-to-end latency per call (≈0.8 s time-to-first-token [2]), oracle invocations dominate query runtime and cost, easily amortizing ACTIS's sub-second tuning overhead. For example, on Review Q0 (N = 10,000), ACTIS spends 10.7 ms tuning, yet lowers the oracle call rate from 65.3% (BARGAIN) to 32.8%, avoiding 3,250 calls to GPT-5.6-terra. Even assuming high concurrency (e.g., 50 simultaneous streams), eliminating these 3,250 calls saves ≈59 s of query execution latency and reduces API expenditure (proxy plus oracle) by 36% ($1.60 vs. $2.50), amortizing the 10.7 ms tuning phase by four orders of magnitude. On the million-row Jackson benchmark, ACTIS requires 3.45 s with B = 100 (or 0.55 s with B = 5,000), but eliminates over 355,000 oracle calls compared to BARGAIN, potentially saving hours of downstream GPU compute.

5 Related Work

Model Cascades. Model cascades [44] were introduced to database query processing to accelerate expensive machine learning predicates over unstructured data [14, 20, 29]. To address the lack of accuracy guarantees in early systems, Kang et al. [21] proposed SUPG, which uses importance sampling and asymptotic confidence bounds to tune a single-threshold cascade for either a precision *or* recall target. Zeighami et al. [50] showed that SUPG's asymptotic guarantees miss targets at small sample sizes and proposed BARGAIN, which provides finite-sample guarantees for single-threshold cascades using non-asymptotic betting martingales.

Recent systems use *two-threshold* cascades to filter negatives and accept positives simultaneously. LOTUS [33] extends SUPG to joint precision and recall targets, while SUPG-IT [26] adapts LOTUS's tuning algorithm to streaming batches. ScaleDoc [52] tunes a two-threshold cascade for a target F1 score using proportional stratified sampling, histogram smoothing, and Pareto-frontier threshold search. Despite claiming statistical guarantees, LOTUS, SUPG-IT, and ScaleDoc rely on analyses containing errors or heuristic approximations (see App. C); our audit finds high failure rates for LOTUS and ScaleDoc (Sec. 4.2). A recent, unpublished code addition by BARGAIN's authors, BARGAIN-PR [39], provides finite-sample validity for two-threshold cascades under a shared target ($\gamma_P = \gamma_R$), but relies on uniform and contiguous-window sampling, tests a conservative upper-tail precision condition, and lacks cost-aware stopping. By contrast, ACTIS supports distinct (γ_P, γ_R) targets and combines anytime-valid sequential testing, decoupled fixed-sequence search, variance-minimizing importance sampling, and Value-of-Information stopping.

Proxy Models for Cascades. Cascade efficiency depends on the discriminative power of its proxy scorer. Prior work has explored diverse proxy architectures, including lightweight classifiers over bag-of-words or domain features [29, 49], task-agnostic embedding indexes [23], small LLM token log-probabilities [33], and classifiers trained on precomputed vector embeddings [5]. ScaleDoc [52] trains a query-aware multilayer perceptron over precomputed LLM document embeddings using two-phase contrastive learning to encourage smooth, monotonic, and bipolar score distributions. These techniques are orthogonal to ACTIS, which treats the proxy scorer as a black box and adapts to its calibration quality.

Beyond Two-Threshold Cascades. Recent work explores richer optimization spaces and non-cascade paradigms for semantic filtering. Shankar et al. [41] introduce *task cascades*, using an LLM agent to assemble cascades that vary the model, operation, and document portion processed at each stage. Stretto [38] jointly optimizes model size, KV-cache compression, and cascade thresholds via gradient descent over a continuous plan relaxation, using marginal Bayesian credible intervals to bound precision and recall. Beyond cascades, Hou et al. [11] propose Clustering-Sampling-Voting (CSV), which clusters tuple embeddings, queries the oracle on cluster samples, and propagates labels through voting with recursive re-clustering. However, CSV bounds only per-cluster accuracy, which is uninformative under class imbalance, while Stretto and CSV omit statistical corrections for adaptive search and partitioning (see App. C).

Semantic Operators and Data Processing Systems. While ACTIS optimizes semantic filters in isolation, parallel work provides specialized approximations with guarantees for other semantic operators, including stratified sampling for approximate aggregations [19, 22] and featurized decomposition for semantic joins [51]. These operators also serve as primitives in declarative semantic data systems [6, 9, 25, 27, 28, 33, 40], which use holistic query optimizers for cost-based model and operator selection [15, 28, 37] or agentic query rewriting [40, 47]. ACTIS complements these system-level optimizers by providing a drop-in physical filter operator with statistical accuracy guarantees.

Routing and Cascades in Machine Learning. Outside databases, query routers [7, 31] and sequential multi-model cascades [3] route inputs across LLMs of varying cost and capability, typically optimizing empirical cost-quality trade-offs without statistical guarantees. In machine learning, Learn-then-Test [1] calibrates post-processing thresholds under finite-sample risk control using fixed-sample p-values and FWER-controlling fixed-sequence testing over a pre-labeled i.i.d. calibration set. Unlike offline calibration, semantic query processing acquires labels online from an expensive oracle, which ACTIS minimizes via anytime-valid testing, importance sampling, and cost-aware stopping.

6 Conclusion

In this paper, we audited the statistical soundness of existing model cascade tuning methods for semantic filters, revealing that state-of-the-art techniques frequently violate their claimed precision and recall guarantees. To deliver dependable guarantees efficiently, we introduced ACTIS, which combines anytime-valid sequential testing, runtime safeguards, decoupled fixed-sequence search, variance-minimizing importance sampling, and Value-of-Information early stopping. Across 42 benchmarks spanning cloud and local proxies, ACTIS never violates the user-specified failure budget, matches or outperforms the only statistically valid baseline on 39 of 42 settings, and reduces oracle invocations by up to 5.1×. More broadly, our findings underscore a critical lesson for semantic data processing systems: to earn the trust expected of database engines, probabilistic accuracy guarantees must withstand rigorous empirical auditing rather than fail silently in practice.

Acknowledgments

This research was supported by the Australian Government through the Office of National Intelligence's funded National Intelligence Discovery Grant program (ID250100316). The views expressed herein are those of the authors and are not necessarily those of the Australian Government or the Office of National Intelligence. This research was also supported by The University of Melbourne's Research Computing Services and the Petascale Campus.

References

[1] Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. 2025. Learn then test: calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19, 2, 1641–1662. doi:10.1214/24-AOAS1998.

[2] Artificial Analysis. 2026. GPT-5.6 Terra (non-reasoning) – Intelligence, Performance & Price Analysis. Version Intelligence Index v4.3.2. Artificial Analysis, Inc. (July 9, 2026). Retrieved Oct. 16, 2026 from https://artificialanalysis.ai/models/gpt-5-6-terra-non-reasoning.

[3] Lingjiao Chen, Matei Zaharia, and James Zou. 2024. FrugalGPT: how to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*. Featured Certification. https://openreview.net/forum?id=cSimKw5p6R.

[4] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Iryna Gurevych and Yusuke Miyao, (Eds.) Association for Computational Linguistics, Melbourne, Australia, (July 2018), 87–96. doi:10.18653/v1/P18-1009.

[5] Yeounoh Chung et al. 2026. 100x cost & latency reduction: performance analysis of AI query approximation using lightweight proxy models: [experiments & analysis]. *Proc. ACM Manag. Data*, 4, 3, Article 125, (May 2026), 23 pages. doi:10.1145/3802002.

[6] Hanjun Dai et al. 2024. UQE: a query engine for unstructured databases. In *Advances in Neural Information Processing Systems*. A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, (Eds.) Vol. 37. Curran Associates, Inc., 29807–29838. doi:10.52202/079017-0938.

[7] Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. 2024. Hybrid LLM: cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*. B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, (Eds.) Vol. 2024, 41348–41366. https://proceedings.iclr.cc/paper_files/paper/2024/file/b47d93c99fa22ac0b377578af0a1f63a-Paper-Conference.pdf.

[8] Anas Dorbani, Sunny Yasser, Jimmy Lin, and Amine Mhedhbi. 2025. Beyond quacking: Deep integration of language models and RAG into DuckDB. *Proc. VLDB Endow.*, 18, 12, (Aug. 2025), 5415–5418. doi:10.14778/3750601.3750685.

[9] Jian He and Vaibhav Sethi. 2025. SQL reimagined for the AI era with BigQuery AI functions. Google Cloud Blog. (Nov. 13, 2025). Retrieved Oct. 8, 2026 from https://cloud.google.com/blog/products/data-analytics/sql-reimagined-for-the-ai-era-with-bigquery-ai-functions.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, (June 2016), 770–778. doi:10.1109/CVPR.2016.90.

[11] Nan Hou, Kangfei Zhao, Jiadong Xie, and Jeffrey Xu Yu. 2026. Beyond linear LLM invocation: an efficient and effective semantic filter paradigm. (2026). https://arxiv.org/abs/2603.04799 arXiv: 2603.04799 [cs.DB].

[12] Ronald A. Howard. 1966. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2, 1, 22–26. doi:10.1109/TSSC.1966.300074.

[13] Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. 2021. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49, 2, pp. 1055–1080. https://www.jstor.org/stable/27169410.

[14] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B. Gibbons, and Onur Mutlu. 2018. Focus: querying large video datasets with low latency and low cost. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. USENIX Association, Carlsbad, CA, (Oct. 2018), 269–286. ISBN: 978-1-939133-08-3. https://www.usenix.org/conference/osdi18/presentation/hsieh.

[15] Yiqian Huang, Shiqi Zhang, Tianyuan Jin, and Xiaokui Xiao. 2026. SCOPE: cost-efficient model selection for compound AI systems under quality constraints. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2* (KDD '26). Association for Computing Machinery, Jeju Island, Republic of Korea, 1922–1931. ISBN: 9798400722592. doi:10.1145/3770855.3818067.

[16] Ramesh Johari, Pete Koomen, Leonid Pekelis, and David Walsh. 2017. Peeking at A/B tests: why it matters, and what to do about it. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '17). Association for Computing Machinery, Halifax, NS, Canada, 1517–1525. ISBN: 9781450348874. doi:10.1145/3097983.3097992.

[17] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8, 64–77. Mark Johnson, Brian Roark, and Ani Nenkova, (Eds.) doi:10.1162/tacl_a_00300.

[18] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. 2012. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning* (ICML'12). Omnipress, Edinburgh, Scotland, 227–234. ISBN: 9781450312851.

