Cut Costs, Not Accuracy: LLM-Powered Data Processing with Guarantees

SEPANTA ZEIGHAMI, UC Berkeley, USA
SHREYA SHANKAR, UC Berkeley, USA
ADITYA PARAMESWARAN, UC Berkeley, USA

Large Language Models (LLMs) are being increasingly used as a building block in data systems to process large text datasets. To do so, LLM model providers offer multiple LLMs with different sizes, spanning various cost-quality trade-offs when processing text at scale. Top-of-the-line LLMs (e.g., GPT-4o, Claude Sonnet) operate with high accuracy but are prohibitively expensive when processing many records. To avoid high costs, more affordable but lower quality LLMs (e.g., GPT-4o-mini, Claude Haiku) can be used to process records, but we need to ensure that the overall accuracy does not deviate substantially from that of the top-of-the-line LLMs. The model cascade framework provides a blueprint to manage this trade-off, by using the confidence of LLMs in their output (e.g., log-probabilities) to decide on which records to use the affordable LLM. However, existing solutions following this framework provide only marginal cost savings and weak theoretical guarantees because of poor estimation of the quality of the affordable LLM's outputs. We present BARGAIN, a method that judiciously uses affordable LLMs in data processing to significantly reduce cost while providing strong theoretical guarantees on the solution quality. BARGAIN employs a novel adaptive sampling strategy and statistical estimation procedure that uses data and task characteristics and builds on recent statistical tools to make accurate estimations with tight theoretical guarantees. Variants of BARGAIN can support guarantees on accuracy, precision, or recall of the output. Experimental results across 8 real-world datasets show that BARGAIN reduces cost, on average, by up to 86% more than state-of-the-art, while providing stronger theoretical guarantees on accuracy of output, with similar gains when guaranteeing a desired level of precision or recall.

CCS Concepts: • **Information systems**;

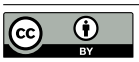
Additional Key Words and Phrases: LLM-Powered Data Processing, AI, Statistical Guarantees

ACM Reference Format:
Sepanta Zeighami, Shreya Shankar, and Aditya Parameswaran. 2025. Cut Costs, Not Accuracy: LLM-Powered Data Processing with Guarantees. *Proc. ACM Manag. Data* 3, 6 (SIGMOD), Article 311 (December 2025), 26 pages. https://doi.org/10.1145/3769776

1 Introduction

LLMs are being increasingly used as a building block in data systems that process large text datasets for tasks such as extraction, filtering, summarization, and question answering [1, 3, 27, 30, 32, 35, 37]. For example, given a set of legal contracts, a lawyer might want to find those that relate to a specific law. To do so, a system can iterate over the contracts and, for each contract, ask an LLM to decide if it relates to the specific law. To obtain the most accurate results possible, users want to use *top-of-the-line LLMs* (e.g., GPT-4o, Claude Sonnet). However, such LLMs are prohibitively expensive at scale. Even a single scan of a few thousand-page long documents by an LLM can cost hundreds of dollars—and gets more expensive with more documents or when multiple scans are needed to address complex or iterative information needs. To support users with budgetary constraints, LLM

Authors' Contact Information: Sepanta Zeighami, UC Berkeley, USA, zeighami@berkeley.edu; Shreya Shankar, UC Berkeley, USA, shreyashankar@berkeley.edu; Aditya Parameswaran, UC Berkeley, USA, adityagp@berkeley.edu.



ACM 2836-6573/2025/12-ART311
https://doi.org/10.1145/3769776

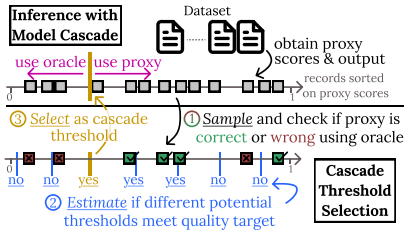


Fig. 1. Overview of Model Cascade

Method	Quality Guarantee	Sample Effectiveness	Threshold Est. Accuracy
SUPG	Asymptotic	Medium	Medium
Naive	Yes	Low	Low
BARGAIN	**Yes**	**High**	**High**

Table 1. Overview of Approaches

companies often also provide *affordable LLMs* (e.g., GPT-4o-mini, Claude Haiku) that are much cheaper but can be less accurate; for instance, GPT-4o-mini is more than 15× cheaper than GPT-4o [31]. However, using the affordable model can reduce answer quality relative to the top-of-the-line model. In such cases, users can often tolerate some marginal quality degradation as long as we can reduce cost substantially, e.g., if the system is guaranteed to match the top-of-the-line LLM's output 90% of the time but at half the cost. The system then needs to decide when to use which LLM to minimize cost while guaranteeing this desired answer quality.

A common paradigm to decide when to use the affordable model and when the more expensive model is *model cascades* [2, 19, 22, 23, 34]. In this paradigm, the more expensive LLM is called the *oracle* while the affordable LLM is called the *proxy*. In model cascades, we additionally have access to *proxy scores* which quantify, for each data record, how confident the proxy is in its output[1]. Proxy scores help decide whether to use the proxy or the oracle to process a record based on a *cascade threshold*, ρ (see Fig. 1, top): the proxy model answer is used for records whose proxy score is more than ρ, and the remaining records are processed with the oracle. The smaller the cascade threshold is, the more frequently the proxy is used to process the records. This means cheaper data processing at potentially worse quality, leading to a classic cost/quality trade off.

A central problem in model cascades is setting the cascade threshold based on users' desired answer quality. In the general case, users provide an *accuracy target*, e.g., outputs should match the oracle 90% of the time, while minimizing the number of oracle invocations. We refer to this problem as *accuracy target (or AT) queries*, where the number of oracle invocations avoided is the *utility* of the approach. Alternatively, in filtering tasks, users may be interested in a desired *recall target* while maximizing precision, or a *precision target* while maximizing recall, both settings introduced by Kang et al. [21], and respectively referred to as RT and PT Queries. We refer to the achieved precision for RT queries and recall for PT queries as the utility for these two queries. Moreover, we collectively refer to the user-specified accuracy, precision, and recall targets in AT, PT, and RT queries as *quality targets*. Similar to [21], the user provides a failure probability, δ, and algorithms must meet quality targets with probability at least $1-\delta$ while maximizing utility. We focus on using LLMs for classification, specifically multiclass classification for AT queries and binary classification for PT and RT queries.

To solve any of AT, PT and RT queries, a common approach [18, 19, 21, 28, 32] is to (1) *sample* and label a subset of records using the oracle, (2) *estimate* if various cascade thresholds meet the desired quality target and (3) *select* a cascade threshold among the ones estimated to meet the quality target, as illustrated in Fig. 1 (bottom). However, most existing methods [18, 19, 28] do not provide *any* guarantees on meeting the quality target. Such methods miss the quality target frequently; e.g., [19, 28] frequently achieve precision below 65% when given a target of 90% [21]. To provide some guarantees, SUPG [21] (also used by [32]), employs Central Limit Theorem (CLT) to estimate if different thresholds meet the quality target from labeled samples. Due to the use of

[1]Proxy scores are provided as part of prediction, e.g., log-probability of LLM outputs

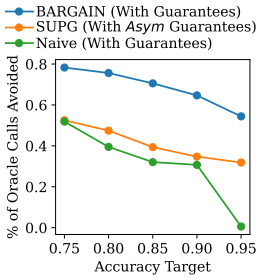


Fig. 2. Summary of AT Query Results

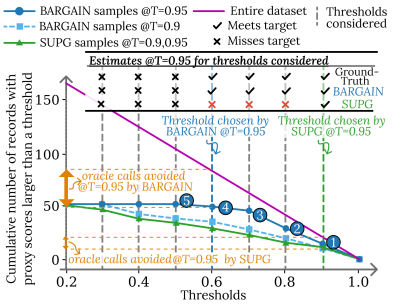


Fig. 3. Example BARGAIN workflow for AT queries

CLT, SUPG meets the quality target *only asymptotically as sample size goes to infinity*. Since the cascade threshold is usually set with small sample sizes, there are cases where SUPG frequently misses the quality target, like other work that don't claim to provide guarantees. Besides weak guarantees, SUPG relies on worst-case analysis that ignores data characteristics. This leads to *inaccurate threshold estimates* that may incorrectly exclude high-utility thresholds that meet the target yielding poor utility. This problem is compounded by SUPG's use of importance sampling (sampling based on proxy scores) that ignores the quality target and label distribution. Ignoring such characteristics leads to *ineffective samples* that don't provide useful information for identifying high-utility and high-quality thresholds.

In this paper, we present *BARGAIN*, an approach for *AI-powered data processing using model cascade with tight theoretical guarantees*. We present novel theoretical and algorithmic insights that use *data and task characteristics* to *sample records effectively* and make *accurate threshold estimations*, thus improving utility while providing rigorous theoretical guarantees. The benefits of BARGAIN are summarized in Table 1, which also shows a Naive approach that achieves the same strong theoretical guarantees as BARGAIN, but through black-box application of statistical tools (i.e., uniform sampling and Hoeffding's inequality) without considering data and task characteristics. To convey a sense of benefits, Fig. 2 shows that BARGAIN provides significantly better utility compared with both the Naive approach and SUPG (results averaged over eight real-world datasets, see Sec. 6). For AT queries, **BARGAIN reduces oracle usage by up to 86% more than SUPG**. Similarly, **BARGAIN improves recall in PT queries by up to 118% and precision in RT queries by up to 19% over SUPG**. BARGAIN achieves these empirical benefits **while providing rigorous theoretical guarantees that SUPG lacks**. This rigor is significant since our results show cases where SUPG misses the target more than 75% of the time when given a failure probability of only 10%.

BARGAIN Overview. We use Fig. 3 to discuss how BARGAIN achieves these benefits on a toy dataset for AT queries (other queries are similar). The blue and green lines in the figure (ignore the large numbered circles, pink line, and orange markings for now) show, for each proxy score, the cumulative number of the samples taken with scores greater than that proxy score by BARGAIN and SUPG respectively, and for accuracy targets $T = 0.9$ and $T = 0.95$. Both approaches take a total of 50 samples in all settings (as shown by the y-intercept), and SUPG has the same distribution of samples for both $T = 0.9$ and $T = 0.95$. The figure also shows the cascade thresholds chosen by the approaches using the samples for $T = 0.95$: 0.6 by BARGAIN and 0.9 by SUPG. The cascade thresholds are chosen based on each approach's estimates, shown in the table embedded in Fig. 3. SUPG wrongly estimates that thresholds between 0.6-0.8 miss the target, but BARGAIN correctly identifies that those thresholds meet the target. Making correct estimates enables BARGAIN to find a better cascade threshold and improve utility. To see the utility, consider the pink line that shows the cumulative number of records in the entire dataset with scores greater than a given threshold.

Note that all samples taken by the approaches are labeled by the oracle. Thus, the *total number of oracle calls avoided* by each approach is the number of records in the entire dataset with a proxy score more than the cascade threshold that are not already sampled. In Fig.3, we visualize this as the difference between the number of cumulative records (pink line) and the cumulative number of samples taken (blue or green line respectively) marked by the orange double arrows. SUPG's cascade threshold results in almost all records being processed by the oracle. BARGAIN instead avoids a significant number of oracle calls.

BARGAIN achieves this, in part, by using adaptive sampling (Step 1 in Fig. 1) to ensure *high sample effectiveness*. BARGAIN iterates through different candidate thresholds (in decreasing order) and, for each threshold, samples sufficient records to accurately estimate if it meets the target before moving to the next. BARGAIN stops sampling once it reaches thresholds estimated to miss the target. This is shown in Fig. 3 for $T = 0.95$ where circles show the iteration at which a threshold was considered. In contrast with SUPG [21] that samples records based on their proxy scores independent of T, BARGAIN's sampling adapts to the label distribution and quality target. In Fig. 3, for $T = 0.95$ (dark blue line), BARGAIN takes enough samples with proxy score larger than 0.8 to correctly estimate that threshold 0.8 meets the target, while SUPG takes fewer such samples and ends up making an incorrect estimate. Both SUPG and BARGAIN sample the same number of records in total (50), but SUPG takes many uninformative samples with proxy scores less than 0.6 that aren't useful for estimating the quality of thresholds larger than 0.6. Fig. 3 also shows samples taken by BARGAIN when $T = 0.9$ (dashed blue line). BARGAIN adapts to this lower target.

Furthermore, BARGAIN's estimation and selection approach (Steps 2 & 3 in Fig. 1) is optimized based on task and data characteristics to achieve *high threshold estimation accuracy* when using the samples. We show that the variance of our samples decreases as the quality of a threshold increases; we take advantage of this low variance to improve our estimation significantly for high-quality thresholds. We use a hypothesis-testing formulation for estimating whether a threshold meets a target and solve it through recent statistical results by Waudby-Smith and Ramdas [42] to obtain significantly more accurate estimates than the Naive approach that uses Hoeffding's inequality [12] with the same guarantees—unlike SUPG's use of CLT with weaker guarantees. Furthermore, we perform an in-depth analysis of statistical events during threshold selection to incorporate useful dataset characteristics into the analysis rather than relying on loose worst-case bounds. This analysis with corresponding modifications to the threshold selection algorithm helps BARGAIN significantly improve utility on real-world datasets. We apply the above intuitions separately to each of AT, PT, and RT queries. For RT queries, we prove a negative result that when the number of true positives on a dataset is small, no approach can achieve high utility while guaranteeing it meets the target. In light of this result, to achieve high utility, we allow users to opt for a relaxation of our quality guarantee designed to match the original guarantees in realistic settings but not in worst-case datasets.

