Controlled Intentional Degradation in Analytical Video Systems

Controlled Intentional Degradation in Analytical Video Systems

Wenjia He
University of Michigan, Ann Arbor
wenjiah@umich.edu

Michael Cafarella
Massachusetts Institute of Technology
michjc@csail.mit.edu

ABSTRACT

It is increasingly affordable for governments to collect video data of public locations. This video can be used for a range of broadly valuable analytical tasks, such as counting traffic, measuring commerce, or detecting accidents. Governments also have a range of policy goals — preserving privacy, reducing bandwidth use, and legal compliance — that may be obtained by degrading the video at some potential cost to analytical accuracy. Ideally, public administrators could employ *controlled intentional video degradation* to achieve policy goals while still obtaining the required analytical accuracy. Unfortunately, the optimal amount of induced degradation is data- and query-dependent, and so is difficult to determine even when public policy preferences are well-known.

We propose a *video degradation-accuracy profiling* model for the problem of controlling the appropriate amount of degradation. It offers administrators a profile that illustrates the tradeoff between increased analytical accuracy and increased amounts of degradation. Computing the true tradeoff curves requires full access to the non-degraded video stream, so a primary technical contribution of this work lies in methods for accurately approximating the curves with only limited information. In addition, we propose a *profile repair* policy to further improve tradeoff curves' accuracy. We describe our prototype system, Smokescreen, plus experiments on two video datasets, two detection models and four aggregate query types. Compared with competing methods, we show our upper bound estimation of analytical error is up to 155% tighter, and Smokescreen enables 88% more accurate tradeoffs.

CCS CONCEPTS

• **Information systems** → **Video search**; **Data analytics**.

KEYWORDS

video query; video degradation; analytical accuracy profile; aggregate query approximation

ACM Reference Format:
Wenjia He and Michael Cafarella. 2022. Controlled Intentional Degradation in Analytical Video Systems. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3514221.3517899

SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

ACM ISBN 978-1-4503-9249-5/22/06…$15.00
https://doi.org/10.1145/3514221.3517899

1 INTRODUCTION

Society is experiencing a vast increase in the availability of video data. This data can be used for a range of public good applications, such as traffic monitoring and gathering commerce data. In these applications, administrators attach importance to analytical accuracy, but may also have competing goals. One goal is to meet system requirements. For example, wireless sensor networks, widely applied for building control, environmental monitoring, etc., suffer from low bandwidth and low power constraints [15, 49]. Another goal is to preserve private information (e.g., facial imagery) captured by video. This information can raise public concern due to potential leakage during shipment of video off-camera or execution of malicious queries. Finally, video surveillance is supposed to obey legal regulations [54]. For instance, according to the EU General Data Protection Regulation [63], face blurring is required when any closed-circuit television (CCTV) footage is shared with a third party.

Generic intentional degradation methods are helpful for these analysis requirements [5, 6, 16, 20, 23, 53, 63, 66]. For example, frame rate reduction can be applied when the storage budget is limited. Frame resolution reduction can ensure legal compliance and is also useful for informal privacy protection. Although video degradation is extremely valuable, it usually does harm to analytical result accuracy, so it has to be done in a careful and controlled way. Unfortunately, no current system reveals how degradation affects analytical accuracy.

In response, we introduce a system for enabling **controlled intentional degradation**. The system has a few basic components:

- A **set of configurable networked cameras** that can collect, modify, and transmit images to a central system for query processing.
- A set of **destructive interventions** available in each camera: decreased resolution, decreased sampling rates, selective image removal, etc. These interventions likely solve system, privacy and legal compliance problems but likely decrease analytical query accuracy.
- A **video query processor** that receives a set of images from the cameras and implements an analytical query. It will be common for this query to include a UDF that embodies a trained neural network.
- A **public administrator** who determines the appropriate degradation/accuracy tradeoff for each query in a workload. This administrator could be an actual individual holding a public office, or a public committee, etc.

Example 1. *Harry is the public administrator for a city that collects surveillance videos of a road. The city wants to compute the average number of cars per frame on weekends so as to extrapolate the average cars per hour in order to schedule construction work. The city wants to maximize individuals' privacy, especially faces, and minimize the energy consumption during video transmission from cameras to*

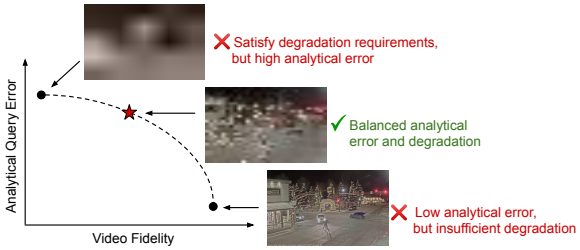

Figure 1: The public administrator must make a query-specific tradeoff that balances degradation requirements with the benefits of accurate analytical queries. Our system does not choose a tradeoff. Rather, it makes the tradeoff curve visible to the administrator.

the central system, but the maintenance department needs a frame-averaged car count that is within 10% of the correct answer. Harry configures the cameras to lower the frame resolution. However, the extremely low resolution has led to a query result that is badly wrong. Without knowing how the frame resolution affects the accuracy of the query, Harry cannot implement the city's preferences.

System Goals — A well-informed tradeoff between destructive interventions and aggregate query accuracy is difficult to make, since interventions can interact in unexpected ways with the query. For example, a slightly-reduced frame sampling rate may not impact a query that counts pedestrians since pedestrians move relatively slowly. But once the sampling rate falls under a particular threshold, the query may become very inaccurate. Therefore, even when the video system operator's degradation goals stay stable, the optimal tradeoff point can change with changes in the query, the destructive interventions, or the video contents. In an ideal world, the video system operator could examine a query-specific *tradeoff curve* (as in Figure 1) to determine an appropriate set of interventions.

Example 2. *Harry submits the weekend car counting query to the system and receives a customized* degradation accuracy tradeoff curve. *By examining the curve, he finds that 128×128 is the lowest resolution that would not cause more than 10% analytical error. The cameras now collect and transmit only this low-resolution information, greatly improving privacy and saving energy while still giving the city maintenance department what it needs.*

Unfortunately, it is not clear how to generate this tradeoff curve. A simple approach would be: run the query on a representative portion of video, run it again on a degraded version of the video, and then compare the resulting query outputs. However, this naïve method presents serious problems:

- Accessing the original video and lightly-degraded video means we cannot conserve systems resources and preserve private data for the examined portion of video.
- It is computationally expensive, because it may need to be performed on many different degradation "knob settings".

The above problems may be acceptable if this examined portion is small and limited, but as mentioned above, we potentially need to recompute tradeoff curves for every new query, model, or video set. This naïve method may have to be applied almost continuously, violating the goals that motivated intentional degradation in the first place. We sidestep all of these problems by computing tradeoff curves without access to the underlying video. Moreover, we show that it can be done in a computationally efficient manner.

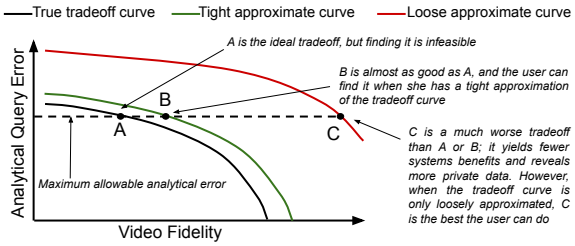

Figure 2: Conceptual diagram of approximate curves with a tight upper bound and a loose upper bound.

Technical Challenge — The main challenge of producing valuable tradeoff curves is to estimate the accuracy of the approximate query answer when video data is modified by any set of destructive interventions. The interventions transform video in different ways. For example, the *reduced frame sampling intervention* samples frames randomly; while the *image removal intervention* samples frames that do not contain restricted objects so that video features are modified non-randomly. The query analytical accuracy should be estimated under both random and non-random interventions.

This estimation problem is difficult because we can only get access to degraded samples instead of the unmodified video. A common solution is to compute the upper bound of the analytical error. Figure 2 shows a conceptual diagram of the true tradeoff curve and approximate tradeoff curves, one with a tight and one with a loose upper bound. Given an analytical error threshold, if the true tradeoff curve were known, an administrator could choose the tradeoff at point A. A tight approximate curve lets the administrator choose a level of degradation at point B; the video here is less degraded than at point A, but the loss in degradation is not too bad. However, with a loose approximation curve, the administrator has no choice but to accept the worst tradeoff at C. As a consequence, we can see the upper bound needs to be tight. Online aggregation [30], stopping algorithms such as EBGS [48] and holistic aggregation approximation methods [40, 45] can provide error upper bounds for a variety of aggregate queries. However, these methods cannot compute sufficiently tight outputs to enable good degradation decisions, especially when video is substantially degraded. Moreover, they are not able to deal with non-random interventions.

Our Approach — We propose new algorithms to provide tight upper bounds of analytical error, allowing us to create better degradation/accuracy tradeoff curves for aggregate queries with `AVG`, `SUM`, `COUNT`, `MAX` and `MIN` functions. These queries' results are computed at a frame level, then aggregated; such queries have been introduced and investigated in previous work [34, 38]. Deduplicated aggregate query types are beyond the scope of this paper. The novelty of our work for each type of destructive intervention is summarized in Table 1.

When the destructive interventions are **random**, for aggregate queries with `AVG`, `SUM` or `COUNT`, we adapt the analytical error estimation method from the empirical Bernstein stopping algorithm [48], and further improve it by relaxing the confidence interval construction requirement and applying the Hoeffding–Serfling inequality [8]. For aggregate queries with `MAX` or `MIN`, we leverage the

Video Scenario	Technical Problem	Our Novelty
Estimate analytical accuracy of video aggregate queries under random destructive interventions, e.g., reduced frame sampling. (Section 2.1)	Provide a tight upper bound of the error of the aggregate result estimation under a certain confidence level when the distribution of models' outputs is unchanged. (Section 2.4)	AVG, SUM, COUNT: Improve the error bound estimation method adapted from the empirical Bernstein stopping algorithm and apply the Hoeffding–Serfling inequality. (Section 3.2.1-3.2.3)
		MAX, MIN: Leverage the normal approximation for hypergeometric distribution to estimate the error bound of extreme quantiles. (Section 3.2.4)
Estimate analytical accuracy of video aggregate queries under non-random interventions, e.g., reduced frame resolution and image removal. (Section 2.1)	Provide a tight upper bound of the error of the aggregate result estimation under a certain confidence level when the distribution of models' outputs may change. (Section 2.4)	Profile repair: Use the randomly sampled correction set to correct possibly wrong error bounds and minimize the correction set size according to its own analytical accuracy, or create tradeoff curves from a similar but less sensitive video. (Section 3.2.5-3.3.1)

Table 1: The video scenarios, technical problems and novelty in our model.

normal approximation for hypergeometric distribution in order to approximate the error of extreme quantiles.

