Privid: Practical, Privacy-Preserving Queries on Public Video

Frank Cangialosi*, Neil Agarwal†, Venkat Arun*, Junchen Jiang¶, Srinivas Narayana§,
Anand Sarwate§, Ravi Netravali†
*MIT CSAIL, †UCLA, ¶University of Chicago, §Rutgers University

ABSTRACT

Camera deployments are becoming increasingly pervasive in public settings. In parallel, advances in computer vision have enabled organizations to leverage this rich visual data to accurately identify and track individuals in real time. While this opens up a large space of interesting and useful applications, it also opens a large potential to intrude on individuals' privacy. Thus, camera owners face a tricky tradeoff: they can either release video to maximize analytics utility but potentially sacrifice privacy, or they can preserve privacy by disallowing analytics. Recent tools have attempted to ease this tradeoff by aiming to remove only private information from video, but these approaches fail to provide any formal guarantees on privacy.

This paper presents Privid, the first system to our knowledge that extends formal differential privacy techniques to the video analytics domain. Unlike prior applications of differential privacy in traditional database systems, video data and analytics queries inherently lack the structure required to hide private information amongst query results in a practical manner. To overcome this, Privid leverages multiple observations about typical public video data (e.g., the persistence of individuals across frames) in order to transform the problem into one that is amenable to traditional differential privacy. In doing so, Privid provides both a formal guarantee of indistinguishability for individuals in a video and a precise bound on how the query will be impacted as a result. Privid supports a large variety of video analytics queries and we show that in many practical cases it is possible to preserve privacy with minimal impact on accuracy.

1. INTRODUCTION

The ubiquitous deployments of high-resolution cameras in public settings [1, 3–5, 7], coupled with steady advances in computer vision techniques [15, 32, 36, 37, 50], have given rise to an abundance of video analytics applications [2, 8]. These applications typically employ analytics pipelines that run computer vision algorithms (often involving deep neural networks or DNNs) on a video's frames in order to detect, recognize, or track various objects; this information can, in turn, be analyzed to answer diverse queries. For example, cities can locate fallen trees after a storm or identify crossings with near-collision incidents in order to assess road safety [11]. Businesses can obtain detailed maps of pedestrian demographics to decide when and where to open new stores [10].

Unfortunately, despite the clear safety and convenience benefits, accurate analytics on public videos[1] enable *privacy* intrusions at an unprecedented level. For instance, by using state-of-the-art facial recognition systems [58], one can automatically identify and track an individual's movements in real-time across a network of cameras. To thwart such tracking, large cities such as San Francisco and Oakland, have outright banned [6, 9] the use of facial recognition on public videos, even for law enforcement purposes. Clearly, there exists an *inherent tension between utility and privacy*, and achieving perfect privacy or utility at the total expense of the other is not ideal. Instead, the desirable outcome is to protect the privacy of individuals in public settings (by hiding their presence in public videos), while enabling a broad range of queries on those videos.

Unsurprisingly, much work has been devoted to this challenge [44]. Across these solutions, the predominant strategy is shared: they attempt to automatically identify and remove (e.g., via black boxes [57], blurring [12], or inpainting the background [16]) all private information from the video before sharing it with the analysts, who can then run arbitrary queries on it. Though promising, such *denaturing* approaches share two fundamental weaknesses. First, object detection algorithms are not 100% accurate and any undetected content cannot be hidden, leaving it in plain sight (e.g., Fig. 2). Second, denaturing precludes certain queries which *safely* operate on private information, i.e., computing aggregate statistics, without revealing the presence of any one individual. For instance, counting the number of cars that use a road daily (versus occasionally) requires a query to read (private) license plates; however, the aggregate counts themselves are safe to release. To make matters worse, it is hard to determine how much private information is missed during denaturing (privacy leakage) or how much adverse impact denaturing causes on query accuracy (utility loss). Thus, it is inherently difficult to balance privacy and utility in any controllable manner.

To overcome these challenges, we present Privid, a system that takes a fundamentally different approach motivated by the privacy preservation techniques used in traditional databases. Rather than the error-prone process of hiding all private information *before* running the query, Privid runs the query over the unmodified input video,[2] but *perturbs the query output* (by adding a controllable amount of noise) before releasing it to the analyst (Fig. 1). Importantly, Privid's output perturbation satisfies Differential Privacy [20] (DP), which offers quantifiable tradeoffs between utility and privacy, regardless of externally available side-channel information. Further, Privid provides a flexible query interface, whereby queries can perform any operations (e.g., filters, joins), with the only restriction being that they must end in an aggregation (i.e., output a single number); we show in §6 that this is amenable to a wide range of video queries.

