

# Accelerating Approximate Aggregation Queries with Expensive Predicates

Daniel Kang\*  
Stanford University  
ddkang@stanford.edu

John Guibas\*  
Stanford University  
jtguibas@stanford.edu

Peter Bailis  
Stanford University  
pbailis@stanford.edu

Tatsunori Hashimoto  
Stanford University  
thashim@stanford.edu

Yi Sun  
University of Chicago  
yi.sun@uchicago.edu

Matei Zaharia  
Stanford University  
matei@cs.stanford.edu

## ABSTRACT

Researchers and industry analysts are increasingly interested in computing aggregation queries over large, unstructured datasets with selective predicates that are computed using expensive deep neural networks (DNNs). As these DNNs are expensive and because many applications can tolerate approximate answers, analysts are interested in accelerating these queries via approximations. Unfortunately, standard approximate query processing techniques to accelerate such queries are not applicable because they assume the result of the predicates are available ahead of time. Furthermore, recent work using cheap approximations (i.e., proxies) do not support aggregation queries with predicates.

To accelerate aggregation queries with expensive predicates, we develop and analyze a query processing algorithm that leverages proxies (ABAE). ABAE must account for the key challenge that it may sample records that do not satisfy the predicate. To address this challenge, we first use the proxy to group records into strata so that records satisfying the predicate are ideally grouped into few strata. Given these strata, ABAE uses pilot sampling and plugin estimates to sample according to the optimal allocation. We show that ABAE converges at an optimal rate in a novel analysis of stratified sampling with draws that may not satisfy the predicate. We further show that ABAE outperforms on baselines on six real-world datasets, reducing labeling costs by up to 2.3 $\times$ .

## PVLDB Reference Format:

Daniel Kang\*, John Guibas\*, Peter Bailis, Tatsunori Hashimoto, Yi Sun, and Matei Zaharia. Accelerating Approximate Aggregation Queries with Expensive Predicates. PVLDB, 14(11): 2341 - 2354, 2021.  
doi:10.14778/3476249.3476285

## 1 INTRODUCTION

Analysts are interested in computing statistics over large, unstructured datasets where only a fraction of the data is of interest (i.e., with a selective predicate) with low computational cost. Machine learning (ML) methods are increasingly used to automatically answer such queries. For example, a media studies researcher may be

interested in computing the average viewership (the statistic) of presidential candidates on TV news (the predicate) [28]. To answer such queries, the researcher may deploy an expensive face detection deep neural network (DNN) to find all faces in the dataset and filter by presidential candidates, e.g.,

```
SELECT AVG(views) FROM video  
WHERE contains_candidate(frame, 'Biden')
```

Critically, these DNNs can be expensive to execute. For example, executing a state-of-the-art face detection DNN on the past year of MSNBC News would cost \$262,000 on cloud compute infrastructure (NVIDIA V100, Amazon Web Services) [60]. Due to limited computational budgets, many organizations cannot exhaustively execute these expensive ML methods over the entirety of the dataset.

Fortunately, many applications can tolerate approximations (as is standard in the approximate query processing (AQP) literature [40]) so answering queries does not require exhaustively executing the expensive DNN. As is standard in AQP, a key requirement with approximate answers are statistical guarantees on query results. For example, the media studies researcher may require such guarantees to make precise claims about bias in TV news. Furthermore, these requirements are standard in scientific analyses. As such, we focus on queries with statistical guarantees in this work.

Unfortunately, standard techniques in AQP, ranging from histograms [47], sketches [7], and others [2], assume that the fields used in the predicates are already available, i.e., as structured records in a database. In contrast, we cannot precompute results as an expensive ML method is required to compute the predicate in our setting, e.g., we would have to execute an expensive face detector on every video frame to answer the query above. Recent work has focused on using cheap approximations (i.e., proxy models) to accelerate queries without having to pre-compute expensive DNNs [4, 29, 32–34]. For example, a proxy for presidential candidates might be a cheap classifier in contrast to a full object detection DNN. Unfortunately, existing work either does not provide statistical guarantees on query accuracy (e.g., NoSCOPE [33], Focus [29], Tahoma [4]) or accelerates other query types (e.g., selection queries [34], aggregation queries without predicates [32], and limit queries [32]).

We propose and analyze ABAE (Aggregation with Expensive BinAry PrEDicates), a query processing algorithm leveraging stratified and pilot sampling [39] to accelerate linear aggregation queries (SUM, COUNT, and AVG) with expensive predicates and statistical guarantees on query accuracy. We further extend ABAE to support common aggregation patterns, including queries with multiple predicates and with group by keys.

\* Marked authors contributed equally.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing [info@vldb.org](mailto:info@vldb.org). Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 14, No. 11 ISSN 2150-8097.  
doi:10.14778/3476249.3476285

ABAE leverages two key opportunities to accelerate such queries: proxy models and stratified sampling. That is, ABAE splits the dataset into disjoint groups (strata), samples within strata, and computes a weighted average to obtain the final answer. ABAE must account for three key challenges, as the predicate results are not available ahead of time: 1) strata selection, 2) budget allocation between strata, and 3) stochastic draws (i.e., sampling a record that may not match the predicate). We provide a principled stratification approach, leverage pilot sampling for budget allocation [39], and provide a novel analysis of stratified sampling with stochastic draws that shows that ABAE converges at an optimal rate.

To address strata selection, ABAE uses the proxy model. We assume the proxy provides information about the likelihood of a record satisfying the predicate [4, 29, 33, 34]. Since the proxy does not give information about the statistic, we stratify records by proxy score quantile. Under a mild monotonicity assumption on the proxy [23], this stratification will group records that are approximately equally likely to match the predicate in the same stratum. Intuitively, if the proxy is perfect (i.e., matches the predicate) and is independent of the statistic, this stratification will minimize the sampling variance. While ABAE performs best when given proxy models which approximate the expensive predicate well, ABAE still returns correct answers regardless of proxy model quality.

Given a stratification, our analysis shows that the optimal allocation depends on two key, per-strata quantities: the fraction of records that match the predicate ( $p_k$ ) and the standard deviation of the statistic within a stratum ( $\sigma_k$ ). Concretely, the optimal allocation is proportional to  $\sqrt{p_k}\sigma_k$ . However, we do not know these quantities ahead of time.

ABAE proceeds in two stages to address this challenge. First, ABAE will estimate  $p_k$  and  $\sigma_k$  using a fraction of the total sampling budget. Then, ABAE will allocate the sampling budget using our plug-in estimates of  $p_k$  and  $\sigma_k$ . We prove that ABAE’s algorithm matches the expected error rates of the optimal stratified sampling allocation given the key quantities. Finally, to provide confidence intervals, ABAE uses a bootstrapping procedure which only adds minimal computational overhead.

We also extend ABAE to support group bys (ABAE-GROUPBY) and complex expressions involving multiple Boolean predicates (ABAE-MULTIPRED). To support group by statements, we adapt our sample allocation strategy to minimize the maximum of the expected mean squared error of the groups (minimax error). We show a numerical optimization procedure can recover the optimal allocation for the minimax error. We also support combining multiple expensive predicates and their respective proxy models through negations, conjunctions, and disjunctions.

Finally, a key challenge in leveraging proxy models is to ensure efficient query answers despite potentially poor proxy model quality. Because ABAE always produces valid results, we only need to address efficiency. To address this challenge, we derive a formula which computes the relative gain of using a given proxy. Then, by using a cheap procedure which can estimate the quantities in the formula, ABAE can calculate expected performance gains of proxy models and select the best proxy model at query time.

We evaluate ABAE and its extensions on six real-world datasets spanning text, images, and video. We show that ABAE outperforms uniform sampling by up to 2.3x. We also provide experiments to

show that our methods for creating confidence intervals, executing group by aggregation queries, and forming complex predicates from multiple proxy models outperforms baselines.

In summary, our contributions are:

- (1) We develop ABAE and its extensions, ABAE-MULTIPRED and ABAE-GROUPBY, to accelerate aggregation queries with expensive predicates via proxy models.
- (2) We provide a theoretical analysis of ABAE and show that it matches the expected error of the optimal stratified sampling algorithm asymptotically.
- (3) We evaluate these techniques on text, image, and video datasets, showing that ABAE significantly outperforms uniform sampling.

## 2 OVERVIEW AND QUERY SEMANTICS

### 2.1 Overview

**Target setting.** ABAE targets aggregation queries that contain one or more predicates that are expensive to evaluate. These predicates typically require executing expensive DNNs or querying human labelers. We assume the statistic can be computed in conjunction with the predicates or is cheap to compute. We support aggregation queries targeting AVG, SUM, and COUNT statistics. We do not support other aggregation types, such as COUNT DISTINCT or MAX.

**Proxies.** We further assume access to a proxy model per predicate, which returns a continuous value between 0 and 1. While not necessary for correctness, high quality proxies will return scores that are correlated with the predicate. These proxies can be orders of magnitude cheaper than oracles (e.g., over 4,000 images/second for the proxy vs 3 fps for the oracle [37]). Thus, as is standard in the literature, we assume these proxies are substantially cheaper than the oracle methods so the proxies can be exhaustively executed over the entire dataset [8, 32, 33, 59].

### 2.2 Examples

**TV news.** Consider a media studies researcher studying how the presence of presidential candidate affects viewership. The researcher is willing to query the expensive DNN at most 10,000 times and computes the average viewership with the following:

---

```
SELECT AVG(views) FROM news
WHERE contains_candidate(frame, 'Biden')
ORACLE LIMIT 10,000 USING proxy(frame)
WITH PROBABILITY 0.95
```

---

where `contains_candidate` is computed via a face detection DNN and the proxy may be trained via specialization [33].

**Traffic analysis.** Consider an urban planner studying traffic patterns. The planner is interested in understanding waiting times at traffic lights and executes the following query

---

```
SELECT AVG(count_cars(frame)) FROM video
WHERE count_cars(frame) > 0
AND red_light(frame)
ORACLE LIMIT 1,000 USING proxy(frame)
WITH PROBABILITY 0.95
```

---

where `count_cars` is computed via an object detection DNN and `red_light` is computed by a human labeler. The proxy could be computed via an embedding index for unstructured data [35].

```

SELECT {AVG | SUM | COUNT} ({field | Expr(field)})
FROM table_name WHERE filter_predicate
[GROUP BY key]
ORACLE LIMIT o USING proxy
WITH PROBABILITY p

```

**Figure 1: Syntax for AB<sub>AE</sub>.** Users provide a statistic to compute, an expensive predicate, an oracle limit, proxy scores, and a success probability. As is standard for aggregation queries, users may specify a group by key.