When the destructive interventions are **non-random**, we propose a *profile repair* strategy. We introduce a *correction set* of video that is only modified by random interventions with the aim of correcting our method's analytical accuracy estimation. We minimize the size of this correction set as much as possible. Administrators may construct correction sets by applying random interventions to the query-specified video. When it is not possible to use only random interventions on the query video (perhaps when the video is especially sensitive), it is still possible to obtain a good approximation: administrators can choose to compute from a separate video set that is similar to the query video, yielding a similar tradeoff curve, and then use this curve to guide non-random interventions applied to the intended query video. Finally, note that the correction set can also improve the accuracy of tradeoff curves for random interventions in some cases.

Contributions — Our contributions are as follows:

- We propose a novel *video degradation-accuracy profiling* model that enables governments to implement well-informed tradeoffs for system, privacy and legal compliance reasons. (Section 2)
- We design novel algorithms for random and non-random destructive interventions to compute tight error bounds of query result estimations for tradeoff curve profiling. Our method can obtain a 155% tighter error bound than the previous state-of-the-art method. (Section 3)
- We embodied these ideas in a prototype software system, Smokescreen, and evaluated it on a range of video datasets and aggregate query types. We show that Smokescreen enables tradeoffs that are 88% more accurate than a method based on previously-known approaches. (Sections 4 and 5)

2 PROBLEM FORMULATION

We introduce the types of video degradation in Section 2.1, the importance of degradation accuracy tradeoff curves in Section 2.2, frequently-used vocabulary in our model in Section 2.3, and the technical problems' formal formulation in Section 2.4.

2.1 Video Degradation

There are often system, privacy and legal compliance requirements in addition to the pure analytical accuracy requirement, so administrators have to balance these competing goals. Intentionally degrading video is a common operation in analytical settings. Here are three ways to do it:

Intervention example 1: Reduced frame sampling — This method reduces the ratio of the randomly sampled frames against the total query-specified frames. With this intervention, time-related privacy (e.g., daily life tracks) will not be revealed [16], and video file size can be reduced to meet system requirements such as a low bandwidth constraint [5, 53] and energy limitations [66].

Intervention example 2: Reduced frame resolution — This method reduces the resolution of processed frames. With this intervention, objects like faces that can be recognized from high-resolution images will not be revealed so as to obey legal regulations [63]; the burden on system resources can also be mitigated [6, 23].

Intervention example 3: Image removal — This method entirely deletes frames that contain restricted objects so as to ensure legal compliance and preserve privacy [20]. Sensitive objects include people, faces, license plates, etc. Any combination of them may be considered to be restricted.

Besides these three examples, there are also other degradation methods, such as noise addition [65], video compression techniques [27], etc. All of these methods can be divided into two categories. **Random** interventions modify video features such that the distribution of models' outputs is unchanged (e.g., reduced frame sampling). **Non-random** interventions modify underlying videos such that the distribution of models' outputs may change (e.g., reduced frame resolution and image removal).

A single intervention type may not meet all the requirements, such as different legal regulations, and may affect analytical accuracy much more than other interventions. For example, previous experiments show that low video resolution can significantly affect the accuracy of some classification models [37]. As a consequence, we allow the administrator to choose a combination of the above three typical intervention examples, covering both random and non-random intervention types. Administrators can tune these degradation knobs (making sample fraction and resolution up or down, and choosing preferred restricted objects) in order to trade analytical accuracy against degradation goals.

2.2 Degradation Accuracy Trade-Off Curves

Two important features about video make the tradeoff problem difficult. First, administrators, perhaps driven by their local government, usually have different preferences about query answer quality and the best degradation level. As a result, it is not feasible to simply fix

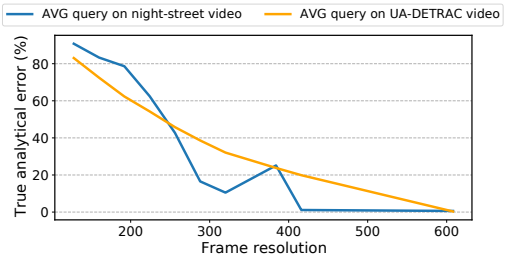

Figure 3: Real degradation accuracy tradeoff curves for the AVG query on two different video datasets.

the intervention settings for all administrators. Second, the shape of the degradation accuracy tradeoff curve changes depending on the query (e.g., calculating the average or the maximum number of cars) and the video content (e.g., collected from the surveillance camera at a downtown intersection or at a narrow road). Figure 3 shows two real degradation accuracy tradeoff curves of queries that compute the average number of cars per frame on night-street video [34] and UA-DETRAC video [64]. YOLOv4 [11] is used in these queries to detect cars. The x-axis describes frame resolution, while the y-axis is the relative error of the estimated query result. These two curves are quite different from each other, illustrating how they are video-dependent.

Therefore, a system that supports administrators in making this crucial degradation/accuracy trade-off must provide video- and query-specific curves.

2.3 Usage Model

In our video degradation-accuracy profiling model, frequently used vocabulary is summarized as follows:

- **Original video**: The raw unaltered video collected by the set of networked cameras. This video has not yet been processed by the destructive interventions. It is never processed directly by the video query processor.
- **Degraded video**: Applying a destructive intervention to the original video will yield a set of degraded video. It can then be analyzed by the video query processor.
- **Profile**: A profile describes a tradeoff between a destructive intervention and analytical accuracy for each unique combination of video corpus, query, and intervention. The profile consists of a set of (degradation, error) pairs; missing values should simply be interpolated by the administrator. The profile shows the error caused by video degradation when compared with the query result derived from non-degraded video, so the error values are computed without regard to the absolute accuracy of the video analysis model.
- **Profile generation**: Our system produces a unique profile for a given video corpus, query, and intervention.
- **Choosing a tradeoff**: Administrators use a profile to select a desired level of destructive intervention. We expect that queries contain video analysis models of high accuracy, or at least that administrators know the approximate accuracy of models. Administrators can adjust the analytical accuracy threshold in the selection process by considering models' inherent accuracy. This selected degradation setting is then applied for query result estimation.

Parameter	Description	Example
D	Video data	Surveillance video
F_{model}	Video analytics model	Car detector
F_A	Aggregate function	Average
f	Reduced sampling	0.1
p	Reduced resolution	128×128
c	Restricted object	Person
$1-\delta$	Confidence level	95%
$X_1, ..., X_N$	Model outputs on original frames	# cars in 1000 frames
$x_1, ..., x_n$	Model outputs on degraded frames	# cars in 100 frames (128×128) with no people contained
$v_1, ..., v_m$	Model outputs on correction set	# cars in randomly sampled 200 frames
Y_{true}	True query answer	Average value of $X_1, ..., X_N$
Y_{approx}	Approximate answer	Average value of $x_1, ..., x_n$
err_b	Upper bound of approximation error	Upper bound of the relative error $\lvert\frac{Y_{approx}-Y_{true}}{Y_{true}}\rvert$

Table 2: Frequently used notation

- **Public preferences**: Preferences that guide the administrator when choosing a tradeoff. Forms of preference include: the minimum allowable analytical error, the maximum allowable frame resolution, and so on.

Consider Harry using our model:

Example 3. *Harry activates our profiling model for his query. During the* **profile generation** *stage, the system produces the* **profiles** *by degrading a representative portion of* **original video** *under multiple sets of interventions and sending the* **degraded video** *to the query processor for analytical error bound estimation, then returns the* **profiles** *to Harry. During the* **choosing a tradeoff** *stage, Harry determines a proper set of interventions according to the* **public preferences**, *so he tunes the knobs and runs the car-counting query on the appropriately degraded video to obtain an approximate query result that is within 10% of the correct result.*

2.4 Technical Formulation

With the above design of the degradation-accuracy profiling model, we still face several technical problems. In the *choosing a tradeoff* stage, algorithms are needed to estimate the query result under destructive interventions. In the *profile generation* stage, algorithms are needed to estimate the analytical accuracy under a broad range of degradation settings; this stage should operate on video that is degraded as much as possible while still yielding a valid profile. These problems can be stated formally, and all of the notation is listed in Table 2.

The video analytical query is characterized by a 3-tuple of parameters (D, F_{model}, F_A). The video data D is queried to collect useful information. Two functions, F_{model} and F_A, represent the video analysis model (e.g., car detector) and the aggregate function (e.g., AVG) in the query. This analysis model's behavior is our definition of the ground truth. The value N is the number of frames that should be sent to F_{model} and F_A in naïve execution. In addition, the 3-tuple of parameters (f, p, c) represent the destructive interventions, which are reduced frame sampling, reduced frame resolution, and restricted objects respectively. The analytical query answer should be executed under these interventions, that is, only n ($n = N \times f$)

frames with resolution p (e.g., 128×128), which do not contain objects c (e.g., person), may be processed by F_{model} and F_A to obtain the approximate query answer Y_{approx}. The value err_b, computed to reflect the analytical accuracy, denotes the upper bound of the relative error of the approximate query result compared with the true result with probability at least $1 - \delta$.

Problem 1: *Given video analytical query* (D, F_{model}, F_A), *compute the approximate query answer* Y_{approx} *and a tight upper bound* err_b *of the approximation error under destructive interventions* (f, p, c).

Problem 2: *In the profile generation stage, compute the profiles while maximizing the interventions.*

3 ALGORITHMS

Now we introduce our novel algorithms to solve the above problems. In Section 3.1, we describe the administration procedure in the stages of profile generation and choosing a tradeoff. In Section 3.2, we propose our query answer and error bound estimation algorithms for frequently used aggregate query types. In Section 3.3, we further discuss the details of profile generation in our model.

3.1 Administration Procedure

In the *profile generation* stage, provided with a query with an analysis model F_{model} and an aggregate function F_A, a tight upper bound of analytical error is computed when the original video D is degraded by every set of intervention candidates. (Intervention candidate selection will be discussed in Section 3.3.2.) These error bounds can form a degradation hypercube with cube slices as multiple two-dimensional arrays that are returned to the administrators in order to choose an appropriate set of interventions. Initially, administrators are only shown three cube slices — obtained by fixing each unseen dimension to the loosest intervention value — visualized as 2D plots. They choose intervention candidates by considering both public preferences (e.g., images that contain people should be removed) and the interventions' effect on analytical accuracy shown in the curves (e.g., resolution 128×128 makes query results too inaccurate), and then adjust the fixed dimensions for more plots, and fine-tune these knobs according to bounded error values. At last, the query result is estimated by running the query on the video D or upcoming videos processed by the determined degradation operations. The algorithms of estimating analytical results and error bounds are described in the following sections.