Contributions. To summarize, our contributions are as follows.

- We present BARGAIN, the first practical method to perform AT, PT and RT queries with rigorous theoretical guarantees.
- We show how to perform task and data-aware sampling, estimation, and threshold selection, each designed to take advantage of the characteristics of different quality metrics and the dataset to provide high utility.
- We present tight theoretical analyses for our algorithms, presenting novel theoretical insights and showing how recent statistical tools can be used in the analysis.

- We perform extensive empirical evaluation, showing that BARGAIN reduces oracle usage by up to 86% for AT queries, improves recall by 118% for PT queries, and precision by 19% for RT queries over the state-of-the-art, SUPG.

We present the necessary background in Sec. 2, BARGAIN for PT queries in Sec. 3, and other queries in Sec. 4, with Sec. 5 discussing how to set the parameters of the methods. We present our empirical results in Sec. 6, and present related work in Sec. 7.

2 Background

2.1 Problem Definition

Setup. Consider a dataset $D = \{x_1, ..., x_n\}$ containing n records. The user wants to process these records with an expensive AI model, e.g., to apply the same prompt to each document using an LLM such as GPT-4o. We call this expensive AI model an oracle $\mathcal{O}$, and let $\mathcal{O}(x_i)$ denote its output for the i-th record, given a fixed prompt. The user also has access to a cheaper AI model, e.g., a smaller LLM such as GPT-4o-mini. We call this model a proxy model $\mathcal{P}$, where $\mathcal{P}(x_i)$ is the output of this model on the i-th record, given a fixed prompt (we discuss extensions to multiple proxies in our technical report [44]). If the user is interested in filtering (or binary classification of the records in) the dataset, then the outputs of both models are in $\{0, 1\}$; we refer to 0 and 1 as the negative and positive class, respectively. In general, model outputs can be arbitrary.

For now, we assume the oracle is expensive while the cost of the proxy is negligible compared to the oracle (our technical report [44] discusses extensions when considering the proxy cost). This often holds in practice where there is more than an order of magnitude cost difference between large and small LLMs (such as for OpenAI and Claude models [31]). Ideally, the user wants to process the dataset D using the oracle, but performing oracle invocations on the entire dataset is expensive. Instead, the user can tolerate some deviation in output quality relative to the oracle to reduce cost, as long as the output does not deviate *too much* from the oracle. We consider accuracy, precision, or recall quality guarantees on the output.

Accuracy Target Queries. In Accuracy Target (AT) queries, the user specifies a desired accuracy target, and the goal is to minimize the number of times the oracle is used while meeting the desired accuracy target. Formally, consider an algorithm A that processes records in D and provides an answer $\hat{y}_i$ for the i-th record x_i. $\hat{y}_i$ is either $\mathcal{P}(x_i)$ or $\mathcal{O}(x_i)$, depending on whether A uses the proxy or the oracle on x_i. Let C be the number of records where A uses the oracle; C is the *cost* of A. Furthermore, consider the answer set $\hat{Y} = \{\hat{y}_1, ..., \hat{y}_n\}$, and define its accuracy as $\mathcal{A}(\hat{Y}) = \sum_{i \in [n]} \frac{\mathbb{I}[\mathcal{O}(x_i) = \hat{y}_i]}{n}$. Then, given an accuracy target T, and failure probability δ, an AT query is the problem of returning an answer set $\hat{Y}$ using minimum number of oracle calls, C, while guaranteeing the accuracy target is met with probability at least $1 - \delta$, that is, $\mathbb{P}(\mathcal{A}(\hat{Y}) \geq T) \geq 1 - \delta$, where the probability is over runs of the algorithm.

Precision/Recall Target Queries. Precision Target (PT) queries and Recall Target (RT) queries were formalized by [21] as follows. In Precision (resp., Recall) Target queries, the user specifies a precision (resp., recall) target and an oracle budget, and our algorithm needs to meet the precision (resp., recall) target and maximize recall (resp., precision). Here, unlike AT queries, the oracle budget is fixed, and the goal is to maximize recall given precision (or vice versa). PT/RT queries only apply to binary classification or filtering where $\mathcal{O}(x_i) \in \{0, 1\}$ for all $i \in [n]$; so our goal is to find the subset of D with positive labels. Formally, given an oracle budget k, consider an algorithm A that performs at most k oracle calls and returns a set of data indexes $\hat{Y} = \{i_1, ..., i_r\}$ for some integer r, where $i_j \in [n]$. $\hat{Y}$ is the set of indexes of records in D that A estimated to be labeled positive.

Precision and recall of this set are defined, respectively, as

$$\mathcal{P}(\hat{Y}) = \sum_{i \in \hat{Y}} \frac{\mathcal{O}(x_i)}{r}, \text{ where } r = |\hat{Y}|, \text{ and}$$

$$\mathcal{R}(\hat{Y}) = \sum_{i \in \hat{Y}} \frac{\mathcal{O}(x_i)}{n^+}, \text{ where } n^+ = \sum_{j \in [n]} \mathcal{O}(x_j).$$

Then, given an oracle budget k, precision target T, and probability of failure δ, a PT query is the problem of returning an answer set $\hat{Y}$ that maximizes recall, $\mathcal{R}(\hat{Y})$, while using at most k oracle calls and guaranteeing the precision target is met with a probability at least $1-\delta$, that is, $\mathbb{P}(\mathcal{P}(\hat{Y}) \geq T) \geq 1-\delta$, where the probability is over runs of the algorithm. RT queries are defined analogously but with precision and recall swapped in the problem definition.

Quality Constraints and Utility. In the remainder of this paper, we collectively refer to *quality constraints* in AT, PT and RT queries as the constraints on accuracy, precision and recall in the respective problem definition. We use the letter $\mathcal{F}$ to refer the quality constraints for all problems (e.g., $\mathcal{F}(Y) \geq T$ means $\mathcal{A}(Y) \geq T$, $\mathcal{P}(Y) \geq T$ and $\mathcal{R}(Y) \geq T$ depending on the situation). We collectively refer to *utility* of the solution for AT, PT and RT queries as the objective optimized in their respective problem definitions. That is, for AT queries, the utility is measured in terms of cost, for PT in terms of recall and for RT queries in terms of precision.

2.2 Model Cascade Framework

To decide when to use the proxy model or the oracle, we follow the model cascade framework [2, 19, 22, 23, 34]. Recall that in PT and RT queries, we are given a fixed oracle budget, while in AT queries, our goal is to minimize the number of oracle calls. Thus, the cascade framework is used differently in each case. Here, we describe the framework for RT and PT queries, following the formalization by [21], and defer the discussion of AT queries to Sec. 4.1.

For PT and RT queries, model cascade relies on *proxy scores* $\mathcal{S}(x) \in [0,1]$ that quantify the proxy model's confidence in the record x being positive (recall that for PT/RT queries model outputs are binary). Such scores are typically produced by the proxy model as part of inference (i.e., the output tokens' probability for LLMs). The framework uses the fixed oracle budget, k, to determine a *cascade threshold* ρ on the proxy scores so that the records with proxy scores more than ρ are deemed positive and those below ρ are deemed negative. More formally, first define, for a set of records S, $S \subseteq D$, and a threshold $\rho \in [0,1]$, $S^\rho = \{x; x \in S, \mathcal{S}(x) > \rho\}$. In our framework, a subset S, $S \subseteq D$, of size at most k is labeled and used to determine the cascade threshold, ρ. Using ρ, we estimate the set of records with positive labels as D^ρ. In practice, D^ρ can additionally be augmented with the observed positive labels in S.

Thus, performing PT queries boils down to finding a cascade threshold, ρ, by labeling a subset $S \subseteq D$ of size k with the oracle, such that $\mathcal{P}(D^\rho)$ meets the precision target, that is $\mathbb{P}(\mathcal{P}(D^\rho) \geq T) \geq 1-\delta$, and maximizes recall, $\mathcal{R}(D^\rho)$. The problem is analogously defined for RT queries. For convenience of notation, we define $\mathcal{P}_D(\rho) = \mathcal{P}(D^\rho)$ and $\mathcal{R}_D(\rho) = \mathcal{R}(D^\rho)$. We further abuse notation and, for any set $S \subseteq D$, define:

$$\mathcal{R}_S(\rho) = \frac{\sum_{x \in S^\rho} \mathbb{I}[\mathcal{O}(x)=1]}{\sum_{x \in S} \mathbb{I}[\mathcal{O}(x)=1]}, \mathcal{P}_S(\rho) = \frac{\sum_{x \in S^\rho} \mathbb{I}[\mathcal{O}(x)=1]}{|S^\rho|}.$$

$\mathcal{P}_S(\rho)$ (resp. $\mathcal{R}_S(\rho)$) is the precision (resp. recall) with respect to S if S^ρ is estimated as the set of records in S with positive labels.

Note that the use of proxy scores to decide whether to use the proxy or not is beneficial only if the model is well-calibrated [39], i.e., there is positive correlation between the value of proxy score and probability of correctness of the proxy. Although our guarantees on output quality always

Notation	Description
D, S	Dataset of all records and a subset of the records
$\mathcal{O}(x)$	Oracle output for a record x
$\mathcal{P}(x), \mathcal{S}(x)$	Proxy output and score for a record x
ρ	Cascade threshold on proxy scores
$\mathcal{A}_S(\rho)$	Accuracy on a set S at threshold ρ
$\mathcal{P}_S(\rho), \mathcal{R}_S(\rho)$	Precision and recall on a set S at threshold ρ
S^ρ	$\{x \in S : \mathcal{S}(x) > \rho\}$ for any set of records, S
S_+	$\{x; x \in S, \mathcal{O}(x) = 1\}$ for any set of records, S
$\mathcal{E}(S, T, \rho, \alpha)$	Estimation function to test if a threshold, ρ, meets T given a sample S with confidence α
C	Set of candidate cascade thresholds to choose from
k	Oracle budget
T	Target precision, recall, or accuracy
δ	Allowed probability of failure to meet the target

Table 2. Summary of mathematical notation

Variant	Quality Metric	Primary Parameters
BARGAIN$_P$-A	Precision	k, T, δ
BARGAIN$_A$-A	Accuracy	T, δ
BARGAIN$_A$-M	Accuracy	T, δ
BARGAIN$_R$-A	Recall	k, T, δ

Table 3. Main BARGAIN Variants

hold even when the models are not calibrated, in such cases, the cascade framework is unlikely to provide high utility, e.g., for PT queries, one obtains low recall even though the precision is guaranteed to meet the precision target.

Notation and Terminology. For a random sample S of D, we refer to metrics calculated on the sample as *observed* metrics (e.g., $\mathcal{P}_S(\rho)$ is the *observed precision* at ρ), and refer to the metrics on the entire dataset as *true* metrics (e.g., $\mathcal{P}_D(\rho)$ is the *true precision*). When sampling points, the oracle labels every sampled point, so we use the terms *sampling* and *labeling* interchangeably. We use $[i]$ to denote the set $\{1, ..., i\}$ for any integer i.

2.3 Statistical Tools

To solve the cascade threshold problem through sampling, we need to estimate, based on observed samples, whether a specific threshold meets the quality target. This estimation can be done using classic concentration bounds such as Hoeffding's or Chernoff's inequality. We instead use recent results by Waudby-Smith and Ramdas [42] that provide tighter bounds (as discussed in [42] and empirically validated in our results). Here, we provide an informal overview of the result by [42] used in our paper. Formal statement of results are presented in our technical report [44].

For a set of i.i.d random variables, $X = \{X_1, .., .X_k\}$ whose true mean is μ, we would like to estimate, using X, whether the true mean is more than a threshold m or not. Theorem 3 of [42] provides a hypothesis testing approach for this estimate. It specifies a boolean function $\mathcal{T}(m, X, \alpha)$, which, with high probability, is 1 when μ is at least m and 0 otherwise. [42] shows that whenever $\mu < m$, $\mathbb{P}\big(\mathcal{T}(m, X, \alpha) = 1\big) \leq \alpha$. That is, $\mathcal{T}$ is unlikely to wrongly estimate the mean is more than m when it is not.

Lemma 2.1 (Informal and Simplified Statement of Theorem 3 by [42]). *Consider the set of i.i.d random variables X with mean μ. For a confidence parameter $\alpha \in [0, 1]$, and any $\mu < m$, we have*

$$\mathbb{P}\big(\mathcal{T}(m, X, \alpha) = 1\big) \leq \alpha, \text{ where} \tag{1}$$

$$\mathcal{T}(m, X, \alpha) \approx \mathbb{I}\big[\mathcal{K}(m, X) \geq \frac{1}{\alpha}\big], \tag{2}$$

$$\mathcal{K}(m, X) \approx \Pi_{i=1}^{k}\Big(1 + \frac{(X_i - m)}{\hat{\sigma}_{i-1}}\sqrt{\log(1/\alpha)}\Big), \tag{3}$$

$$\hat{\sigma}_i^2 = \frac{1/4 + \sum_{j=1}^{i}(X_j - \hat{\mu}_j)^2}{i+1}, \ \hat{\mu}_i = \frac{1/2 + \sum_{j=1}^{i} X_j}{i+1}.$$

Note that $\approx$ means *we have dropped some of the terms from the definition* to convey high-level intuition (see our technical report [44] for exact formulas). Above, $\hat{\sigma}_i^2$ (resp., $\hat{\mu}_i$) is a term analogous to empirical variance (resp., mean). $\mathcal{K}(m, X)$, informally, quantifies whether the sequence of X_i consistently exceeds m, when normalized by the variance $\hat{\sigma}_i^2$; larger values show observations

exceed m more frequently (thus suggesting $\mu > m$). The use of empirical variance $\hat{\sigma}_i^2$ contrasts with Hoeffding's inequality that only relies on the empirical mean. Taking standard deviation into account significantly improves the bounds when standard deviation is small, as our experiments show (see Sec. 3.2 for comparison).