**Analyzing historical newspaper scans.** Consider political scientists that are interested in computing statistics (e.g., fraction of articles with positive sentiment) over editorials (i.e., the predicate) in historical newspaper scans. Computing these statistics requires executing expensive OCR and text processing DNNs.

### 2.3 Query Syntax and Semantics

We show the query syntax for AB<sub>AE</sub> in Figure 1. As with standard AQP systems, AB<sub>AE</sub> accepts a sampling budget and a probability of error and will return an approximate answer to the query and a confidence interval (CI). Our CI semantics are the standard frequentist CI semantics provided by other AQP systems [2]. In particular, our CI semantics are valid regardless of proxy quality.

In contrast to standard AQP systems, AB<sub>AE</sub> assumes that the predicate is expensive to evaluate. We refer to the methods to execute the predicates as “oracles” [32, 34]. These oracles typically involve executing an expensive DNN and post-processing the result, e.g., executing Mask R-CNN to extract object types and positions from a frames of video and filtering by frames that contain at least two cars. Other use cases may require a human labeler. We further assume that the statistic is either cheap to compute or can be extracted by post-processing the oracle results.

To accelerate these queries, the user also provides a proxy function that computes per-record proxy scores for each predicate. These proxy scores are ideally correlated with the result of the predicate and substantially cheaper than the oracle predicates. Nonetheless, our algorithms will provide valid results even if the proxy scores are of poor quality: proxy correlation will only affect performance, not correctness.

AB<sub>AE</sub> aims to return query results that minimize the mean squared error (MSE) between the approximate result and the result when exhaustively executing the query. AB<sub>AE</sub> further aims to return CIs that are as tight as possible while maintaining the probability of success.

### 2.4 Query Formalism

Formally, let  $\mathcal{D} = \{x_i\}$  be the set of data records,  $O(x) \in \{0, 1\}$  be the oracle predicate, and  $X_i = f(x_i) \in \mathbb{R}$  be the expression the query aggregates over. Let  $\mathcal{D}^+ = \{x \in \mathcal{D} : O(x) = 1\}$ . Finally, let  $N$  be the sample budget.

AB<sub>AE</sub> computes  $\mu = \sum_{x \in \mathcal{D}^+} f(x) / |\mathcal{D}^+|$  via an approximation,  $\hat{\mu}$ , with a fixed sampling budget  $N$ . We measure query result quality by the MSE, i.e.,  $|\mu - \hat{\mu}|^2$ . AB<sub>AE</sub> returns a CI  $[\underline{\mu}, \bar{\mu}]$ . AB<sub>AE</sub> further aims to minimize the length of the CI  $\bar{\mu} - \underline{\mu}$  subject to  $\mu \in [\underline{\mu}, \bar{\mu}]$  with

**Table 1: Summary of notation.**

Symbol	Description
$\mathcal{D}$	Universe of data records
$\mathcal{S}$	Stratification, i.e., $k$ strata
$\mathcal{P}(x)$	Proxy model
$N$	User-specified sampling budget
$K$	Number of strata
$O(x)$	Oracle predicate
$X_{k,i}$	$i$ th sample from stratum $k$
$p_k$	Predicate positive rate
$p_{\text{all}}$	$\sum_k p_k$
$w_k$	Normalized $p_k$ , i.e., $p_k / p_{\text{all}}$
$\mu_k$	$\mathbb{E}[X_{i,k}]$
$\mu_{\text{all}}$	$\sum p_k \mu_k / p_{\text{all}}$
$\sigma_k^2$	$\text{Var}[X_{i,k}]$
$N_1$	Number of samples in Stage 1
$N_2$	Number of samples in Stage 2

the specified probability and sample budget, over randomizations of the query procedure.

## 3 ALGORITHM DESCRIPTION AND QUERY PROCESSING

We describe AB<sub>AE</sub> for accelerating aggregation queries with expensive predicates. We first describe accelerating queries with a single predicate. We then describe three natural extensions: queries with a group by key, queries with multiple predicates, and estimating proxy quality.

### 3.1 AB<sub>AE</sub> with a Single Predicate

**Overview.** AB<sub>AE</sub> leverages stratified sampling and pilot sampling [39] to accelerate aggregation queries with expensive predicates. Namely, AB<sub>AE</sub> splits the dataset into disjoint subsets called strata. Then, AB<sub>AE</sub> allocates sampling budget to the strata and combines the per-strata estimates to give the final estimate.

Our setting involves three distinct challenges. First, since not all records satisfy the predicate, we may not sample a valid record. This change, while seemingly simple, changes the optimal allocation and requires new theoretical analysis to prove convergence rates. Second, we must construct the strata without knowing which records satisfy the predicate. Third, we do not know  $p_k$  (the predicate positive rate) and  $\sigma_k$  (the standard deviation), which are necessary for computing the optimal allocation.

To address these issues, we leverage a two-stage sampling algorithm. AB<sub>AE</sub> first estimates the key quantities necessary for optimal allocation:  $p_k$  and  $\sigma_k$  (also known as pilot sampling). AB<sub>AE</sub> then uses these estimates to allocate sampling budget in the Stage 2. We show in Section 4 that AB<sub>AE</sub> achieves an optimal rate.

**Formal description.** Recall that  $p_k$  is the predicate positive rate and that  $\sigma_k^2$  is the variance of the statistic. Furthermore, recall that  $\mathcal{D}$  is the full dataset,  $O(x)$  is the oracle predicate, and  $X_i$  are the samples. Denote  $X_{k,i}$  to be the  $i$ th *positive* sample from stratum  $k$ .

Additionally, denote  $K$  to be the number of strata,  $N_1$  to be the number of samples in Stage 1, and  $N_2$  to be the number of samples in Stage 2, which are parameters to AB<sub>AE</sub>. AB<sub>AE</sub> will compute

**Algorithm 1** Pseudocode for ABAE. ABAE proceeds in two stages. It first estimates  $p_k$  and  $\sigma_k$ . It then samples according to the estimated optimal allocation,  $\hat{T}_k = \sqrt{\hat{p}_k \hat{\sigma}_k} / \sum_{i=1}^K \sqrt{\hat{p}_i \hat{\sigma}_i}$ .

---

```

1: function ABAEINIT( $\mathcal{D}, \mathcal{P}, K$ )
2:    $\mathcal{D} \leftarrow \text{Sort}(\mathcal{D}, \text{key} = \text{lambda } x : \mathcal{P}(x))$ 
3:    $\mathcal{S}_1, \dots, \mathcal{S}_K \leftarrow \text{StratifyByQuantile}(\mathcal{D}, K)$ 
4:   return  $\mathcal{S}$ 
5:
6: function ABAESAMPLE( $\mathcal{S}, \mathcal{O}, K, N_1, N_2, \text{SampleFn}$ )
7:   for each k in  $[1, \dots, K]$  do ▷ Stage 1
8:      $R_k^{(1)} \leftarrow \text{SampleFn}(\mathcal{S}_k, N_1)$  ▷  $R_k$  are sampled records
9:      $X_k^{(1)} \leftarrow \{f(x) \mid x \in R_k^{(1)}, \mathcal{O}(x) = 1\}$ 
10:     $\hat{\mu}_k \leftarrow \sum_{i=1}^{|X_k^{(1)}|} X_{k,i}^{(1)} / |X_k^{(1)}|$  if  $|X_k^{(1)}| > 0$  else 0
11:     $\hat{p}_k \leftarrow |X_k^{(1)}| / |R_k^{(1)}|$ 
12:     $\hat{\sigma}_k^2 \leftarrow \sum_{i=1}^{|X_k^{(1)}|} \frac{(X_{k,i}^{(1)} - \hat{\mu}_k)^2}{|X_k^{(1)}| - 1}$  if  $|X_k^{(1)}| > 1$  else 0
13:   for each k in  $[1, \dots, K]$  do
14:      $\hat{T}_k \leftarrow \sqrt{\hat{p}_k \hat{\sigma}_k} / \sum_{i=1}^K \sqrt{\hat{p}_i \hat{\sigma}_i}$ 
15:   for each k in  $[1, \dots, K]$  do ▷ Stage 2
16:      $R_k^{(2)} \leftarrow R_k^{(1)} + \text{SampleFn}(\mathcal{S}_k, \lfloor N_2 \hat{T}_k \rfloor)$ 
17:      $X_k^{(2)} \leftarrow X_k^{(1)} + \{f(x) \mid x \notin R_k^{(1)}, x \in R_k^{(2)}, \mathcal{O}(x) = 1\}$ 
18:      $\hat{p}_k \leftarrow |X_k^{(2)}| / |R_k^{(2)}|$ 
19:      $\hat{\mu}_k \leftarrow \sum_{i=1}^{|X_k^{(2)}|} X_{k,i}^{(2)} / |X_k^{(2)}|$  if  $|X_k^{(2)}| > 0$  else 0
20:   return  $\sum_{k=1}^K \hat{p}_k \hat{\mu}_k / \sum_{k=1}^K \hat{p}_k, R^{(2)}$ 
21:
22: function ABAE( $\mathcal{D}, \mathcal{O}, \mathcal{P}, K, N_1, N_2$ )
23:    $\mathcal{S} \leftarrow \text{ABAEINIT}(\mathcal{D}, \mathcal{P}, K)$ 
24:    $\text{SampleFn} \leftarrow \text{SampleWithoutReplacement}$ 
25:    $\hat{\mu}, R^{(2)} \leftarrow \text{ABAESAMPLE}(\mathcal{S}, \mathcal{O}, K, N_1, N_2, \text{SampleFn})$ 
26:   return  $\hat{\mu}$ 

```

---

several other quantities, including  $p_{\text{all}} = \sum_k p_k$ ,  $w_k = p_k / p_{\text{all}}$  the normalized  $p_k$ , and  $\mu_k = \mathbb{E}[X_{k,i}]$  the per stratum mean. We summarize the notation in Table 1.

We present the pseudocode for the sampling algorithm in Algorithm 1. ABAE creates the strata by ordering the records by proxy score and splitting into  $K$  strata by quantile.

ABAE will then perform a two-stage sampling procedure. In Stage 1, ABAE samples  $N_1$  samples from each of the  $K$  strata to estimate  $p_k$  and  $\sigma_k$ , which are the key quantities for determining optimal allocation. In Stage 2, ABAE will allocate the remaining samples proportional to our estimates of the optimal allocation.

ABAE construct plugin estimates for  $p_k$  and  $\mu_k$ , denoted  $\hat{p}_k$  and  $\hat{\mu}_k$  respectively. To compute its final estimates, ABAE will use all the samples from Stage 1 and Stage 2 to compute  $\hat{p}_k$  and  $\hat{\mu}_k$ . ABAE will return the estimate  $\sum_k \hat{p}_k \hat{\mu}_k / \sum_k \hat{p}_k$  as the approximate answer.