3.2 Query Answer and Error Bound Estimation

We describe our estimation algorithms for frequently used aggregate functions. We first address the case of reduced frame sampling in Section 3.2.1 through 3.2.4. Then in Section 3.2.5, we introduce the profile repair strategy for non-random interventions.

3.2.1 AVG Function.

The aggregate function `AVG()` is applied to calculate the frame-level average value of a user-defined vision model's outputs on video frames. In Example 1, the public administrator, Harry, applies this function to collect the average number of cars per frame in order to learn how busy the road is. Let $X_1, X_2, ..., X_N$ denote the outputs of N frames with mean μ and range R. Due to the reduced frame sampling intervention f, only n frames are randomly sampled, yielding outputs $x_1, x_2, ..., x_n$. The relative error of the approximate query result Y_{approx} compared with the true query result μ, $|\frac{Y_{approx}-\mu}{\mu}|$, is used as the analytical accuracy metric, so we aim to compute the upper bound of this relative error.

Many research efforts have focused on this computation. When the sample size is relatively large, sample mean approximately obeys the normal distribution according to the central limit theorem, so the upper bound of absolute error between sample mean and true mean can be derived [30], and then the upper bound of relative error can be obtained by dividing the lower bound of the query result. However, it is highly probable that the administrator chooses the sample fraction to be a small value. In other words, the central limit theorem will become useless exactly in the scenarios where our system aims to be the most useful. Online aggregation [30] also provides another more conservative bound from Hoeffding's inequality [31]. Besides these classic approaches, a tighter upper bound can be derived from the Hoeffding-Serfling inequality [8] proposed recently, which assumes sampling without replacement instead of $i.i.d.$ sampling. Moreover, early stopping algorithms — determining a stopping point when the error is within some threshold — can also be adapted for the error bound estimation. The empirical Bernstein stopping algorithm [48] provides a new query result estimation instead of sample mean, yielding a tighter error bound. We further improve this method by relaxing the confidence interval construction requirement and applying the tight Hoeffding-Serfling inequality [8], which is more suitable for a small sample size than the empirical Bernstein bound [7] in the original version. This estimation mechanism is shown in Algorithm 1.

Algorithm 1: AVG()

Input: Aggregate query (D, F_{model}, AVG), Intervention f, δ

`// Sample model outputs`

$x_1, x_2, ..., x_n = F_{model}(\text{Sample}(D, f))$;

`// Calculate Hoeffding-Serfling bound` I

Compute sample range R and sample mean $\bar{x}_n$;

$\rho_n = \min\{(1 - \frac{n-1}{N}), (1 - \frac{n}{N})(1 + \frac{1}{n})\}$;

$I = R\sqrt{\frac{\rho_n \log(2/\delta)}{2n}}$;

`// Compute approximate result and error bound`

$UB = |\bar{x}_n| + I$;

$LB = \max(0, |\bar{x}_n| - I)$;

$Y_{approx} = sgn(\bar{x}_n) \cdot \frac{2UB \cdot LB}{UB + LB}$;

$err_b = \frac{UB - LB}{UB + LB}$;

Output: Y_{approx}, err_b

Hoeffding-Serfling inequality states that with probability at least $1 - \delta$, $\bar{x}_n - \mu \leq R\sqrt{\frac{\rho_n \log(1/\delta)}{2n}}$, where $\bar{x}_n$ is the sample mean: $\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$, $\{x_i\}$ is sampled without replacement, and $\rho_n = \min\{(1 - \frac{n-1}{N}), (1 - \frac{n}{N})(1 + \frac{1}{n})\}$. Similarly, with this confidence level, $\bar{x}_n - \mu \geq -R\sqrt{\frac{\rho_n \log(1/\delta)}{2n}}$. Due to union bound, with probability at least $1 - \delta$, $|\bar{x}_n - \mu| \leq R\sqrt{\frac{\rho_n \log(2/\delta)}{2n}}$. We denote this bound as I, so $(\bar{x}_n - I, \bar{x}_n + I)$ is a $1 - \delta$ confidence interval for μ. In contrast to the empirical Bernstein stopping algorithm, we do not need to simultaneously construct the intervals for all $n \in \mathbb{N}^+$ but just for the sample size n

under $1-\delta$ confidence level. As a result, this confidence interval can be smaller by our construction. Correspondingly, we set LB to $\max(0, |\bar{x}_n| - I)$ and UB to $|\bar{x}_n| + I$ rather than the definitions in the stopping algorithm.

Theorem 3.1. *The approximate query answer Y_{approx} and the error bound err_b with probability at least $1-\delta$ are as follows:*

$$Y_{approx} = sgn(\bar{x}_n) \cdot \frac{2UB \cdot LB}{UB + LB}, \tag{1}$$

$$err_b = \frac{UB - LB}{UB + LB}. \tag{2}$$

Proof. With probability at least $1-\delta$,

$$|Y_{approx}| = \frac{2UB \cdot LB}{UB + LB} = (1 + err_b) LB \le (1 + err_b)|\mu|, \tag{3}$$

$$|Y_{approx}| = \frac{2UB \cdot LB}{UB + LB} = (1 - err_b) UB \ge (1 - err_b)|\mu|. \tag{4}$$

When $LB = 0$, it can be derived that $Y_{approx} = 0$ and $err_b = 1$, so err_b is the error bound. When $LB \neq 0$, the inequality $|\bar{x}_n| > I \ge |\bar{x}_n - \mu|$ holds true, so $sgn(Y_{approx}) = sgn(\bar{x}_n) = sgn(\mu)$, where $sgn()$ is the sign function. We can obtain the following inequality:

$$\left| \frac{Y_{approx} - \mu}{\mu} \right| = \frac{||Y_{approx}| - |\mu||}{|\mu|} \le err_b \tag{5}$$

Therefore, the above theorem holds true. □

3.2.2 SUM Function.

The aggregate function `SUM()` is applied to calculate the sum of the model's outputs in each frame. This function can be used to compute the sum of all cars seen in each frame in a time period. It captures both car number and car speed information, which is valuable for determining the road congestion level. The parameters and the error metric are the same as that in Section 3.2.1. In this case, $Y_{true} = N\mu$, so we compute the upper bound of the relative error $|\frac{Y_{approx} - N\mu}{N\mu}|$. We assume that the length of video is known before any processing. According to the conclusion in Section 3.2.1, we define $Y_{approx} = sgn(\bar{x}_n) \cdot \frac{2UB \cdot LB}{UB + LB} \cdot N$ and $err_b = \frac{UB - LB}{UB + LB}$ to make err_b the error bound with probability at least $1-\delta$.

3.2.3 COUNT Function.

The aggregate function `COUNT()` is applied to calculate the number of frames that satisfy the query predicate. This function can be used to compute the number of frames (i.e., the length of time) when there are varying levels of cars. It would be helpful to decide when congestion is low enough to close a single lane. Although this seems like a new problem, we can redefine it as the estimation problem for `SUM`. For each frame i, if the predicate model returns TRUE, we assign an associated value 1 to X_i; otherwise, 0 is assigned to X_i. Therefore, the count problem is transformed to calculating the sum of X_i, and the conclusion in Section 3.2.2 can be directly applied here.

3.2.4 MAX/MIN Function.

The aggregate function `MAX()` or `MIN()` is applied to calculate extreme values in the frame-level outputs. This function can be used to compute the maximum/minimum number of cars that exist in one frame in order to detect the most/least crowded moment. Unfortunately, it is hard to estimate and analyze extreme values just by sampling, because only the extreme value itself in the samples seems to be related to the true result. Therefore, we use rth-quantile to estimate the result of `MAX()` and `MIN()` (when r is close to 1 or 0). The goal is transformed into estimating the rth-quantile in the outputs, $X_1, X_2, ..., X_N$. There are n frames randomly sampled without replacement for processing, yielding $x_1, x_2, ..., x_n$. For quantiles, BlinkDB [3] uses the same relative error metric as other aggregate query types. However, this metric is substantially affected by the hidden distribution, especially for extreme quantiles. As a result, the ranks rather than the actual values are compared, that is, the relative error between the ranks of Y_{true} and Y_{approx} in the original array, $|\frac{rank(Y_{approx}) - rank(Y_{true})}{rank(Y_{true})}|$, is used to reflect the accuracy. This metric is also compatible with the definition of ϵ-approximate quantile [44].

Previous works [40, 45] have designed sampling-based algorithms to estimate quantiles in wireless sensor networks and for business intelligence applications. The classic approach [45] proposed an estimation based on Stein's lemma. A recent work [40] made estimates based on the central limit theorem. However, there are two problems in these algorithms. First, the inequality bound is too loose during the derivation process. Second, they assume random sampling with replacement, which is less reasonable than our non-replacement assumption. Both of them lead to loose upper bounds. We make improvements based on recent work [40] (the novelty is summarized in Table 1). We propose the quantile approximation algorithm as follows, shown in Algorithm 2, and compare our algorithm with the better approach [45] between the above two previous works in Section 5.2.1.

Algorithm 2: MAX() or MIN()

Input: Aggregate query (D, F_{model}, F_A), Intervention f, Extreme percentage r, δ

`// Sample model outputs`

$x_1, x_2, ..., x_n = F_{model}(\text{Sample}(D, f))$;

`// Compute approximate result and error bound`

$sortList = \text{Sort}(x_1, x_2, ..., x_n)$;

$Y_{approx} = sortList[n \cdot r]$;

$\hat{F}_{\hat{k}} = sortList.\text{count}(Y_{approx}) \,/\, n$;

if F_A == MAX **then**

  $err_b = \left(\frac{\phi_{\frac{\delta}{2}} \sqrt{r(1-r)} \sqrt{\frac{N-n}{n(N-1)}} + \hat{F}_{\hat{k}}}{\hat{F}_{\hat{k}}} + 1\right) \cdot \frac{\hat{F}_{\hat{k}}}{r}$

end

else

  $err_b = \left(\frac{\phi_{\frac{\delta}{2}} \sqrt{(r+\hat{F}_{\hat{k}})[1-(r+\hat{F}_{\hat{k}})]} \sqrt{\frac{N-n}{n(N-1)}} + \hat{F}_{\hat{k}}}{\hat{F}_{\hat{k}}} + 1\right) \cdot \frac{\hat{F}_{\hat{k}}}{r}$

end

Output: Y_{approx}, err_b

Let $\{s_1, s_2, ...\}$ be the sorted distinct values in $X_1, X_2, ..., X_N$. Each s_i occurs N_i times in this array and n_i times in the sampled array, the frequency of which is $F_i = \frac{N_i}{N}$ and $\hat{F}_i = \frac{n_i}{n}$ respectively. According to the definition of rth-quantile, $Y_{true} = \min_i\{s_i : \sum_{j=1}^{i} F_j \ge r\}$. Let Y_{true} and Y_{approx} be the kth and $\hat{k}$th distinct value, i.e., $Y_{true} = s_k$ and $Y_{approx} = s_{\hat{k}}$.