2.4 Overview and Outline

Armed with the statistical tools from Sec. 2.3, BARGAIN solves the cascade threshold problem for AT, PT, and RT queries. We present BARGAIN for PT queries in Sec. 3, discussing alternatives and solution components in depth. We extend BARGAIN to AT and RT queries, respectively, in Secs. 4.1 and 4.2. Our main proposed BARGAIN variants for each of the queries are shown in Table 3. The variants in the table perform adaptive sampling; for ease of exposition and comparison purposes, we also present other variants (not listed in the table) that perform uniform sampling. All variants take the quality target, T, and confidence parameter, δ, as input, while BARGAIN$_P$-A and BARGAIN$_R$-A additionally require the oracle budget, k, upfront. All methods additionally use other parameters which need not be set by the users and modifying their default values have limited impact on utility. We discuss how these parameters are set in Sec. 5.

Use-cases of Variants. BARGAIN$_A$ is suitable for multi-class classification, as well as binary classification without a maximum oracle budget constraint but when the goal is to minimize the total budget used. BARGAIN$_P$ and BARGAIN$_R$ on the other hand respect a maximum oracle budget and thus are useful when user wants to stay within a budgetary constraint. Moreover, BARGAIN$_P$ and BARGAIN$_R$ are useful for binary classification tasks with class imbalance, e.g., when there is expected to be few records in the positive class, while BARGAIN$_A$ is additionally applicable when classes are balanced. Finally, BARGAIN$_A$-A and BARGAIN$_A$-M differ in that BARGAIN$_A$-A chooses a single cascade threshold for *all* output classes while BARGAIN$_A$-M chooses a cascade threshold *per class*. As such, BARGAIN$_A$-M is beneficial if the proxy is differently calibrated for different output classes, which may occur if it is more difficult to correctly estimate one class but not another.

3 BARGAIN for PT Queries

In this section, we discuss BARGAIN for solving PT queries. We use Fig. 4 (top half) as a running example, where we perform a PT query with target $T = 0.75$ on the dataset, D. That is, our goal is to find a cascade threshold with precision at least 0.75 w.h.p while maximizing recall. ① Red/blue squares show records $x \in D$, plotted at $\mathcal{S}(x)$ on the proxy score axis and colored based on $\mathcal{O}(x)$. The vertical lines $\rho_1, ..., \rho_{10}$ are *candidate thresholds* to choose the final cascade threshold from. Bottom of Fig. 4 ② shows the true precision, $\mathcal{P}_D(\rho_i)$, and ③ true recall, $\mathcal{R}_D(\rho_i)$, for the candidate thresholds. *candidate thresholds* are precisely the proxy scores of records in S, $\mathrm{C} = \{\mathcal{S}(x); x \in S\}$, because $\mathcal{P}_S(\rho)$ (i.e., the observed precision at threshold ρ) changes only at these values and is constant otherwise.

Overview of BARGAIN for PT Queries. Our framework for PT queries consists of three main components, as detailed in Fig. 1 and discussed here at a high level: (1) A *sampling* component samples a set of k records, S, from D, to be labeled by the oracle. (2) An *estimation* component estimates for any candidate threshold whether it is expected to meet the precision target. This is done by defining a boolean *estimation function* $\mathfrak{E}(S, T, \rho, \alpha)$, which given a sample set S, the target T, and a candidate threshold ρ, returns 1 if $\mathcal{P}_D(\rho) \geq T$ is expected to hold and 0 otherwise. $\mathfrak{E}$ depends on a *confidence parameter* α that quantifies the probability that $\mathfrak{E}$ makes wrong estimates, discussed later. (3) A *selection* component selects, among the thresholds expected to meet the target, the one with the highest recall. This step repeatedly uses $\mathfrak{E}$ to estimate if different candidate thresholds

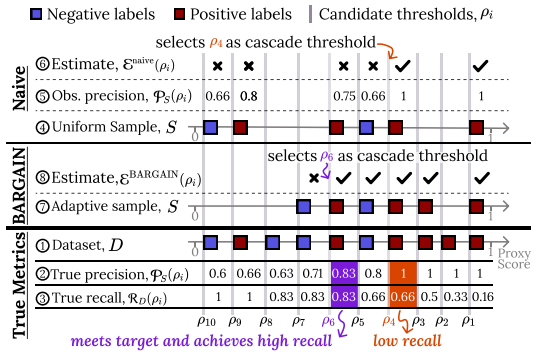


Fig. 4. An example of cascade approaches for PT queries given precision target $T = 0.75$.

meet the requirement before choosing the one with the highest recall among ones where $\mathcal{E}$ returns true.

Next, to illustrate the framework and associated challenges, we describe a naive approach following the framework. We then present BARGAIN$_P$-U, which keeps the uniform sampling procedure of the naive method but improves upon estimation and selection procedures. We then present BARGAIN$_P$-A, our final algorithm that incorporates adaptive sampling to improve utility.

3.1 Warm-up: Naive Algorithm with Guarantees

Sampling and Estimation. A naive algorithm samples a set S of k records uniformly at random from D. The estimation function $\mathcal{E}$ uses this set to estimate if different candidate thresholds meet the precision target. We want $\mathcal{E}$ to have a *low false positive probability*:

$$\mathbb{P}_{S\sim D}(\mathcal{E}(S, T, \rho, \alpha) = 1) \leq \alpha \text{ if } \mathcal{P}_D(\rho) < T, \tag{4}$$

for any confidence parameter $\alpha \in [0, 1]$. False positives must be avoided as they can lead to the selection of a threshold incorrectly estimated to have true precision above T. To design $\mathcal{E}$, we can use Hoeffding's inequality to relate the observed precision, $\mathcal{P}_S(\rho)$ to the true precision, $\mathcal{P}_D(\rho)$. By doing so, we show

$$\mathcal{E}^{\text{naive}}(S, T, \rho, \alpha) = \mathbb{I}\big[\mathcal{P}_S(\rho) \geq T + \Delta\big], \text{ where } \Delta = \sqrt{\frac{\log(1/\alpha)}{2|S^\rho|}} \tag{5}$$

satisfies Eq. 4, where $\mathbb{I}$ is the indicator function and S^ρ is the subset of S with proxy scores more than ρ. $\mathcal{E}^{\text{naive}}$ estimates that a threshold ρ meets the target whenever the observed precision, $\mathcal{P}_S(\rho)$ is more than $T + \Delta$. Note that Δ serves as an adjustment factor for sampling. That is, it is not sufficient for observed precision $\mathcal{P}_S(\rho) \geq T$ to guarantee $\mathcal{P}_D(\rho) \geq T$, but we instead need $\mathcal{P}_S(\rho) \geq T + \Delta$. Fig. 4 shows an example at threshold ρ_9 where $\mathcal{P}_S(\rho) \geq T$, but $\mathcal{P}_D(\rho)$ is not. We can use Hoeffding's inequality to show the following.

Proposition 3.1. *For any $\rho \in [0, 1]$ with $\mathcal{P}_D(\rho) < T$, we have $\mathbb{P}(\mathcal{P}_S(\rho) \geq T + \sqrt{\frac{\log(1/\alpha)}{2|S^\rho|}}) \leq \alpha$, where S is i.i.d and uniformly sampled from D and $\alpha \in [0, 1]$ is a confidence parameter. Consequently, $\mathbb{P}(\mathcal{E}^{naive}(S, T, \rho, \alpha) = 1) \leq \alpha$.*

Fig. 4 shows an example of this procedure. The figure depicts a sample S ④ and the observed precision at different candidate thresholds ρ_i ⑤. In the example, we see that $\mathcal{E}^{\text{naive}}$ estimates $\rho_i \in \{\rho_1, ..., \rho_4\}$ meet the target while $\mathcal{E}^{\text{naive}}(S, \rho_i, T, \alpha) = 0$ for all other thresholds ⑥. We note that it is possible to use Chernoff's inequality to define the estimation function instead of Hoeffding's. In our technical report [44] we discuss how the bound can be applied and present results comparing

the use of Chernoff's and Hoeffding's inequality. We saw marginal differences between the two and thus present only Hoeffding's inequality here because it is simpler to apply.

Selection. The naive approach for threshold selection chooses the smallest threshold in the candidate set as the cascade threshold:

$$\rho^* = \min\{\rho; \rho \in \mathrm{C}, \mathcal{E}^{\text{naive}}(S, T, \rho, \alpha) = 1\}. \tag{6}$$

ρ^* is the ρ that maximizes recall (returns the most records as "positive") among all ρ that satisfy $\mathcal{E}^{\text{naive}}$. Finding ρ^* requires $|\mathrm{C}|$ applications of $\mathcal{E}^{\text{naive}}$. To guarantee ρ^* meets the target, *all* applications of $\mathcal{E}^{\text{naive}}$ must return 0 for candidate thresholds with $\mathcal{P}_D(\rho) < T$. Prop. 3.1 shows this holds for a single application of $\mathcal{E}^{\text{naive}}$. We use union bound across all applications of $\mathcal{E}^{\text{naive}}$ to ensure the probability that any of $\mathcal{E}^{\text{naive}}$ return 0 for candidate thresholds with $\mathcal{P}_D(\rho) < T$ is small. Doing so proves:

Proposition 3.2. *Let $\mathcal{E}$ be a function with false positive probability bounded by α (as defined in Eq. 4) when sampling a set S from D. Setting ρ^* as Eq. 6, we have $\mathbb{P}(\mathcal{P}_D(\rho^*) < T) \leq |\mathrm{C}|\alpha$.*

Prop. 3.2 shows the probability of the selection method failing to meet the target is proportional to $|\mathrm{C}|$, the candidate set's size.

Final Algorithm and Guarantees. Since we want $|\mathrm{C}|\alpha = \delta$, we set $\alpha = \frac{\delta}{|\mathrm{C}|}$. Therefore, we select the cascade threshold as

$$\rho_S^{\text{naive}} = \min\{\rho; \rho \in \mathrm{C}, \mathcal{E}^{\text{naive}}(S, T, \rho, \frac{\delta}{|\mathrm{C}|}) = 1\}. \tag{7}$$

ρ_S^{naive} uses $\mathcal{E}^{\text{naive}}$ for estimation with $\alpha = \frac{\delta}{|\mathrm{C}|}$. This ensures, using Prop. 3.2, that the probability of ρ_S^{naive} missing the target is bounded by δ. Indeed, combining Props. 3.1 and 3.2 shows:

Lemma 3.3. *For cascade threshold ρ_S^{naive} returned by the naive method, we have $\mathbb{P}_{S\sim D}(\mathcal{P}_D(\rho_S^{naive}) < T) \leq \delta$.*

Discussion. Consider our running example, in Fig. 4, where the naive algorithm chooses ρ_4 as the cascade threshold, since it is the smallest threshold with $\mathcal{E}^{\text{naive}}(S, \rho_i, T, \frac{\delta}{|\mathrm{C}|}) = 1$. This cascade threshold leads to recall 66%, well below the best possible of 83% at ρ_6 while meeting the precision requirement. This low recall can be attributed to the *high false negative rate* of $\mathcal{E}^{\text{naive}}$ which incorrectly precludes the algorithm from selecting thresholds that meet the target and have better utility; e.g., in Fig. 4, $\mathcal{E}^{\text{naive}}$ incorrectly estimates that ρ_5 and ρ_6 don't meet the target. To improve on the naive approach, BARGAIN leverages novel sampling, estimation, and selection methods that significantly reduce the false negative rate and ultimately provide much better utility. As Fig. 4 shows, BARGAIN performs adaptive sampling and uses a more accurate estimation function and threshold selection mechanism. These significantly improve recall while providing the same theoretical guarantees.

3.2 BARGAIN$_P$-U: Data-Aware Estimation

As described above, the low recall achieved by the Naive approach can be attributed to the estimator $\mathcal{E}^{\text{naive}}(S, T, \rho, \frac{\delta}{|\mathrm{C}|})$ frequently returning 0 when $\mathcal{P}_D(\rho) \geq T$. BARGAIN$_P$-U uses a better estimation function $\mathcal{E}$ and selection algorithm to remedy this. This selection algorithm performs a tighter analysis across applications of $\mathcal{E}$ to avoid splitting the failure probability δ into $\frac{\delta}{|\mathrm{C}|}$ for each application of $\mathcal{E}$. Both improvements allow BARGAIN$_P$-U to identify better valid candidate thresholds, ρ, where $\mathcal{P}_D(\rho) \geq T$, thus improving the recall. We describe these two components in more detail before presenting the final BARGAIN$_P$-U algorithm. BARGAIN$_P$-U performs uniform sampling; we incorporate adaptive sampling in Sec. 3.3.