As the final estimates are sensitive to the estimate of  $p_k$ , i.e.,  $\hat{p}_k$ , we find that reusing samples between stages dramatically improves performance (Section 5.3).

We defer the proofs of convergence and rates to Section 4.

**Algorithm 2** Bootstrap procedure for computing CIs. We resample existing samples over both stages of the algorithm.

---

```

1: function BOOTSTRAP( $R^{(2)}, \mathcal{O}, K, N_1, N_2, \beta, \alpha$ )
2:   for each b in  $[1, \dots, \beta]$  do ▷  $\beta$  is # of bootstrap trials
3:     for each k in  $[1, \dots, K]$  do
4:        $R_k^* \leftarrow \text{SampleWithReplacement}(R_k^{(2)}, |R_k^{(2)}|)$ 
5:        $X_k^* \leftarrow \{f(x) \mid x \in R_k^*, \mathcal{O}(x) = 1\}$ 
6:        $\hat{p}_k^* \leftarrow |X_k^*| / |R_k^*|$ 
7:        $\hat{\mu}_k^* \leftarrow \sum_{i=1}^{|X_k^*|} X_{k,i}^* / |X_k^*|$  if  $|X_k^*| > 0$  else 0
8:        $\hat{\mu}_b \leftarrow \sum_{k=1}^K \hat{p}_k^* \hat{\mu}_k^* / \sum_{k=1}^K \hat{p}_k^*$ 
9:     return  $\text{Percentile}(\alpha/2, \hat{\mu}), \text{Percentile}(1 - \alpha/2, \hat{\mu})$ 
10:
11: function ABAEWITHCI( $\mathcal{D}, \mathcal{O}, \mathcal{P}, K, N_1, N_2, \beta, \alpha$ )
12:    $\mathcal{S} \leftarrow \text{ABAEINIT}(\mathcal{D}, \mathcal{P}, K)$ 
13:    $\text{SampleFn} \leftarrow \text{SampleWithoutReplacement}$ 
14:    $\hat{\mu}, R^{(2)} \leftarrow \text{ABAESAMPLE}(\mathcal{S}, \mathcal{O}, K, N_1, N_2, \text{SampleFn})$ 
15:   return  $\hat{\mu}, \text{BOOTSTRAP}(R^{(2)}, \mathcal{O}, K, N_1, N_2, \beta, \alpha)$ 

```

---

**Confidence intervals.** We use the non-parametric bootstrap [16] to compute confidence intervals, which resamples existing samples. Since the per-stratum samples from both stages of ABAE are independent and identically distributed (i.i.d.), we resample from samples across both stages.

We present the pseudocode for the bootstrap procedure in Algorithm 2. ABAE bootstraps across both stages of the sampling algorithm to form CIs. We formally show the validity of the bootstrap in an extended technical report [36]. We further show that our procedure produces CIs that are nominally correct in Section 5.

In standard AQP, the bootstrap is considered an expensive procedure as it requires resampling and recomputing the statistic. However, in our setting, we assume that the oracle predicate is expensive to execute. As a result, the bootstrap is computationally cheap compared to the cost of obtaining the samples. Concretely, in several of our experiments, executing 1,000 bootstrap trials using unoptimized Python code on a single CPU core is as expensive as executing 2,500 oracle calls on an NVIDIA T4 accelerator, which corresponds to under 0.3% of a medium-sized dataset.

**Setting parameters.** ABAE requires setting two parameters: the fraction of samples between Stage 1 and 2 and the number of strata. We recommend using 30-50% of samples in Stage 1 and  $K$  to be maximal such that every strata receives at least 100 samples in Stage 1. Our experiments show that ABAE is not sensitive to these parameters, but that these settings tend to do best (Section 5.3).

### 3.2 Group Bys

We extend ABAE to support queries with group by keys (ABAE-GROUPBY). We focus on minimizing the maximum error (minimax error) over groups. Other objectives can be supported (e.g., the sum of errors), but we defer optimizing other objectives to future work. As an example of a query with a group by key, consider:

---

```

SELECT COUNT(frame), person FROM VIDEO
WHERE person IN ('Biden', 'Trump')
GROUP BY person

```

---

where, for simplicity, we assume that person is a virtual field extracted by an expensive face detection DNN.

We assume that the group by key is expensive to compute and consider two different scenarios. In the first scenario, a single oracle determines the group key directly. For example, in the query above, a single oracle would directly classify a person as Biden or Trump. In the second scenario, there is an oracle *per group*, where each oracle can classify whether a record belongs to a specific group key or not. For example, in the query above, we must execute two oracles: one to classify whether or not a person is Biden and another for Trump. We have found this scenario to be common when practitioners do not build their own models. We separate these two cases as they have different optimization objectives.

To optimize the minimax error, we formulate the allocation of the samples to minimize the minimax error as a non-linear optimization problem. Concretely, consider the multiple oracle setting. Given a stratification of group  $g$  (with a total of  $G$  groups), denote the error of this stratification as  $\text{Err}(g)/N$ , for  $N$  samples. Namely,  $\text{Err}(g)/N$  is equivalent to the error when using a single predicate. Then, we aim to optimize the following objective

$$\mathcal{L} = \min_{\Lambda \in [0,1]^G, \sum_{l=1}^G \Lambda_l = 1} \left( \max_g \frac{\text{Err}(g)}{\Lambda_l N} \right) \quad (1)$$

where  $\Lambda$  is a weight vector corresponding to the sample allocation between groups. We show that this optimization problem can be numerically optimized using the Nelder-Mead simplex algorithm [53].

Concretely, ABAE-GROUPBY proceeds as follows:

- (1) Sample uniformly at random to estimate the quantities needed to compute  $\text{Err}(g)$ .
- (2) Solve Eq. 1 via the Nelder-Mead simplex algorithm [53].
- (3) Sample according to the allocation in the previous step.
- (4) Return the combined estimates.

We defer the full formulation to Section 4.5.

### 3.3 Complex Predicates

In addition to a queries with a single predicate, we extend ABAE to support queries that contain any number of conjunctions, disjunctions, and negations (ABAE-MULTIPRED). Presently, we assume that ABAE-MULTIPRED receives as input a set of per-record proxy scores per predicate. We show an example of such a query in Section 2.2.

To answer such queries, ABAE-MULTIPRED will combine the proxy scores from the predicates to obtain a single, per-record set of scores that ideally indicates the likelihood of a record matching the whole expression. ABAE-MULTIPRED combines the proxy scores by transforming the expression into an arithmetic expression with the following substitutions:

- (1) Negations are replaced by subtraction from one.
- (2) Conjunctions are replaced by products.
- (3) Disjunctions are replaced by max.

ABAE-MULTIPRED’s approach will return exact results if the proxies are perfectly calibrated and perfectly sharp. While this assumption does not hold in practice, we show that our approach works well in practice in Section 5.

### 3.4 Selecting Proxies

In some applications, users may have to select between several viable proxies for an expensive predicate. For example, suppose the user wishes to filter emails by spam. The user can provide several rule-based proxies in the form of detecting keywords, such as “money” or “\$”.

A key question is which proxy will provide the lowest MSE for a given budget. To estimate performance improvements, ABAE will use the samples from Stage 1. For each proxy, ABAE will construct the strata and estimate the corresponding  $p_k$  and  $\sigma_k$ . ABAE will use the MSE formula for the perfect information, deterministic draws setting to estimate the optimal achievable MSE (Proposition 2). Given these estimates, ABAE will take the top proxy as the proxy to use in the query. We note that this procedure can reuse samples and only adds negligible computational overhead.

Although the formula for the perfect information, deterministic draws setting does not directly apply, we find it is a good predictor of relative performance.

Finally, ABAE can combine proxies by sampling randomly in Stage 1 and using these samples to train a logistic regression model using the proxies as features and the predicate as the target.

## 4 THEORETICAL ANALYSIS

We present a statistical analysis of ABAE and its extensions. We first show that a related sampling procedure achieves rate  $O\left(\frac{1}{N}\right)$  assuming perfect knowledge of  $p_k$  and  $\sigma_k$ . We then show that our sampling procedure matches the rate of the optimal strategy. Finally, we show that our optimization procedure for allocation for group by keys is optimal for the deterministic setting.

We provide the intuition and theorem statements in this manuscript. We defer the full proofs to an extended technical report [36].

### 4.1 Notation and Preliminaries

**Notation.** Recall the notation in Table 1. We emphasize that  $X_{k,i}$  is the  $i$ th *positive* sample from stratum  $k$ , i.e., the  $i$ th sample that satisfies the predicate. Furthermore, recall that  $\mu_k$  is the per-stratum mean,  $p_{\text{all}} = \sum p_k$  be the sum of the  $p_k$ , and  $\mu_{\text{all}}$  be the overall mean. Finally, recall that  $w_k = p_k/p_{\text{all}}$ , the normalized predicate positive rate, which corresponds to the weighting of  $\mu_k$  to  $\mu_{\text{all}}$ .

**Assumptions and properties.** We assume  $X_{k,i}$  is sub-Gaussian with nonzero standard deviation, a standard assumption for stratified sampling [9]. Sums of sub-Gaussian variables converge with quantitative rates and this assumption widely holds in practice. In particular, centered, bounded random variables are sub-Gaussian. The sub-Gaussian assumption gives the existence of universal constants such that  $\mathbb{E}[|X_{k,i}|] \leq C^{(\mu)}$  and  $\text{Var}[X_{k,i}] \leq C^{(\sigma^2)}$ .

We further assume that  $p_{\text{all}} \geq C^{p_{\text{all}}} > 0$ , which enforces that at least one stratum has non-vanishing  $p_k$ .

### 4.2 Optimal Allocation with Deterministic Draws

We first analyze the setting where we assume perfect knowledge of  $p_k$  and  $\sigma_k$  and that we receive a deterministic, per-stratum number of draws given a sampling budget. Specifically, given a budget

of  $T_k N$  per stratum, we assume that we receive  $p_k T_k N$  samples, rounded up. We prove the optimal allocation under a continuous relaxation and the rate when using this optimal allocation.