[19] Daniel Kang, Peter Bailis, and Matei Zaharia. 2019. BlazeIt: optimizing declarative aggregation and limit queries for neural network-based video analytics. *Proc. VLDB Endow.*, 13, 4, (Dec. 2019), 533–546. doi:10.14778/3372716.3372725.

[20] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: optimizing neural network queries over video at scale. *Proc. VLDB Endow.*, 10, 11, (Aug. 2017), 1586–1597. doi:10.14778/3137628.3137664.

[21] Daniel Kang, Edward Gan, Peter Bailis, Tatsunori Hashimoto, and Matei Zaharia. 2020. Approximate selection with guarantees using proxies. *Proc. VLDB Endow.*, 13, 12, (July 2020), 1990–2003. doi:10.14778/3407790.3407804.

[22] Daniel Kang, John Guibas, Peter Bailis, Tatsunori Hashimoto, Yi Sun, and Matei Zaharia. 2021. Accelerating approximate aggregation queries with expensive predicates. *Proc. VLDB Endow.*, 14, 11, (July 2021), 2341–2354. doi:10.14778/3476249.3476285.

[23] Daniel Kang, John Guibas, Peter D. Bailis, Tatsunori Hashimoto, and Matei Zaharia. 2022. TASTI: semantic indexes for machine learning-based queries over unstructured data. In *Proceedings of the 2022 International Conference on Management of Data* (SIGMOD '22). Association for Computing Machinery, Philadelphia, PA, USA, 1934–1947. ISBN: 9781450392495. doi:10.1145/3514221.3517897.

[24] Udesh Kumarasinghe, Tyler Liu, Ahmed Mahmood, Chunwei Liu, and Walid G. Aref. 2026. iPDB: SQL with ML and LLM predicates (towards a database engine for AI). *Proc. VLDB Endow.*, 19, 12, (Sept. 2026), 4782–4785. doi:10.14778/3827998.3828121.

[25] Yiming Lin, Madelon Hulsebos, Ruiying Ma, Shreya Shankar, Sepanta Zeighami, Aditya G. Parameswaran, and Eugene Wu. 2025. Querying templatized document collections with large language models. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, (May 2025), 2422–2435. doi:10.1109/ICDE65448.2025.00183.

[26] Paweł Liskowski and Kyle Schmaus. 2026. Streaming model cascades for semantic SQL. (2026). https://arxiv.org/abs/2604.00660 arXiv: 2604.00660 [cs.DB].

[27] Paweł Liskowski et al. 2026. Cortex AISQL: a production SQL engine for unstructured data. In *Companion of the International Conference on Management of Data* (SIGMOD Companion '26). Association for Computing Machinery, India, 400–412. ISBN: 9798400724503. doi:10.1145/3788853.3803093.

[28] Chunwei Liu et al. 2025. Palimpzest: optimizing AI-powered analytics with declarative query processing. In *Proceedings of the 15th Conference on Innovative Data Systems Research (CIDR)* (CIDR '25). CIDR, 7 pages.

[29] Yao Lu, Aakanksha Chowdhery, Srikanth Kandula, and Surajit Chaudhuri. 2018. Accelerating machine learning inference with probabilistic predicates. In *Proceedings of the 2018 International Conference on Management of Data* (SIGMOD '18). Association for Computing Machinery, Houston, TX, USA, 1493–1508. ISBN: 9781450347037. doi:10.1145/3183713.3183751.

[30] Willi Maurer, Ludwig Hothorn, and Walter Lehmacher. 1995. Multiple comparisons in drug clinical trials and preclinical assays: a-priori ordered hypotheses. In *Biometrie in der chemisch-pharmazeutischen Industrie*. Vol. 6. J. Vollmar, (Ed.) Fischer Verlag, Stuttgart, 3–18.

[31] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Kadous, and Ion Stoica. 2025. RouteLLM: learning to route LLMs from preference data. In *International Conference on Learning Representations*. Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, (Eds.) Vol. 2025, 34433–34448. https://proceedings.iclr.cc/paper_files/paper/2025/file/5503a7c69d48a2f86fc00b3dc09de686-Paper-Conference.pdf.

[32] OpenAI. 2026. Pricing | OpenAI API. OpenAI. (July 30, 2026). Retrieved Oct. 8, 2026 from https://developers.openai.com/api/docs/pricing.

[33] Liana Patel, Siddharth Jha, Melissa Pan, Harshit Gupta, Parth Asawa, Carlos Guestrin, and Matei Zaharia. 2025. Semantic operators and their optimization: Enabling LLM-based data processing with accuracy guarantees in LOTUS. *Proc. VLDB Endow.*, 18, 11, (July 2025), 4171–4184. doi:10.14778/3749646.3749685.

[34] Kangkang Qi, Dongyang Xie, Wenbo Li, Hao Zhang, Yuanyuan Zhu, Jeffrey Xu Yu, and Kangfei Zhao. 2026. Sema: a high-performance system for LLM-based semantic query processing. *Proc. VLDB Endow.*, 19, 11, (Sept. 2026), 3231–3244. doi:10.14778/3836663.3836685.

[35] Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. 2023. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38, 4, 576–601. doi:10.1214/23-STS894.

[36] Herbert Robbins. 1970. Statistical methods related to the law of the iterated logarithm. *The Annals of Mathematical Statistics*, 41, 5, 1397–1409. doi:10.1214/aoms/1177696786.

[37] Matthew Russo, Chunwei Liu, Sivaprasad Sudhir, Gerardo Vitagliano, Michael Cafarella, Tim Kraska, and Samuel Madden. 2026. Abacus: a cost-based optimizer for semantic operator systems. *Proc. VLDB Endow.*, 19, 5, (May 2026), 1060–1073. doi:10.14778/3796195.3796215.

[38] Gabriele Sanmartino, Matthias Urban, Paolo Papotti, and Carsten Binnig. 2026. The Stretto execution engine for LLM-augmented data systems. (2026). https://arxiv.org/abs/2602.04430 arXiv: 2602.04430 [cs.DB].

[39] Shreya Shankar. 2026. Add BARGAIN_PR for combined precision + recall guarantees. ucbepic/BARGAIN, Pull Request #1. EPIC Lab at UC Berkeley. (June 8, 2026). Retrieved Oct. 11, 2026 from https://github.com/ucbepic/BARGAIN/pull/1.

[40] Shreya Shankar, Tristan Chambers, Tarak Shah, Aditya G. Parameswaran, and Eugene Wu. 2025. DocETL: agentic query rewriting and evaluation for complex document processing. *Proc. VLDB Endow.*, 18, 9, (May 2025), 3035–3048. doi:10.14778/3746405.3746426.

[41] Shreya Shankar, Sepanta Zeighami, and Aditya Parameswaran. 2026. Task cascades for efficient unstructured data processing. *Proc. ACM Manag. Data*, 4, 1, Article 88, (Apr. 2026), 26 pages. doi:10.1145/3786702.

[42] Zhaoze Sun, Chengliang Chai, Qiyan Deng, Kaisen Jin, Xinyu Guo, Han Han, Ye Yuan, Guoren Wang, and Lei Cao. 2025. QUEST: Query optimization in unstructured document analysis. *Proc. VLDB Endow.*, 18, 11, (July 2025), 4560–4573. doi:10.14778/3749646.3749713.

[43] Jean Ville. 1939. *Étude critique de la notion de collectif*. Ph.D. Dissertation. Université de Paris. http://eudml.org/doc/192893.

[44] P. Viola and M. Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. IEEE, Kauai, Hawaii, USA. doi:10.1109/CVPR.2001.990517.

[45] Ian Waudby-Smith, David Arbour, Ritwik Sinha, Edward H. Kennedy, and Aaditya Ramdas. 2024. Time-uniform central limit theory and asymptotic confidence sequences. *The Annals of Statistics*, 52, 6, 2613–2640. doi:10.1214/24-AOS2408.

[46] Ian Waudby-Smith and Aaditya Ramdas. 2024. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B:*

Statistical Methodology, 86, 1, (Feb. 2024), 1–27. eprint: https://academic.oup.com/jrsssb/article-pdf/86/1/1/56961777/qkad009.pdf. doi:10.1093/jrsssb/qkad009.

[47] Lindsey Linxi Wei, Shreya Shankar, Sepanta Zeighami, Yeounoh Chung, Fatma Ozcan, and Aditya G. Parameswaran. 2026. Multi-objective agentic rewrites for unstructured data processing. *Proc. VLDB Endow.*, 19, 11, (Sept. 2026), 3062–3075. doi:10.14778/3836663.3836673.

[48] Peter H. Westfall and Alok Krishen. 2001. Optimally weighted, fixed sequence and gatekeeper multiple testing procedures. *Journal of Statistical Planning and Inference*, 99, 1, 25–40. doi:https://doi.org/10.1016/S0378-3758(01)00077-5.

[49] Zhihui Yang, Zuozhi Wang, Yicong Huang, Yao Lu, Chen Li, and X. Sean Wang. 2022. Optimizing machine learning inference queries with correlative proxy models. *Proc. VLDB Endow.*, 15, 10, (June 2022), 2032–2044. doi:10.14778/3547305.3547310.

[50] Sepanta Zeighami, Shreya Shankar, and Aditya Parameswaran. 2025. Cut costs, not accuracy: LLM-powered data processing with guarantees. *Proc. ACM Manag. Data*, 3, 6, Article 311, (Dec. 2025), 26 pages. doi:10.1145/3769776.

[51] Sepanta Zeighami, Shreya Shankar, and Aditya Parameswaran. 2026. Featurized-decomposition join: low-cost semantic joins with guarantees. *Proc. VLDB Endow.*, 19, 11, (Sept. 2026), 3048–3061. doi:10.14778/3836663.3836672.

[52] Hengrui Zhang, Yulong Hui, Yihao Liu, and Huanchen Zhang. 2026. ScaleDoc: scaling LLM-based predicates over large document collections. *Proc. ACM Manag. Data*, 4, 3, Article 229, (May 2026), 26 pages. doi:10.1145/3802106.

[53] Junhao Zhu, Lu Chen, Xiangyu Ke, Ziquan Fang, Tianyi Li, Yunjun Gao, and Christian S. Jensen. 2026. Beyond relational: semantic-aware multi-modal analytics with LLM-native query optimization. *Proc. ACM Manag. Data*, 4, 1, Article 14, (Apr. 2026), 25 pages. doi:10.1145/3786628.