3.2.1 Estimation. Observe that the true precision at a threshold ρ is the mean of the random variables $\mathbb{I}[\mathcal{O}(x) = 1]$ for $x \in S^\rho$. That is, $\mathbb{E}[\mathbb{I}[\mathcal{O}(x) = 1]] = \mathcal{P}_D(\rho)$, because $x \in S^\rho$ is a uniform sample from D^ρ, so

$$\mathbb{P}(\mathcal{O}(x) = 1) = \sum_{x' \in D^\rho} \frac{\mathbb{I}[\mathcal{O}(x') = 1]}{|D^\rho|} = \mathcal{P}_D(\rho).$$

Therefore, to estimate whether $\mathcal{P}_D(\rho) \geq T$ or not, we can use the hypothesis test of whether the true mean of the random variables in S^ρ is more than T or not. This hypothesis testing formulation enables us to use Lemma 2.1 to design our estimation function. Our approach results in a new function $\mathcal{E}^{\text{BARGAIN}}$ that helps test whether $\mathcal{P}_D(\rho) \geq T$. $\mathcal{E}^{\text{BARGAIN}}$ performs the hypothesis test, $\mathcal{T}$, from Lemma 2.1, to decide if the true precision is below the threshold T. The following Lemma shows the guarantees of $\mathcal{E}^{\text{BARGAIN}}$. Define the set of random variables $S_O^\rho = \{\mathbb{I}[\mathcal{O}(x) = 1]; x \in S^\rho\}$.

Lemma 3.4. *For a confidence parameter $\alpha \in [0, 1]$ and any $\rho \in [0, 1]$ where $\mathcal{P}_D(\rho) < T$,*

$$\mathbb{P}(\mathcal{E}^{\text{BARGAIN}}(S, T, \rho, \alpha) = 1) \leq \alpha, \quad \textit{where} \tag{8}$$

$$\mathcal{E}^{\text{BARGAIN}}(S, T, \rho, \alpha) = \mathcal{T}(T, S_O^\rho, \alpha), \tag{9}$$

and $\mathcal{T}$ is the hypothesis test defined in Lemma 2.1.

Benefits. Eq. 8 shows that $\mathcal{E}^{\text{BARGAIN}}$ performs our desired hypothesis test of whether $\mathcal{P}_D(\rho) \geq T$ or not with bounded false positive probability. That is, $\mathcal{E}^{\text{BARGAIN}}$ is unlikely to return 1 for thresholds where $\mathcal{P}_D(\rho) < T$. As such, we use $\mathcal{E}^{\text{BARGAIN}}$ instead of $\mathcal{E}^{\text{naive}}$, replacing Prop. 3.1 in the naive approach with Lemma 3.4 in BARGAIN$_P$-U.

The benefit of $\mathcal{E}^{\text{BARGAIN}}$ over $\mathcal{E}^{\text{naive}}$ can be attributed to the use of observed variance in $\mathcal{E}^{\text{BARGAIN}}$ to provide significantly better estimates, especially so when observations have low variance. $\mathcal{E}^{\text{BARGAIN}}$ takes both observed means and variances into account (via $\hat{\sigma}_i$ and $\hat{\mu}_i$ in Lemma 2.1), but $\mathcal{E}^{\text{naive}}$ only depends on the observed mean (i.e., the observed precision in Eq. 5). We empirically show the benefits of $\mathcal{E}^{\text{BARGAIN}}$ in Fig. 5, where we plot

$$T_\rho^* = \max\{T; T \in [0, 1], \mathcal{E}(S, T, \rho, \alpha) = 1\} \tag{10}$$

at different values of ρ for fixed random samples, S and for $\mathcal{E}$ as either $\mathcal{E}^{\text{BARGAIN}}$ or $\mathcal{E}^{\text{naive}}$. T_ρ^* denotes the largest target T that a threshold ρ will be estimated to meet using $\mathcal{E}$, given S and α. The closer T_ρ^* is to $\mathcal{P}_D(\rho)$ the more accurate $\mathcal{E}$ is. In fact, for any target $T \in (T_\rho^*, \mathcal{P}_D(\rho)]$, the estimate of $\mathcal{E}$ is a false negative, so how close T_ρ^* is to $\mathcal{P}_D(\rho)$ determines the accuracy of $\mathcal{E}$ at different targets. To plot T_ρ^*, we use one of our real-world datasets, Court (details in Sec. 6), as D, set $\alpha = 0.1$, and for each threshold, ρ, sample a (large enough) set, S so that S^ρ contains 50 records. We keep the size of S^ρ fixed to ensure the number of samples does not impact the trends across thresholds (e.g., $|S^\rho| = 50$ in Eq. 5 for all ρ). Fig. 5 also plots $\mathcal{P}_D(\rho)$ and $\mathcal{P}_S(\rho)$, and the shaded region around $\mathcal{P}_S(\rho)$ shows its observed standard deviation across 10 runs.

Fig. 5 shows $\mathcal{E}^{\text{BARGAIN}}$ provides increasingly better estimates than $\mathcal{E}^{\text{naive}}$ as $\mathcal{P}_D(\rho)$ increases. Indeed, if the user provides target $T = 0.9$, $\mathcal{E}^{\text{naive}}$ estimates that no target meets the threshold. Instead, $\mathcal{E}^{\text{BARGAIN}}$ correctly identifies thresholds larger than ~0.5 that meet the target, because it takes advantage of the lower variance in the observed samples when the true precision is high. For instance, when the true precision is 1, observed variance will be zero because all observed samples at the threshold will always be positive. Indeed, in our technical report [44] we show that the variance in observations decreases as the true precision increases; thus $\mathcal{E}^{\text{BARGAIN}}$ makes more accurate estimates when true precision is high. Meanwhile, $\mathcal{E}^{\text{naive}}$ provides the same estimates independent of the variance.

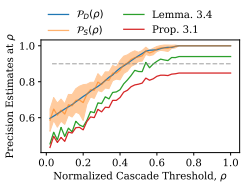


Fig. 5. Comparison of different estimation methods

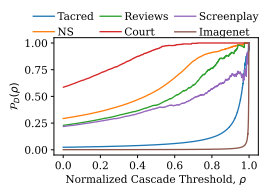


Fig. 6. Ground-truth precision at different thresholds

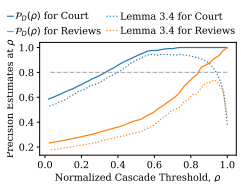


Fig. 7. Estimation with a fixed uniform sample

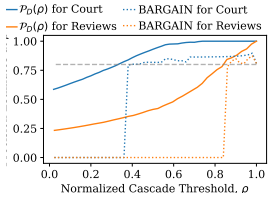


Fig. 8. Estimation from sampling with BARGAIN$_P$-A

3.2.2 Threshold Selection. The Naive algorithm selected the smallest threshold among all the candidate thresholds estimated to meet the precision target. Prop. 3.2 showed that this leads to the probability of missing the target increasing in the number of candidate thresholds, $|\mathrm{C}|$. Next, we show that we can incorporate data characteristics into the selection algorithm and analysis to avoid this dependence on $|\mathrm{C}|$.

Selection Procedure. The following equation presents our selection method (used in place of Eq. 6) to improve utility. It uses a tolerance parameter η that, as discussed later, helps us incorporate data characteristics into the algorithm:

$$\rho^* = \min\{\rho; \rho \in \mathrm{C}, \mathfrak{E}(S,T,\rho,\alpha) = 1, \gamma_\rho \leq \eta\}, \text{ where} \tag{11}$$
$$\gamma_\rho = \sum_{\rho' \in \mathrm{C}, \rho' \geq \rho} \mathbb{I}[\mathfrak{E}(S,T,\rho',\alpha) = 0].$$

γ_ρ denotes the number of candidate thresholds greater than or equal to ρ that $\mathfrak{E}$ estimates to miss the target. Thus, ρ^* is the smallest candidate threshold where at most η larger thresholds are estimated to miss the target. When $\eta = |\mathrm{C}|$, ρ^* is the smallest threshold that meets the target (same as Eq. 6), but when $\eta = 0$, ρ^* is the smallest candidate threshold such that both itself and all larger thresholds are estimated to meet the target. The failure probability of Eq. 11 to meet the target now depends on η not $|\mathrm{C}|$:

Lemma 3.5. *Let $\mathfrak{E}$ be a function with false positive probability bounded by α (as defined in Eq. 4) when sampling a set S from D. Setting ρ^* as Eq. 11, we have $\mathbb{P}(\mathcal{P}_D(\rho^*) < T) \leq (\eta + 1)\alpha$.*

Benefits. The tolerance parameter η allows us to capture data characteristics, and, in real-world settings, can be set to a small value to obtain low failure probability using Lemma 3.5. A suitable value for η depends on how often, in a dataset, the true precision oscillates around the target. For instance, if after the largest ρ with true precision below T, $\mathcal{P}_D(\rho) < T$, all thresholds $\rho' < \rho$ also have $\mathcal{P}_D(\rho) < T$, then there is no benefit in considering thresholds less than ρ, and it is sufficient to set $\eta = 0$. Fig. 6 that plots $\mathcal{P}_D(\rho)$ for six different real-world datasets (from Sec. 6) shows this is the case in real-world datasets, where true precision is often monotonically decreasing. Thus, the above argument implies setting $\eta = 0$ is sufficient to obtain a good utility. We note that setting $\eta > 0$ may be beneficial in cases where $\mathcal{P}(D)$ drops below T for η different candidate thresholds, after which $\mathcal{P}(D)$ rises up again to above T. Fig. 6 shows this is rarely the case in real-world datasets. We emphasize that the choice of η only affects the utility of the methods. Our guarantees, as Lemma 3.5 shows, hold for all possible datasets. Given the above observations, and to simplify our results, in the rest of this paper, we set $\eta = 0$. Generalization to $\eta > 0$ is straightforward and discussed in our technical report [44].

3.2.3 BARGAIN$_P$-U Algorithm and Guarantees. BARGAIN$_P$-U performs uniform sampling, and uses the above estimation and threshold selection procedures, as stated in Alg. 1. It iterates through candidate thresholds in decreasing order and stops after reaching the first threshold estimated by $\mathfrak{E}^{\text{BARGAIN}}$ to miss the target. If such a threshold is $\mathrm{C}[i]$, it returns $\mathrm{C}[i-1]$, which was estimated to meet the target at the previous iteration. The early stopping is possible because $\eta = 0$ in threshold selection. Combining Lemmas 3.4 and 3.5 proves:

Algorithm 1 BARGAIN$_P$-U

```
S ← Sample k records from D uniformly at random
Sort C in descending order
for i in |C| do
    ρ ← C[i]
    if 𝓔^BARGAIN(S, T, ρ, δ) = 0 then
        return C[i − 1]
return C[|C|]
```

Algorithm 2 BARGAIN$_P$-A

```
Sort C_M in descending order
for i in M do
    ρ ← C_M[i]
    S ← ∅
    while 𝓔^BARGAIN(S, T, ρ, δ) = 0 do
        if Out of sampling budget then
            return C_M[i − 1]
        Sample a record uniformly from D^ρ and add to S
return C_M[M]
```

Lemma 3.6. *Let $\rho_S^{BARGAIN_P\text{-}U}$ be the cascade threshold found by Alg. 1. We have $\mathbb{P}_{S\sim D}(\mathcal{P}_D(\rho_S^{BARGAIN_P\text{-}U}) < T) \leq \delta$.*

3.2.4 Discussion. The improved estimation and selection methods in BARGAIN$_P$-U improve upon the naive approach, but its utility is still hampered by uniform sampling. To see why, observe that applications of $\mathcal{E}^{\text{BARGAIN}}$ for different ρ values use the same sample set S but different subsets, S^ρ, of S to make an estimate for each ρ. Given a fixed set, S, the size of S^ρ decreases the larger ρ gets. This causes two problems. First, for large ρ, $\mathcal{E}^{\text{BARGAIN}}$ has access to a smaller set to perform estimation. Using fewer observations, $\mathcal{E}^{\text{BARGAIN}}$ will be less confident about the true precision; thus, to provide the same low false positive guarantees (as defined in Eq. 4) it has to err on the safe side and estimate that thresholds don't meet the target more frequently. This is plotted in Fig. 7, showing T_ρ^* (as defined in Eq. 10) for different values of ρ, but for a fixed uniform sample from D (for two real-world datasets, Court and Reviews). We see that Lemma 3.4 provides poor estimates for large ρ, incorrectly estimating that larger thresholds don't meet many targets. Comparing Figs. 7 and 5 shows that this is indeed due to decreasing sample sizes, where Fig. 5 plots the same quantity for Court dataset, but when the sample size is fixed (with $k_\rho = 50$) for all thresholds. Second, even though uniform sampling leads to few samples for large candidate thresholds, it, on the other hand, *wastes* many samples for estimation when the candidate threshold is small. In fact, when BARGAIN$_P$-U stops at a candidate threshold, ρ, any sample with proxy score less than ρ is simply not used during estimation at all.

3.3 BARGAIN$_P$-A: Task-Aware Sampling

In this section, we present BARGAIN$_P$-A, our final algorithm for PT Queries that builds on BARGAIN$_P$-U, but additionally performs adaptive sampling to maximally utilize the oracle budget.

BARGAIN$_P$-A. BARGAIN$_P$-A combines the sampling, estimation and selection steps to direct the sampling budget to both sample enough records when needed and avoid wasting samples when not. To do so, BARGAIN$_P$-A follows a similar algorithm as BARGAIN$_P$-U, but instead of sampling S upfront, it samples records on the go and interleaved with the estimation and selection steps.