**PROPOSITION 1.** *Suppose  $p_k$  and  $\sigma_k$  are known and we receive  $B_k = p_k T_k N$  samples per stratum (up to rounding effects). Then, the choice  $T_k = T_k^*$  that minimizes the MSE for the unbiased estimator  $\hat{\mu}_{all} = \sum_k p_k \hat{\mu}_k / \sum_k p_k$  is*

$$T_k^* = \frac{\sqrt{p_k} \sigma_k}{\sum_{i=1}^K \sqrt{p_i} \sigma_i} \quad (2)$$

**PROPOSITION 2.** *Suppose the conditions in Proposition 1 hold. Then, the squared error under the allocation  $T_k^*$  is*

$$\mathbb{E}[(\hat{\mu}_{all} - \mu_{all})^2 | B_k = p_k T_k^* N] = \sum_{k=1}^K \frac{w_k^2 \sigma_k^2}{p_k T_k^* N} \quad (3)$$

$$= \frac{1}{N p_{all}^2} \cdot \left( \sum_{k=1}^K \sqrt{p_k} \sigma_k \right)^2 \quad (4)$$

Intuitively, these propositions say for deterministic draws, the optimal allocation downweights the standard importance sampling allocation by a factor of  $\sqrt{p_k}$ . The resulting MSE decreases linearly with respect to the sample budget and a scaling factor.

We note that uniform sampling with deterministic draws converges at rate  $\frac{\sigma^2}{N p_{avg}}$ , where  $p_{avg} = \sum p_k / K$ . As a result, stratified sampling offers room for improvement. For example, suppose  $p_1 = 1$ ,  $p_k = 0$  for  $k \neq 1$ , and that  $\sigma_k = 1$  for all  $k$ . This corresponds to a perfect proxy and conditionally independent draws and statistic. Then, uniform sampling converges at rate  $\frac{K}{N}$ , in contrast to stratified sampling's rate of  $\frac{1}{N}$ . This corresponds to a  $K$ -fold improvement in rate.

### 4.3 ABAE with a Single Predicate

We analyze ABAE's two stage sampling algorithm, in which we do not know  $p_k$  and  $\sigma_k$ . We provide the theorem statement, but defer the full proof to an extended technical report [36]. We assume that  $N_2$  is suitably large relative to  $N_1$  for the remainder of the paper.

**THEOREM 4.1.** *With high probability over the draws made in Stage 1 and in expectation in Stage 2,*

$$\mathbb{E}[(\hat{\mu}_{all} - \mu_{all})^2] \leq O\left(\frac{1}{N_1} + \frac{1}{N_2} + \frac{\sqrt{N_1}}{\sqrt{N_2}} \cdot \frac{1}{N_2}\right) \quad (5)$$

Furthermore, if  $N_1 = N_2$

$$\mathbb{E}[(\hat{\mu}_{all} - \mu_{all})^2] \leq O\left(\frac{1}{N}\right) \quad (6)$$

### 4.4 Understanding ABAE

We provide an overall proof sketch of the analysis of ABAE and highlight several aspects of the analysis of broader interest.

**4.4.1 Proof Sketch.** Our proof strategy proceeds as follows. We first show that our estimates  $\hat{p}_k$  and  $\hat{\sigma}_k$  converge to  $p_k$  and  $\sigma_k$  in a quantitative way (i.e., with a specific rate). As a result, our estimate for the optimal allocation will also converge in a quantitative way.

Given the estimate for the optimal allocation, we show that the number of draws in Stage 2 for all strata will approach the

deterministic number of draws, for  $p_k$  large enough (larger than  $\frac{1}{N_2}$ ). We then show that the error converges appropriately for the strata with  $p_k$  large enough and that the error for the remaining strata becomes negligible. As a result, our final estimate converges with rate  $O(\frac{1}{N})$ .

**4.4.2 Challenges.** We describe several challenges in the analysis of ABAE. Prior work has focused on known, deterministic per-strata costs and variances. In contrast, our problem does not have a cost, but rather a stochastic probability of receiving useful information. We study this stochastic draw case and prove that using pilot sampling with plug-in estimates [39] is valid and near optimal.

**Unknown  $p_k$  and  $\sigma_k$ .** Most work in stratified sampling assumes that features of the data distributions within each stratum are known and constructs optimal allocations of samples using this information. In our setting, these quantities must be estimated, which may not be possible when  $p_k$  is small. For example, if  $p_k = \frac{1}{N^2}$  for some stratum, then we may not draw even a single positive record from that stratum, making  $p_k$  and  $\sigma_k$  impossible to estimate.

**Stochastic draws.** In contrast to standard stratified sampling, we may draw a record that does not satisfy the predicate. As a result, for a fixed number of draws, the number of records matching a predicate is stochastic. Most work on stratified sampling assumes a deterministic allocation of samples to strata.

When the number of draws for some arbitrary  $M$  from a stratum and the probability  $p_k$  of matching the predicate are both large, the number of positive records concentrates around  $p_k M$  and the resulting estimator has similar properties to one with  $p_k M$  deterministic draws. However, if  $p_k M$  is small, this analysis breaks down.

**Fractional allocations.** To show that ABAE converges at the optimal rate, we compare to the setting of deterministic draws and perfect information. Given perfect information of  $p_k$  and  $\sigma_k$ , the optimal allocation is given by Proposition 1 and its MSE is given by Proposition 2. However, this allocation cannot be achieved in general, as it results in fractional sampling. Nonetheless, we show that our sampling procedure, which rounds down the ideal fractional allocations, achieves the same  $O(\frac{1}{N})$  rate. Thus, rounding does not affect the convergence rate of our procedure.

**4.4.3 Statistical Intuition.** Our primary tool for dealing with unknown quantities and stochastic draws is dividing the strata into groups: where  $p_k$  is large and where  $p_k$  is small. Since the number of positive draws is Binomial, we apply standard convergence to the total number of positive draws when  $p_k$  is large. For stratum where  $p_k$  is small, the contribution of that stratum to the total error is at most  $p_k C^{(\mu)}$ , which does not increase the asymptotic error. To illustrate our technique, consider the following proposition.

**PROPOSITION 3.** *Recall that  $N_1$  and  $N_2$  are the number of samples in Stages 1 and 2 respectively. With high probability in Stage 1 and if  $N_1$  is a constant multiple of  $N_2$  as  $N$  grows, the MSE of the error in Stage 2 can be written as*

$$\mathbb{E}[(\hat{\mu}_{all} - \mu_{all})^2] = \sum_{k=1}^K \hat{w}_k^2 \text{Var}(\hat{\mu}_k) + O\left(\frac{1}{N_1} + \frac{1}{N_2}\right) \quad (7)$$

where  $\hat{w}_k = \hat{p}_k / \sum \hat{p}_k$ .

As shown in Eq. 7, the overall MSE is bounded above by the sum of  $\hat{w}_k^2 \text{Var}(\hat{\mu}_k)$ , which are per-strata quantities. We then bound these quantities for strata where  $p_k$  is (quantitatively) large or small. Specifically, define  $p_* = \frac{2 \ln(1/\delta) + 2 \sqrt{\ln(1/\delta) + 2}}{N_1} = O\left(\frac{1}{N_1}\right)$  for failure probability  $\delta$ . Furthermore, we assume that  $N_1$  is a constant multiple of  $N_2$  as  $N$  grows. We divide the strata into cases based on whether  $p_k > p_*$  or  $p_k \leq p_*$ .

Consider the case where  $p_k > p_*$ . By standard concentration arguments, the number of positive samples in Stage 2 concentrates to its expectation, which is large. Thus,  $\hat{w}_k^2 \text{Var}(\hat{\mu}_k)$  decays at rate (approximately)  $O\left(\frac{1}{N_2}\right)$  by standard concentration arguments. For  $p_k \leq p_*$ , we can directly bound the contribution. To understand this, consider the following proposition.

PROPOSITION 4.

$$\hat{w}_k^2 \text{Var}[\hat{\mu}_k] \leq \hat{w}_k^2 \left( \mathbb{E} \left[ \frac{\sigma_k^2}{B_k^{(2)}} \mid B_k^{(2)} > 0 \right] + P(B_k^{(2)} = 0) \mu_k^2 \right) \quad (8)$$

$$\leq O \left( \frac{1}{N_1} + \frac{1}{N_2} + \frac{\sqrt{N_1}}{\sqrt{N_2}} \cdot \frac{1}{N_2} \right) \quad (9)$$

where  $B_k^{(2)}$  is the number of positive draws in Stage 2.

PROOF SKETCH. The key challenge is bounding quantities involving  $B_k^{(2)}$ . Suppose counterfactually that  $B_k^{(2)}$  were deterministic: then the expression would correspond to the standard variance of an i.i.d. estimator. Namely, the variance if an i.i.d. estimator decays as  $1/B_k^{(2)}$ . However, since we obtain a stochastic number of draws, we must condition on the event of non-zero draws and take the expectation. Since the draws are binomial in distribution, the leading order converges a.s. to its mean value, which would give the desired bound in this toy setting.

We now adapt this strategy to account for  $B_k^{(2)}$  being stochastic in ABAE. For  $p_k > p_*$ ,  $B_k^{(2)}$  is approximately  $p_k T_k^* N_2$  with high probability. As a result, with high probability, each stratum had sufficient samples to form estimates. We can complete the proof similarly to the toy setting with deterministic  $B_k^{(2)}$ .

However, if  $p_k < p_*$ , we may not draw the requisite number of samples. For example, if  $p_k = \frac{1}{N_2}$ , we would not obtain any samples on average. Thus, our analysis must consider the case where  $p_k < p_*$  separately. When  $p_k$  is small, we can directly bound the contribution of the sum. Namely,  $\hat{w}_k^2 = O(1/N^2)$  as  $p_k \leq \frac{C_1}{N}$  and the remainder of the quantities are bounded by a constant.

Thus, the overall bound follows from considering the strata where  $p_k$  is small and where  $p_k$  is large.  $\square$

**4.4.4 Overview of Techniques.** We briefly describe two techniques used to prove the necessary bounds. First, we leverage exponential tail bounds on sums of Bernoulli random variables (Lemma 1 [11]). Both the upper and lower tail bounds are required to show that  $\sqrt{\hat{p}_k}$  converges to  $\sqrt{p_k}$ : this requires stronger tail bounds than showing  $\hat{p}_k$  converges to  $p_k$ . Second, we use quantitative, exponential tail bounds on Binomial random variables [54] to bound the number of positive draws.

## 4.5 Analyzing Group By

Recall that we aim to optimize the minimax error for queries with a group by clause. Suppose there are  $G$  groups and that we have a proxy per group. As in the case with a single predicate, each proxy induces a stratification over the dataset. Given these  $G$  stratifications, ABAE-GROUPBY estimates the quantities necessary for optimal allocation and executes Stage 2 of ABAE on each stratification appropriately. Thus, the key question is how to allocate samples between the stratifications. We first demonstrate how to allocate samples in the perfect information, deterministic setting and use our plug-in estimates for allocation estimation.