A Technical Details for Section 3

This appendix provides the mathematical derivations, hyperparameter configurations, and formal proofs supporting the ACTIS framework in Sec. 3. Specifically, App. A.1 details the affine transformations used to map importance-weighted recall and precision increments onto $[0,1]$ for the non-asymptotic betting test (Sec. 3.2). App. A.2 specifies the hyperparameter settings for both anytime-valid testing subroutines, including the proxy-surrogate horizon variance estimator and null-boundary error clamping for the asymptotic Gaussian mixture test. App. A.3 derives the single-step variance floor (Safeguard 1) and the instrumental-adjusted null failure probability (Safeguard 3) introduced in Sec. 3.3. Appendices A.4 and A.5 provide the formal proofs of Propositions 3.1 and 3.2, respectively. Appendix A.6 specifies the warmup threshold $n_{\min}$ used by the Value-of-Information stopping rule (Sec. 3.5). Finally, App. A.7 formalizes the joint variance objective, proves Proposition 3.3, and derives the variance-minimizing component weights in Sec. 3.6.

A.1 Rescaling for the Non-Asymptotic Betting-Based Test

When implementing RecallMet and PrecisionMet using non-asymptotic betting martingales (Sec. 2.2.1), test increments must lie within the unit interval $[0,1]$. However, the raw recall and precision increments f_R and f_P (Eqs. (13) and (15)) violate this condition: they take negative values on constraint violations (false negatives and false positives, respectively) and can exceed 1 under non-uniform importance weights. We therefore map each raw increment $Z \in \{f_R, f_P\}$ into $[0,1]$ via an affine transformation $Z \mapsto (Z + \text{shift})/\text{scale}$, which shifts the corresponding null hypothesis threshold from 0 to $m := \text{shift}/\text{scale} \in (0,1)$.

To define shift and scale parameters that guarantee $[0,1]$ support across the workload $\mathcal{D}$, we establish upper bounds on the importance weights:

$$w^{\geq}_{\max}(\tau) := \max_{x \in \mathcal{D}:\ \mathrm{A}(x) \geq \tau} w(x), \quad w^{<}_{\max}(\tau) := \max_{x \in \mathcal{D}:\ \mathrm{A}(x) < \tau} w(x). \tag{19}$$

Under uniform sampling, $w(x) \equiv 1$ yielding $w^{\geq}_{\max}(\tau) = w^{<}_{\max}(\tau) = 1$ for all thresholds.

For recall, the increment $f_R(X, Y; \check{\tau})$ is bounded from below by $-\gamma_R w^{<}_{\max}(\check{\tau})$ and bounded from above by $(1-\gamma_R) w^{\geq}_{\max}(\check{\tau})$. Defining

$$\begin{aligned} \text{shift}_R(\check{\tau}) &:= \gamma_R w^{<}_{\max}(\check{\tau}) \\ \text{scale}_R(\check{\tau}) &:= (1-\gamma_R) w^{\geq}_{\max}(\check{\tau}) + \gamma_R w^{<}_{\max}(\check{\tau}) \end{aligned}$$

ensures $[0,1]$-bounded increments, which are evaluated against the rescaled target margin $m_R(\check{\tau}) := \text{shift}_R(\check{\tau})/\text{scale}_R(\check{\tau})$.

Similarly, for precision, the increment $f_P(X, Y; \check{\tau}, \hat{\tau})$ is bounded from below by $-\gamma_P w^{\geq}_{\max}(\hat{\tau})$ and bounded from above by $(1-\gamma_P) w^{\geq}_{\max}(\check{\tau})$. Defining

$$\begin{aligned} \text{shift}_P(\hat{\tau}) &:= \gamma_P w^{\geq}_{\max}(\hat{\tau}) \\ \text{scale}_P(\check{\tau}, \hat{\tau}) &:= (1-\gamma_P) w^{\geq}_{\max}(\check{\tau}) + \gamma_P w^{\geq}_{\max}(\hat{\tau}) \end{aligned}$$

ensures $[0,1]$-bounded increments, which are evaluated against the rescaled target margin $m_P(\check{\tau}, \hat{\tau}) := \text{shift}_P(\hat{\tau})/\text{scale}_P(\check{\tau}, \hat{\tau})$.

A.2 Hyperparameter Settings for Anytime-Valid Tests

This section details how hyperparameters are configured for the two anytime-valid test subroutines in Section 3.2.

A.2.1 Non-Asymptotic Betting-Based Test. For the betting test, predictable betting fractions λ_i in Eq. (9) are evaluated using the standard default hyperparameter settings recommended by Waudby-Smith and Ramdas [46]:

- Truncation scale parameter $c = 1/2$.
- Prior mean $\tilde{\mu}_0 = 1/2$ and prior variance $\tilde{\sigma}^2_0 = 1/4$ for the regularized sample variance estimator $\tilde{\sigma}^2_t$, initialized with one pseudo-observation.

A.2.2 Asymptotic Gaussian Mixture Test. To optimize statistical power over a planned tuning horizon n_0, the baseline prior variance parameter v in Eq. (10) is set to the estimated cumulative process variance $v^{\text{est}} := n_0 \tilde{\sigma}^2$ with $n_0 = \min(1000, \max(50, N/10))$. The critical operating point for Type I error control is the boundary of the null hypothesis where $\Delta_R(\check{\tau}) = 0$ or $\Delta_P(\check{\tau}, \hat{\tau}) = 0$. In the interior of the null ($\Delta < 0$), drift is negative, impeding rejection; in the alternative ($\Delta > 0$), positive drift accelerates rejection. We therefore evaluate the process variance σ^2 specifically at the null boundary.

Proposition A.1 (Process Variance at the Boundary Null). *At the boundary of the null hypotheses $H_{0,R}(\check{\tau})$ and $H_{0,P}(\check{\tau}, \hat{\tau})$, the process variances are given by:*

$$\sigma^2_R(\check{\tau}) = (1-\gamma_R)^2 \,\mathrm{TPR}_w(\check{\tau}) + \gamma_R^2 \,\mathrm{FNR}_w(\check{\tau}), \tag{20}$$

$$\sigma^2_P(\check{\tau}, \hat{\tau}) = (1-\gamma_P)^2 \,\mathrm{TPR}_w(\check{\tau}) + \gamma_P^2 \,\mathrm{FPR}_w(\hat{\tau}), \tag{21}$$

where the importance-weighted true positive, false negative, and false positive components are defined as:

$$\begin{aligned} \mathrm{TPR}_w(\check{\tau}) &:= \mathop{\mathrm{E}}_{X \sim q, Y}[w(X)^2 Y \mathbf{1}_{\{\mathrm{A}(X) \geq \check{\tau}\}}], \\ \mathrm{FNR}_w(\check{\tau}) &:= \mathop{\mathrm{E}}_{X \sim q, Y}[w(X)^2 Y \mathbf{1}_{\{\mathrm{A}(X) < \check{\tau}\}}], \\ \mathrm{FPR}_w(\hat{\tau}) &:= \mathop{\mathrm{E}}_{X \sim q, Y}[w(X)^2 (1-Y) \mathbf{1}_{\{\mathrm{A}(X) \geq \hat{\tau}\}}]. \end{aligned}$$

Proof. For brevity, we let $f = f_R(X, Y; \check{\tau})$ and $f = f_P(X, Y; \check{\tau}, \hat{\tau})$ denote the increments for $H_{0,R}(\check{\tau})$ and $H_{0,P}(\check{\tau}, \hat{\tau})$ respectively. At the boundary of the null hypothesis, the expected increment is zero ($\mathrm{E}_q[f] = 0$). The variance reduces to the uncentered second moment: $\mathrm{Var}_q(f) = \mathrm{E}_q[f^2]$. Since oracle labels and indicator functions are $\{0,1\}$-valued, we have $Y^2 = Y$, $(1-Y)^2 = 1-Y$, and $\mathbf{1}^2_{\{\cdot\}} = \mathbf{1}_{\{\cdot\}}$.

For recall, squaring the increment Eq. (13) yields:

$$\begin{aligned} f_R(X, Y; \check{\tau})^2 &= w(X)^2 Y^2 \left(\mathbf{1}_{\{\mathrm{A}(X)\geq\check{\tau}\}} - \gamma_R\right)^2 \\ &= w(X)^2 Y \left((1-\gamma_R)^2 \mathbf{1}_{\{\mathrm{A}(X)\geq\check{\tau}\}} + \gamma_R^2 \mathbf{1}_{\{\mathrm{A}(X)<\check{\tau}\}}\right). \end{aligned}$$

Taking expectations with respect to $X \sim q$ and $Y \sim \mathrm{O}(X)$ yields Eq. (20).

For precision, note that because $\hat{\tau} \geq \check{\tau}$, the event $\mathrm{A}(X) \geq \hat{\tau}$ implies $\mathrm{A}(X) \geq \check{\tau}$. Moreover, an item cannot be simultaneously positive and negative ($Y(1-Y) = 0$), so cross-product terms vanish. Squaring the precision increment Eq. (15) yields:

$$\begin{aligned} f_P(X, Y; \check{\tau}, \hat{\tau})^2 = w(X)^2 \big((1-\gamma_P)^2 Y \mathbf{1}_{\{\mathrm{A}(X)\geq\check{\tau}\}} \\ + \gamma_P^2 (1-Y) \mathbf{1}_{\{\mathrm{A}(X)\geq\hat{\tau}\}}\big). \end{aligned}$$

Taking expectations under $X \sim q$ and $Y \sim \mathrm{O}(X)$ yields Eq. (21). □

Proxy Surrogate Variance with Null Boundary Clamping. Prior to cascade tuning, ground-truth oracle labels are unobserved. We estimate the variance components across the workload $\mathcal{D}$ by substituting proxy confidence scores $\mathrm{A}(x)$ as surrogates for oracle expectations $\mathrm{E}[Y \mid x]$:

$$\widetilde{\mathrm{TPR}}_w(\check{\tau}) \coloneqq \frac{1}{N} \sum_{x\in\mathcal{D}} w(x)\, \mathrm{A}(x) \mathbf{1}_{\{\mathrm{A}(x)\geq\check{\tau}\}}, \tag{22}$$

$$\widetilde{\mathrm{FNR}}_w(\check{\tau}) \coloneqq \frac{1}{N} \sum_{x\in\mathcal{D}} w(x)\, \mathrm{A}(x) \mathbf{1}_{\{\mathrm{A}(x)<\check{\tau}\}}, \tag{23}$$

$$\widetilde{\mathrm{FPR}}_w(\hat{\tau}) \coloneqq \frac{1}{N} \sum_{x\in\mathcal{D}} w(x)(1-\mathrm{A}(x)) \mathbf{1}_{\{\mathrm{A}(x)\geq\hat{\tau}\}}. \tag{24}$$