That is, for each candidate threshold ρ, in decreasing order, when checking whether $\mathcal{E}^{\text{BARGAIN}}$ is true or false, it *continuously samples* new records in range $[\rho, 1]$ until it obtains $\mathcal{E}^{\text{BARGAIN}}(S, T, \rho, \delta) = 1$. Only after it obtains $\mathcal{E}^{\text{BARGAIN}}(S, T, \rho, \delta) = 1$ the algorithm considers the next (smaller) candidate threshold, for which the same process is repeated until it runs out of oracle budget. Note that for a candidate threshold, ρ with $\mathcal{P}_D(\rho) > T$, given enough records $\mathcal{E}^{\text{BARGAIN}}$ will eventually estimate that it meets the target. However, if $\mathcal{P}_D(\rho) < T$, with high probability $\mathcal{E}^{\text{BARGAIN}}$ will return 0, in which case BARGAIN$_P$-A continues sampling until it runs out of budget. Since we do not sample a set S apriori, we cannot use the candidate threshold set C. Define, for a system parameter M,

$$\mathrm{C}_M = \{\mathcal{S}(x_i); i = \lfloor \frac{j}{M} n \rfloor, j \in [M]\}, \tag{12}$$

as the candidate threshold set of size M, where $x_1, .., x_n \in D$, are sorted according to their proxy scores, so that C_M contains every $\frac{j}{M}$-th percentile of proxy scores of D as the candidate threshold set. We discuss the impact of M as well as other potential choices for the candidate threshold set in our technical report [44].

BARGAIN$_P$-A as presented in Alg. 2, iterates over candidates in C_M in descending order, and at the i-th iteration for $\rho = \mathrm{C}_M[i]$ checks if $\mathcal{E}^{\text{BARGAIN}}$ returns true (Line 5). If $\mathcal{E}^{\text{BARGAIN}}$ is false, it samples a new record from D^ρ (i.e., from records with proxy score $> \rho$) and checks $\mathcal{E}^{\text{BARGAIN}}$ again with this new sample. This sampling and estimation continues until either (1) $\mathcal{E}^{\text{BARGAIN}}$ returns 1, so the algorithm is confident that ρ meets the target and moves on to the next candidate threshold, or (2) the algorithm runs out of oracle budget, wherein the algorithm returns the last candidate threshold estimated to meet the target. Note that when $\mathcal{P}_D(\rho) < T$, it is unlikely that $\mathcal{E}^{\text{BARGAIN}}$ will return true. Thus, when the algorithm reaches a candidate threshold where $\mathcal{P}_D(\rho) < T$, it keeps sampling new records and eventually runs out of budget, thus returning the last candidate threshold for which it is estimated that $\mathcal{P}_D(\rho) \geq T$.

Benefits. To see the benefits of BARGAIN$_P$-A, Fig 8 plots the same quantity T^*_ρ across thresholds, as defined in Eq. 10, but now after the sampling procedure of BARGAIN$_P$-A. Note that T^*_ρ is plotted for the sample set obtained at the end of every iteration. Comparing Fig 8 and Fig. 7, we see significant benefits where the sampling method improves the estimates for larger values of ρ. Furthermore, the sharp drop in Fig 8 for small ρ is due to the algorithm not sampling any more records after it chooses the cascade threshold, so it avoids wasting samples for candidate thresholds that will not be selected.

Guarantees and Proof Overview. BARGAIN$_P$-A provide the same required theoretical guarantees as BARGAIN$_P$-U (see our technical report [44]). We omit the theorem for BARGAIN$_P$-A for the sake of space but discuss overview of the analysis. Alg. 2 fails to meet the target if $\mathcal{E}^{\text{BARGAIN}}$ wrongly estimates a threshold meets the target at any iterations of the algorithm. To bound the probability of this event, we ensure that (1) the estimates for any threshold ρ can be wrong with a bounded probability and use this result to show (2) the total probability of making a wrong estimate across all thresholds is bounded by δ. To show (1), observe that every estimate by $\mathcal{E}^{\text{BARGAIN}}$ for a threshold ρ uses samples taken uniformly from D^ρ. Thus, the probability that a single estimate is incorrect is bounded by Lemma 3.4. Furthermore, the repeated estimation while sampling for the same threshold is accounted for through [42] which shows that the estimation procedure in Lemma 3.4 is *anytime valid*, that is, the estimation can be performed repeatedly during sampling, while still providing the same bound on the overall probability of making a wrong estimate. Then, to show (2), we use Lemma 3.5 which uses the union bound to account for the total probability of making a wrong estimate across all thresholds.

Sampling without Replacement and Reuse. So far, the discussion uses sampling with replacement, additionally with S being set to $\emptyset$ per new threshold considered. We show that we can

Algorithm 3 BARGAIN$_A$-A

```
Sort C_M in descending order
for i in |C_M| do
    ρ ← C_M[i]
    S ← ∅
    while E_A^BARGAIN(S, T, ρ, δ) = 0 do
        if avg(S_A^ρ) − std(S_A^ρ) ≥ T and |S_A^ρ| ≥ c then
            return C_M[i − 1]
        Sample a record uniformly from D^ρ and add to S
return C_M[M]
```

sample without replacement and obtain the same theoretical guarantees but with a slightly different estimation function. Our technical report [44] discusses how to modify $\mathcal{E}^{\text{BARGAIN}}$, and shows we obtain the same guarantees. BARGAIN$_P$-A uses sampling without replacement. Furthermore, the sampling process in Line 8 of Alg. 2can reuse the oracle labels from samples from previous iterations, discussed in detail in our technical report [44].

4 AT and RT Queries

BARGAIN applies similar intuitions as PT queries to solve AT and RT queries. Theoretical statements are deferred to our technical report [44].

4.1 BARGAIN$_A$ for AT Queries

4.1.1 Model Cascade for AT Queries. We first introduce the model cascade framework for AT queries. For AT queries, because the output is not necessarily binary, the proxy score is defined more generally as the model's confidence in its output (not necessarily the positive class). Furthermore, AT queries do not come with a fixed predefined budget, but our goal is to determine the cascade threshold while minimizing cost. To perform AT queries following the model cascade framework, first, a subset S, $S \subseteq D$, is labeled and used to determine the cascade threshold, ρ. Given such a threshold, the set $D^\rho \setminus S$ is labeled by the proxy while the set $D \setminus (S \cup D^\rho)$ is labeled by the oracle. We denote the set of estimated labels by $\hat{Y} = \{\mathscr{P}(x_i); x_i \in D^\rho \setminus S\} \cup \{\mathscr{O}(x_i); x_i \in (D \setminus D^\rho) \cup S)\}$. The cost, i.e., the total number of oracle calls, is $C = |D| - |D^\rho \setminus S|$. Thus, under model cascade, for AT queries, we need to find a cascade threshold, ρ, by labeling a subset S of D with the oracle, such that $\hat{Y}$ as defined above meets $\mathbb{P}(\mathcal{A}(\hat{Y}) \geq T) \geq 1 - \delta$, while cost, $C = |D| - |D^\rho \setminus S|$, is minimized. For convenience of notation, for a set $S \subseteq D$, we define:

$$\mathcal{A}_S(\rho) = \frac{\sum_{x \in S^\rho} \mathbb{I}[\mathscr{O}(x) = \mathscr{P}(x)]}{|S^\rho|}.$$

$\mathcal{A}_S(\rho)$ is the accuracy of the proxy model *when processing records in S^ρ only*. We study $\mathcal{A}_D(\rho)$ to provide our theoretical guarantees.

4.1.2 BARGAIN$_A$-A.. Our solution to AT queries is similar to PT queries. We use the same adaptive sampling procedure as BARGAIN$_P$-A, but our method differs from BARGAIN$_P$-A in two ways. First, we estimate accuracy at each iteration, not precision, by defining an estimation function $\mathcal{E}_A^{\text{BARGAIN}}$, analogous to $\mathcal{E}^{\text{BARGAIN}}$, but instead to estimate accuracy. Second, recall that for PT queries, we continuously sampled records until we ran out of oracle budget. For AT queries, we do not have a fixed budget. Instead, we stop sampling when we determine *it's not worth sampling more to obtain a better estimate* at the threshold under consideration. We first discuss our estimation function before presenting the BARGAIN$_A$-A algorithm.

Estimation. We present a new estimation function $\mathcal{E}_A^{\text{BARGAIN}}$ to estimate accuracy, in place of $\mathcal{E}^{\text{BARGAIN}}$ for precision. Recall that our adaptive sampling procedure for PT queries, when considering a threshold ρ, samples an $x \in S$ uniformly from D^ρ (see line 8 in Alg. 2). Using the same sampling procedure and random variables $x \in S$, but considering $\mathbb{I}[\mathcal{O}(x) = \mathcal{P}(x))]$, we have

$$\mathbb{E}[\mathbb{I}[\mathcal{O}(x) = \mathcal{P}(x))]] = \sum_{x' \in D^\rho} \frac{\mathbb{I}[\mathcal{O}(x') = \mathcal{P}(x'))]}{|D^\rho|} = \mathcal{A}_D(\rho).$$

Thus, defining the set, $S_A^\rho = \{\mathbb{I}[\mathcal{O}(x) = \mathcal{P}(x))]; x \in S^\rho\}$, to estimate whether $\mathcal{A}_D(\rho) \geq T$ at a threshold ρ we use the observations in S_A^ρ to estimate their true mean:

$$\mathcal{E}_A^{\text{BARGAIN}}(S, T, \rho, \alpha) = \mathcal{T}(T, S_A^\rho, \alpha),$$

where $\mathcal{T}$ is the hypothesis test from Lemma 2.1. $\mathcal{E}_A^{\text{BARGAIN}}$ returns 1 if ρ is estimated to meet the target. It provides the same guarantees for AT (see technical report [44]) as $\mathcal{E}^{\text{BARGAIN}}$ for PT queries.

BARGAIN$_A$-A Algorithm. We present BARGAIN$_A$-A in Alg. 3. The algorithm follows the same sampling procedure as BARGAIN$_P$-A, iteratively considering candidate thresholds in decreasing order and estimating whether each threshold meets the accuracy target. It now uses $\mathcal{E}_A^{\text{BARGAIN}}$ as the estimation function instead of $\mathcal{E}^{\text{BARGAIN}}$ to estimate accuracy. Furthermore, as the algorithm iterates through the thresholds, we stop sampling at a threshold ρ if $\texttt{avg}(S_A^\rho) - \texttt{std}(S_A^\rho) \geq T$, but do so only after $|S_A^\rho| \geq c$ for some parameter c so that the mean and standard deviation of S_A^ρ are meaningful. This follows the intuition that if T is within one standard deviation of the mean of S_A^ρ, it will be difficult to estimate if the true mean, $\mathcal{A}_D(\rho)$, of the observation in S_A^ρ, meets the target without needing a large number of new samples. Thus, we terminate the algorithm and return the smallest threshold estimated to meet the target so far as the candidate threshold. This modification is done in Line 6 of Alg. 3 to decide if the algorithm should continue sampling or not. BARGAIN$_A$-A provides the required theoretical guarantees.

4.1.3 BARGAIN$_A$-M and Other Details. We design BARGAIN$_A$-M, an extension to BARGAIN$_A$-A for classification tasks to *set the cascade thresholds per class* which may help when proxy scores for some classes are more helpful than others. If there are r classes, BARGAIN$_A$-M runs BARGAIN$_A$-A r times, and determines a different cascade threshold for each class. When the proxy predicts a class, the cascade threshold for that specific class is used for prediction. Details are discussed in our technical report [44]. Finally, our technical report [44] also discusses how to extend the analysis to take the data records labeled by the oracle into account when analyzing the final accuracy of the algorithms to improve utility.

4.2 BARGAIN$_R$ for RT Queries

Next, we discuss RT queries; we start with uniform samples and then extend the discussion to adaptive sampling.

4.2.1 BARGAIN$_R$-U with Uniform Samples. Consider a uniform sample S, and denote by $S_+ = \{x; x \in S, \mathcal{O}(x) = 1\}$, subset of S with positive labels, and define D_+ analogously for D. We next describe BARGAIN$_R$-U's estimation and threshold selection procedures.

Estimation. Here, we define an estimation function, $\mathcal{E}_R^{\text{BARGAIN}}$ to estimate whether a candidate threshold meets the recall target or not. To do so, note that a random variable $x \in S_+$, i.e., an observed sample with positive label, is a uniform sample from D_+ so

$$\mathbb{E}[\mathbb{I}[\mathcal{S}(x) \geq \rho]] = \sum_{x' \in D_+} \frac{\mathbb{I}[\mathcal{S}(x') \geq \rho]}{|D^+|} = \mathcal{R}_D(\rho).$$

Thus, the set $S_+^\rho = \{\mathbb{I}[\mathcal{S}(x) \geq \rho]; x \in S_+\}$, consists of i.i.d random variables with mean $\mathcal{R}_D(\rho)$. We define $\mathcal{E}_R^{\text{BARGAIN}}$, analogous to $\mathcal{E}^{\text{BARGAIN}}$, to use observations in S_+^ρ to estimate their true mean:

$$\mathcal{E}_R^{\text{BARGAIN}}(S, T, \rho, \alpha) = \mathcal{T}(T, S_+^\rho, \alpha),$$

where $\mathcal{T}$ is the hypothesis test from Lemma 2.1. $\mathcal{E}_R^{\text{BARGAIN}}$ returns 1 if ρ is estimated to meet the target, i.e., $\mathcal{R}_D(\rho) \geq T$. It provides the same guarantees for RT as $\mathcal{E}^{\text{BARGAIN}}$ for PT queries.