Theorem 3.2. *The approximate quantile Y_{approx} and error bound err_b with probability at least $1-\delta$ can be constructed as follows.*

$$Y_{approx} = \min_i\{s_i : \sum_{j=1}^{i} \hat{F}_j \geq r\}. \tag{6}$$

When the aggregate function is MAX, r is close to 1,

$$err_b = \left(\frac{\phi_{\frac{\delta}{2}}\sqrt{r(1-r)}\sqrt{\frac{N-n}{n(N-1)}} + F_k}{\min_{\hat{k}+1\leq i\leq k-1 \text{ or } k+1\leq i\leq \hat{k}-1}\hat{F}_i} + 1\right) \cdot \frac{\max_{\hat{k}+1\leq i\leq k \text{ or } k+1\leq i\leq \hat{k}} F_i}{r}. \tag{7}$$

And when the aggregate function is MIN, r is close to 0,

$$err_b = \left(\frac{\phi_{\frac{\delta}{2}}\sqrt{(r+F_k)[1-(r+F_k)]}\sqrt{\frac{N-n}{n(N-1)}} + F_k}{\min_{\hat{k}+1\leq i\leq k-1 \text{ or } k+1\leq i\leq \hat{k}-1}\hat{F}_i} + 1\right) \cdot \frac{\max_{\hat{k}+1\leq i\leq k \text{ or } k+1\leq i\leq \hat{k}} F_i}{r}. \tag{8}$$

Proof sketch: The error metric satisfies the inequality:

$$error = \frac{|\sum_{i=1}^{k} F_i - \sum_{i=1}^{\hat{k}} F_i|}{\sum_{i=1}^{k} F_i} \leq \frac{|k-\hat{k}|\max_{\hat{k}+1\leq i\leq k \text{ or } k+1\leq i\leq \hat{k}} F_i}{r}. \tag{9}$$

According to the definition, we have $\sum_{i=1}^{\hat{k}-1}\hat{F}_i < r \leq \sum_{i=1}^{k} F_i$ and $\sum_{i=1}^{k-1} F_i < r \leq \sum_{i=1}^{\hat{k}}\hat{F}_i$. So when $\hat{k} > k$, $\hat{k} - k < \frac{\sum_{i=1}^{k} F_i - \sum_{i=1}^{\hat{k}}\hat{F}_i}{\min_{k+1\leq i\leq \hat{k}-1}\hat{F}_i} + 1$, and when $k > \hat{k}$, $k - \hat{k} < \frac{\sum_{i=1}^{k}\hat{F}_i - \sum_{i=1}^{k} F_i + F_k}{\min_{\hat{k}+1\leq i\leq k-1}\hat{F}_i} + 1$. Therefore,

$$error < \left(\frac{|\sum_{i=1}^{k}\hat{F}_i - \sum_{i=1}^{k} F_i| + F_k}{\min_{\hat{k}+1\leq i\leq k-1 \text{ or } k+1\leq i\leq \hat{k}-1}\hat{F}_i} + 1\right) \cdot \frac{\max_{\hat{k}+1\leq i\leq k \text{ or } k+1\leq i\leq \hat{k}} F_i}{r}. \tag{10}$$

Because $\sum_{i=1}^{k}\hat{F}_i = \frac{\sum_{i=1}^{k} n_i}{n}$, and $\sum_{i=1}^{k} n_i$ obeys hypergeometric distribution, we can obtain that $\mathbb{E}[\sum_{i=1}^{k}\hat{F}_i] = \sum_{i=1}^{k} F_i$ and $Var[\sum_{i=1}^{k}\hat{F}_i] = \sum_{i=1}^{k} F_i(1-\sum_{i=1}^{k} F_i)\cdot\frac{N-n}{n(N-1)}$. It has been demonstrated that there is a normal approximation for the hypergeometric distribution when $N, n, \sum_{i=1}^{k} N_i, \sum_{i=1}^{k} n_i$ are large [19, 50], so there exists an asymptotic normal distribution: $\frac{\sum_{i=1}^{k}\hat{F}_i - \sum_{i=1}^{k} F_i}{\sqrt{Var[\sum_{i=1}^{k}\hat{F}_i]}} \sim \mathcal{N}(0,1)$. When r is close to 1, $Var[\sum_{i=1}^{k}\hat{F}_i] \leq r(1-r)\cdot\frac{N-n}{n(N-1)}$, so

$$P\left(|\sum_{i=1}^{k}\hat{F}_i - \sum_{i=1}^{k} F_i| \geq \phi_{\frac{\delta}{2}}\sqrt{r(1-r)}\sqrt{\frac{N-n}{n(N-1)}}\right) \leq \delta, \tag{11}$$

where $\phi_{\frac{\delta}{2}}$ is the Z-score. Therefore $P(error \geq err_b) \leq \delta$ is satisfied. When r is close to 0, $Var[\sum_{i=1}^{k}\hat{F}_i] \leq (r+F_k)[1-(r+F_k)]\cdot\frac{N-n}{n(N-1)}$. Similarly, the theorem holds true.

In the formula of err_b, F_i (for any $i \in \mathbb{N}^+$), k, and $\hat{k}$ are unknown. Ideally, $\hat{F}_i$ and F_i, k and $\hat{k}$ should be close, so we use $\hat{F}_{\hat{k}}$ to estimate F_k, $\min_{\hat{k}+1\leq i\leq k-1 \text{ or } k+1\leq i\leq \hat{k}-1}\hat{F}_i$, and $\max_{\hat{k}+1\leq i\leq k \text{ or } k+1\leq i\leq \hat{k}} F_i$ above. It needs to be noted that a distribution approximation is utilized in the above proof. Although it holds true when sample size is large, the derived error bound is still valid experimentally in Section 5 even when sample size is vary small.

3.2.5 Managing Combinations of Random and Non-random Interventions through Profile Repair.

We have provided the error bound for different aggregate functions for random interventions. However, the algorithms cannot be directly applied when there are non-random interventions because sampled outputs from videos degraded by non-random interventions can be systematically wrong in one direction. Under this circumstance, these sampled outputs are not enough for an accurate error bound. A correction set, $v_1, v_2, ..., v_m$, obtained from processing videos degraded by only random interventions, is required to repair the biased bound. Its construction is elaborated in Section 3.3.1. Once it is constructed, it can be used for correcting error bounds of any combination of interventions. Our algorithm is shown in Algorithm 3, and the proof sketches for the error bounds are presented below (Equation (12) and (13)). These proofs leverage the error bound conclusion connected with random interventions (see Theorem 3.1 and 3.2). That is, the error bound of the correction set under a certain confidence level has been proved, and it is utilized in the inequality derivation below. Further, note there is no distributional assumption of the outputs from videos degraded by non-random interventions.

Algorithm 3: Managing a Combination of Random and Non-random Interventions

Input: Aggregate query (D, F_{model}, F_A), Destructive interventions (f, p, c), δ, r, m

```
// Compute approximate result and error bound of
   the degraded video and the correction set
```

Y_{approx}, err_b = resultErrorEst(D, F_{model}, F_A, f, p, c, δ, r);

$Y_{approx}(\mathbf{v})$, $err_b(\mathbf{v})$ = resultErrorEst(D, F_{model}, F_A, m/len(D), None, None, δ, r);

```
// Correct the error bound of degraded video
```

if F_A == AVG or SUM or COUNT **then**

  $err_b = \frac{(1+err_b(\mathbf{v}))|Y_{approx} - Y_{approx}(\mathbf{v})|}{|Y_{approx}(\mathbf{v})|} + err_b(\mathbf{v})$

end

if F_A == MAX or MIN **then**

  $\sum_{i=1}^{\hat{k}}\hat{F}_i$ = Rank of Y_{approx} in correction set / m;

  $\sum_{i=1}^{\hat{k}(\mathbf{v})}\hat{F}_i$ = Rank of $Y_{approx}(\mathbf{v})$ in correction set / m;

  $err_b = \frac{|\sum_{i=1}^{\hat{k}}\hat{F}_i - \sum_{i=1}^{\hat{k}(\mathbf{v})}\hat{F}_i|}{r} + err_b(\mathbf{v})$

end

Output: err_b

For the aggregate function `AVG()`, we assume that $v_1, v_2, ..., v_m$ are randomly sampled outputs without replacement. The approximate answer $Y_{approx}(\mathbf{v})$ to estimate μ and the error bound $err_b(\mathbf{v})$ obtained only from the correction set as in Equation (1) and (2) can satisfy $\frac{|Y_{approx}(\mathbf{v})-\mu|}{|\mu|} \leq err_b(\mathbf{v})$, with probability at least $1-\delta$. So when non-random interventions exist, the error bound for the approximate result Y_{approx} can be derived as follows:

$$\begin{aligned}\frac{|Y_{approx}-\mu|}{|\mu|} &\leq \frac{|Y_{approx} - Y_{approx}(\mathbf{v})| + |Y_{approx}(\mathbf{v}) - \mu|}{|\mu|} \\ &\leq \frac{(1+err_b(\mathbf{v}))|Y_{approx} - Y_{approx}(\mathbf{v})|}{|Y_{approx}(\mathbf{v})|} + err_b(\mathbf{v}).\end{aligned} \tag{12}$$

Since it is derived from the error bound of the correction set, this error bound also holds true with probability at least $1-\delta$. And for

other functions, SUM() and COUNT(), because the error metric is the same, the corrected error bound can be derived similarly.