If the proxy model is miscalibrated, raw surrogate error estimates $\widetilde{\mathrm{FNR}}_w$ or $\widetilde{\mathrm{FPR}}_w$ could artificially inflate or deflate the estimated variance. To improve robustness, we constrain these error terms to be consistent with the theoretical null boundary:

Proposition A.2 (Null Boundary Error Bounds). *At the boundary of the null hypotheses $H_{0,R}(\check{\tau})$ and $H_{0,P}(\check{\tau}, \hat{\tau})$, the importance-weighted error components satisfy:*

$$\mathrm{FNR}_w(\check{\tau}) \leq \rho_R(\check{\tau}) \left(\frac{1-\gamma_R}{\gamma_R}\right) \mathrm{TPR}_w(\check{\tau}), \tag{25}$$

$$\mathrm{FPR}_w(\hat{\tau}) \leq \rho_P(\check{\tau}, \hat{\tau}) \left(\frac{1-\gamma_P}{\gamma_P}\right) \mathrm{TPR}_w(\check{\tau}), \tag{26}$$

where the partition weight disparity ratios are:

$$\rho_R(\check{\tau}) \coloneqq \frac{\max_{x:\ \mathrm{A}(x)<\check{\tau}} w(x)}{\min_{x:\ \mathrm{A}(x)\geq\check{\tau}} w(x)}, \quad \rho_P(\check{\tau}, \hat{\tau}) \coloneqq \frac{\max_{x:\ \mathrm{A}(x)\geq\hat{\tau}} w(x)}{\min_{x:\ \mathrm{A}(x)\geq\check{\tau}} w(x)}.$$

Proof. At the null boundary for recall, $\Delta_R(\check{\tau}) = \mathrm{TPR}(\check{\tau}) - \gamma_R\, \pi = 0$. Since $\pi = \mathrm{TPR}(\check{\tau}) + \mathrm{FNR}(\check{\tau})$, this implies $\mathrm{FNR}(\check{\tau}) = \frac{1-\gamma_R}{\gamma_R} \mathrm{TPR}(\check{\tau})$. Bounding the weighted components:

$$\mathrm{FNR}_w(\check{\tau}) = \mathop{\mathrm{E}}_{X\sim p,Y}[w(X) Y \mathbf{1}_{\{\mathrm{A}(X)<\check{\tau}\}}] \leq \left(\max_{x:\ \mathrm{A}(x)<\check{\tau}} w(x)\right) \mathrm{FNR}(\check{\tau}),$$

$$\mathrm{TPR}_w(\check{\tau}) = \mathop{\mathrm{E}}_{X\sim p,Y}[w(X) Y \mathbf{1}_{\{\mathrm{A}(X)\geq\check{\tau}\}}] \geq \left(\min_{x:\ \mathrm{A}(x)\geq\check{\tau}} w(x)\right) \mathrm{TPR}(\check{\tau}).$$

Combining these inequalities with the null boundary relation yields Eq. (25). An identical derivation holds for precision, establishing Eq. (26). □

Clamping the surrogate error components by their theoretical null limits yields the process variance estimators:

$$\begin{aligned} \widetilde{\sigma}_R^2(\check{\tau}) &\coloneqq (1-\gamma_R)^2 \widetilde{\mathrm{TPR}}_w(\check{\tau}) \\ &\quad + \gamma_R^2 \min\left(\widetilde{\mathrm{FNR}}_w(\check{\tau}),\ \rho_R(\check{\tau}) \left(\frac{1-\gamma_R}{\gamma_R}\right) \widetilde{\mathrm{TPR}}_w(\check{\tau})\right), \\ \widetilde{\sigma}_P^2(\check{\tau}, \hat{\tau}) &\coloneqq (1-\gamma_P)^2 \widetilde{\mathrm{TPR}}_w(\check{\tau}) \\ &\quad + \gamma_P^2 \min\left(\widetilde{\mathrm{FPR}}_w(\hat{\tau}),\ \rho_P(\check{\tau}, \hat{\tau}) \left(\frac{1-\gamma_P}{\gamma_P}\right) \widetilde{\mathrm{TPR}}_w(\check{\tau})\right). \end{aligned}$$

The baseline estimated horizon variances are then $v_R^{\mathrm{est}} = n_0 \widetilde{\sigma}_R^2(\check{\tau})$ and $v_P^{\mathrm{est}} = n_0 \widetilde{\sigma}_P^2(\check{\tau}, \hat{\tau})$. When Safeguard 1 is active (Sec. 3.3), these are floored via Eq. (16) during cascade tuning.

A.3 Finite-Sample Safeguard Derivations

This section provides derivations for the runtime safeguards introduced in Section 3.3.

A.3.1 Proof of Single-Step Boundary Safety (Safeguard 1).

Proposition A.3 (Single-Step Boundary Safety). *Let M_t be the Gaussian mixture process in Eq. (10) with rejection threshold $1/\alpha$ for significance level $\alpha \in (0,1)$. Suppose the increments Z_t are bounded from above by b and the prior variance satisfies:*

$$v \geq v^{\mathit{floor}} \coloneqq \frac{2b^2}{\log(1/\alpha)}.$$

For any first observation $Z_1 \leq b$ (at sample size $t = 1$, where $V_1 = 0$), the accumulated wealth satisfies:

$$M_1 \leq \left(\frac{1}{\alpha}\right)^{1/4} < \frac{1}{\alpha}.$$

Consequently, no single observation can ever cross the rejection boundary on the first sample.

Proof. At sample size $t = 1$, the cumulative sample mean is $\bar{Z}_1 = Z_1$. The empirical variance accumulator evaluates to:

$$V_1 = (Z_1 - \bar{Z}_1)^2 = 0,$$

and the cumulative drift evaluates to $S_1 = Z_1$ (assuming $m = 0$). If $Z_1 \leq 0$, the process yields $M_1 = 0$. For any positive increment $Z_1 \in (0, b]$, substituting $V_1 = 0$ into Eq. (10) yields $M_1 = \exp(Z_1^2/(2v))$. Since $Z_1 \leq b$ and $v \geq 2b^2/\log(1/\alpha)$, we obtain:

$$M_1 \leq \exp\left(\frac{b^2}{2 \cdot \frac{2b^2}{\log(1/\alpha)}}\right) = \exp\left(\frac{\log(1/\alpha)}{4}\right) = \left(\frac{1}{\alpha}\right)^{1/4} < \frac{1}{\alpha}.$$

□

A.3.2 Derivation of Instrumental-Adjusted Failure Probability (Safeguard 3).

Proposition A.4 (Instrumental-Adjusted Null Failure Probability). *Let $F, S \subset \mathcal{D} \times \{0,1\}$ denote disjoint failure and success subsets, and let $A \coloneqq F \cup S$ denote the active subset. Let $\pi_F \coloneqq \mathrm{E}_p[\mathbf{1}_{\{(X,Y)\in F\}}]$ and $\pi_S \coloneqq \mathrm{E}_p[\mathbf{1}_{\{(X,Y)\in S\}}]$ denote their population masses under nominal uniform distribution p. Define the average instrumental densities over the failure and success regions:*

$$\bar{q}_F \coloneqq \frac{\sum_{x\in\mathcal{D}} q(x)\,\mathrm{E}[\mathbf{1}_{\{(x,Y)\in F\}} \mid x]}{N\pi_F},$$
$$\bar{q}_S \coloneqq \frac{\sum_{x\in\mathcal{D}} q(x)\,\mathrm{E}[\mathbf{1}_{\{(x,Y)\in S\}} \mid x]}{N\pi_S}.$$

Under the boundary null hypothesis $\pi_F/\pi_S = (1-\gamma)/\gamma$, the conditional probability of drawing a failure under instrumental distribution q given that an active record is sampled is:

$$p_0 \coloneqq \Pr_{X\sim q,Y}\big[(X,Y)\in F \,\big|\, (X,Y)\in A\big] = \frac{(1-\gamma)\eta}{\gamma + (1-\gamma)\eta}, \tag{27}$$

where $\eta \coloneqq \bar{q}_F/\bar{q}_S$ is the instrumental density ratio.

Proof. Applying Bayes' rule and the definition of conditional probability:

$$p_0 = \frac{\Pr_q[(X,Y)\in F]}{\Pr_q[(X,Y)\in F] + \Pr_q[(X,Y)\in S]} = \frac{\bar{q}_F\pi_F}{\bar{q}_F\pi_F + \bar{q}_S\pi_S} = \frac{(\bar{q}_F/\bar{q}_S)(\pi_F/\pi_S)}{1 + (\bar{q}_F/\bar{q}_S)(\pi_F/\pi_S)}.$$

Substituting the boundary null condition $\pi_F/\pi_S = (1-\gamma)/\gamma$ and instrumental density ratio $\eta = \bar{q}_F/\bar{q}_S$ yields Eq. (27). □

Instrumental Ratio for Recall (η_R). For recall at threshold $\check{\tau}$, failures (false negatives) reside strictly in the proxy rejection partition $\mathcal{D}_0(\check{\tau}) = \{x\colon \mathrm{A}(x) < \check{\tau}\}$, while successes (true positives) reside in the cascade acceptance partition $\mathcal{D} \setminus \mathcal{D}_0(\check{\tau}) = \{x\colon \mathrm{A}(x) \geq \check{\tau}\}$. Prior to cascade tuning, the conditional label prevalence $\mathrm{E}[Y \mid x]$ within each partition is unobserved. Under an uninformative prior that treats conditional label density as homogeneous within each partition, the unknown constants factor out of the numerator and denominator sums. This reduces $\bar{q}_F$ and $\bar{q}_S$ to the unweighted arithmetic means of $q(x)$ across the respective partitions:

$$\eta_R(\check{\tau}) \coloneqq \frac{|\mathcal{D}_0(\check{\tau})|^{-1}\sum_{x\in\mathcal{D}_0(\check{\tau})} q(x)}{|\mathcal{D}\setminus\mathcal{D}_0(\check{\tau})|^{-1}\sum_{x\in\mathcal{D}\setminus\mathcal{D}_0(\check{\tau})} q(x)}.$$

Substituting $\eta_R(\check{\tau})$ into Eq. (27) with $\gamma = \gamma_R$ yields the adjusted null failure probability $p_{0,R}(\check{\tau})$.