For this section, we index the stratification by  $l$ , the group by  $g$ , and the strata by  $k$ . Thus,  $p_{l,g,k}$ ,  $\sigma_{l,g,k}$ , and  $\mu_{l,g,k}$  denote the predicate positive rate, the standard deviation, and the stratum mean in stratification  $l$ , group  $g$ , and strata  $k$  respectively.

To accelerate group by queries, we rely on ABAE as a subroutine. When we say that we execute an instance of ABAE with a stratification  $l$ , we mean that we stratify the dataset using the proxy for group  $l$  and that we allocate samples across strata optimally for computing the statistic associated with that group.

We now analyze the two cases described in Section 3.2.

**Single Oracle.** In this scenario, recall that we can identify the group key with a single oracle model. To accelerate this query, we will execute  $G$  instances of ABAE's Stage 2 for each stratification  $l$ .

Since a single oracle model identifies the group key, applying ABAE's allocation for a given group gives us estimates for the other groups for free. As a result, we can reuse these estimates across all groups to obtain combined estimators. Thus, by uniformly sampling in Stage 1, ABAE-GROUPBY can obtain estimates  $\hat{p}_{l,g,k}$ ,  $\hat{\sigma}_{l,g,k}$ , and  $\hat{\mu}_{l,g,k}$ . Given these quantities, we can estimate the error using Proposition 2. To account for the multiple estimators across stratifications, we aggregate estimates via inverse-variance weighting, which minimizes the variance [25].

Given the estimates from Stage 1, we estimate the optimal allocation across stratifications with the following objective, where we allocate  $\Lambda_l \cdot N_2$  samples to stratification  $l$ :

$$\mathcal{L} = \min_{\Lambda \in [0,1]^G, \sum_{l=1}^G \Lambda_l = 1} \left( \max_g \left( \sum_{l=1}^G \left( \frac{1}{\Lambda_l N_2} \sum_{k=1}^K \frac{\hat{w}_{l,g,k}^2 \hat{\sigma}_{l,g,k}^2}{\hat{p}_{l,g,k} \hat{T}_{l,k}} \right)^{-1} \right)^{-1} \right) \quad (10)$$

The constraint that  $\sum_{l=1}^G \Lambda_l = 1$  ensures at most  $N_2$  samples are used in Stage 2. The term in the inner sum is the estimated MSE using our plug-in estimates.

This objective follows from the per stratification and per group error which can be calculated using Proposition 2 and using that inverse-variance weighting achieves the least error among all weighted averages which can be calculated as  $(\sum_i 1/\sigma_i^2)^{-1}$ .

Standard tools in convex optimization [6] show that the objective and constraints are convex. Thus, the optimization problem has a unique minimizer. We use the Nelder-Mead simplex algorithm [53] to numerically compute the minimizer.

**Multiple Oracles.** In this scenario, determining which group, if any, a record belongs to requires  $G$  oracle models (one for each group), which we assume has similar costs. In contrast to the setting

with a single oracle model, we do not obtain estimates for other groups when sampling a given group.

As we cannot obtain estimates for other groups in this setting, the oracle for group  $g$  is only applied to samples from stratification  $g$ . Namely, we only consider elements where  $g = l$ . Specifically, we need only estimate  $\hat{p}_{g,g,k}$ ,  $\hat{\sigma}_{g,g,k}$ , and  $\hat{\mu}_{g,g,k}$ . Hence, we will have a single final estimate  $\hat{\mu}_{all,g,g}$  for each group  $g$ . Using Proposition 2 we can estimate the error of  $\hat{\mu}_{all,g,g}$  as function of the number of samples we allocate towards the instance of ABAE associated with stratification and group  $g$ .

We allocate  $N_l$  samples in Stage 1 for each group. To optimize for the minimax error, we will Stage 2 for stratification  $l$  with  $\Lambda_l \cdot N_2$  samples such that  $\sum_{l=1}^G \Lambda_l = 1$ . We will optimize for the optimal values of  $\Lambda$  with the following objective:

$$\mathcal{L} = \min_{\Lambda \in [0,1]^G, \sum_{l=1}^G \Lambda_l = 1} \left( \max_g \frac{1}{\Lambda_l N_2} \sum_{k=1}^K \frac{\hat{w}_{g,g,k}^2 \hat{\sigma}_{g,g,k}^2}{\hat{p}_{g,g,k} \hat{T}_{g,k}} \right) \quad (11)$$

This objective follows from the formula for estimating error provided in Proposition 2. We further note that the objective in Equation 10 reduces to the objective in Equation 11 when  $p_{l,g,k} = 1$ ,  $\sigma_{l,g,k} = \infty$  for  $l \neq g$ .

As before, we can use standard tools in convex optimization to show that the objective and constraints are convex, so the optimization problem has a unique minimizer. We also use the Nelder-Mead simplex algorithm to find the minimizer.

Asymptotic optimality of this objective follows from Theorem 4.1. As the per-group objectives are asymptotically optimal, we can apply the union bound across the  $G$  groups, which shows the convergence of the overall objective.

## 4.6 Discussion

We have shown that ABAE achieves the same asymptotic rate as the optimal allocation strategy for deterministic draws. To our knowledge, the setting of stochastic draws and our convergence proofs are novel. However, we defer extensions to future work. For example, our analysis is asymptotic: finite sample bounds with exact constants would compare against uniform sampling more precisely. Additionally, a bandit algorithm that updates the estimates of  $p_k$  and  $\sigma_k$  per sample draw may provide non-asymptotic improvements.

## 5 EVALUATION

We evaluate ABAE and its extensions on six real world datasets and synthetic datasets. We first describe the experimental setup and baselines. We then demonstrate that ABAE outperforms baselines in all settings we consider. We also show that ABAE’s sample reuse is effective and ABAE is not sensitive to hyperparameters.

### 5.1 Experimental Setup

**Datasets, target DNNs, and proxies.** We consider six real world datasets, including text, still images, and videos (Table 2). We additionally consider synthetic datasets for some settings.

We used the `night-street` (also known as `jackson`) and `taipei` video datasets, which are commonly used for video analytics evaluation [8, 31–33, 59]. We executed the following query:

---

```
SELECT AVG(count_cars(frame)) FROM video
```

---



---

```
WHERE count_cars(frame) > 0
```

---

which computes the average number of cars in the video, conditioning on cars present. We use Mask R-CNN to compute the oracle filter [26]. We use an efficient index for the proxy scores [35].

We used the `celeba` dataset [42], an image dataset of celebrity faces that contains annotations of celebrity names and other attributes, such as hair color. We executed the following query:

---

```
SELECT PERCENTAGE(is_smiling(img)) FROM images
WHERE hair_color(img) = 'blonde'
```

---

which computes the fraction of images where the celebrity is smiling conditioned on the celebrity having blonde hair. We used the human labels in the `celeba` dataset as the ground truth. We used a specialized MobileNetV2 [50] as the proxy.

We used the TREC public spam corpora from 2005 (`trec05p`) [13]. We used the SPAM25 subset. We executed the following query:

---

```
SELECT AVG(NB_LINKS(text)) FROM emails
WHERE is_spam(text)
```

---

which computes the average number of links for spam emails. We used human labels as ground truth. We used a manual, keyword-based proxy based on the presence of words (e.g., “money,” “please”).

We used Amazon movie reviews and posters, which was generated from the Amazon reviews dataset [46]. We scraped the movie posters from the metadata and excluded reviews that did not have posters. We executed the following query:

---

```
SELECT AVG(rating) FROM movies
WHERE face_exists(posters) AND gender(posters) = 'female'
```

---

which computes the average rating of posters with a female actress. We use MT-CNN to extract faces [62] and VGGFace pretrained from deepface [52] to classifier gender as the ground truth. We use a specialized MobileNetV2 as a proxy [50].

We used the Amazon reviews dataset [46] which is a dataset of textual reviews from Amazon. We subset to the office supplies reviews. We executed the following query:

---

```
SELECT AVG(rating) FROM data
WHERE sentiment(review) = 'strongly_positive'
```

---

which computes the average rating of reviews with strongly positive sentiment. We use a BERT-based sentiment classifier provided by FlairNLP to compute the oracle filter [3] and the NLTK sentiment predictor, a simple rule-based classifier, for the proxy [30].

**Metrics.** Our primary metric is the *RMSE* of the true and estimated values: we use the RMSE so that the units are on the same scale as the original value. We additionally compare the number of samples required to achieve a particular error target in some experiments. We measure the cost in terms of oracle predicate invocations as it is the dominant cost of query execution by orders of magnitude.

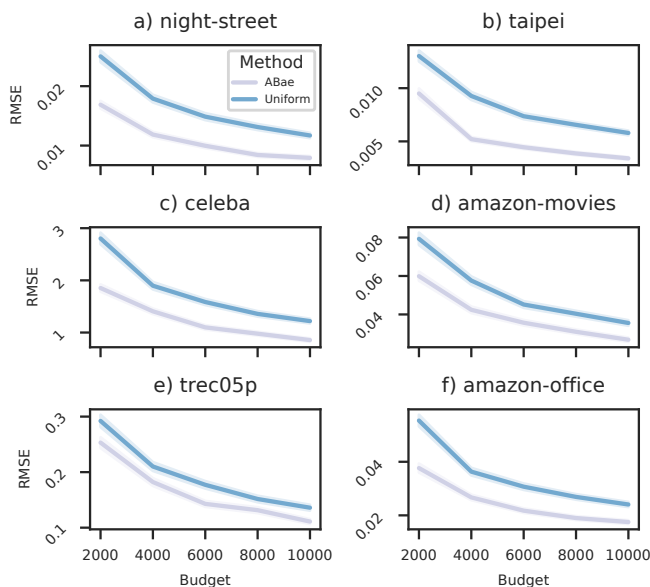
**Methods evaluated.** We compare ABAE to uniform sampling as it is applicable without precomputing predicate results. A range of standard AQP techniques are not applicable to our setting, since the results of the predicate are not available at ingest time. For example, techniques that create histograms [14, 47, 49] or sketches [20, 21] as ingest time are not applicable.

**Implementation.** We implement ABAE’s sampling procedure in Python for ease of integration with deep learning frameworks.