For the aggregate function MIN() or MAX(), the approximate rth-quantile $Y_{approx}(\mathbf{v})$ and the error bound $err_b(\mathbf{v})$ obtained only from the correction set as in Equation (6), (7), and (8) can satisfy $\frac{|\sum_{i=1}^{\hat{k}(\mathbf{v})} F_i - \sum_{i=1}^{k} F_i|}{\sum_{i=1}^{k} F_i} \leq err_b(\mathbf{v})$ with probability at least $1-\delta$, where $Y_{approx}(\mathbf{v})$ is the $\hat{k}(\mathbf{v})$th distinct value. So

$$\frac{|\sum_{i=1}^{\hat{k}} F_i - \sum_{i=1}^{k} F_i|}{\sum_{i=1}^{k} F_i} \leq \frac{|\sum_{i=1}^{\hat{k}} F_i - \sum_{i=1}^{\hat{k}(\mathbf{v})} F_i| + |\sum_{i=1}^{\hat{k}(\mathbf{v})} F_i - \sum_{i=1}^{k} F_i|}{\sum_{i=1}^{k} F_i} \leq \frac{|\sum_{i=1}^{\hat{k}} F_i - \sum_{i=1}^{\hat{k}(\mathbf{v})} F_i|}{r} + err_b(\mathbf{v}), \tag{13}$$

with probability at least $1-\delta$. In this formula, the true rank difference $|\sum_{i=1}^{\hat{k}} F_i - \sum_{i=1}^{\hat{k}(\mathbf{v})} F_i|$ is unknown, so we use the rank difference $|\sum_{i=1}^{\hat{k}} \hat{F}_i - \sum_{i=1}^{\hat{k}(\mathbf{v})} \hat{F}_i|$ between Y_{approx} and $Y_{approx}(\mathbf{v})$ in the correction set to estimate it.

3.3 Discussion

In this section, we further discuss details of the profile generation stage.

3.3.1 Correction Set Construction.
As introduced in Section 3.2.5, the correction set is necessary for estimating the analytical accuracy of non-random interventions. It can also improve the error bound of random interventions when the correction set can provide substantially more information than the degraded video, as shown in Section 5.2.2. Constructing the correction set requires access to videos with random interventions alone; non-random interventions are not permissible. However, using only random interventions is feasible in many cases. Instead of using a non-random intervention, the administrator might be willing to apply a random one at a very high degradation level. (For example, they may choose a lower sampling rate instead of lowering frame resolution.) In addition, since the correction set is only required in the profile generation stage, it may be acceptable to permit a lower level of degradation for just a limited amount of time.

The correction set should still be degraded by the random interventions as much as possible; in the context of reduced frame sampling, that means minimizing the set's size. However, there is a limit to how much degradation can be introduced, since we want to ensure a tight error bound for the downstream process. According to the definition of the corrected bound in Equation (12) and (13), when $err_b(\mathbf{v})$ is smaller, the corrected error bound is tighter. Therefore, we need to achieve low $err_b(\mathbf{v})$ while using as few frames in the correction set as possible, i.e., picking the elbow of the curve of $err_b(\mathbf{v})$ against m, the size of the correction set. In our design, we use a simple heuristic to determine the size: the correction set's size is increased gradually by 1% of the total size of the original video to output $err_b(\mathbf{v})$. Once the difference between the current and the previous output is less than 2%, which means the value $err_b(\mathbf{v})$ does not change much with the correction set's size (i.e., the elbow), or the current size reaches the size limit defined by the administrator, we stop growing the correction set.

If pure random interventions are not allowed or only substantial random interventions are allowed, it may be that no correction set or only a small correction set is possible. In that case, an alternative method to approximate the tradeoff curve may be generating profiles on less privacy-sensitive video at another time. Videos at different times might interact with analytical models in different ways, but they are expected to be visually similar and will yield roughly similar profiles. Experiments in Section 5.3.2 demonstrate that similar profiles arise from visually-similar video collections.

3.3.2 Intervention Candidate Design and Time Complexity.
Our system first considers many possible sets of destructive interventions (f, p, c). For the reduced frame sampling f, similar to the correction set design, we consider sample fractions at 1% intervals. For reduced frame resolution p and image removal c, we uniformly generate ten frame resolutions and all combinations of possibly sensitive classes. Then administrators filter out the intervention candidates that cannot satisfy degradation goals.

The total time of profile generation includes time for the neural network model to process frames plus the analytical error estimation time. Estimation time is usually negligible compared with the network's image processing time (discussed in Section 5.3.1). Model processing time is $O(N_{model} \cdot T_{model})$, where N_{model} is the total number of model invocations, and T_{model} is the averaged processing time on each frame, including loading, transformation and inference. An early stopping and reuse strategy can be applied to decrease N_{model}. For each resolution candidate, the error bound is estimated for frame sampling rate candidates in ascending order. In this way, model outputs for frames sampled at a low rate can be reused for the outputs at a high rate, and the estimation process can stop early when the error bound decreases slowly. As a result, the profile generation overhead is modest.

4 SYSTEM PROTOTYPE

We implemented a prototype system, Smokescreen, in Python. This system ran on a 64-core (2.10GHz) Intel Xeon Gold 6130 server with 512 GB RAM and 4 GeForce GTX 1080 Ti GPUs. It embodies our novel algorithms and contains three main components: 1) video frame processor, 2) analytical result and error bound estimator, and 3) correction set and intervention candidate design.

Video frame processor — This component processes video frames by calling the UDFs in queries. We use YOLOv4 [11] and Mask R-CNN [28] as two built-in models for detection UDFs. YOLOv4 has been implemented based on a neural network framework, Darknet [55], written in C and CUDA. This model is invoked through a Python interface in this component. And we directly apply a Mask R-CNN implementation based on Keras and Tensorflow [1]. The processed video frames are from decoded videos which are stored on a disk for downstream processing. Only one frame can be loaded, resized, and processed at a time (i.e., no batch computation), and all the model inference procedures run on a GPU.

Analytical result and error bound estimator — This component consists of our estimation algorithms in Section 3. The cost of the estimation itself is relatively small.

Correction set and intervention candidate design — This component determines the correction set size and the sets of intervention candidates, working in the profile generation stage. By calling the above estimation component to process video frames of different sizes, the error bound differences are computed and the size of the correction set is determined as stated in Section 3.3.1. This component also interacts with administrators to collect intervention candidates, which will then be sent to the error bound estimator.

5 EXPERIMENTS

We evaluate three core claims about Smokescreen:

(1) For random destructive interventions, our algorithm can provide a tighter analytical error bound than competing methods. This holds true for every aggregate query type. (Section 5.2.1)
(2) For both random and non-random destructive interventions, the correction set can improve the performance of error bound estimation. And our technique can efficiently determine an appropriate correction set size. (Section 5.2.2 - 5.2.3)
(3) The discussion of profile generation time and profile similarity between similar videos is demonstrated. (Section 5.3)

5.1 Experimental Setting

We describe our workloads, baselines, and accuracy metrics.

Workloads — We evaluated our system on multiple workloads. Each workload consists of a video dataset, a trained neural network to process video frames, an aggregate function to collect useful information, and a set of destructive interventions. Every workload was run 100 times, and the experimental results below are the averaged results of 100 trials of the following workloads unless stated otherwise.

- **Video dataset** — The video set is one of either night-street video or UA-DETRAC video. The night-street video is surveillance video of a street in Jackson Hole at night, which is released by the BlazeIt project [34]. It contains 973k frames in total and the frame rate is 30 FPS. We selected one out of every fifty frames (19463 frames) to construct our dataset. The UA-DETRAC video [64] is recorded at Beijing and Tianjin in China. It contains 40 sequences (56k frames) in its test dataset and the frame rate is 25 FPS. We selected 12 sequences (15210 frames) for our experiments.
- **Neural network model** — We used Mask R-CNN [28] for night-street video and YOLOv4 [11] for UA-DETRAC video to detect cars. The detection threshold was set to be 0.7 for both of the models. Although the confidence output associated with each detected object can further improve the detection accuracy when averaged over frames, we just utilized the object output alone in each frame for simplicity because we assume the model output as the ground truth and our work does not try to improve the model's standalone accuracy.
- **Aggregate function** — The aggregate function is one of `AVG`, `SUM`, `MAX`, or `COUNT`. In our experiments, they were used to compute the average, sum, maximum of the number of cars in frames, and count the number of frames that contain cars respectively. For `MAX`, our system estimates 0.99 quantile as an approximation of the maximum value.
- **Destructive intervention** — A set of destructive interventions is composed of reduced frame sampling fraction, reduced frame resolution, and the restricted class for image removal. We assumed the video with the original length and the highest resolution as the original video. We set the highest resolution to be 640×640 for Mask R-CNN and 608×608 for YOLOv4. In our experiments, the sample fraction can be any value less than one, the frame resolution should be lower than the highest value and meet models' requirements (e.g., the default structure of Mask R-CNN can only handle the resolution in multiples of 64), and restricted classes include "person" and "face". We detected "person" by applying YOLOv4 with detection threshold 0.7, and detected "face" by applying MTCNN [69] with threshold 0.8. Restricting "person" is usually a more strict intervention because people can appear in cameras with unclear faces. According to the detector, 2761 frames (14.18%) contain "person" and 782 frames (4.02%) contain "face" in night-street data; 10018 frames (65.86%) contain "person" and 377 frames (2.48%) contain "face" in UA-DETRAC data. These contained classes for each frame were stored as prior information.

Baselines — We evaluated against the first four baselines for `AVG`, `SUM`, and `COUNT`, and evaluated against the last baseline for `MAX`.

- **EBGS** — The EBGS algorithm [48] is widely used for early stopping when estimated error is within some small number. We directly used it to estimate the query result and error bound instead of using the stopping mechanism.
- **Hoeffding-Serfling** — The upper bound of absolute error can be derived from the Hoeffding-Serfling inequality [8]. We divided it by the lower bound of the query result in order to obtain the upper bound of relative error for comparison.
- **Hoeffding** — Online aggregation [30] provides the upper bound of absolute error from Hoeffding's inequality. Then we processed it in the same way as above.
- **CLT** — Online aggregation [30] also provides the upper bound of absolute error from the central limit theorem. Then we processed it in the same way as above.
- **Stein** — [45] minimizes the sample size that can ensure the ϵ-approximate extreme quantile based on Stein's lemma. We directly used it to derive the error bound.

Accuracy Metrics — As introduced in Section 3.2, the relative error of the approximate query result compared with the true result was used as the accuracy metric when querying with aggregate functions `AVG`, `SUM` or `COUNT`, and the relative error of the approximate result's true rank compared with the true result's true rank was used when querying with `MAX` in our experiments. We treated the query result without destructive interventions as the true result. Our algorithms computed upper bounds of these relative errors and we compared them with the true relative errors.

5.2 Analytical Result and Accuracy Estimation

We show that our query result and accuracy estimation algorithms are effective across a range of video data, models and aggregate query types for both random and non-random interventions.