Instrumental Ratio for Precision (η_P). For precision at $(\check{\tau},\hat{\tau})$, failures (false positives) and successes (true positives) can both occur throughout the cascade acceptance region. We use the proxy scores as a surrogate for the oracle probability: an item with proxy score $\mathrm{A}(x)$ has surrogate false-positive probability $1-\mathrm{A}(x)$ and surrogate true-positive probability $\mathrm{A}(x)$. Weighting instrumental draws by these proxy surrogates gives:

$$\bar{q}_{\mathrm{FP}}(\hat{\tau}) \coloneqq \frac{\sum_{x\colon \mathrm{A}(x)\geq\hat{\tau}} q(x)(1-\mathrm{A}(x))}{\sum_{x\colon \mathrm{A}(x)\geq\hat{\tau}}(1-\mathrm{A}(x))},$$
$$\bar{q}_{\mathrm{TP}}(\check{\tau}) \coloneqq \frac{\sum_{x\colon \mathrm{A}(x)\geq\check{\tau}} q(x)\,\mathrm{A}(x)}{\sum_{x\colon \mathrm{A}(x)\geq\check{\tau}} \mathrm{A}(x)}.$$

To guard against proxy miscalibration, we clamp the ratio by the extremal instrumental density range over the active regions:

$$\eta_{P,\min}(\check{\tau},\hat{\tau}) \coloneqq \frac{\min_{x\colon \mathrm{A}(x)\geq\hat{\tau}} q(x)}{\max_{x\colon \mathrm{A}(x)\geq\check{\tau}} q(x)},$$
$$\eta_{P,\max}(\check{\tau},\hat{\tau}) \coloneqq \frac{\max_{x\colon \mathrm{A}(x)\geq\hat{\tau}} q(x)}{\min_{x\colon \mathrm{A}(x)\geq\check{\tau}} q(x)},$$
$$\eta_P(\check{\tau},\hat{\tau}) \coloneqq \mathrm{clip}\left(\frac{\bar{q}_{\mathrm{FP}}(\hat{\tau})}{\max(\bar{q}_{\mathrm{TP}}(\check{\tau}), 10^{-12})},\ \eta_{P,\min},\ \eta_{P,\max}\right).$$

Evaluating Eq. (27) with $\gamma = \gamma_P$ and clamping the result to $p_{0,P} \coloneqq \min\big(0.50,\ \frac{(1-\gamma_P)\eta_P}{\gamma_P+(1-\gamma_P)\eta_P}\big)$ yields the precision null failure probability. The 0.5 ceiling prevents a miscalibrated proxy from asserting that cascade positive predictions are more likely to fail than succeed under the null ($\eta_P \geq \gamma_P/(1-\gamma_P)$, e.g., ≥ 9 for $\gamma_P \geq 0.9$).

A.4 Proof of Proposition 3.1

Consider the true positive rate $\mathrm{TPR}(\check{\tau}) = \mathrm{E}[Y\mathbf{1}_{\{\mathrm{A}(X)\geq\check{\tau}\}}]$. For any $\check{\tau} \leq \check{\tau}'$, the indicator satisfies $\mathbf{1}_{\{\mathrm{A}(X)\geq\check{\tau}\}} \geq \mathbf{1}_{\{\mathrm{A}(X)\geq\check{\tau}'\}}$ almost surely. Since $Y \geq 0$, we have $Y\mathbf{1}_{\{\mathrm{A}(X)\geq\check{\tau}\}} \geq Y\mathbf{1}_{\{\mathrm{A}(X)\geq\check{\tau}'\}}$, which implies $\mathrm{TPR}(\check{\tau}) \geq \mathrm{TPR}(\check{\tau}')$. Subtracting the constant $\gamma_R\,\pi$ preserves the inequality, yielding $\Delta_R(\check{\tau}) \geq \Delta_R(\check{\tau}')$, proving (i).

Now consider the false positive rate $\mathrm{FPR}(\hat{\tau}) = \mathrm{E}[(1-Y)\mathbf{1}_{\{\mathrm{A}(X)\geq\hat{\tau}\}}]$. Following an identical argument, $1-Y \geq 0$ implies $\mathrm{FPR}(\hat{\tau}) \geq \mathrm{FPR}(\hat{\tau}')$ for any $\hat{\tau} \leq \hat{\tau}'$. Multiplying by $-\gamma_P < 0$ reverses the inequality: $-\gamma_P\,\mathrm{FPR}(\hat{\tau}) \leq -\gamma_P\,\mathrm{FPR}(\hat{\tau}')$. Adding the constant $(1-\gamma_P)\,\mathrm{TPR}(\check{\tau})$ yields $\Delta_P(\check{\tau},\hat{\tau}) \leq \Delta_P(\check{\tau},\hat{\tau}')$, completing the proof of (ii).

A.5 Proof of Proposition 3.2

A joint constraint violation occurs if $\check{\tau}^\star$ violates recall or $\hat{\tau}^\star$ violates precision:

$$\mathrm{FWER} \leq \Pr[\check{\tau}^\star \text{ violates recall}] + \Pr[\hat{\tau}^\star \text{ violates precision}].$$

In Stage 1, fixed-sequence testing over $\check{G}$ ensures that the probability of rejecting any true recall null hypothesis is bounded by $\delta_R = \delta/2$.

In Stage 2, for each fixed $k \in \{1,\ldots,\check{m}\}$, let E_k be the event that the backward fixed-sequence scan over $\hat{G}$ with fixed lower threshold $\check{\tau}_k$ commits a Type I error. By fixed-sequence FWER control, $\Pr[E_k] \leq \alpha_P = \delta_P/\check{m}$ unconditionally. Since the selected threshold satisfies $\check{\tau}^\star \in \{\check{\tau}_1,\ldots,\check{\tau}_{\check{m}}\}$, a Stage 2 error implies that $\bigcup_{k=1}^{\check{m}} E_k$ occurred. By the union bound:

$$\Pr[\hat{\tau}^\star \text{ violates precision}] \leq \Pr\left[\bigcup_{k=1}^{\check{m}} E_k\right] \leq \sum_{k=1}^{\check{m}} \Pr[E_k] \leq \check{m}\alpha_P = \delta_P.$$

Combining both stages yields $\mathrm{FWER} \leq \delta_R + \delta_P = \delta/2 + \delta/2 = \delta$.

Because RecallMet and PrecisionMet are anytime-valid tests (Sec. 2.2), each individual test controls Type I error uniformly over all stopping times $t \geq 1$, ensuring the FWER bound δ holds at termination of the repeat–until loop in Alg. 1.

A.6 Warmup Threshold for Adaptive Stopping

On highly selective queries, initial tuning batches may contain zero or very few positive records ($Y = 1$), yielding degenerate empirical variance estimates for recall and optimistic thresholds. To ensure sufficient positive coverage before permitting ShouldStop (Sec. 3.5) to halt via $\mathrm{Savings}(B) \leq \mathrm{Cost}(B)$ (prior to reaching the

sample cap $n_{\max}$), ACTIS sets the minimum positive requirement $n_{\min}$ adaptively from the proxy scores $\mathcal{S}$ and recall error budget $\delta_R = \delta/2$:

$$n_{\min} = \min\Big(N,\ \max\big(5, \lceil 8\log(1/\delta_R)\rceil\big),\ \max\big(5, \lceil \hat{\pi}_{\text{floor}} N\rceil\big)\Big),$$

where $\hat{\pi}_{\text{floor}} = \text{clip}(\hat{\pi}/2,\ 10^{-4},\ 10^{-2})$ is a conservative lower bound on positive prevalence derived from the mean proxy score $\hat{\pi} = \frac{1}{N}\sum_{x\in\mathcal{D}} \mathrm{A}(x)$. Here, $\lceil 8\log(1/\delta_R)\rceil$ corresponds to the sub-Gaussian sample size required to bound the concentration radius $\sqrt{\frac{2\log(1/\delta_R)}{n_+}}$ of the positive sample below $1/2$, $\lceil \hat{\pi}_{\text{floor}} N\rceil$ caps the requirement on small or highly selective workloads, and the floor of 5 enforces the standard minimum cell count for stable empirical variance estimation.

A.7 Variance-Minimizing Instrumental Distribution: Derivations and Defensive Mechanisms

A.7.1 Variance Objective and Proof of Proposition 3.3. Evaluating whether candidate thresholds satisfy the recall and precision constraints requires estimating three constituent rates defined in Sec. 2.1: the true positive rate $\mathrm{TPR}(\check{\tau})$, the false positive rate $\mathrm{FPR}(\hat{\tau})$, and the positive prevalence π. The importance-weighted estimators of these three constituent rates are:

$$\widetilde{\mathrm{TPR}}(\check{\tau}) = \frac{1}{n}\sum_{i=1}^{n} w(X_i) Y_i \mathbf{1}_{\{\mathrm{A}(X_i)\ge\check{\tau}\}},$$

$$\widetilde{\mathrm{FPR}}(\hat{\tau}) = \frac{1}{n}\sum_{i=1}^{n} w(X_i)(1-Y_i) \mathbf{1}_{\{\mathrm{A}(X_i)\ge\hat{\tau}\}},$$

$$\widetilde{\pi} = \frac{1}{n}\sum_{i=1}^{n} w(X_i) Y_i,$$

where $w(X) = p(X)/q(X)$ is the importance weight for record X drawn from instrumental distribution q. To optimize estimation efficiency jointly across the precision and recall constraints, we seek an instrumental distribution q that minimizes the weighted average of the empirical estimation variances across all candidate thresholds in $\check{G}$ and $\hat{G}$:

$$J(q) \coloneqq \frac{\lambda_{\text{TPR}}}{\check{m}} \sum_{\check{\tau}\in\check{G}} \mathrm{Var}_q[\widetilde{\mathrm{TPR}}(\check{\tau})] + \frac{\lambda_{\text{FPR}}}{\hat{m}} \sum_{\hat{\tau}\in\hat{G}} \mathrm{Var}_q[\widetilde{\mathrm{FPR}}(\hat{\tau})] + \lambda_\pi \mathrm{Var}_q[\tilde{\pi}], \tag{28}$$

where $\lambda_{\text{TPR}}, \lambda_{\text{FPR}}, \lambda_\pi > 0$ are weighting coefficients that balance the relative emphasis placed on each constituent rate.