Threshold Selection. BARGAIN$_R$-U selects cascade threshold

$$\rho_S^{\text{BARGAIN}_R-U} = \max\{\rho; \rho \in \mathrm{C}, \mathcal{E}_R^{\text{BARGAIN}}(S, T, \rho, \delta) = 1\}, \tag{13}$$

the largest ρ for which $\mathcal{E}_R^{\text{BARGAIN}}$ returns 1. To maximize precision, Eq. 13 selects the maximum over the thresholds since larger thresholds are expected to have higher precision (as Fig. 6 shows). ρ_S^{BARGAIN} provides the required theoretical guarantees. The proof shows that since recall is a monotonically decreasing function, we can take the minimum over $\rho \in \mathrm{C}$ without splitting the failure probability into $\frac{\delta}{|\mathrm{C}|}$ for each application of $\mathcal{E}_R^{\text{BARGAIN}}$.

4.2.2 BARGAIN$_R$-A with Adaptive Samples. When the total number of positives in D is much smaller than n, a uniform sample will contain few, if any, positive labels, and BARGAIN$_R$-U will likely select a low-utility threshold. BARGAIN$_R$-A complements BARGAIN$_R$-U with a pre-filtering step to focus sampling only on records expected to contain positive labels using the notion of *positive density* that quantifies how positive labels are distributed. We first discuss *positive density* before presenting BARGAIN$_R$-A.

Positive Density. To study how positive labels are distributed in our dataset, define *positive density* at a threshold ρ, $d_r(\rho)$ as

$$d_r(\rho) = \frac{\sum_{x \in D_r^\rho} \mathbb{I}[\mathcal{O}(x) = 1]}{|D_r^\rho|}, \; D_r^\rho = \{x; x \in D, \mathcal{S}(x) \in [\rho, \rho + r)\}.$$

where D_r^ρ contains the records *near* the threshold ρ for a *resolution parameter* r, and positive density denotes the fraction of records *near* ρ (i.e., in D_r^ρ) that are positive. Positive density can be seen as an approximation to $\mathbb{P}(\mathcal{O}(x) = 1 | \mathcal{S}(x) = \rho)$ (known as correctness likelihood [39]), the probability that a sample is positive given proxy score ρ, where the resolution r is used to include enough records for this approximation. Fig. 9 shows positive density for multiple real-world datasets. For datasets Onto and Imagenet, positive density for most proxy scores is zero, with non-zero density only at very large scores. BARGAIN$_R$-A uses this observation to improve utility.

BARGAIN$_R$-A Algorithm. BARGAIN$_R$-A takes advantage of this low positive density at small proxy scores to find a *cutoff*, ρ_P, on proxy scores, such that positive labels are expected to have proxy scores only in $[\rho_P, 1]$. It then runs BARGAIN$_R$-U on D^{ρ_P} (subset of D with proxy score at least ρ_P) to obtain the cascade threshold. BARGAIN$_R$-A finds ρ_P, by estimating $d_r(\rho)$ at different thresholds to find the subset of D with high positive density. It sets ρ_P as the largest cutoff where any threshold ρ with *high positive density*, that is $d_r(\rho) \geq \beta$ for a *minimum positive density* parameter $\beta \geq 0$, is estimated to be in the range $\rho \in [\rho_P, 1]$. We discuss the role of β later.

BARGAIN$_R$-A is presented in Alg. 4. It consists of two stages. The first stage performs a binary search on proxy scores to find ρ_P, and the second stage runs BARGAIN$_R$-U on D^{ρ_P} to find the cascade threshold. To find the cutoff, ρ_P, using binary search BARGAIN$_R$-A estimates whether $d_r(\rho) < \beta$ starting from the midpoint of possible values, $\rho = 0.5$. To do so, it uses a function $\mathcal{E}_d^{\text{BARGAIN}}$, analogous to $\mathcal{E}^{\text{BARGAIN}}$ but now to estimate positive density, discussed later. If at any iteration $\mathcal{E}_d^{\text{BARGAIN}}$ returns 1, the algorithm estimates that ρ has low positive density, and it proceeds to check the density at $(1 + \rho)/2$. This continues until the algorithm runs out of sampling budget or if it finds a threshold where it estimates that $d_r(\rho) \geq \beta$. Finally, the algorithm sets ρ_P as the largest

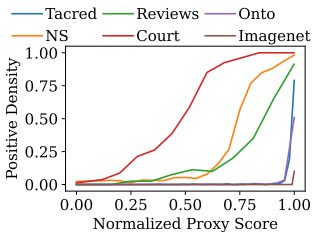


Fig. 9. Positive density in real-world datasets

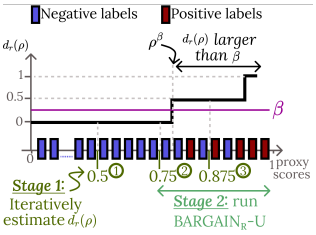


Fig. 10. BARGAIN$_R$-A example

Algorithm 4 BARGAIN$_R$-A

$k_1, k_2 \leftarrow k/2$
$\delta_1, \delta_2 \leftarrow \delta/2$
$\rho_P \leftarrow 0, \rho \leftarrow 0.5$
while $k_1 \geq 0$ **do**
  $S \leftarrow \emptyset$
  while $\mathcal{E}_d^{\text{BARGAIN}}(S, \beta, \rho, \delta_1) = 0$ **and** $k_1 \geq 0$ **do**
    Sample a record uniformly from D_r^ρ and add to S
    $k_1 -= 1$
  if $\mathcal{E}_d^{\text{BARGAIN}}(S, \beta, \rho, \delta_1) = 0$ **then**
    break
  $\rho_P \leftarrow \rho, \rho \leftarrow (1+\rho)/2$
return BARGAIN$_R$-U sampling k_2 points over $[\rho_P, 1]$ with δ_2

ρ where it estimated $d_r(\rho) < \beta$. It runs BARGAIN$_R$-U on D^{ρ_P} to find the final cascade threshold. BARGAIN$_R$-A assigns half of the sampling budget to the binary search and half to find the cascade threshold. It also divides δ into two and uses half for binary search for the cutoff and half for finding cascade threshold to bound the total probability of failure. An example of BARGAIN$_R$-A is presented in Fig. 10 (r is set so that D_r^ρ contains 2 points at all ρ), where we see BARGAIN$_R$-A first estimates $d_r(\rho)$ at $\rho = 0.5$, $\rho = 0.75$ (where $d_r(\rho) < \beta$), and finally at $\rho = 0.875$ where $d_r(\rho) \geq \beta$ so the search stops and BARGAIN$_R$-A selects $\rho_P = 0.75$. It then runs BARGAIN$_R$-U on points with proxy scores larger than 0.75.

$d_r(\rho)$ can be seen as the precision of the set D_r^ρ, and we estimate whether $d_r(\rho) \geq \beta$ using the function $\mathcal{E}_d^{\text{BARGAIN}}(S, \beta, \rho, \alpha)$, defined analogously to $\mathcal{E}^{\text{BARGAIN}}$. We defer exact definition of $\mathcal{E}_d^{\text{BARGAIN}}$ to our technical report [44]. Here, we note that $\mathcal{E}_d^{\text{BARGAIN}}$ guarantees that, if $d_r(\rho) \geq \beta$ and given a sample set S, $\mathbb{P}(\mathcal{E}_d^{\text{BARGAIN}}(S, \beta, \rho, \alpha) = 1) \leq \alpha$, for any confidence parameter α. That is, if ρ meets the minimum positive density, $\mathcal{E}_d^{\text{BARGAIN}}$ is unlikely to return 1.

Guarantees and Role of β. The guarantees BARGAIN$_R$-A provides depends on β. If $\beta = 0$, BARGAIN$_R$-A provides the same guarantees as BARGAIN$_R$-U, but is also unlikely to improve on its utility, leading to poor utility when the total number of positive labels is low in a dataset. In fact, in our technical report [44], we show the following impossibility result: consider *any approach* that samples records with probability monotonically increasing in the proxy scores and guarantees meeting the recall target on all possible datasets; such an approach *must provide low precision for datasets with few positive records*. Due to this impossibility result, we allow users to optionally set $\beta > 0$ to improve the utility of the solution but weaken the theoretical guarantees. To understand the role of β, for a dataset, D, let $\rho^\beta \in [0, 1]$ be the smallest proxy score such that $d_r(\rho) \geq \beta$ for all $\rho \in [\rho^\beta, 1]$ (Fig. 10 shows an example). Define $\texttt{dense}_\beta(D) = \{x; x \in D, \mathcal{S}(x) \geq \rho^\beta\}$, and $\mathcal{R}_D^\beta(\rho) = \mathcal{R}_{\texttt{dense}_\beta(D)}(\rho)$, that is, $\mathcal{R}_D^\beta$ is the recall on the set $\texttt{dense}_\beta(D)$, a subset of D with minimum positive density β at all proxy scores. BARGAIN$_R$-A guarantees $\mathcal{R}_D^\beta(\rho) \geq T$ with high probability,

	Review	Court	Screen	Wiki	Onto	Imagenet	Tacred	NS
$\mathbf{n}$	855	1,000	1,000	1,000	11,165	50,000	22,631	973,085
$\frac{\mathbf{n}^+}{\mathbf{n}}$	0.23	0.59	0.22	0.25	0.02	0.001	0.02	0.29

Table 4. Dataset characteristics

but can potentially ignore any positive labels in D but not in $\mathsf{dense}_\beta(D)$, i.e., positive labels present where positive density is low. In real-world datasets, as Fig. 9 shows, positive labels are densely distributed towards high proxy scores so that $\mathcal{R}_D^\beta(\rho)$ and $\mathcal{R}_D(\rho)$ are often very similar for small β. As such, our experiments show that setting β as a fixed small value across all datasets allows us to obtain $\mathcal{R}_D(\rho) \geq T$ in practice. Sec. 6.3 empirically evaluates the impact of β.

5 Parameter Setting

Here, we discuss the role of system parameters in BARGAIN.

Number of Candidate Thresholds, M. M controls the number of candidate thresholds BARGAIN$_A$ and BARGAIN$_P$ consider. Increasing M (i.e., having more candidate thresholds) increases the chance of finding a threshold with high utility; however, considering many thresholds requires more samples and could be wasteful if most do not improve utility. To understand the trade-offs, note that there are most $\frac{n}{M}$ records between any two consecutive candidate thresholds in C_M (by definition, see Eq. 12), and that BARGAIN returns the i-th threshold $\rho_i \in \mathrm{C}_M$ when the $i+1$-th threshold, ρ_{i+1} is estimated to not meet the target. If the precision or accuracy of the proxy monotonically increases in the proxy scores (which, according to Fig. 6, typically holds in practice), then increasing M can lead to choosing a threshold between ρ_i and ρ_{i+1}, but not smaller. This means that, for AT queries (similar argument holds for PT queries, see our technical report [44]), the fraction of records processed by the proxy can improve by $\frac{1}{M}$ at most, showing diminishing returns as M increases (e.g., any $M > 20$ leads to at most 5% increase in utility over $M = 20$). So large values of M are unlikely to provide significantly better utility and the cost incurred by sampling more records to evaluate many thresholds when M is large may offset any such gains. We show this empirically in our technical report [44] and recommend $M = 20$ as the default value.

Minimum Number of Samples per Threshold, c. The parameter c controls the minimum number of samples taken by BARGAIN$_A$ variants at a threshold before it decides a threshold does not meet the target. If c is too small, the algorithm might prematurely decide a threshold does not meet the target (before having observed sufficient samples), while if c is too large it might spend too many samples at a threshold that does not meet the target, thus wasting samples. Setting c as a small constant fraction of data size ensures that not too many samples are wasted during estimation relative to total processing cost. This is supported by our experiments (see our technical report [44]) that show setting c to 1% to 5% of data size performs well across datasets, and in general, utility of BARGAIN is not very sensitive to c as long as it is not very large compared with data size.

Tolerance Parameter, η. As BARGAIN$_A$ and BARGAIN$_P$ variants iterate through candidate thresholds, they terminate after they consider η thresholds that do not meet the target (see Eq. 11). η is a tolerance parameter that can be set based on whether the quality metric is expected to be monotonic or not. That is, whether, after estimating that a threshold ρ does not meet the target, thresholds smaller than ρ are expected to also not meet the target. In real-world datasets, and as discussed in Sec. 3.2.2, this monotonicity property is expected to hold, and as such we set $\eta = 0$ by default. Experiments in our technical report [44] validate this.