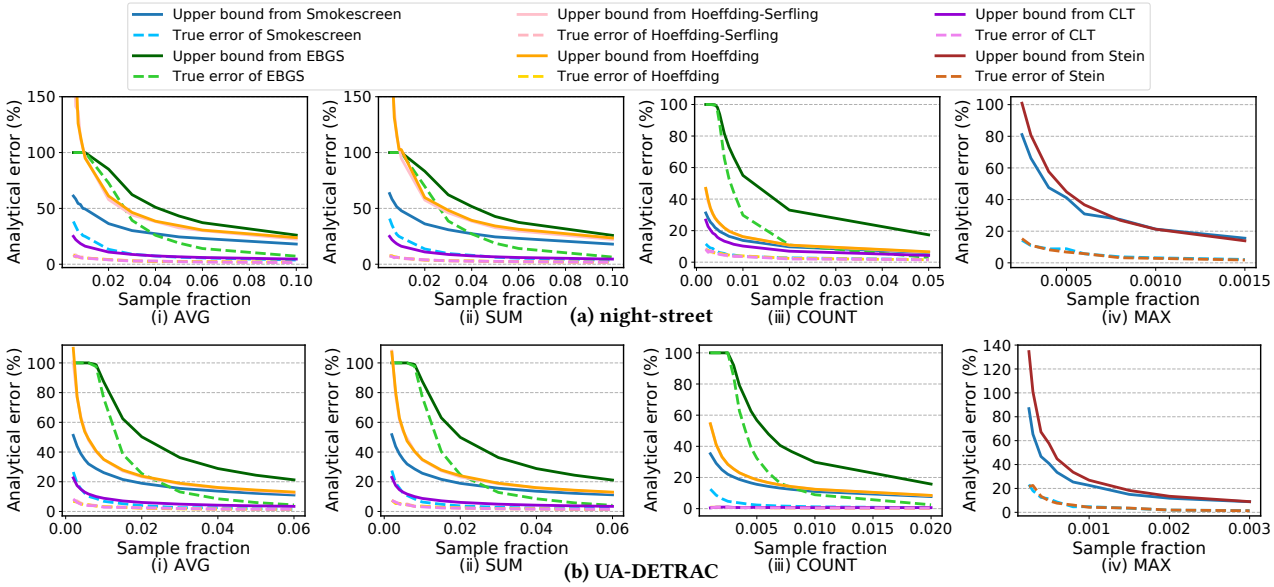

Figure 4: The true relative error of estimated query result (dashed lines) and error bound (solid lines) computed from Smokescreen and baselines for each aggregate query type on two datasets

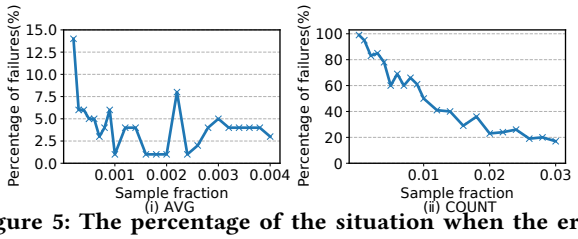

Figure 5: The percentage of the situation when the error bound from CLT is smaller than the true error in 100 trials

5.2.1 Managing Random Interventions.

Summary — For random destructive interventions, our basic algorithms can provide good estimated analytical results and tighter upper bounds of relative errors compared with reliable competing methods for every aggregate query type. Our error bound can be up to 154.70% tighter than baselines, and the tight bound can enable tradeoffs that are 88% more accurate.

Overview — We evaluated our query result and error estimation algorithms against competing methods on four aggregate query types, two video datasets and two models. When we varied the sample fraction, we did not tune other destructive interventions. And when we varied the frame resolution or restricted class, the sample fraction was set to be 0.5. For better comparison with baselines, no correction set was used in this experiment.

Results — We show the true relative error of estimated query result and the error bound computed by each method in Figure 4. It shows the results varying with the reduced frame sampling intervention. From the true analytical error of Smokescreen (blue dashed lines), we can find that for every aggregate function, the sample fraction increases, the true estimation error goes down and approaches zero. Since the curves have flattened, for the four query types, we end them when the fraction is 0.1, 0.1, 0.05, 0.0015 for night-street video, and 0.06, 0.06, 0.02, 0.003 for UA-DETRAC video. The curves indicate that our algorithm can collect useful information from the samples and show good performance even when the sample fraction is relatively small. When looking at the upper bound from Smokescreen (blue solid lines), we can find that they are always higher than the true error curves, which means that our error estimation algorithm for random interventions truly provides an upper bound of the error.

For AVG, SUM and COUNT, the query result and error estimation of Smokescreen are always better than EBGS. When compared with Hoeffding and Hoeffding-Serfling, although our result estimation is less precise, a tight error bound is the more important goal. Our error bound can be up to 154.70% tighter (Due to the range of the y-axis, it is not shown in the figure). All of these algorithms, Smokescreen, EBGS, Hoeffding and Hoeffding-Serfling, can ensure these upper bound estimations are greater than the true errors with at least 95% probability. It seems that CLT can provide an even tighter bound than ours. However, CLT can be brittle and unreliable: it cannot always obtain a bound at the 95% confidence level especially when the sample size is small. Figure 5 shows the percentage of situations when CLT's error bound is smaller than the true error on UA-DETRAC video in 100 trials. These upper bound estimations would provide misleading information for administrators to determine a set of interventions that yield large error beyond expectations. For MAX, our query result estimation is the same as Stein's, but our error bound is tighter when the sample fraction is small.

When non-random interventions are applied, none of the above techniques can provide correct upper bounds, that is, the error estimation cannot be guaranteed to be greater than the true error. These destructive interventions will be handled in the next section.

5.2.2 Managing Combinations of Random and Non-random Interventions.

Summary — Our error correction algorithm can provide a true error bound when non-random interventions exist, and can further

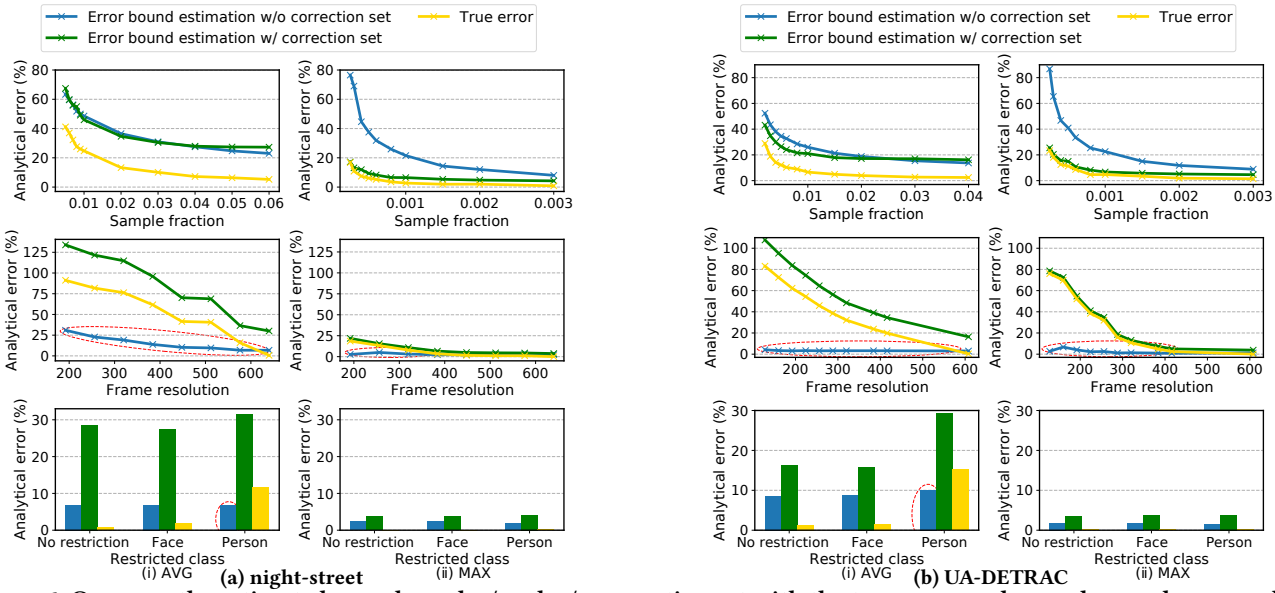

Figure 6: Compare the estimated error bound w/ and w/o correction set with the true error under random and non-random destructive interventions on two video datasets

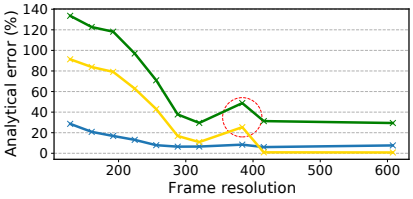

Figure 7: Apply YOLOv4 to compute the average number of cars in night-street video. The relative error is abnormally large when resolution is 384×384. The legend is the same as that in Figure 6.

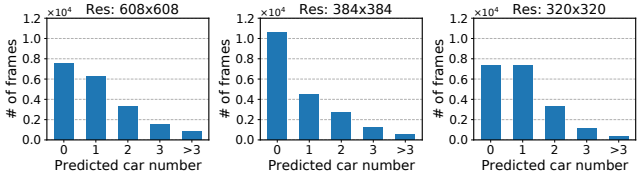

Figure 8: Car number distribution predicted by YOLOv4 in night-street video data

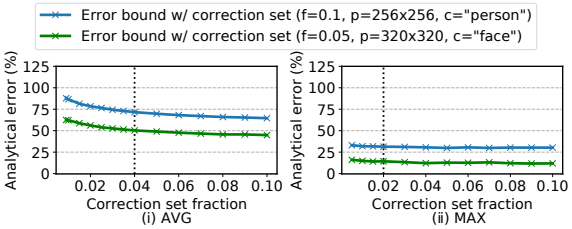

Figure 9: The error bound estimation with different correction set sizes for two sets of destructive interventions on UA-DETRAC video

improve basic algorithms' bound estimations for random interventions in some cases.

Overview — In order to test our error correction algorithm, we compared error bound estimation computed with and without the correction set with the true error under each set of interventions. Because the algorithms for SUM and COUNT are almost the same as that for AVG, we only tested AVG and MAX functions. We set the sizes of the correction sets according to the correction set construction strategy in Section 3.3.1: 6% of the original frames for function AVG and 2% for function MAX for night-street video, and 4% for AVG and 2% for MAX for UA-DETRAC video. (We will explain more in Section 5.2.3.) Only one kind of intervention was tuned at a time and the other two were fixed. When testing the combination situation when both random and non-random interventions exist, we set the sample fraction to be 0.5 while varying non-random interventions. The only exception was that we set the sample fraction to be 0.1 when changing the restricted class for UA-DETRAC video, because the number of frames that do not contain "person" is less than half of the total number.