For any estimator of the form $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} w(X_i) g(X_i, Y_i)$, its variance decomposes as the uncentered second moment minus the squared expectation $\mathrm{E}[g(X,Y)]^2$, which is independent of q:

$$\mathrm{Var}_q[\tilde{\mu}] = \frac{1}{n}\left(\mathop{\mathrm{E}}_{X\sim p, Y}\left[\frac{p(X)}{q(X)} g(X,Y)^2\right] - \mathrm{E}[g(X,Y)]^2\right).$$

Minimizing $J(q)$ is therefore equivalent to minimizing the weighted sum of uncentered second moments:

$$J(q) \propto \mathrm{E}\left[\frac{h(\mathrm{A}(X), Y)}{q(\mathrm{A}(X))}\right],$$

where:

$$h(S,Y) = \frac{\lambda_{\text{TPR}}}{\check{m}} \sum_{\check{\tau}\in\check{G}} \left(Y \mathbf{1}_{\{S\ge\check{\tau}\}}\right)^2 + \frac{\lambda_{\text{FPR}}}{\hat{m}} \sum_{\hat{\tau}\in\hat{G}} \left((1-Y)\mathbf{1}_{\{S\ge\hat{\tau}\}}\right)^2 + \lambda_\pi Y^2.$$

Since the oracle labels and indicators are binary, we have $Y^2 = Y$, $(1-Y)^2 = 1-Y$, and $\mathbf{1}^2_{\{\cdot\}} = \mathbf{1}_{\{\cdot\}}$. Averaging across the candidate grids yields:

$$h(S,Y) = \lambda_{\text{TPR}}\, Y \frac{\check{m}(S)}{\check{m}} + \lambda_{\text{FPR}}\,(1-Y)\frac{\hat{m}(S)}{\hat{m}} + \lambda_\pi\, Y.$$

Conditioning on the proxy score $S = \mathrm{A}(X)$ and invoking the calibration assumption ($\mathrm{E}[Y \mid S] = S$ and $\mathrm{E}[1-Y \mid S] = 1-S$), we evaluate the conditional expectation:

$$\begin{aligned} c(S) &\coloneqq \mathrm{E}[h(S,Y) \mid \mathrm{A}(X) = S] \\ &= \lambda_{\text{TPR}}\, S \frac{\check{m}(S)}{\check{m}} + \lambda_{\text{FPR}}\,(1-S)\frac{\hat{m}(S)}{\hat{m}} + \lambda_\pi\, S. \end{aligned}$$

The optimization problem amounts to choosing q that minimizes $\sum_{x\in\mathcal{D}} \frac{c(\mathrm{A}(x))}{q(x)}$ subject to $\sum_{x\in\mathcal{D}} q(x) = 1$ with $q(x) > 0$. By the Cauchy-Schwarz inequality:

$$\left(\sum_{x\in\mathcal{D}} \frac{c(\mathrm{A}(x))}{q(x)}\right)\left(\sum_{x\in\mathcal{D}} q(x)\right) \ge \left(\sum_{x\in\mathcal{D}} \sqrt{c(\mathrm{A}(x))}\right)^2,$$

with equality holding if and only if $\frac{\sqrt{c(\mathrm{A}(x))}}{\sqrt{q(x)}} \propto \sqrt{q(x)}$, which uniquely implies:

$$q^*(x) = \frac{\sqrt{c(\mathrm{A}(x))}}{\sum_{x'\in\mathcal{D}} \sqrt{c(\mathrm{A}(x'))}}.$$

A.7.2 Derivation of Component Weights. To specify the component weights $\lambda_{\text{TPR}}, \lambda_{\text{FPR}}, \lambda_\pi$, we standardize the recall and precision constraints by their operational tolerance scales. The recall and precision margins can be expressed directly in terms of the component rates:

$$\Delta_R(\check{\tau}) = \mathrm{TPR}(\check{\tau}) - \gamma_R\,\pi = (1-\gamma_R)\,\mathrm{TPR}(\check{\tau}) - \gamma_R\,\mathrm{FNR}(\check{\tau}), \tag{29}$$

$$\Delta_P(\check{\tau},\hat{\tau}) = (1-\gamma_P)\,\mathrm{TPR}(\check{\tau}) - \gamma_P\,\mathrm{FPR}(\hat{\tau}). \tag{30}$$

For recall in Eq. (29), satisfying target $\gamma_R \approx 1$ depends on detecting rare false negatives; dividing by the slack scale $1-\gamma_R$ standardizes the true positive term to unit coefficient: $\mathrm{TPR}(\check{\tau}) - (\frac{\gamma_R}{1-\gamma_R})\,\mathrm{FNR}(\check{\tau}) \ge 0$. For precision in Eq. (30), the risk stems from false positives; dividing by γ_P standardizes the false positive term to unit coefficient: $(\frac{1-\gamma_P}{\gamma_P})\,\mathrm{TPR}(\check{\tau}) - \mathrm{FPR}(\hat{\tau}) \ge 0$.

The variance of a linear combination of estimators scales quadratically with its linear coefficients. Since every false negative is a positive ($\mathrm{FNR} \le \pi$), we assign the squared false-negative sensitivity to the prevalence term $\tilde{\pi}$ as a threshold-independent surrogate in $J(q)$. Reading off the squared sensitivity coefficients across the standardized constraints yields:

$$\lambda_{\text{TPR}} = 1 + \left(\frac{1-\gamma_P}{\gamma_P}\right)^2, \quad \lambda_{\text{FPR}} = 1, \quad \lambda_\pi = \left(\frac{\gamma_R}{1-\gamma_R}\right)^2.$$

Under these weights, sampling priority concentrates on discovering true positives for recall ($\lambda_{\text{TPR}} \approx 1$) and detecting false positives for

precision ($\lambda_{\text{FPR}} = 1$), while scaling up prevalence discovery under tight recall targets ($\gamma_R \to 1$).

B Additional Details on the Evaluation

This appendix provides comprehensive dataset statistics and complete empirical results complementing Sec. 4.

Benchmark Statistics. Table 4 summarizes all 42 benchmarks evaluated in our study, spanning 23 workloads across pre-LLM computer vision and NLP tasks [21], general text corpora [50], and technical documents [52]. Alongside workload size N, filtered record counts, and ground-truth positive prevalence π, we characterize proxy quality using Precision-Recall AUC (PR-AUC), calibration bias ($\mathrm{E}[\mathrm{A}(X) - Y]$), Expected Calibration Error (ECE, computed over 10 equal-width bins on $[0, 1]$), and score saturation rate (the fraction of records assigned exact scores of 0 or 1). Note that while cloud-hosted GPT-5.6-luna achieves higher PR-AUC than local Llama-3.2-1B across all text workloads, it frequently produces saturated confidence scores (up to 55.8% of records), motivating the tail-tolerance clipping in Sec. 3.6.

Soundness Audit. Table 5 disaggregates the statistical soundness audit from Sec. 4.2 (Table 2) across individual error budgets $\delta \in \{0.01, 0.05, 0.10\}$ and accuracy targets $\gamma \in \{0.70, 0.80, 0.90, 0.95\}$. Across all twelve (δ, γ) configurations, both ACTIS and ACTIS-U achieve 0.0% certified unsoundness, whereas LOTUS and ScaleDoc exhibit high failure rates that worsen as γ increases. The ablation of ACTIS without runtime safeguards incurs unsoundness on 2.4%–7.1% of benchmarks across all regimes, confirming that our finite-sample safeguards (Sec. 3.3) are essential for maintaining strict Type I error control under asymptotic testing.

Complete Performance Results. Table 6 reports the mean oracle call rate (OCR) and CPU tuning latency across all 42 benchmarks at $\gamma = 0.95$ and $\delta = 0.05$, expanding upon the representative subset in Table 3.

C Statistical Soundness of Prior Work

In this appendix, we briefly discuss potential statistical and algorithmic issues in existing semantic filter approximation methods that claim probabilistic accuracy guarantees: LOTUS [33], SUPG-IT [26], ScaleDoc [52], Stretto [38], and Clustering-Sampling-Voting (CSV) [11]. We emphasize that this discussion is non-exhaustive and reflects issues identified from a reading of the respective papers and, for the baselines evaluated in Sec. 4 (LOTUS and ScaleDoc), a cursory inspection of their released code. We have not attempted to catalog every possible issue or empirically isolate the causal impact of each individual factor on the failure rates observed in Sec. 4.2.

C.1 LOTUS

LOTUS [33] extends SUPG [21] to tune a two-threshold cascade for simultaneous precision and recall targets using a single importance-sampled pilot set of size s. While Patel et al. [33] state that the joint guarantee is achieved by splitting the failure probability ($\delta/2$ per metric) and applying normal confidence bounds with a Bonferroni correction over s, the paper does not provide a formal theorem or proof. In their tuning procedure and released implementation, several issues can undermine coverage: (i) thresholds are selected using asymptotic Gaussian confidence bounds with plug-in empirical Bernoulli standard deviations $\hat{\sigma}_k = \sqrt{\hat{\mu}_k(1 - \hat{\mu}_k)}$, which collapse to zero ($\hat{\sigma}_k = 0$) whenever the top-k highest-scoring tuning records are all positive ($\hat{\mu}_k = 1$), yielding a zero-width confidence interval that can select an overly aggressive upper threshold on a small streak of positives; (ii) the lower (recall) threshold is tuned via a two-step heuristic that first identifies an uncorrected empirical threshold $\check{\tau}_0$, computes normal bounds on the subsets above and below $\check{\tau}_0$ (scaling both standard errors by the total sample size $\sqrt{s}$ rather than the subset sizes) to form a "corrected" recall target, and re-scans for $\check{\tau}$ without uniform concentration across the candidate grid; and (iii) although the pilot sample is drawn from a non-uniform importance distribution biased toward high proxy scores, the precision tuning step in the released code computes unweighted empirical sums of the oracle labels, omitting the importance-sampling weights.

C.2 SUPG-IT

SUPG-IT [26] adapts LOTUS's two-threshold tuning algorithm to streaming environments, iteratively refining the cascade thresholds as data batches arrive rather than estimating them in a single pass. Although SUPG-IT aims to provide probabilistic guarantees, its statistical foundations inherit and compound the limitations of SUPG and LOTUS. Specifically, its bounds rely on asymptotic Gaussian approximations with plug-in variance estimates, employ an ad-hoc "target clipping" heuristic to bypass extreme values in small samples, and repeatedly apply fixed-sample confidence bounds to an adaptive, sequential thresholding process without time-uniform corrections for continuous monitoring.