Applying DP in a *practical* manner to video analytics settings is challenging—this is the primary focus of this paper, and to date, has been unexplored. DP crucially involves quantifying how *sensitive* the query output is to the presence of any one individual in the video—this value dictates how much noise must be added to ensure privacy. Techniques to determine sensitivity are well understood in database settings which run transparent

[1] In this paper, public videos refer to videos that observe public settings, but are not publicly accessible.

[2] We focus on fixed-location static cameras, but note that the presented concepts can be generalized to fixed-location pan-tilt-zoom (PTZ) cameras.

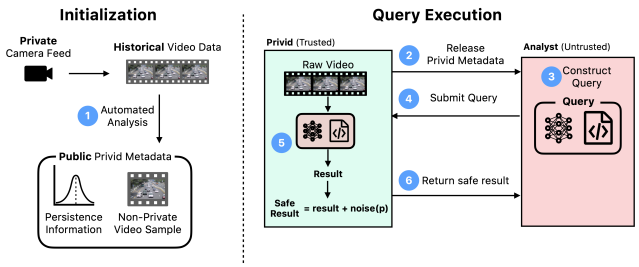


Figure 1: *Overview of query execution with* Privid. Once-per-camera initialization*: (1) Video owner analyzes historical data from the camera of interest (§5.1) to collect persistence metadata (and an optional sample frame to help with query construction), which does not leak privacy.* Execution process per query*: (2) They release this metadata to the analyst, who (3) chooses their query parameters to best balance noise and utility for their specific query. (4) The analyst submits their query and parameters to* Privid*, which (5) executes it over the raw video, (6) adds noise based on the persistence metadata, and returns the (safe) aggregate noisy result.*

queries on data that is structured to easily delineate private information (e.g., one individual per row, or an ID column). However, video analytics lack these properties. In particular, private information is not directly delineated in raw video data, and instead must be extracted based on semantic meaning derived from running imperfect vision algorithms (or human analysis) on groups of pixels or frames. Further, given the diverse range of queries, and the difficulty in fully introspecting their operations (especially their DNN components), queries must be viewed as completely *untrusted black boxes*; in essence, the query opaquely defines the data extraction technique, data schema, and the operation over that data. Without understanding precisely how a query computes over video data, the only way to satisfy DP is to assume the worst case (i.e., largest) sensitivity, and consequently, noise, which would eliminate any utility. Privid tackles these issues by making two key contributions in the way that it applies DP to video analytics.

First, Privid identifies and exploits a key property that videos of public places exhibit: most individuals *persist* in view of a camera for a *shorter* duration than the timescales over which many queries operate (we empirically validate this in §5.1). As a result, query outputs are often not very *sensitive* to the presence of any one individual, e.g., a query may count people once per hour, but the average person persists for only 30 seconds. Based on this observation, Privid defines the sensitivity of a query in terms of a persistence threshold, whereby it temporally splits query execution and adds noise proportional to that threshold, bounding the amount that an individual can influence the output; all individuals with persistence smaller than the chosen threshold are protected with DP. Fig. 1 provides an overview of this process. Of course, such persistence-derived perturbation relies on object detection and tracking (to determine persistence values) whose imperfections plague denaturing approaches. Two factors help us here. First, the maximum persistence of all private individuals (a single value) is easier to obtain (and less susceptible to errors) than the exact locations of all individuals per frame, since the persistence information essentially coarsens the latter along both space and time. Second, unlike image denaturing that may cause arbitrary impact on DNN output, Privid's persistence threshold value is tunable, enabling the video owner to formally trade off between privacy and utility.

Second, Privid identifies and leverages two *video-specific* opportunities to further reduce the amount of noise that must be added to ensure the same level of privacy. First, while most people stay in view of the camera only briefly, a select few may persist for longer. The amount of noise needed to protect these people would make many queries impractical (see §5.1). We empirically observe that such lingering individuals tend to concentrate in small areas (e.g., pedestrians waiting at a crossroad to cross the street or people sitting on a bench). Privid leverages this uneven distribution by automatically adding static masks over these areas if a query does not depend on that information. Second, we observe that since public cameras tend to cover a large area, at any point in time, an individual typically only occupies a relatively small portion of a video frame. Building off of this, (when possible) Privid spatially splits queries to operate over subsets of each frame independently. This limits the amount of a query output that an individual can impact, thereby enabling Privid to further bound the amount of noise required to achieve privacy.