	Reviews	Court Opinion	Screenplay	Wiki Talk	Onto	Imagenet	Tacred	NS
			(a) Percentage of Oracle Calls Avoided for AT Queries					
SUPG	3.2	24.4	1.5	4.4	73.5	84.7	80.1	7.5
Naive	0.0 (-100.0)	0.0 (-100.0)	6.8 (+362)	0.0 (-100.0)	73.6 (+0.1)	84.6 (-0.1)	79.9 (-0.2)	0.0 (-100.0)
BARGAIN$_A$-A	**41.8 (+1218)**	48.0 (+96.5)	0.0 (-100.0)	**48.1 (+984)**	97.7 (+32.9)	**99.7 (+17.8)**	97.0 (+21.2)	65.3 (+765)
BARGAIN$_A$-M	36.7 (+1060)	**58.6 (+139)**	**11.1 (+655)**	42.9 (+865)	**98.9 (+34.6)**	**99.9 (+18.0)**	**99.2 (+24.0)**	**75.7 (+903)**
			(b) Observed Recall for PT Queries					
SUPG	59.4	75.6	44.2	54.9	49.1	89.7	33.5	10.2
Naive	47.0 (-20.9)	39.9 (-47.2)	40.2 (-9.1)	40.3 (-26.6)	3.4 (-93.1)	0.7 (-99.2)	1.7 (-95.0)	3.9 (-61.6)
BARGAIN$_P$-U	54.2 (-8.7)	**86.5 (+14.4)**	40.2 (-9.2)	43.1 (-21.5)	3.5 (-92.8)	1.0 (-98.9)	1.7 (-95.1)	4.1 (-60.1)
BARGAIN$_P$-A	**89.0 (+49.8)**	84.2 (+11.4)	**73.4 (+65.9)**	**86.3 (+57.2)**	**88.2 (+79.8)**	**100 (+11.5)**	**61.5 (+83.3)**	**44.6 (+337)**
			(c) Observed Precision for RT Queries					
SUPG	34.3	77.5	27.9	45.9	13.2 [MT]	90.3	11.1	45.3
Naive	22.9 (-33.1)	74.8 (-3.5)	21.8 (-21.8)	24.8 (-46.0)	2.5 (-81.0)	0.1 (-99.9)	2.4 (-78.8)	31.3 (-31.0)
BARGAIN$_R$-U	**40.9 (+19.2)**	**82.7 (+6.7)**	**32.6 (+16.8)**	**52.3 (+14.0)**	2.5 (-81.0)	0.1 (-99.9)	2.4 (-78.8)	**59.1 (+30.5)**
BARGAIN$_R$-A	39.7 (+15.9)	**82.0 (+5.9)**	**33.0 (+18.2)**	50.9 (+10.9)	**28.0 (+112)**	**97.8 (+8.3)**	**22.0 (+97.8)**	56.1 (+23.8)

Table 5. Observed Utility for AT, PT and RT Queries with Target $T = 0.9$. +/- shows percentage change over SUPG, [MT] means the method missed the target, and all methods otherwise met the target.

6 Experiments

We present our experimental setup in Sec 6.1, comparison with baselines in Sec. 6.2, analysis of sensitivity of approaches to user parameters in Sec. 6.3, and robustness of the approaches to random noise and adversarial settings in Sec. 6.4.

6.1 Setup

Datasets and Tasks. We perform experiments on 8 different datasets. We use all 4 of the datasets used in [21] (obtained from [20]), Imagenet, night-street (NS), Tacred, and OntoNotes (Onto), with the first two datasets on image classification, and the latter two on text classification. These datasets use non-LLM deep learning models for image and text processing. To evaluate our approach for data processing using LLMs, we additionally consider 4 new datasets: Reviews (Steam game reviews, obtained from Kaggle [17]), Court (US court opinions since 1970 [8]), Screenplay (popular movie scripts from Kaggle [16]), and Wiki (Wikipedia talk page discussions amongst editors [6]). We randomly sampled 1,000 examples from each source, though Reviews contains only 855 examples because many reviews did not pass the OpenAI toxicity filter. The LLM tasks involve determining game reference comparisons (Reviews), court ruling reversals (Court), decisions based on false information (Screenplay), and whether discussions led to edit reversions (Wiki). The specific prompts used are described in our technical report [44]. Throughout, gpt4o-mini is used as the proxy model and gpt-4o as the oracle. Table 4 shows the number of records, together with $\frac{n^+}{n}$ denoting the fraction of records that have positive labels in each dataset.

Methods. We compare BARGAIN against two baselines: SUPG [21], the state-of-the-art model cascade method with *asymptotic* guarantees, and Naive, a baseline we designed to provide the same guarantees as BARGAIN, but using simple statistical tools. We use the open-source implementation of SUPG [20]. SUPG was only designed for PT and RT queries, and uses importance sampling independent of target and data characteristics. To extend it to AT queries, we use the PT method to solve the AT query by changing the calculated metric during estimation. Since AT queries do not use an apriori budget while SUPG takes a fixed budget as input, we run SUPG for 10 different budgets and report the result with the best utility. Naive, for all queries, takes a uniform sample and uses Hoeffding's inequality to estimate quality, as in Sec. 3.1. For AT queries, we use the same method as for SUPG discussed above to determine the sample size for Naive. We additionally performed experiments using an alternative naive variant that uses Eq. 11 (with $\eta = 0$). This variant performed similarly to the one in Sec. 3.1; results are presented in our technical report [44].

Metrics. To evaluate the methods for PT and RT queries, we respectively report the true recall and precision for the methods at the threshold returned by the algorithm. For AT queries, we

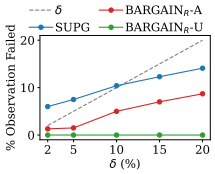


Fig. 11. Meeting target in Onto Dataset

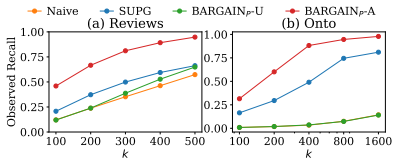


Fig. 12. Impact of k in PT Queries

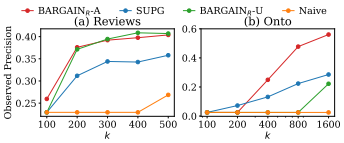


Fig. 13. Impact of k in RT Queries

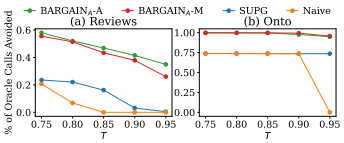


Fig. 14. Impact of T in AT Queries

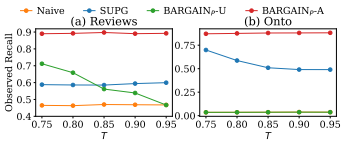


Fig. 15. Impact of T in PT Queries

count the total number of points, n_{proxy} that only the proxy was evaluated on, and report the fraction $\frac{n_{proxy}}{n}$ as the *percentage of oracle calls avoided*. Reported results are averaged across 50 runs. We performed 50 runs to empirically evaluate whether the methods meet the quality target with the desired probability. We also report observed variances across the 50 runs in our technical report [44]; our results show BARGAIN has lower variance than SUPG.

Default Parameters. Unless otherwise stated, we set $T = 0.9$, $\delta = 0.1$ and $k = 400$. For BARGAIN$_P$ variants and Naive $M = 20$ for PT and AT queries, and for BARGAIN$_R$-A, we set $\beta = 0.02$ and $r = 150$. We show the impact of parameters in Sec. 6.3, with further results in our technical report [44]. We use SUPG with default parameters [21].

6.2 Comparison across Datasets

Table 5 shows baseline comparison across all datasets, showing significant benefits to BARGAIN over SUPG and the Naive method. First, consider AT queries. We see that both BARGAIN$_A$ variants significantly reduce oracle usage compared with SUPG. For many datasets, SUPG only uses the proxy model prediction less than 10% of the time, while **BARGAIN increases proxy model usage up to 10× over SUPG**. Note that there is no clear winner between BARGAIN$_A$-A and BARGAIN$_A$-M, where BARGAIN$_A$-A chooses a single threshold for all classes while BARGAIN$_A$-M chooses different thresholds for different classes. BARGAIN$_A$-A performs better when a single cascade threshold for all classes provides sufficiently good utility, since this threshold can be determined with fewer labels. However, if a single threshold cannot be applied to all classes (e.g., in Screenplay), BARGAIN$_A$-M, which chooses a per-class threshold, will perform better.

For PT and RT queries, we see BARGAIN$_P$-A and BARGAIN$_R$-A significantly improve upon SUPG. We see that BARGAIN$_P$-U and BARGAIN$_R$-U, the BARGAIN variants with uniform sampling do improve upon the Naive approach but, depending on the dataset, often perform worse than SUPG which uses importance sampling. This is particularly visible on Onto, Imagenet and Tacred datasets that, as Table 4 shows, have very low number of positive labels compared to the full dataset, and thus, uniform sampling is unlikely to find where the positives are located. Nonetheless, improved sampling procedures in BARGAIN$_P$-A and BARGAIN$_R$-A help BARGAIN provide the significant advantage over SUPG, **improving recall in PT queries by up to 80% and precision in RT queries by up to 98%.**

All results in Table 5 empirically meet the required target except for SUPG on Onto dataset for RT queries. Fig. 11 further illustrates this, showing percentage of runs across 1,000 runs where SUPG returned recall below the target, $T = 0.9$. When δ is small, i.e., the allowed failure probability is low, SUPG fails to provide the desired guarantees. In Sec. 6.4, we show that this is not an isolated incident and construct datasets where SUPG frequently misses the target.

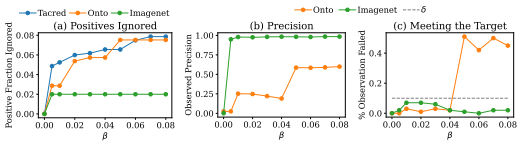


Fig. 16. Impact of β

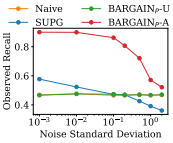


Fig. 17. Impact of noise in PT Queries

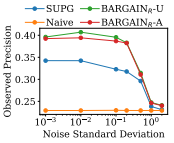


Fig. 18. Impact of noise in RT Queries

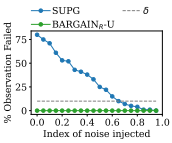


Fig. 19. Meeting target in Noisy Data

6.3 Sensitivity to Parameters

Here, we use Review, our dataset for LLM-powered data processing, and Onto, a dataset from [21] with a low positive rate (see Table 4 to study the sensitivity of approaches to user parameters.

Varying k. Fig. 12-13 show the impact of budget size, k, for AT and PT queries on two datasets, Reviews and Onto. BARGAIN$_P$-A and BARGAIN$_R$-A outperform SUPG across budget sizes, although the gap between methods is smaller when the budget size is small, cases where no method provide good utility due to limited budget.

Varying T. Figs. 14-15 show the sensitivity of BARGAIN to the target T for AT and PT queries (results for RT queries are similar and omitted here for the sake of space) on Reviews and Onto datasets. BARGAIN outperforms SUPG across values of T.

Varying β. For RT queries, BARGAIN$_R$-A depends on the parameter β, the minimum positive density parameter. For datasets with large positive rates, i.e., all datasets except Tacred, Onto and Imagenet, any small value of β, e.g., $\beta < 0.1$, does not impact the performance. The dataset's positive density at most thresholds is larger than β, so BARGAIN$_R$-A simply performs BARGAIN$_R$-U on the entire domain without reducing the range (i.e., sets $\rho^{\beta} = 0$). Here, we investigate the impact of β on datasets with low positive rates, i.e., Tacred, Onto, and Imagenet. First, in Fig. 16 (a), we show what fraction of the positive records may be *ignored* by BARGAIN$_R$-A at a particular β, that is the ratio between the number of positive labels in $D \setminus \text{dense}_{\beta}(D)$ and the total number of positives in D (see Sec. 4.2 for definitions). As Fig. 16 (a) shows, in real-world datasets, only a small fraction of the positive labels are not present in a dense subset of D, justifying use of dense positive labels to provide guarantees in real-world settings. Fig. 16 (b) and (c) evaluate BARGAIN$_R$-A at different β values (results for Tacred are similar to Onto are omitted for visual clarity), showing significant improvements in precision even at very low β values. However, as β increases, BARGAIN$_R$-A may stop meeting the target, T, as it starts to ignore too many positive points. For Onto, this happens at $\beta \geq 0.05$. We note that even though BARGAIN$_R$-A stops to meet the target at the probability needed by δ, the observed recall is often very close to the target since the deviation from the target is bounded based on the fraction of positive labels ignored by the algorithm, which as Fig. 16 (a) shows, is small.

Other parameters. Additional parameters across BARGAIN variants are M, c and η. Our technical report [44] shows experiments on the impact of these parameters on the utility. We observe that changing the parameters has little impact on the utility of BARGAIN and a large set of values perform well across datasets.

6.4 Robustness

Impact of Random Noise in Proxy Scores. We add Gaussian noise to the proxy scores in the Review datasets to study the impact of model calibration. The larger the magnitude of the noise is, the less correlated the proxy scores will be with the correctness of proxy prediction. The results of this experiment are plotted in Figs. 17 and 18, where we measure utility as we increase the standard deviation of the noise injected. In general, BARGAIN outperforms other methods. When the noise

magnitude becomes large, the proxy scores lose all correlation with proxy correctness, and all approaches perform similarly to Naive and provide little utility.