C.3 ScaleDoc

ScaleDoc [52] tunes a two-threshold cascade to meet a target F1 score using proportional stratified sampling and histogram smoothing, and claims a finite-sample $1 - \delta$ guarantee based on Bernstein's inequality (see their Proposition 1). However, the theoretical analysis contains several statistical flaws: the bound fails to correct for multiple comparisons across the threshold search space, applies standard i.i.d. concentration inequalities to non-i.i.d. stratified samples, analyzes raw empirical indicator means whereas the actual algorithm relies on smoothed (and jittered) score histograms, and treats population variances as known without accounting for the uncertainty of variance estimation. Furthermore, neither the threshold-selection algorithm (Algorithm 2 in Zhang et al. [52]) nor the released code takes δ as an input or subtracts the derived Bernstein safety margin ϵ; instead, the authors informally argue that the one-bin discretization step in their search acts as a conservative buffer to cover ϵ, even though a fixed bin width does not scale with sample size, label variance, or δ.

C.4 Stretto

Stretto [38] considers a broader space of filter execution plans—jointly tuning model size, KV-cache compression ratio, and cascade thresholds—by optimizing execution cost over a continuous relaxation of the plan space via gradient descent, subject to precision and recall constraints. Noting that frequentist confidence bounds are difficult to construct over a large plan space, the authors use

Table 4: Summary statistics across all 42 benchmarks: workload size (N), records excluded by API content filters (Missing), ground-truth positive prevalence (Pos. rate), area under the proxy precision-recall curve (PR-AUC), mean calibration bias $\mathrm{E}[\mathrm{A}(X) - Y]$ (Cal. Bias), expected calibration error over 10 equal-width bins (ECE), and fraction of records with saturated proxy scores in $\{0, 1\}$ (Sat. Rate).

Dataset	N	Query	Missing	Pos. rate π	Proxy Model	PR-AUC	Cal. Bias	ECE	Sat. Rate
Onto	11,165	—	0	0.025	UFT [4]	0.830	+0.008	0.009	0.000
ImageNet	50,000	—	0	0.001	ResNet-50 [10]	0.982	+0.000	0.000	0.000
Jackson	973,085	—	0	0.292	ResNet-50 [10]	0.880	+0.017	0.023	0.007
TACRED	22,631	—	0	0.024	SpanBERT [17]	0.794	+0.001	0.007	0.000
Court	1,000	—	4	0.650	GPT-5.6-luna	0.897	-0.057	0.201	0.255
					Llama-3.2-1B	0.645	+0.292	0.292	0.000
Review	10,000	—	8	0.059	GPT-5.6-luna	0.823	+0.035	0.042	0.509
					Llama-3.2-1B	0.124	+0.618	0.618	0.000
Screenplay	2,858	—	4	0.865	GPT-5.6-luna	0.965	+0.058	0.086	0.082
					Llama-3.2-1B	0.886	+0.085	0.104	0.000
Wiki	10,000	—	1	0.342	GPT-5.6-luna	0.868	-0.166	0.173	0.243
					Llama-3.2-1B	0.362	+0.289	0.289	0.000
PubMed	10,000	0_{ext}	0	0.006	GPT-5.6-luna	0.024	-0.005	0.006	0.173
					Llama-3.2-1B	0.004	+0.503	0.503	0.000
		1_{ext}	0	0.615	GPT-5.6-luna	0.999	+0.055	0.055	0.416
					Llama-3.2-1B	0.867	+0.032	0.138	0.000
		2_{ext}	0	0.584	GPT-5.6-luna	0.996	-0.012	0.020	0.160
					Llama-3.2-1B	0.731	-0.177	0.177	0.000
		0	0	0.790	GPT-5.6-luna	0.987	+0.043	0.060	0.460
					Llama-3.2-1B	0.932	-0.285	0.285	0.000
		4	0	0.680	GPT-5.6-luna	0.991	+0.054	0.065	0.422
					Llama-3.2-1B	0.884	-0.022	0.066	0.000
		10	0	0.122	GPT-5.6-luna	0.995	+0.010	0.011	0.558
					Llama-3.2-1B	0.895	+0.084	0.117	0.000
BigPatent	10,000	0_{ext}	0	0.121	GPT-5.6-luna	0.988	+0.007	0.012	0.512
					Llama-3.2-1B	0.714	+0.019	0.091	0.000
		1_{ext}	0	0.676	GPT-5.6-luna	0.966	-0.243	0.243	0.061
					Llama-3.2-1B	0.723	-0.366	0.366	0.000
		2_{ext}	0	0.379	GPT-5.6-luna	0.952	-0.030	0.080	0.307
					Llama-3.2-1B	0.575	-0.196	0.196	0.000
		0	0	0.231	GPT-5.6-luna	0.986	+0.022	0.028	0.477
					Llama-3.2-1B	0.840	-0.126	0.126	0.000
		4	0	0.061	GPT-5.6-luna	0.214	-0.042	0.058	0.442
					Llama-3.2-1B	0.052	+0.069	0.085	0.000
		10	0	0.056	GPT-5.6-luna	0.936	-0.013	0.014	0.556
					Llama-3.2-1B	0.282	+0.088	0.088	0.000
GovReport	10,000	0	2	0.207	GPT-5.6-luna	0.982	+0.023	0.027	0.492
					Llama-3.2-1B	0.723	+0.114	0.136	0.000
		4	2	0.199	GPT-5.6-luna	0.837	+0.116	0.136	0.156
					Llama-3.2-1B	0.375	+0.281	0.281	0.000
		10	2	0.146	GPT-5.6-luna	0.905	-0.009	0.040	0.437
					Llama-3.2-1B	0.575	+0.198	0.198	0.000

Table 5: Statistical soundness across 42 benchmarks disaggregated by error budget δ and accuracy target γ. Using 95% Clopper-Pearson confidence intervals $[\tilde{\delta}_{\text{LB}}, \tilde{\delta}_{\text{UB}}]$ over 1,000 trials, we report the percentage of benchmarks certified *sound* ($\tilde{\delta}_{\text{UB}} < \delta$), *inconclusive* ($\delta \in [\tilde{\delta}_{\text{LB}}, \tilde{\delta}_{\text{UB}}]$), or *unsound* (in red; $\tilde{\delta}_{\text{LB}} > \delta$). $\tilde{\delta}_{\text{unsound}}$ and $\tilde{\delta}_{\max}$ denote the mean empirical failure rate over certified unsound benchmarks and the maximum empirical failure rate across all benchmarks, respectively.

			Percentage Certified				
δ	γ	Method	Sound	Inconcl.	Unsound	$\tilde{\delta}_{\text{unsound}}$	$\tilde{\delta}_{\max}$
0.01	0.70	ACTIS	100.0	0.0	0.0	—	0.000
		ACTIS (w/o safeguards)	97.6	0.0	2.4	0.069	0.069
		ACTIS-U	100.0	0.0	0.0	—	0.001
		BARGAIN	85.7	14.3	0.0	—	0.005
		ScaleDoc	69.0	4.8	26.2	0.345	1.000
		LOTUS	28.6	0.0	71.4	0.792	1.000
	0.80	ACTIS	100.0	0.0	0.0	—	0.003
		ACTIS (w/o safeguards)	97.6	0.0	2.4	0.129	0.129
		ACTIS-U	100.0	0.0	0.0	—	0.001
		BARGAIN	88.1	11.9	0.0	—	0.006
		ScaleDoc	54.8	4.8	40.5	0.329	0.999
		LOTUS	21.4	0.0	78.6	0.869	1.000
	0.90	ACTIS	100.0	0.0	0.0	—	0.001
		ACTIS (w/o safeguards)	95.2	0.0	4.8	0.083	0.146
		ACTIS-U	100.0	0.0	0.0	—	0.000
		BARGAIN	90.5	9.5	0.0	—	0.006
		ScaleDoc	31.0	2.4	66.7	0.336	1.000
		LOTUS	7.1	7.1	85.7	0.866	1.000
	0.95	ACTIS	97.6	2.4	0.0	—	0.004
		ACTIS (w/o safeguards)	85.7	9.5	4.8	0.157	0.217
		ACTIS-U	100.0	0.0	0.0	—	0.000
		BARGAIN	100.0	0.0	0.0	—	0.003
		ScaleDoc	21.4	0.0	78.6	0.317	0.934
		LOTUS	4.8	4.8	90.5	0.841	1.000
0.05	0.70	ACTIS	100.0	0.0	0.0	—	0.030
		ACTIS (w/o safeguards)	97.6	0.0	2.4	0.102	0.102
		ACTIS-U	100.0	0.0	0.0	—	0.002
		BARGAIN	100.0	0.0	0.0	—	0.020
		ScaleDoc	76.2	7.1	16.7	0.516	1.000
		LOTUS	28.6	0.0	71.4	0.810	1.000
	0.80	ACTIS	97.6	2.4	0.0	—	0.043
		ACTIS (w/o safeguards)	97.6	0.0	2.4	0.160	0.160
		ACTIS-U	100.0	0.0	0.0	—	0.001
		BARGAIN	100.0	0.0	0.0	—	0.027
		ScaleDoc	61.9	9.5	28.6	0.450	0.999
		LOTUS	21.4	2.4	76.2	0.904	1.000
	0.90	ACTIS	100.0	0.0	0.0	—	0.007
		ACTIS (w/o safeguards)	95.2	2.4	2.4	0.173	0.173
		ACTIS-U	100.0	0.0	0.0	—	0.002
		BARGAIN	100.0	0.0	0.0	—	0.030
		ScaleDoc	42.9	7.1	50.0	0.436	1.000
		LOTUS	16.7	0.0	83.3	0.908	1.000
	0.95	ACTIS	100.0	0.0	0.0	—	0.007
		ACTIS (w/o safeguards)	92.9	0.0	7.1	0.174	0.232
		ACTIS-U	100.0	0.0	0.0	—	0.002
		BARGAIN	100.0	0.0	0.0	—	0.022
		ScaleDoc	26.2	9.5	64.3	0.376	0.934
		LOTUS	7.1	2.4	90.5	0.862	1.000
0.10	0.70	ACTIS	100.0	0.0	0.0	—	0.061
		ACTIS (w/o safeguards)	97.6	0.0	2.4	0.125	0.125
		ACTIS-U	100.0	0.0	0.0	—	0.004
		BARGAIN	100.0	0.0	0.0	—	0.039
		ScaleDoc	85.7	0.0	14.3	0.591	1.000
		LOTUS	28.6	4.8	66.7	0.871	1.000
	0.80	ACTIS	100.0	0.0	0.0	—	0.065
		ACTIS (w/o safeguards)	97.6	0.0	2.4	0.186	0.186
		ACTIS-U	100.0	0.0	0.0	—	0.002
		BARGAIN	100.0	0.0	0.0	—	0.048
		ScaleDoc	73.8	2.4	23.8	0.522	0.999
		LOTUS	23.8	0.0	76.2	0.909	1.000
	0.90	ACTIS	100.0	0.0	0.0	—	0.041
		ACTIS (w/o safeguards)	97.6	0.0	2.4	0.190	0.190
		ACTIS-U	100.0	0.0	0.0	—	0.003
		BARGAIN	100.0	0.0	0.0	—	0.049
		ScaleDoc	52.4	4.8	42.9	0.492	1.000
		LOTUS	16.7	0.0	83.3	0.916	1.000
	0.95	ACTIS	100.0	0.0	0.0	—	0.009
		ACTIS (w/o safeguards)	92.9	2.4	4.8	0.264	0.283
		ACTIS-U	100.0	0.0	0.0	—	0.004
		BARGAIN	100.0	0.0	0.0	—	0.039
		ScaleDoc	38.1	7.1	54.8	0.426	0.934
		LOTUS	7.1	2.4	90.5	0.874	1.000