**Table 2: Summary of datasets, predicates, target DNNs, and proxies.**

Dataset	Size	Predicate	Target DNN	Proxy model
night-street	973,136	At least one car	Mask R-CNN [26]	TASTI [35]
taipei	1,187,850	At least one car	Mask R-CNN [26]	TASTI [35]
celeba [42]	202,599	Blonde hair	Human labels	MobileNetV2 [50]
Amazon movie posters [46]	35,815	Contains woman	MT-CNN [62], VGGFace [52]	MobileNetV2 [50]
trec05p [13]	52,578	Is spam	Human labels	Keyword-based
Amazon office supplies [46]	800,144	Strong positive sentiment	FlairNLP BERT sentiment [3]	NLTK sentiment [30]



**Figure 2: Sampling budget vs RMSE for uniform sampling and ABae, with the standard deviation shaded. ABae outperforms on all budgets and datasets we evaluated on. ABae can outperform by up to 1.5× on RMSE at a fixed budget and achieve the same error with up to 2× fewer samples.**

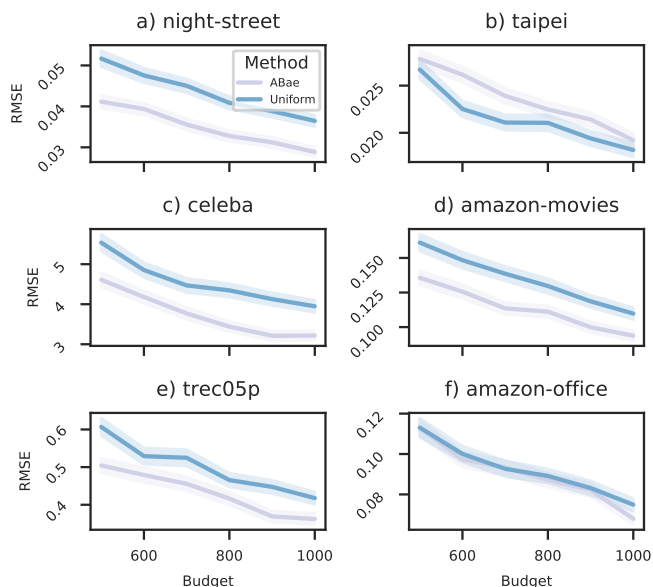
Our open-sourced code is available at <https://github.com/stanford-futuredata/abae>.

## 5.2 End-to-end Performance

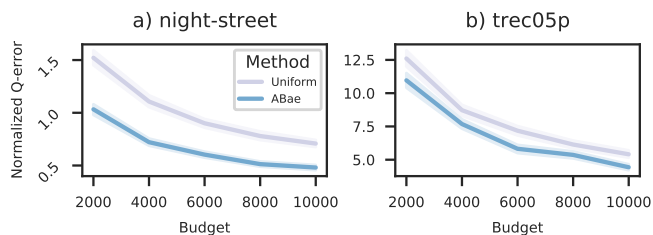
**Single predicate.** We show that ABae outperforms uniform sampling on the metric of RMSE. For each dataset and query, we executed ABae and random sampling for sampling budgets of 2,000 to 10,000 in increments of 2,000. We used five strata and allocated half budget to each stage. We used a failure probability of 5% for every condition. We ran every condition 1,000 times.

As shown in Figure 2, ABae outperforms for every dataset, query, and budget setting we consider. ABae can achieve up to 2.3× improvements in RMSE at a fixed budget or up to 2× fewer samples at a fixed error rate. We additionally show that ABae outperforms uniform sampling at low sampling budgets (500-1,000) in Figure 3.

We further show that ABae outperforms on Q-error [44], which is a relative error metric that penalizes under- and over-estimation symmetrically. We show the normalized Q-error (i.e.,  $100 \times (q - 1)$ ), which roughly indicates percent error in Figure 4. As shown, ABae



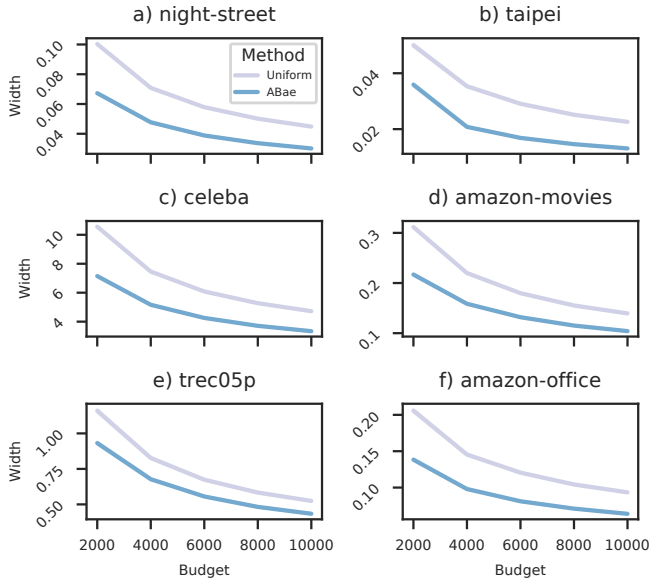
**Figure 3: Low sampling budgets vs RMSE for uniform sampling and ABae, with the standard deviation shaded. We see that even at small sample sizes, ABae outperforms or matches uniform sampling in all cases.**



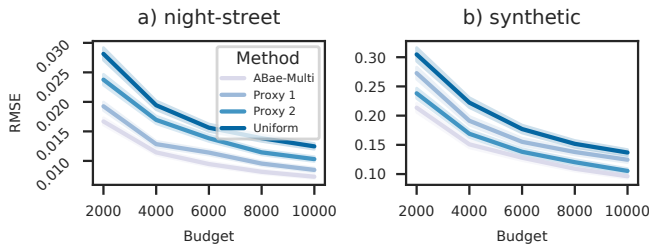
**Figure 4: Sampling budget vs normalized Q-error for uniform sampling and ABae, with the standard deviation shaded. We see that ABae outperforms on Q-error. The same trends hold for all other datasets.**

outperforms on the two datasets we show—ABae also outperforms on all the other datasets by 14-70%, which we omit for brevity. ABae similarly outperform on relative error by 13-76%.

We further show that ABae outperforms on the metric of confidence interval (CI) width. For each dataset and query, we executed ABae and random sampling with the parameters above. We ran every condition 1,000 times.



**Figure 5: Sampling budget vs CI width for uniform sampling and ABae with the standard deviation of the width shaded (not visible in many plots). ABae can outperform by up to 1.5× on CI width at a fixed budget and achieve the same width with up to 2× fewer samples.**



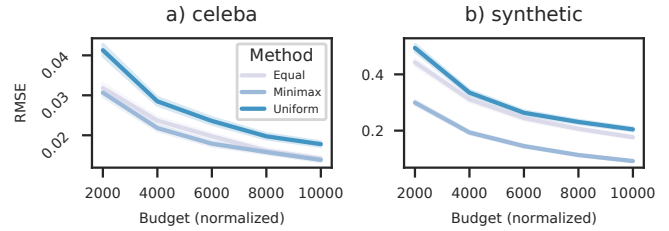
**Figure 6: Sampling budget vs RMSE for uniform sampling and ABae-MULTIPRED, with the standard deviation shaded. As shown, ABae-MULTIPRED outperforms on both queries and all budgets we evaluated on.**

ABae outperforms for every dataset, query, and budget setting we consider (Figure 5). ABae can outperform by up to 1.5× on CI width at a fixed budget. Furthermore, to achieve the same confidence interval width, ABae can use up to 2× fewer samples. Finally, ABae satisfies the nominal coverage across all datasets and settings.

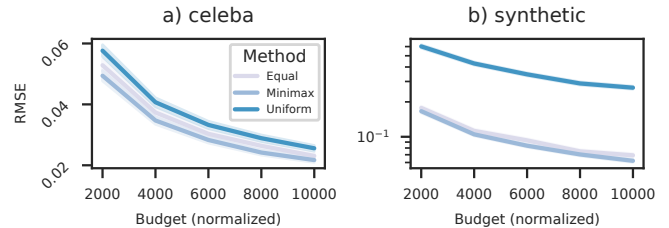
**Multiple predicates.** We show that ABae-MULTIPRED outperforms random sampling and using a proxy for a single predicate. For the night-street dataset, we executed the following query:

```
SELECT AVG(count_cars(frame)) FROM video
WHERE count_cars(frame) > 0
AND red_light(frame)
```

The positive rate is 0.17. We additionally executed a query on a synthetic dataset with five strata and two predicates. For each proxy, we draw the  $p_k$  values from a Beta distribution. As shown in



**Figure 7: Normalized sampling budget vs max RMSE for uniform sampling and ABae-GroupBy with a single oracle (standard deviation shaded). ABae-GroupBy outperforms on both queries and on all budgets we evaluated on.**



**Figure 8: Normalized sampling budget vs max RMSE for uniform sampling and ABae-GroupBy with multiple, per-group oracle, with the standard deviation shaded. The y-axis is on a log-scale. The budget is normalized by the total number of groups. ABae-GroupBy outperforms on both queries and on all budgets we evaluated on.**

Figure 6, ABae-MULTIPRED outperforms on both queries and every budget setting we consider.

**Group bys (single oracle).** We show that ABae-GroupBy outperforms random sampling in the single oracle setting. For the celeba dataset, we executed the following query:

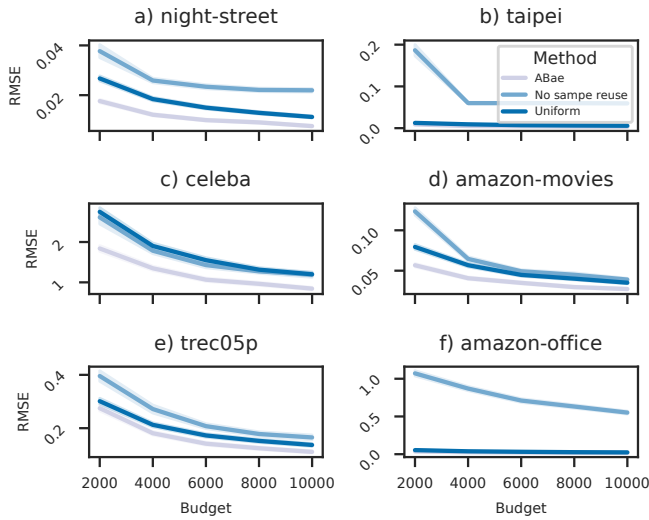
```
SELECT PERCENTAGE(is_smiling(image)) FROM images
WHERE HAIR_COLOR(image) = "gray"
OR HAIR_COLOR(image) = "blond"
GROUP BY HAIR_COLOR(image)
```

We additionally executed a synthetic query where the statistic was distributed normally and the predicate was generated as a Bernoulli with the proxy probability. The synthetic dataset had four groups with a positive rate of 3.3%, 3.3%, 3.4%, and 3.5% respectively.

For these queries, we normalized the budget by the total number of groups. We measured the maximum RMSE over all groups. As shown in Figure 7, ABae-GroupBy outperforms on both queries and all budget settings we consider.

**Group bys (multiple oracles).** We show that ABae-GroupBy outperforms random sampling when a separate oracle is required for each group by key. For the celeba dataset, we executed the same query as for group bys with a single oracle. We additionally executed a synthetic query where the statistic was distributed normally and the predicate was generated as a Bernoulli with the proxy probability. The synthetic dataset had four groups with positive rates of 16%, 12%, 9%, and 5% respectively.