Adversarial Setting for Meeting the Quality Target. In this experiment, we adversarially modify the Imagenet dataset to show how frequently SUPG, which achieves only asymptotic guarantees, can fail to meet the target and make the case for strong non-asymptotic theoretical guarantees that BARGAIN achieves. Specifically, on the Imagenet dataset, we sort the records in the increasing order of proxy scores and denote the i-th record in this order as x_i. To modify the dataset, we change the label of records x_i,..., x_{i+100} and set them to be positive, and vary i in this experiment. For instance, when $i = 0$, we set the label of the 100 records with the lowest proxy scores to be positive. Fig. 19 shows the result of this experiment. We see that SUPG misses the target more than 75% of the times, significantly more than the 10% allowed (since $\delta = 0.1$). On the other hand, BARGAIN$_P$-U meets the target throughout. Putting this in the context of the results in Table 5, this lack of guarantees for SUPG explains why it can provide high utility on datasets with a low number of positives, but BARGAIN$_P$-U that does provide guarantees provides poor utility. Our method BARGAIN$_R$-A strikes a balance between the two, by allowing the user to quantifiably (through the parameter β) relax the guarantee requirement to provide high utility on real-world datasets, unlike SUPG that may unpredictably fail to meet the target. We also note that although the results here are in a synthetically generated setting, existing work shows that LLMs may be uncalibrated in practice [14, 24], and thus rigorous theoretical guarantees are needed to ensure robustness.

7 Related Work

LLM-Powered Data Management. LLMs have revolutionized data management research and applications [10], reshaping how our community approaches longstanding challenges. LLMs have been broadly used in two ways: (1), developing "point" solutions for challenging problems such as data discovery [11, 41], data extraction and cleaning [3, 25, 29, 30, 38], query planning [36], and text to SQL [33]; and (2) creating flexible query processing frameworks that incorporate LLMs in open-ended ways [1, 26, 27, 32, 35, 37, 40, 43]. BARGAIN reduces costs while guaranteeing quality, making it applicable to any LLM-based component across these systems.

Cost-Efficient ML-Powered Data Processing. Many techniques have been used to reduce the cost of ML-powered data processing. Some strategies include optimizing for specific *types* of queries, like aggregations [18], or building *indexes* to reduce online query processing time [4, 23]. More recently, [15] performs profiling to estimate model accuracies for different tasks to decide which model to use, and Huang et al. [13] ensembles various model answers to generate the final output. *Model cascade*, which routes queries through cheaper proxy models before using expensive oracle models, has been widely adopted in traditional ML and deep learning, particularly for video analytics [2, 5, 9, 18, 19, 28]. In particular, recent work [7, 32] has used this framework for LLM-powered data processing; [7] without providing guarantees while [32] uses SUPG [21] to provide theoretical guarantees.

Kang et al. [21] demonstrated that model cascades approaches that lack theoretical guarantees frequently miss quality targets (e.g., achieving precision below 65% when targeting 90%). This motivated SUPG [21], a method to set cascade thresholds with theoretical guarantees and improved utility over prior work [19, 28], but as already discussed, SUPG's guarantees only hold asymptotically. Our approach, BARGAIN, improves upon SUPG by presenting stronger theoretical guarantees, especially for LLM-powered data processing tasks, and significantly better empirical utility confirmed by our experiments. Current LLM-powered data processing frameworks seeking theoretical guarantees [32] rely on SUPG, underscoring the value of our improved methodology. BARGAIN

can be integrated into any existing LLM-powered data processing frameworks to substantially reduce costs while maintaining quality guarantees.

8 Conclusion

We studied the problem of low-cost LLM-powered data processing through model cascade while providing quality guarantees. We studied accuracy, precision, and recall quality guarantees, and presented BARGAIN to decide the model cascade threshold while providing tight theoretical guarantees and good utility. BARGAIN uses adaptive sampling to label records, hypothesis testing to estimate whether different cascade thresholds meet the target, and choose the cascade threshold based on the estimates with a tight theoretical analysis. We empirically showed BARGAIN significantly improves utility over the state-of-the-art. We plan to extend BARGAIN to open-ended tasks which requires further consideration on how to calculate proxy scores, as well as *semantic join* [32] and entity matching operations. The latter cases can be formulated as a filter on the cross product of two datasets. BARGAIN can be applied as is, but additional optimizations are possible by considering properties the operations (e.g., transitivity in entity matching).

Acknowledgments

We acknowledge support from grants DGE-2243822, IIS-2129008, IIS-1940759, and IIS-1940757 awarded by the National Science Foundation, funds from the State of California, an NDSEG Fellowship, funds from the Alfred P. Sloan Foundation, as well as EPIC lab sponsors: Adobe, Google, G-Research, Microsoft, PromptQL, Sigma Computing, Bridgewater, and Snowflake. Compute credits were provided by Azure, Modal, NSF (via NAIRR), and OpenAI.

References

[1] Eric Anderson, Jonathan Fritz, Austin Lee, Bohou Li, Mark Lindblad, Henry Lindeman, Alex Meyer, Parth Parmar, Tanvi Ranade, Mehul A Shah, et al. 2024. The Design of an LLM-powered Unstructured Analytics System. *arXiv preprint arXiv:2409.00847* (2024).

[2] Michael R Anderson, Michael Cafarella, German Ros, and Thomas F Wenisch. 2019. Physical representation-based predicate optimization for a visual analytics database. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 1466–1477.

[3] Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. Language models enable simple systems for generating structured views of heterogeneous data lakes. *arXiv preprint arXiv:2304.09433* (2023).

[4] Favyen Bastani and Samuel Madden. 2022. OTIF: Efficient tracker pre-processing over large video datasets. In *Proceedings of the 2022 International Conference on Management of Data*. 2091–2104.

[5] Jiashen Cao, Karan Sarkar, Ramyad Hadidi, Joy Arulraj, and Hyesoon Kim. 2022. Figo: Fine-grained query optimization in video analytics. In *Proceedings of the 2022 International conference on management of data*. 559–572.

[6] Jonathan P Chang, Caleb Chiam, Liye Fu, Andrew Wang, Justine Zhang, and Cristian Danescu-Niculescu-Mizil. 2020. ConvoKit: A Toolkit for the Analysis of Conversations. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 57–60.

[7] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176* (2023).

[8] courtlistener. [n.d.]. Court Opinion. Retrieved April, 2025 from courtlistener.com

[9] Dujian Ding, Sihem Amer-Yahia, and Laks Lakshmanan. 2022. On Efficient Approximate Queries over Machine Learning Models. *Proceedings of the VLDB Endowment (PVLDB)* 16, 4 (2022), 918–931.

[10] Raul Castro Fernandez, Aaron J Elmore, Michael J Franklin, Sanjay Krishnan, and Chenhao Tan. 2023. How large language models will disrupt data management. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3302–3309.

[11] Juliana Freire, Grace Fan, Benjamin Feuer, Christos Koutras, Yurong Liu, Eduardo Pena, Aécio Santos, Cláudio Silva, and Eden Wu. [n.d.]. Large Language Models for Data Discovery and Integration: Challenges and Opportunities. *Data Engineering* ([n. d.]), 3.

[12] Wassily Hoeffding. 1994. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding* (1994), 409–426.

[13] Keke Huang, Yimin Shi, Dujian Ding, Yifei Li, Yang Fei, Laks Lakshmanan, and Xiaokui Xiao. 2025. ThriftLLM: On Cost-Effective Selection of Large Language Models for Classification Queries. *arXiv preprint arXiv:2501.04901* (2025).
[14] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics* 9 (2021), 962–977.
[15] Saehan Jo and Immanuel Trummer. 2024. SMART: Automatically Scaling Down Language Models with Accuracy Guarantees for Reduced Processing Fees. *arXiv preprint arXiv:2403.13835* (2024).
[16] Kaggle. [n.d.]. Screenplay. Retrieved April, 2025 from https://www.kaggle.com/datasets/gufukuro/movie-scripts-corpus
[17] Kaggle. [n.d.]. Steam game reviews. Retrieved April, 2025 from https://www.kaggle.com/datasets/najzeko/steam-reviews-2021
[18] Daniel Kang, Peter Bailis, and Matei Zaharia. [n.d.]. BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. *Proceedings of the VLDB Endowment* 13, 4 ([n. d.]).
[19] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: Optimizing Neural Network Queries over Video at Scale. *Proceedings of the VLDB Endowment* 10, 11 (2017).
[20] Daniel Kang, Edward Gan, Peter Bailis, Tatsunori Hashimoto, and Matei Zaharia. [n.d.]. SUPG code and datasets. Retrieved Mar, 2025 from https://github.com/stanford-futuredata/supg
[21] Daniel Kang, Edward Gan, Peter Bailis, Tatsunori Hashimoto, and Matei Zaharia. 2020. Approximate Selection with Guarantees using Proxies. *Proceedings of the VLDB Endowment* 13, 11 (2020).
[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 (2021), 2341–2354. https://api.semanticscholar.org/CorpusID:237012342
[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*. 1934–1947.
[24] Sanyam Kapoor, Nate Gruver, Manley Roberts, Arka Pal, Samuel Dooley, Micah Goldblum, and Andrew Wilson. 2024. Calibration-tuning: Teaching large language models to know what they don't know. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*. 1–14.
[25] Alexander W Lee, Justin Chan, Michael Fu, Nicolas Kim, Akshay Mehta, Deepti Raghavan, and Ugur Cetintemel. 2025. Semantic Integrity Constraints: Declarative Guardrails for AI-Augmented Data Processing Systems. *arXiv preprint arXiv:2503.00600* (2025).
[26] Yiming Lin, Madelon Hulsebos, Ruiying Ma, Shreya Shankar, Sepanta Zeigham, Aditya G Parameswaran, and Eugene Wu. 2024. Towards accurate and efficient document analytics with large language models. *arXiv preprint arXiv:2405.04674* (2024).
[27] Chunwei Liu, Matthew Russo, Michael Cafarella, Lei Cao, Peter Baille Chen, Zui Chen, Michael Franklin, Tim Kraska, Samuel Madden, and Gerardo Vitagliano. 2024. A declarative system for optimizing ai workloads. *arXiv preprint arXiv:2405.14696* (2024).
[28] 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*. 1493–1508.
[29] Zan Ahmad Naeem, Mohammad Shahmeer Ahmad, Mohamed Eltabakh, Mourad Ouzzani, and Nan Tang. 2024. RetClean: Retrieval-Based Data Cleaning Using LLMs and Data Lakes. *Proceedings of the VLDB Endowment* 17, 12 (2024), 4421–4424.
[30] Avanika Narayan, Ines Chami, Laurel Orr, Simran Arora, and Christopher Ré. 2022. Can foundation models wrangle your data? *arXiv preprint arXiv:2205.09911* (2022).
[31] openai. 2025. OpenAI pricing. Retrieved Mar, 2025 from https://openai.com/api/pricing/
[32] Liana Patel, Siddharth Jha, Carlos Guestrin, and Matei Zaharia. 2024. Lotus: Enabling semantic queries with llms over tables of unstructured and structured data. *arXiv preprint arXiv:2407.11418* (2024).
[33] Mohammadreza Pourreza, Hailong Li, Ruoxi Sun, Yeounoh Chung, Shayan Talaei, Gaurav Tarlok Kakkar, Yu Gan, Amin Saberi, Fatma Ozcan, and Sercan O Arik. 2024. Chase-sql: Multi-path reasoning and preference optimized candidate selection in text-to-sql. *arXiv preprint arXiv:2410.01943* (2024).
[34] Matthew Russo, Tatsunori Hashimoto, Daniel Kang, Yi Sun, and Matei Zaharia. 2023. Accelerating Aggregation Queries on Unstructured Streams of Data. *Proceedings of the VLDB Endowment* 16, 11 (2023), 2897–2910.
[35] Shreya Shankar, Tristan Chambers, Tarak Shah, Aditya G Parameswaran, and Eugene Wu. 2024. DocETL: Agentic Query Rewriting and Evaluation for Complex Document Processing. *arXiv preprint arXiv:2410.12189* (2024).
[36] Matthias Urban and Carsten Binnig. 2024. Demonstrating CAESURA: Language Models as Multi-Modal Query Planners. In *Companion of the 2024 International Conference on Management of Data*. 472–475.

[37] Matthias Urban and Carsten Binnig. 2024. ELEET: Efficient Learned Query Execution over Text and Tables. *Proc. VLDB Endow* 17 (2024), 13.
[38] David Vos, Till Döhmen, and Sebastian Schelter. 2022. Towards parameter-efficient automation of data wrangling tasks with prefix-tuning. In *NeurIPS 2022 First Table Representation Workshop*.
[39] Cheng Wang. 2023. Calibration in deep learning: A survey of the state-of-the-art. *arXiv preprint arXiv:2308.01222* (2023).
[40] Jiayi Wang and Guoliang Li. 2025. Aop: Automated and interactive llm pipeline orchestration for answering complex queries. CIDR.
[41] Qiming Wang and Raul Castro Fernandez. 2023. Solo: Data Discovery Using Natural Language Questions Via A Self-Supervised Approach. *Proceedings of the ACM on Management of Data* 1, 4 (2023), 1–27.
[42] 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 (2024), 1–27.
[43] Sepanta Zeighami, Yiming Lin, Shreya Shankar, and Aditya Parameswaran. 2025. LLM-Powered Proactive Data Systems. *arXiv preprint arXiv:2502.13016* (2025).
[44] Sepanta Zeighami, Shreya Shankar, and Aditya Parameswaran. 2025. Cut Costs, Not Accuracy: LLM-Powered Data Processing with Guarantees (Technical Report). *arXiv preprint arXiv:2509.02896* (2025).

Received April 2025; revised July 2025; accepted August 2025