Results — Figure 6 shows the error bounds with and without the correction set under each set of interventions for AVG and MAX functions. In the second and third rows of Figure 6, when the frame resolution is low or the restricted class is "person", the error bound without correction set (blue curve or blue bar), circled in red, can be lower than the true error (yellow curve or yellow bar), so they are wrong and will mislead administrators. It happens because low-resolution objects are hard to be detected by neural network models and the existences of "person" and "car" are very likely to be correlated, both yielding systematic error in samples. Fortunately, the error correction algorithm can solve this problem: the error bound with correction set (green curve or green bar) is always higher than the true error. From the first row, it shows that the correction set is also helpful for random interventions when the size of the set is much larger than the size of the degraded video (that is, it provides more information). When there is only the random

intervention, the tighter of the error bounds with and without the correction set is used as the error estimation.

Besides Mask R-CNN, we also applied YOLOv4 to detect cars in night-street video, and we noticed an abnormal situation when querying the average number of cars with frame resolution interventions, shown in Figure 7. The estimation error under resolution 384×384, marked in the red circle, is even larger than that under lower resolutions. To find out the reason, we show the predicted car number distribution, i.e., the number of frames that are predicted to contain certain number of cars, in resolution 608×608 (ground truth), 384×384, and 320×320, in Figure 8. It shows that the distribution under resolution 320×320 is similar to the true distribution, while that under resolution 384×384 deviates substantially from the truth. Therefore, the neural network's large prediction error causes the inaccurate result estimation. If not provided with degradation profiles, administrators might unknowingly select this bad intervention that keeps video's good fidelity while yielding a high estimation error. Fortunately, our algorithms can detect this counter-intuitive situation to help administrators make a reasonable tradeoff.

5.2.3 *Correction Set Size.*

Summary — Our algorithm, which determines an appropriate correction set size through the change of its error bound with its size, is effective in real cases so that checking the correction set's performance under every set of interventions can be avoided.

Overview — In this experiment, in order to verify that an appropriate correction set size can be directly obtained from its error bound without considering multiple destructive interventions, we tested two sets of interventions and all four aggregation functions on two datasets. These representative sets of interventions were randomly selected: (1) sample fraction 0.1, frame resolution 256×256 and restricted class "person"; and (2) sample fraction 0.05, frame resolution 320×320 and restricted class "face".

Results — The curves of error bound estimation that varies with correction set fraction for AVG and MAX on UA-DETRAC video are shown in Figure 9, and other cases are similar. In this figure, the x-axis, the correction set fraction, is the proportion of the correction set size to the length of the original video. When it is larger, error bounds become smaller and approach true errors. When the fraction is large enough, the slopes of these curves are close to zero, which means more correction data would not make the estimation more precise, so we should stop increasing the size. According to the mechanism in Section 3.3.1, the determined fractions are shown as dotted vertical lines. We can find that even though two sets of destructive interventions' curves are different, the determined fractions are appropriate choices for both of them because their slopes have dropped down at the intersections with the dotted lines.

5.3 Other Experiments

In this section, we discuss the profile generation time and profile similarity between similar videos.

5.3.1 *Profile Generation Time.*

Summary — The total time of profile generation is dominated by network model processing time, which is determined by the model and intervention candidates.

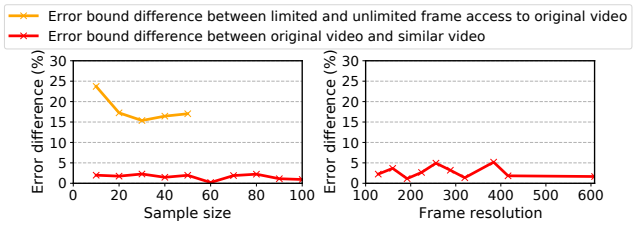

Figure 10: Compare two error bound differences when using or not using a similar video.

Overview — We evaluated the total time of profile generation for the analytical query that employs YOLOv4 to compute the average number of cars in UA-DETRAC video. In the profiles, we set the highest resolution to be 608×608, and ten resolutions were selected as the intervention candidates. And we set the loosest image removal intervention to be no restricted class. As shown in Section 5.2.3, the determined correction set fraction is 0.04 in this case. We also set this value as the highest sample fraction.

Results — YOLOv4 needs to be invoked 6084 times to process 4% of the total frames under every resolution setting, and the total time is around three minutes. Compared with the model processing time, our estimation stage takes only tens of milliseconds for each set of degradation interventions, so the profile generation time is dominated by the former. When the neural network model, the video content, or the intervention candidate settings are different, the profiling time would vary.

5.3.2 *Profile Similarity between Similar Videos.*

Summary — Similar profiles can be generated from a similar video to guide the tradeoff in the original video.

Overview — We computed the profiles of the AVG analytical query with YOLOv4 on two video sequences selected from UA-DETRAC dataset. One video (MVI_40771), denoted as video A and set as the original video, is from a traffic monitoring camera at a busy intersection. Another video (MVI_40775), denoted as video B, is captured by the same camera at a different time, and is visually similar to the original video. They contain 1720 frames and 975 frames respectively. We tested reduced frame sampling and reduced resolution interventions and set the correction set size as 500 for both video A and B. We also tested multiple degradation settings for video A when at most 50 randomly sampled frames can be accessed, which may happen due to a high degradation requirement. We compared the target profiles of video A when 500 frames are sampled as the correction set with other profiles by computing the absolute differences.

Results — Figure 10 shows the profile differences. In the left figure, the reduced frame sampling intervention is applied with the fixed resolution 608×608. The total number of frames are different in the two video sequences, so we use the sample size instead of the sample fraction as the x-axis for better comparison. And we only show the results when the size is less than 100 because the error bound differences only slightly change beyond this area. The limited frame access (up to 50 frames) to video A causes an incomplete and loose error bound estimation, yielding the substantial difference (orange line) compared with the target profile. Fortunately, when enough frames (500 frames) in video B are accessed, the error bound

differences between this similar video and video A (red line) are close to zero. In the right figure, the resolution is varied with the fixed sample size 500. Similarly, the error bound differences between video A and B are very small and always within 5%.

6 RELATED WORK

Some research areas are relevant to our system, and we discuss them in depth below:

Video Data Management — As video data has comprised a major part of all information about the world, a variety of database management systems for video analytics are emerging. Most projects have focused on optimizing video queries [4, 9, 29, 32–35, 43, 52, 68], while some have focused on video compression and storage [17, 46]. These works may partially solve system problems, such as power constraints and storage limitations, but other requirements like privacy cannot be satisfied. Some of them have also considered the quality of query results [17, 35]. However, none of the existing video data management systems have generated degradation-accuracy tradeoff profiles.

Database with Privacy and System Requirements — A large amount of literature has explored the privacy-preserving problem in data publishing, summarized by [21]. Numerous models are proposed for guaranteeing k-anonymity [39, 59], ϵ-differential privacy [14, 18], etc. Apart from the non-interactive data publishing, studies have integrated the privacy protection into the query processing engine [10, 61]. For the system requirements, such as bandwidth management, energy saving, and storage capacity limitation, popular solutions include database compression [24, 57] and adjusting hardware and software configurations [62]. However, some techniques are only suitable for specific database types like relational databases, and the relationship between analytical accuracy and these methods has not been examined.

Aggregate Query Approximation — Approximate query processing (AQP), aiming to approximate aggregate query answers in online analytical processing, has been researched for decades [13]. AQP methods comprise two categories: online aggregation and offline synopsis generation [41]. Works about online aggregation [2, 12, 30, 51, 67] select samples online to estimate the answers of aggregate functions, such as `COUNT`, `SUM`, and `AVG`. The estimation performance can be further improved by recent developments in concentration inequality [8, 36, 47, 58] which have been used in many areas, such as solving the multi-armed bandit problem [42]. Methods about offline synopsis generation [56] can be applied to more aggregate functions but require prior knowledge. Other than the above distributive and algebraic aggregate functions, holistic aggregate function approximation, such as `MEDIAN` and `RANK`, is widely studied in the areas of data streaming and sensor networks in order to save sorting time and storage space. These estimation algorithms mainly rely on summary statistics [22, 25, 26, 60]; only some of the works are based on sampling [40, 45].

7 CONCLUSION AND FUTURE WORK

As video data of public locations is increasingly collected and analyzed, how to balance the analytical query accuracy and other competing goals becomes a problem. In summary, we present a novel video degradation-accuracy profiling model which is able to produce accuracy/degradation tradeoff curves so that administrators can determine a set of appropriate destructive interventions. In addition, we implemented our prototype system, Smokescreen, and verified its good performance on real-world video datasets.

For the analytical accuracy estimation problem in this work, we modeled random and non-random interventions as shown in Table 1. This modeling is not restricted to videos — if other scenarios for other data types can be modeled as the same technical problems in Table 1, our algorithms are also applicable. If videos' unique properties are exploited — for example, a sequence of frames are so similar that part of frames can be skipped from processing — the quality of the estimated error bound can be further improved.

In this work, we focus on the video aggregate queries with frame-level detection models. Even though they can cover a variety of cases, there exists another type of model, which processes frame sequences, e.g., a RNN model for action recognition and detection. Because reducing the sampling rate likely decreases the accuracy of the model's outputs, simply considering it as a random intervention seems inappropriate. In this situation, both of our algorithms for random and non-random interventions cannot be directly applied. In addition, besides the four commonly used aggregate functions, `AVG`, `SUM`, `MAX`, and `COUNT` in our work, more aggregate types can be explored, such as `VAR`. We believe examining the degradation-accuracy profiling problem for more neural network model types and aggregate functions, as well as exploiting videos' unique properties, are promising future projects.

ACKNOWLEDGMENTS

We would like to thank our anonymous reviewers and Rui Liu from University of Michigan–Ann Arbor for their valuable comments and feedback. This material is based upon work supported by the Federal Highway Administration under contract number 693JJ319000009. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Federal Highway Administration.

REFERENCES

[1] Waleed Abdulla. 2017. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. https://github.com/matterport/Mask_RCNN.
[2] Sameer Agarwal, Henry Milner, Ariel Kleiner, Ameet Talwalkar, Michael Jordan, Samuel Madden, Barzan Mozafari, and Ion Stoica. 2014. Knowing when you're wrong: building fast and reliable approximate query processing systems. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 481–492.
[3] 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 *Proceedings of the 8th ACM European Conference on Computer Systems*. 29–42.
[4] 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.
[5] Ronnie T Apteker, James A Fisher, Valentin S Kisimov, and Hanoch Neishlos. 1995. Video acceptability and frame rate. *IEEE multimedia* 2, 3 (1995), 32–40.
[6] Azeem Aqil, Ahmed OF Atya, Srikanth V Krishnamurthy, and George Papageorgiou. 2015. Streaming lower quality video over LTE: How much energy can you save?. In *2015 IEEE 23rd International Conference on Network Protocols (ICNP)*. IEEE, 156–167.
[7] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2007. Tuning bandit algorithms in stochastic environments. In *International conference on algorithmic learning theory*. Springer, 150–165.