For these queries, we normalized the budget by the total number of groups as extracting the group key requires executing multiple



**Figure 9: Lesion study when removing sample reuse and all components (i.e., uniform sampling) of ABaE. As shown, both sample reuse and allocation between strata are critical for performance on all datasets.**

models. We measured the maximum RMSE over all groups. As shown in Figure 8, ABaE-GROUPBY outperforms on both queries and all budget settings we consider.

**Discussion of results.** To contextualize our results, we first note that relative errors for some of the datasets are as high as 12% (Figure 4b). As a result, a 2× decrease in error (or number of samples at a fixed error) represents a substantial improvement. Several of our ongoing collaborations at Stanford University and elsewhere require expert human labeling as part of scientific studies. Requiring 2× fewer human labels is a substantial decrease in expert time.

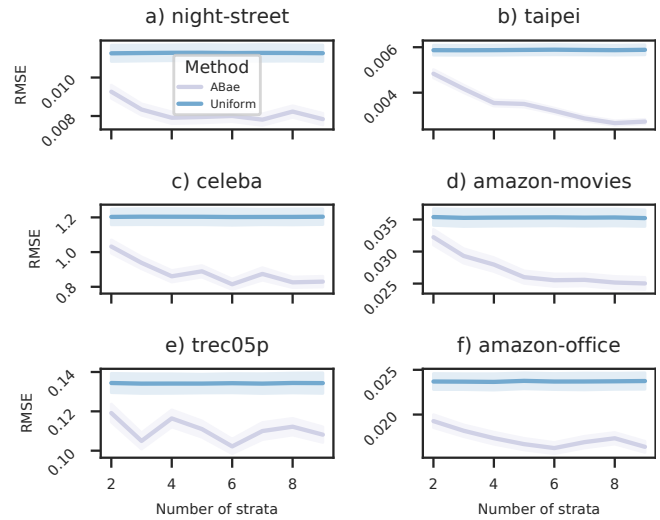
### 5.3 Lesion and Sensitivity Analysis

**Lesion study.** We performed a lesion study in which we removed sample reuse and our two stage procedure (i.e., uniform sampling). Specifically, we executed ABaE, ABaE without sample reuse, and random sampling on all datasets. We used 10,000 samples for all conditions and ran 1,000 trials. For ABaE, we used five strata and allocated half of the samples in each stage.

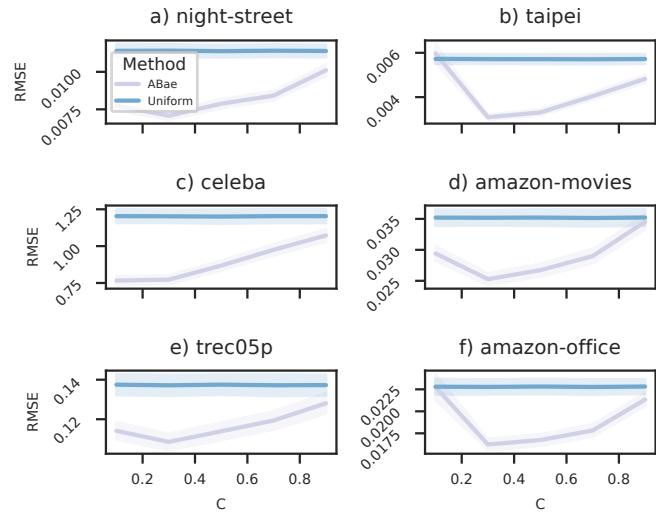
As shown in Figure 9, both parts of ABaE are necessary for high performance. In particular, removing sample reuse substantially harms query performance. Sample reuse contributes to more accurate estimates of  $p_k$ , which is critical for low error.

**Sensitivity analysis.** We analyzed the sensitivity of ABaE to the number of strata ( $K$ ) and the fraction of samples between Stage 1 and Stage 2 ( $C$ ). We executed ABaE when varying  $K$  and  $C$  with a budget of 10,000 and ran 1,000 trials for each condition and dataset.

We varied  $K$  from two to 10 and compared to uniform sampling. As shown in Figure 10, ABaE outperforms on all choices of  $K$  for all datasets. We find that, perhaps surprisingly, the number of strata does not affect strongly performance relative to the performance of uniform sampling. However, in our datasets, more strata tends to perform better.



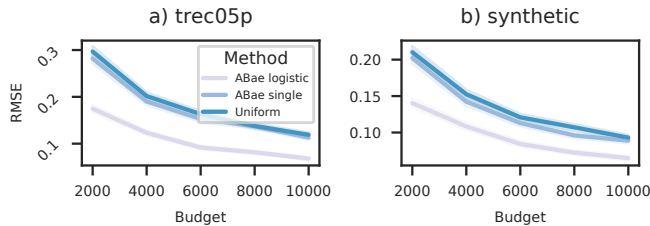
**Figure 10: Sensitivity analysis of ABaE to the number of strata. As shown, ABaE outperforms on all values of  $K$  we evaluated on compared to uniform sampling.**



**Figure 11: Sensitivity analysis of ABaE to the fraction of samples between Stage 1 and Stage 2. As shown, ABaE generally outperforms, except for extreme values of  $C$ .**

We varied  $C$  from 0.1 to 0.9 in increments of 0.2. We compared to uniform sampling. As shown in Figure 11, ABaE outperforms for  $C$  between 0.3 and 0.7, a wide range of values. Several datasets underperform on extreme values of  $C$  (0.1 and 0.9), but these are outside of our recommended values of  $C$ .

**Combining proxies.** We analyzed if ABaE’s method of combining proxies via logistic regression can improve performance. We used keyword-based proxies for the trec05p dataset and a synthetic dataset. For the synthetic dataset, we generated Bernoulli random variables and the proxies were the Bernoulli parameters with noise.



**Figure 12: Uniform sampling, ABAE with a single proxy, and ABAE combining proxies. As shown, ABAE’s method of combining proxies outperforms baselines.**

We show results for uniform sampling, single proxy ABAE, and ABAE with the combined proxy in Figure 12. As shown, ABAE can combine proxies, effectively “ignoring” low quality proxies.

## 6 RELATED WORK

**AQP.** A recent survey on AQP techniques categorizes AQP into offline and online methods [40]. The closest work in AQP are offline methods, which use pre-computed samples [1, 2], histograms [14, 47, 49], wavelets [22], and sketches [20, 21] to accelerate approximate queries. These techniques involve computation at ingest time to generate a synopsis based on expected query workload [40], but assume the records are already present as structured data. Since we assume the predicates are expensive to compute, we cannot compute these synopses at ingest time.

ABAE is also related to online methods, where the statistic is computed on the fly. For example, online aggregation provides shrinking confidence intervals as the query is executed [27]. These techniques also largely rely on precomputed information, e.g., indexes.

**Surveying and optimal allocation.** Work in the surveying and classical sampling literature have long studied stratified sampling [45, 51]. If the strata variances and costs are known, the optimal allocation is proportional to  $\frac{\sigma_k}{\sqrt{c_k}}$ , for costs  $c_k$  [39]. The closest work we are aware of are algorithms for stratified sampling where the strata variances are not known [5, 9]. This work does not consider the case of stochastic draws, as we do.

Other work in surveying considers stratified sampling with non-responses, such as non-responses to mail surveys [24]. This literature is largely concerned with estimating the bias from non-responses or focusing on which subpopulations to follow up with, e.g., with phone calls [18, 38, 57]. In this work, not all of the population satisfies the predicate, so the non-response model is different.

A common technique in scientific studies and surveying is pilot sampling, in which a small sample is used for preliminary analysis. Pilot sampling is commonly used in randomized trials [55] and to estimate various quantities for downstream sampling (strata variances, sampling costs, feasibility, etc.) [12, 15, 17, 39]. In our setting, each sample has a stochastic probability of giving useful information instead of a cost, which is not covered by prior work. However, the allocations in ABAE coincide with those made by defining hypothetical costs proportional to  $\frac{1}{p_k}$ . To show that this allocation remains valid in our setting, we prove that Stage 1 of ABAE is optimal with high probability, something which is not handled by standard survey sampling theory.

**Stratified sampling in data management systems.** Several traditional systems and algorithms use stratified sampling to accelerate sampling [1, 2, 10]. In the data management literature, using stratified sampling typically requires pre-computation, which is not applicable for the same reasons as described above.

Recent work learns machine learning models, such as classification DNNs or generative adversarial networks (GANs), over relational data to improve AQP [41, 56, 58]. These approaches approximate the data (e.g., the result of a predicate or overall data statistics), which can be faster than directly accessing the data. However, they do not apply to complex unstructured data sources that we consider in this work.

**Proxies.** Recent work has focused on accelerated DNN-based queries by using proxies. Many systems have been developed to accelerate certain classes of queries using proxies, including selection without statistical guarantees [4, 29, 33, 43], selection with statistical guarantees [34], aggregation queries without predicates [32], and limit queries [32]. The work on selection does not directly apply to our setting, since these systems do not guarantee that the records selected are independent of the statistic we wish to compute. The BLAZEIT system accelerates aggregation queries over the entire dataset by using proxies as a control variate, but does not consider aggregation queries with predicates [32]. We add to this body of literature by designing an algorithm for stochastic draws to address that many aggregation queries contain predicates.

**DNN-based queries.** Other work aims to accelerate other DNN-based queries. Several systems aim to improve execution speeds or latency of DNNs on accelerators [37, 48, 61], which are complementary to our work. Other work assumes that the target DNN is not expensive to execute or that extracting bounding boxes is not expensive [19, 63]. We have found that many applications require accurate and expensive target DNNs, so we focus on reducing executing the target DNN via sampling.

## 7 CONCLUSION

To reduce the cost of approximate aggregation queries with expensive predicates, we introduce stratified sampling algorithms leveraging proxies. We provide proofs of convergence for stratified sampling with stochastic draws, which corresponds to our setting. We show that ABAE achieves optimal rates. We further extend ABAE to answer queries with multiple predicates and group by keys. We show that our algorithms outperform baselines by up to 2.3× on a wide range of domains and predicates.