We evaluated Privid using a variety of queries and three representative public video streams spanning 12 hours each. Overall, we find that Privid's video-specific insights allow for a significant reduction (597× - 3093×) in the amount of noise necessary to satisfy DP. This, in turn, results in Privid providing high accuracy while preserving the privacy of all individuals in our video streams. We also present results highlighting the flexibility of Privid's query interface, and quantify how video and query properties affect the privacy versus utility tradeoff with Privid. We will open-source Privid post publication.

Note. In this paper, we trust the video owner to execute Privid correctly, while allowing for maliciously-crafted queries; we justify and elaborate upon this assumption in the threat model (§2.2). Prior work has shown how to use cryptography and/or trusted hardware to compute on video while ensuring that *nobody* has unrestricted access to the video [23, 26, 47, 52–54]. This complementary line of work reduces trust on the video owner, but trusts the query itself to not reveal undesirable information.

2. MOTIVATION

2.1 Privacy in Public Video Analytics

The last decade has seen dramatic advances in computer vision that have made it practical to process large amounts of video data (live or offline) automatically. An *analyst* seeks to answer a *query* about a segment of video, such as "how many cars cross this highway each hour?" or "what fraction of cars use this highway every week?" In this paper, we focus on typical queries in *public* video analytics.[3]

There are growing privacy concerns over highly accurate analytics of videos collected in public domains. The subjects in view (individuals) do not explicitly consent to this monitoring and may be unaware of it in the first place. For example, the NYTimes was able to track and identify thousands of individuals walking through Bryant Park by simply running the publicly-available video stream through Amazon's commercial facial recognition service [19]. As camera deployments increase [1–5, 7] and cover more area (city streets, campuses, parks, etc), they are able to capture an increasing portion of our daily lives, activities, and behaviors. At the logical extreme, they could be used to reconstruct an entire timeline of all of the places we visited and with whom.

In simple terms, privacy in public video analytics is the ability for an individual to remain anonymous in public places.

[3] We consider video analytics as distinct from *video security*, which requires pinpoint answers (e.g., the exact location of an identity) and is thus directly at odds with privacy.

In that sense, the space of privacy in public video analytics entails four logically distinct entities:

1. **Individual**, whose behavior and activity are recorded by the camera and thus whose privacy is at stake.
2. **Video Owner**, who controls the camera and thus the video data it records.
3. **Analyst**, who wishes to run queries over the video.
4. **Compute Provider**, who executes the analyst's query.

If the video owner wishes to intrude on the privacy of citizens, there is little that can be done other than legislative measures that prevent the recording in the first place. Thus, we focus on scenarios where the video owner has an interest in (and is willing to) protect the privacy of citizens.

In most cases, the "video owner" is logically an organization, such as a transportation department or an enterprise, rather than an individual. The analyst may be an employee of the organization acting on their behalf. We assume that the analyst cannot view the videos directly and that the analyst's intentions cannot be trusted.

2.2 Threat Model

In Privid, the video owner and compute provider are trusted. The adversary can arbitrarily corrupt an any number of analysts. Privid prevents the analyst from answering the following question with more confidence than a configurable threshold (we will give a rigorous definition in §4.2) about video segments collected from a public camera:

"Was individual x visible to this camera at any point during time window W?"

for *any* x and a system-wide parameter W (e.g. a given day).

We make minimal assumptions on the query from an analyst. For instance, a malicious analyst who wants to reveal individual x could ask to count the number of traffic lights, but use an object detector that has been trained to classify individual x as "traffic light". Even when the analyst is truthful about the intent, a query can still reveal an individual's presence by not just their face, but also by their clothes, a distinctive bicycle or a parked car, which can be strongly associated to individuals' presence. Moreover, one could also ask the following two queries (which are safe when viewed individually) and take the difference: (1) Count total number of people and (2) Count total number of people excluding individual x. We protect against many (but not all) ways by which an adversary can detect an individual.

Therefore, Privid should prevent privacy leakage from queries that *indirectly* infer the presence of an individual with high confidence. While there have been solutions that target specific query formats, we believe Privid's threat model has broader application. In public video analytics, an analyst may be a third-party entity who wishes the video owner to run their query as a blackbox for proprietary reasons (e.g., they use a neural network that could reveal sensitive training data) or their expressed intention may not be trusted.