marginal Bayesian credible intervals to lower-bound precision and recall. However, the use of marginal Bayesian credible intervals does not circumvent the need to correct for adaptive selection over the search space: evaluating the final execution plan on the same sample used for gradient-based optimization introduces optimism (selection) bias into the resulting guarantees.

C.5 Clustering-Sampling-Voting (CSV)

Departing from model cascades, CSV [11] groups tuples in an embedding space, queries the oracle on a sampled subset of each cluster, and assigns labels to the remaining items via majority voting, dynamically triggering recursive re-clustering when sampled labels within a cluster exhibit high variance. Beyond only aiming to bound per-cluster accuracy rather than dataset-wide precision or recall, its theoretical analysis contains several statistical errors: the proofs misapply fixed-sample concentration inequalities to random stopping times, ignore the adaptivity of recursive re-clustering (which invalidates the bounds by conditioning sub-populations on previously observed labels), and omit a union bound to control the global failure probability across the partitioned dataset.

Table 6: Oracle call rate (%) and tuner latency (ms) across all benchmarks for precision/recall target $\gamma = 0.95$ and error budget $\delta = 0.05$. Red* indicates certified unsoundness; boldface highlights the best result among statistically sound methods.

Benchmark			Oracle Call Rate (%)						Tuner Latency (ms)				
Workload	**Proxy**	**N**	**Ideal**	**ACTIS**	**ACTIS-U**	**BARGAIN**	**ScaleDoc**	**LOTUS**	**ACTIS**	**ACTIS-U**	**BARGAIN**	**ScaleDoc**	**LOTUS**
Onto	UFT	11k	6.2	**32.1**	61.3	62.6	28.4*	63.2*	17.2	42.6	**11.2**	4.4*	1.6*
ImageNet	ResNet-50	50k	0.0	19.6	100.0	100.0	10.1	**5.1**	48.1	191.6	123.2	7.1	**6.2**
Jackson	ResNet-50	973k	35.4	**45.0**	46.4	81.5	58.7	25.7*	3854.3	10110.6	3580.0	**109.6**	171.3*
TACRED	SpanBERT	23k	6.5	**32.1**	54.4	51.3	43.5*	35.9*	43.7	69.1	**22.2**	5.5*	3.0*
Court	GPT-5.6-luna	1k	44.5	**80.2**	88.1	94.0	60.8*	22.3*	3.6	5.6	**2.2**	3.3*	0.6*
	Llama-3.2-1B	1k	85.2	**96.8**	97.6	98.6	95.4*	67.3*	6.1	8.0	**2.6**	4.3*	0.7*
Review	GPT-5.6-luna	10k	10.2	**32.8**	51.7	65.3	22.8*	7.4*	**10.7**	18.4	12.3	4.3*	1.6*
	Llama-3.2-1B	10k	89.0	98.1	**96.9**	97.9	94.6*	77.2*	25.4	47.0	**17.1**	4.7*	1.6*
Screenplay	GPT-5.6-luna	3k	15.3	**49.7**	53.3	93.4	73.0*	34.0*	13.8	23.1	**6.5**	5.1*	1.1*
	Llama-3.2-1B	3k	53.9	**84.1**	85.5	97.5	94.8	52.0*	44.2	87.2	4.1	**3.9**	0.9*
Wiki	GPT-5.6-luna	10k	37.0	71.1	**68.6**	83.1	82.3*	26.1*	69.5	129.1	**13.7**	3.4*	1.6*
	Llama-3.2-1B	10k	91.7	**97.0**	97.2	98.0	95.2*	55.6*	45.5	88.3	**20.5**	5.4*	2.4*
PubMed Q0	GPT-5.6-luna	10k	0.0	**15.3**	15.8	24.8	42.3	9.4*	**3.9**	4.5	5.2	6.2	1.8*
	Llama-3.2-1B	10k	37.7	61.6	62.1	92.1	**56.1**	48.8*	85.6	214.4	27.2	**5.0**	1.7*
PubMed $Q0_{ext}$	GPT-5.6-luna	10k	100.0	100.0	100.0	100.0	**94.9**	55.9*	14.1	36.2	20.3	**4.1**	1.8*
	Llama-3.2-1B	10k	97.9	**100.0**	**100.0**	**100.0**	98.5*	89.5*	21.0	48.3	**19.8**	5.0*	1.7*
PubMed $Q1_{ext}$	GPT-5.6-luna	10k	0.0	11.4	10.0	16.1	29.7	**9.4**	3.9	3.9	2.6	5.1	**1.7**
	Llama-3.2-1B	10k	50.1	71.9	**71.8**	85.1	73.8	67.4*	77.2	174.7	13.3	**5.2**	1.8*
PubMed $Q2_{ext}$	GPT-5.6-luna	10k	0.0	10.7	11.1	16.5	**5.6**	9.3*	7.1	12.8	**3.1**	6.5	1.8*
	Llama-3.2-1B	10k	76.5	**89.2**	89.8	95.7	89.6	57.1*	67.9	150.5	19.6	**5.2**	1.9*
PubMed Q4	GPT-5.6-luna	10k	0.0	**15.2**	15.4	23.1	38.9	9.4*	**4.2**	5.1	4.9	5.8	1.8*
	Llama-3.2-1B	10k	51.1	71.8	71.8	90.6	**70.9**	61.3*	83.6	188.2	19.9	**5.2**	1.8*
PubMed Q10	GPT-5.6-luna	10k	0.0	**11.6**	24.8	19.6	9.7*	7.5*	3.5	3.5	**2.7**	4.8*	1.6*
	Llama-3.2-1B	10k	13.2	57.3	**50.1**	64.6	36.0*	61.1	68.0	37.2	12.1	5.5*	**1.7**
BigPatent Q0	GPT-5.6-luna	10k	1.8	**15.6**	21.8	22.6	20.6	8.6*	**3.7**	4.2	5.3	6.4	2.3*
	Llama-3.2-1B	10k	30.1	52.8	53.5	64.0	61.6	**46.2**	36.4	56.2	14.5	5.3	**1.9**
BigPatent $Q0_{ext}$	GPT-5.6-luna	10k	0.7	**14.3**	26.5	21.2	11.7*	7.6*	**3.4**	3.8	4.9	6.2*	2.4*
	Llama-3.2-1B	10k	60.2	85.9	**84.8**	94.0	86.4	71.6*	69.3	113.9	18.9	**5.4**	1.9*
BigPatent $Q1_{ext}$	GPT-5.6-luna	10k	18.5	43.6	**41.0**	56.8	52.4	9.2*	43.5	79.2	14.7	**4.7**	2.6*
	Llama-3.2-1B	10k	80.3	90.8	91.2	98.0	**89.8**	39.9*	72.1	161.2	20.8	**5.5**	1.9*
BigPatent $Q2_{ext}$	GPT-5.6-luna	10k	17.0	**36.7**	38.4	44.9	65.0*	9.0*	11.3	19.4	**10.5**	5.2*	2.6*
	Llama-3.2-1B	10k	80.9	**90.9**	91.4	94.5	92.7	54.3*	56.0	112.8	19.7	**5.2**	1.9*
BigPatent Q4	GPT-5.6-luna	10k	99.1	100.0	99.7	**99.4**	98.9*	79.1*	**21.2**	34.3	24.3	5.1*	2.5*
	Llama-3.2-1B	10k	95.2	99.1	99.1	**99.0**	98.6*	72.0*	29.2	50.8	**20.4**	5.3*	1.8*
BigPatent Q10	GPT-5.6-luna	10k	2.6	**53.0**	66.1	78.3	85.1*	5.3*	**17.6**	26.5	19.2	5.3*	2.7*
	Llama-3.2-1B	10k	68.8	90.7	**90.3**	92.2	85.7*	83.2*	38.1	55.7	**19.1**	5.6*	1.8*
GovReport Q0	GPT-5.6-luna	10k	2.0	**15.6**	23.2	21.3	18.9	8.5*	4.9	6.9	**4.5**	4.9	1.8*
	Llama-3.2-1B	10k	65.9	**84.0**	84.3	90.3	79.0*	54.8*	70.7	117.1	**17.4**	5.4*	2.0*
GovReport Q4	GPT-5.6-luna	10k	28.1	**49.0**	53.5	62.5	41.2*	15.7*	14.8	21.8	**12.0**	4.8*	1.9*
	Llama-3.2-1B	10k	83.3	93.1	93.3	95.7	**89.8**	63.0*	50.4	78.5	18.2	**5.3**	2.0*
GovReport Q10	GPT-5.6-luna	10k	12.5	**35.6**	43.1	52.7	77.4*	7.9*	13.3	17.9	**10.2**	4.6*	1.9*
	Llama-3.2-1B	10k	62.6	84.1	**83.9**	86.9	75.4*	62.8*	58.3	81.5	**16.7**	5.6*	2.0*