## ACKNOWLEDGMENTS

This research was supported in part by affiliate members and other supporters of the Stanford DAWN project—Ant Financial, Facebook, Google, Infosys, NEC, and VMware—as well as Toyota Research Institute (“TRI”), Northrop Grumman, Amazon Web Services, Cisco, and the NSF under CAREER grant CNS-1651570. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. TRI provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

## REFERENCES

- [1] Swarup Acharya, Phillip B Gibbons, and Viswanath Poosala. 1999. Aqua: A fast decision support systems using approximate query answers. In *PVLDB*. 754–757.
- [2] Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden, and Ion Stoica. 2013. BlinkDB: queries with bounded errors and bounded response times on very large data. In *EuroSys*. ACM, 29–42.
- [3] Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*. 1638–1649.
- [4] Michael R Anderson, Michael Cafarella, Thomas F Wenisch, and German Ros. 2019. Predicate Optimization for a Visual Analytics Database. *ICDE* (2019).
- [5] Bouhari Arouna. 2004. Adaptive Monte Carlo method, a variance reduction technique. (2004).
- [6] Stephen Boyd, Stephen P Boyd, and Lieven Vandenbergh. 2004. *Convex optimization*. Cambridge university press.
- [7] Vladimir Braverman and Rafail Ostrovsky. 2013. Generalizing the layering method of Indyk and Woodruff: Recursive sketches for frequency-based vectors on streams. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 58–70.
- [8] Christopher Canel, Thomas Kim, Giulio Zhou, Conglong Li, Hyeontaek Lim, David Andersen, Michael Kaminsky, and Subramanya Dulluro. 2019. Scaling Video Analytics on Constrained Edge Nodes. *SysML* (2019).
- [9] Alexandra Carpentier, Remi Munos, and András Antos. 2015. Adaptive strategy for stratified Monte Carlo sampling. *J. Mach. Learn. Res.* 16 (2015), 2231–2271.
- [10] Surajit Chaudhuri, Gautam Das, and Vivek Narasayya. 2007. Optimized stratified sampling for approximate query processing. *TODS* (2007).
- [11] Fan Chung and Linyuan Lu. 2002. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics* 6, 2 (2002), 125–145.
- [12] William G Cochran. 2007. *Sampling techniques*. John Wiley & Sons.
- [13] Gordon V Cormack and Thomas R Lynam. 2005. TREC 2005 Spam Track Overview. In *TREC*. 500–274.
- [14] Graham Cormode, Antonios Deligiannakis, Minos Garofalakis, and Andrew McGregor. 2009. Probabilistic histograms for probabilistic data. *Proceedings of the VLDB Endowment* 2, 1 (2009), 526–537.
- [15] Jonathan S Denne and Christopher Jennison. 1999. Estimating the sample size for at-test using an internal pilot. *Statistics in Medicine* 18, 13 (1999), 1575–1585.
- [16] Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.
- [17] Sandra M Eldridge, Gillian A Lancaster, Michael J Campbell, Lehana Thabane, Sally Hopewell, Claire L Coleman, and Christine M Bond. 2016. Defining feasibility and pilot studies in preparation for randomised controlled trials: development of a conceptual framework. *PLoS one* 11, 3 (2016), e0150205.
- [18] FL Fillion. 1975. Estimating bias due to nonresponse in mail surveys. *Public Opinion Quarterly* 39, 4 (1975), 482–492.
- [19] Daniel Y Fu, Will Crichton, James Hong, Xinwei Yao, Haotian Zhang, Anh Truong, Avani Narayan, Maneesh Agrawala, Christopher Ré, and Kayvon Fatahalian. 2019. ReKall: Specifying video events using compositions of spatiotemporal labels. *arXiv preprint arXiv:1910.02993* (2019).
- [20] Edward Gan, Peter Bailis, and Moses Charikar. 2020. Coopstore: Optimizing precomputed summaries for aggregation. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2174–2187.
- [21] Minos Garofalakis, Johannes Gehrke, and Rajeev Rastogi. 2002. Querying and mining data streams: you only get one look a tutorial. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*. 635–635.
- [22] Sudipto Guha and Boulos Harb. 2005. Wavelet synopsis for data streams: minimizing non-euclidean error. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 88–97.
- [23] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*. PMLR, 1321–1330.
- [24] Morris H Hansen and William N Hurwitz. 1946. The problem of non-response in sample surveys. *J. Amer. Statist. Assoc.* 41, 236 (1946), 517–529.
- [25] Joachim Hartung, Guido Knapp, and Bimal K Sinha. 2011. *Statistical meta-analysis with applications*. Vol. 738. John Wiley & Sons.
- [26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *ICCV*. IEEE, 2980–2988.
- [27] Joseph M Hellerstein, Peter J Haas, and Helen J Wang. 1997. Online aggregation. In *Acm Sigmod Record*, Vol. 26. ACM, 171–182.
- [28] James Hong, Will Crichton, Haotian Zhang, Daniel Y Fu, Jacob Ritchie, Jeremy Barenholtz, Ben Hannel, Xinwei Yao, Michaela Murray, Geraldine Moriba, et al. 2020. Analyzing Who and What Appears in a Decade of US Cable TV News. *arXiv preprint arXiv:2008.06007* (2020).
- [29] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Paramvir Bahl, Matthai Philipose, Phillip B Gibbons, and Onur Mutlu. 2018. Focus: Querying Large Video Datasets with Low Latency and Low Cost. *OSDI* (2018).
- [30] Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAI Conference on Web and Social Media*, Vol. 8.
- [31] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica. 2018. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM, 253–266.
- [32] Daniel Kang, Peter Bailis, and Matei Zaharia. 2019. BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. *PVLDB* (2019).
- [33] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: optimizing neural network queries over video at scale. *PVLDB* 10, 11 (2017), 1586–1597.
- [34] Daniel Kang, Edward Gan, Peter Bailis, Tatsunori Hashimoto, and Matei Zaharia. 2020. Approximate Selection with Guarantees using Proxies. *PVLDB* (2020).
- [35] Daniel Kang, John Guibas, Peter Bailis, Tatsunori Hashimoto, and Matei Zaharia. 2020. Task-agnostic Indexes for Deep Learning-based Queries over Unstructured Data. *arXiv preprint arXiv:2009.04540* (2020).
- [36] Daniel Kang, John Guibas, Peter Bailis, Yi Sun, Tatsunori Hashimoto, and Matei Zaharia. 2021. Proof: Accelerating Approximate Aggregation Queries with Expensive Predicates. *arXiv preprint arXiv:2107.12525* (2021).
- [37] Daniel Kang, Ankit Mathur, Teja Veeramacheni, Peter Bailis, and Matei Zaharia. 2021. Jointly optimizing preprocessing and inference for DNN-based visual analytics. *PVLDB* (2021).
- [38] Mohammad GM Khan, EA Khan, and MJ Ahsan. 2008. Optimum allocation in multivariate stratified sampling in presence of non-response. *Journal of the Indian Society of Agricultural Statistics* 62, 1 (2008), 42–48.
- [39] Leslie Kish. 1965. *Survey sampling*. Number 04; HN29, K5.
- [40] Kaiyu Li and Guoliang Li. 2018. Approximate query processing: What is new and where to go? *Data Science and Engineering* 3, 4 (2018), 379–397.
- [41] Wanxin Li. 2020. Supporting Database Constraints in Synthetic Data Generation based on Generative Adversarial Networks. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2875–2877.
- [42] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.
- [43] Yao Lu, Aakanksha Chowdhery, Srikanth Kandula, and Surajit Chaudhuri. 2018. Accelerating Machine Learning Inference with Probabilistic Predicates. In *SIGMOD*. ACM, 1493–1508.
- [44] Guido Moerkotte, Thomas Neumann, and Gabriele Steidl. 2009. Preventing bad plans by bounding the impact of cardinality estimation errors. *Proceedings of the VLDB Endowment* 2, 1 (2009), 982–993.
- [45] Dankit K Nassiuma. 2001. Survey sampling: Theory and methods.
- [46] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 188–197.
- [47] Gregory Piatetsky-Shapiro and Charles Connell. 1984. Accurate estimation of the number of tuples satisfying a condition. *SIGMOD* (1984).
- [48] Alex Poms, William Crichton, Pat Hanrahan, and Kayvon Fatahalian. 2018. Scanner: Efficient Video Analysis at Scale (To Appear). (2018).
- [49] Viswanath Poosala, Peter J Haas, Yannis E Ioannidis, and Eugene J Shekita. 1996. Improved histograms for selectivity estimation of range predicates. *ACM Sigmod Record* 25, 2 (1996), 294–305.
- [50] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.
- [51] Richard L Scheaffer, William Mendenhall III, R Lyman Ott, and Kenneth G Gerow. 2011. *Elementary survey sampling*. Cengage Learning.
- [52] Sefik Ilkin Serengil and Alper Ozpinar. 2020. LightFace: A Hybrid Deep Face Recognition Framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 23–27. <https://doi.org/10.1109/ASYU50717.2020.9259802>
- [53] Danny C Sorensen. 1982. Newton’s method with a model trust region modification. *SIAM J. Numer. Anal.* 19, 2 (1982), 409–426.
- [54] Robert Tarjan. 2009. 15-359: Probability and Computing Lecture 10: More Chernoff Bounds, Sampling, and the Chernoff + Union Bound method. <http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/>. [Online; accessed 11-Feb-2021].
- [55] Lehana Thabane, Jinhui Ma, Rong Chu, Ji Cheng, Afisi Ismaila, Lorena P Rios, Reid Robson, Marroon Thabane, Lora Giangregorio, and Charles H Goldsmith. 2010. A tutorial on pilot studies: the what, why and how. *BMC medical research methodology* 10, 1 (2010), 1–10.
- [56] Saravanan Thirumuruganathan, Shohedul Hasan, Nick Koudas, and Gautam Das. 2020. Approximate query processing for data exploration using deep generative models. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1309–1320.

- [57] Rahul Varshney, MJ Ahsan, et al. 2012. An optimum multivariate stratified double sampling design in presence of non-response. *Optimization Letters* 6, 5 (2012), 993–1008.
- [58] Brett Walenz, Stavros Sintos, Sudeepa Roy, and Jun Yang. 2019. Learning to sample: Counting with complex queries. *arXiv preprint arXiv:1906.09335* (2019).
- [59] Tiantu Xu, Luis Materon Botelho, and Felix Xiaozhu Lin. 2019. VStore: A Data Store for Analytics on Large Videos. In *Proceedings of the Fourteenth EuroSys Conference 2019*. ACM, 16.
- [60] Changzheng Zhang, Xiang Xu, and Dandan Tu. 2018. Face detection using improved faster rcnn. *arXiv preprint arXiv:1802.02142* (2018).
- [61] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J Freedman. 2017. Live Video Analytics at Scale with Approximation and Delay-Tolerance. In *NSDI*, Vol. 9. 1.
- [62] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503.
- [63] Yuhao Zhang and Arun Kumar. 2019. Panorama: a data system for unbounded vocabulary querying over video. *Proceedings of the VLDB Endowment* 13, 4 (2019), 477–491.