[8] Rémi Bardenet, Odalric-Ambrym Maillard, et al. 2015. Concentration inequalities for sampling without replacement. *Bernoulli* 21, 3 (2015), 1361–1385.

[9] Favyen Bastani, Songtao He, Arjun Balasingam, Karthik Gopalakrishnan, Mohammad Alizadeh, Hari Balakrishnan, Michael Cafarella, Tim Kraska, and Sam Madden. 2020. MIRIS: Fast Object Track Queries in Video. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1907–1921.

[10] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. 2005. Practical privacy: the SuLQ framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 128–138.

[11] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934* (2020).

[12] Badrish Chandramouli, Jonathan Goldstein, and Abdul Quamar. 2013. Scalable progressive analytics on big data in the cloud. *Proceedings of the VLDB Endowment* 6, 14 (2013), 1726–1737.

[13] Surajit Chaudhuri, Bolin Ding, and Srikanth Kandula. 2017. Approximate query processing: No silver bullet. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 511–519.

[14] Rui Chen, Noman Mohammed, Benjamin CM Fung, Bipin C Desai, and Li Xiong. 2011. Publishing set-valued data via differential privacy. *Proceedings of the VLDB Endowment* 4, 11 (2011), 1087–1098.

[15] Maggie Xiaoyan Cheng, Lu Ruan, and Weili Wu. 2005. Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, Vol. 4. IEEE, 2638–2645.

[16] Ji Dai, Jonathan Wu, Behrouz Saghafi, Janusz Konrad, and Prakash Ishwar. 2015. Towards privacy-preserving activity recognition using extremely low temporal and spatial resolution cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 68–76.

[17] Maureen Daum, Brandon Haynes, Dong He, Amrita Mazumdar, Magdalena Balazinska, and Alvin Cheung. 2020. TASM: A Tile-Based Storage Manager for Video Analytics. *arXiv preprint arXiv:2006.02958* (2020).

[18] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*. Springer, 1–19.

[19] Willliam Feller. 2008. *An introduction to probability theory and its applications, vol 2.* John Wiley & Sons.

[20] Douglas A Fidaleo, Hoang-Anh Nguyen, and Mohan Trivedi. 2004. The networked sensor tapestry (NeST) a privacy enhanced software architecture for interactive analysis of data in video-sensor networks. In *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*. 46–53.

[21] Benjamin CM Fung, Ke Wang, Rui Chen, and Philip S Yu. 2010. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (Csur)* 42, 4 (2010), 1–53.

[22] Edward Gan, Jialin Ding, Kai Sheng Tai, Vatsal Sharan, and Peter Bailis. 2018. Moment-based quantile sketches for efficient high cardinality aggregation queries. *arXiv preprint arXiv:1803.01969* (2018).

[23] Wilson S Geisler and Jeffrey S Perry. 1998. Real-time foveated multiresolution system for low-bandwidth video communication. In *Human vision and electronic imaging III*, Vol. 3299. International Society for Optics and Photonics, 294–305.

[24] Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1998. Compressing relations and indexes. In *Proceedings 14th International Conference on Data Engineering*. IEEE, 370–379.

[25] Michael Greenwald and Sanjeev Khanna. 2001. Space-efficient online computation of quantile summaries. *ACM SIGMOD Record* 30, 2 (2001), 58–66.

[26] Michael B Greenwald and Sanjeev Khanna. 2004. Power-conserving computation of order-statistics over sensor networks. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 275–285.

[27] Ajit D Gupte, Bharadwaj Amrutur, Mahesh M Mehendale, Ajit V Rao, and Madhukar Budagavi. 2011. Memory bandwidth and power reduction using lossy reference frame compression in video encoding. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 2 (2011), 225–230.

[28] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[29] Wenjia He, Michael R Anderson, Maxwell Strome, and Michael Cafarella. 2020. A Method for Optimizing Opaque Filter Queries. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1257–1272.

[30] Joseph M Hellerstein, Peter J Haas, and Helen J Wang. 1997. Online aggregation. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. 171–182.

[31] Wassily Hoeffding. 1994. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*. Springer, 409–426.

[32] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B Gibbons, and Onur Mutlu. 2018. Focus: Querying large video datasets with low latency and low cost. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*. 269–286.

[33] 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*. 253–266.

[34] Daniel Kang, Peter Bailis, and Matei Zaharia. 2018. BlazeIt: optimizing declarative aggregation and limit queries for neural network-based video analytics. *arXiv preprint arXiv:1805.01046* (2018).

[35] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. Noscope: optimizing neural network queries over video at scale. *arXiv preprint arXiv:1703.02529* (2017).

[36] Vladimir Koltchinskii and Karim Lounici. 2017. Concentration inequalities and moment bounds for sample covariance operators. *Bernoulli* 23, 1 (2017), 110–133.

[37] Michał Koziarski and Bogusław Cyganek. 2018. Impact of low resolution on image recognition with deep neural networks: An experimental study. *International Journal of Applied Mathematics and Computer Science* 28, 4 (2018), 735–744.

[38] Tony CT Kuo and Arbee LP Chen. 2000. Content-based query processing for video databases. *IEEE Transactions on Multimedia* 2, 1 (2000), 1–13.

[39] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. 2006. Mondrian multidimensional k-anonymity. In *22nd International conference on data engineering (ICDE'06)*. IEEE, 25–25.

[40] Ji Li, Siyao Cheng, Zhipeng Cai, Jiguo Yu, Chaokun Wang, and Yingshu Li. 2017. Approximate holistic aggregation in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 13, 2 (2017), 1–24.

[41] 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.

[42] Rui Liu, Tianyi Wu, and Barzan Mozafari. 2019. A bandit approach to maximum inner product search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4376–4383.

[43] Yao Lu, Aakanksha Chowdhery, and Srikanth Kandula. 2016. Optasia: A relational platform for efficient large-scale video analytics. In *Proceedings of the Seventh ACM Symposium on Cloud Computing*. 57–70.

[44] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G Lindsay. 1998. Approximate medians and other quantiles in one pass and with limited memory. *ACM SIGMOD Record* 27, 2 (1998), 426–435.

[45] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G Lindsay. 1999. Random sampling techniques for space efficient online computation of order statistics of large datasets. *ACM SIGMOD Record* 28, 2 (1999), 251–262.

[46] Amrita Mazumdar, Brandon Haynes, Magdalena Balazinska, Luis Ceze, Alvin Cheung, and Mark Oskin. 2019. Vignette: Perceptual Compression for Video Storage and Processing Systems. *arXiv preprint arXiv:1902.01372* (2019).

[47] Florence Merlevède, Magda Peligrad, and Emmanuel Rio. 2009. Bernstein inequality and moderate deviations under strong mixing conditions. In *High dimensional probability V: the Luminy volume*. Institute of Mathematical Statistics, 273–292.

[48] Volodymyr Mnih, Csaba Szepesvári, and Jean-Yves Audibert. 2008. Empirical bernstein stopping. In *Proceedings of the 25th international conference on Machine learning*. 672–679.

[49] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, et al. 2021. A variegated look at 5G in the wild: performance, power, and QoE implications. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 610–625.

[50] WL Nicholson et al. 1956. On the normal approximation to the hypergeometric distribution. *The annals of mathematical statistics* 27, 2 (1956), 471–483.

[51] Frank Olken. 1993. *Random sampling from databases*. Ph.D. Dissertation. University of California, Berkeley.

[52] Alex Poms, Will Crichton, Pat Hanrahan, and Kayvon Fatahalian. 2018. Scanner: Efficient video analysis at scale. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

[53] Xin Qi, Qing Yang, David T Nguyen, Gang Zhou, and Ge Peng. 2015. LBVC: towards low-bandwidth video chat on smartphones. In *Proceedings of the 6th ACM Multimedia Systems Conference*. 1–12.

[54] Qasim Mahmood Rajpoot and Christian Damsgaard Jensen. 2015. Video surveillance: Privacy issues and legal compliance. In *Promoting Social Change and Democracy Through Information Technology*. IGI global, 69–92.

[55] Joseph Redmon. 2013. Darknet: Open source neural networks in c.

[56] Mirek Riedewald, Divyakant Agrawal, et al. 2000. pCube: Update-efficient online aggregation with progressive feedback and error bounds. In *Proceedings. 12th International Conference on Scientific and Statistica Database Management*. IEEE, 95–108.

[57] Mark A Roth and Scott J Van Horn. 1993. Database compression. *ACM Sigmod Record* 22, 3 (1993), 31–39.

[58] Mark Rudelson and Roman Vershynin. 2013. Hanson-wright inequality and sub-gaussian concentration. *Electronic Communications in Probability* 18 (2013), 1–9.

[59] Pierangela Samarati and Latanya Sweeney. 1998. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. (1998).

[60] Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. 2004. Medians and beyond: new aggregation techniques for sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor*

systems. 239–249.

[61] Bhavani Thuraisingham. 2005. Privacy constraint processing in a privacy-enhanced database management system. *Data & Knowledge Engineering* 55, 2 (2005), 159–188.

[62] Dimitris Tsirogiannis, Stavros Harizopoulos, and Mehul A Shah. 2010. Analyzing the energy efficiency of a database server. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. 231–242.

[63] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10 (2017), 3152676.

[64] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. 2020. UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking. *Computer Vision and Image Understanding* (2020).

[65] Takayuki Yamada, Seiichi Gohshi, and Isao Echizen. 2012. Use of invisible noise signals to prevent privacy invasion through face recognition from camera images. In *Proceedings of the 20th ACM international conference on Multimedia*. 1315–1316.

[66] Jiadi Yu, Haofu Han, Hongzi Zhu, Yingying Chen, Jie Yang, Yanmin Zhu, Guangtao Xue, and Minglu Li. 2014. Sensing human-screen interaction for energy-efficient frame rate adaptation on smartphones. *IEEE Transactions on Mobile Computing* 14, 8 (2014), 1698–1711.

[67] Kai Zeng, Shi Gao, Barzan Mozafari, and Carlo Zaniolo. 2014. The analytical bootstrap: a new method for fast error estimation in approximate query processing. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 277–288.

[68] 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 *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*. 377–392.

[69] 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.