The only assumption Privid makes is that the amount of time x is *visible* to a camera is less than some configurable threshold. We justify this assumption in §4 and explain how to determine such a threshold in practice to strike a more desirable balance between privacy and query result accuracy than existing solutions.

3. GOALS AND CHALLENGES

We begin this section by outlining a set of goals (§3.1) that a usable system satisfying our threat model in the previous section should meet. We then overview the approach taken by the majority of prior work on preserving privacy in videos—*denaturing*—and argue that it is fundamentally unable to satisfy these goals (§3.2). In contrast, we motivate Privid's approach based on Differential Privacy. While DP conceptually resolves the fundamental issues of denaturing, video data is not amenable to the direct application of standard approaches to achieving differential privacy (§3.3).

3.1 Design Goals

G1 (bring your own models): Analysts should be able to bring their own models and computation to run over video.

We aim to enable highly flexible computations over video data by analysts. For example, our system should allow analysts to bring their own deep-learning models to recognize objects of their choosing, without requiring expensive re-training to enable the privacy guarantees of our system. Recent work on video processing [29] has shown that there is no one-size-fits-all model that works in all situations, even when the task, such as object recognition, is the same. Variations in scenery, time of day, lighting conditions, and subject distance necessitate different models to achieve high accuracy and efficiency. A large class of privacy-leaking computations we choose to protect against are neural network models and image-processing functions [38]. There has been significant research attention to training neural network models that protect private information in the *training data,* e.g., [49]. However, to our knowledge, there is relatively little research on understanding how a given neural network leaks information about its *test data* through its outputs. Given the rate of innovation in image processing models, and the specialization required to achieve good performance (e.g., attention to time of day, lighting, *etc.*), we do not consider approaches that require specialized neural-net training to protect privacy, as it will significantly hamper the usability of our system. Given the current limitations in human abilities to "interpret" the actions of neural network models, unfortunately, allowing analysts to bring their own models will significantly obscure the intention of the analyst from the video owner, effectively preventing any (manual or automatic) vetting of queries before execution.

G2 (private-obliviousness): Eschew accurate identification of all private information in queried videos.

Naturally, in order to protect sensitive private information in a database, one has to answer the question *what information is private?* in the first place. This question is particularly challenging in the video context due to the data representation (image frames are matrices of pixels as opposed to, e.g., database rows with unique identifiers for people) and data volume (a single day of 30fps video from one camera consists of 2.6M frames). While a human may be able to manually label private information in a handful of frames, it is simply impractical to have humans label all private information in public video feeds that we wish to analyze, given the data volumes and the explosive growth of available video data over time. Great strides have been made in automatically identifying objects in images and videos; however, even state-of-the-art object detection techniques are still imperfect and frequently produce erroneous classification labels [42, 51]. In particular, accuracy typically degrades significantly when lighting conditions are poor or when objects are small in the scene, both of which are common in public videos [61]. Worse, while it is clear that while some information must be private (e.g., an individual's face), it is difficult to determine *all* objects that might be used to identify an individual. What if an adversarial analyst has access to auxiliary information that a video owner does not? For example, an individual may carry

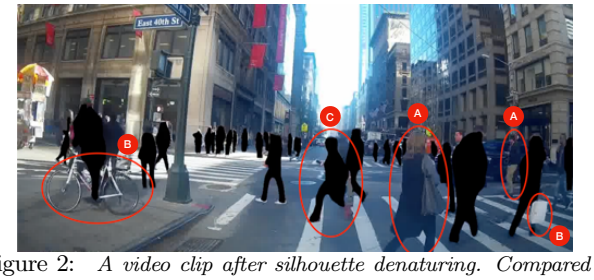


Figure 2: *A video clip after silhouette denaturing. Compared to boxes, people are now easier to count and they obstruct less of the background, but their gait is revealed which may be enough to identify them.*

a particular bag, ride a particular bike, or walk a particular pet that uniquely identifies them within a specific location (Fig. 2, B). An ideal system must not require the privacy guarantees to hinge on the video owner accurately identifying all private information in their video—either manually or automatically.

G3 (quantifiable guarantees): Provide a quantifiable trade-off between utility and privacy.

While allowing expressive analyst queries, it is fundamentally impossible to simultaneously achieve complete privacy and error-free query results, since the information queried by an analyst may itself be private. We seek the more feasible goal of achieving a *quantifiable trade-off* between the two aspects: the utility of the query (expressed in terms of the error in the query output) and the privacy guarantee (expressed in terms of the probability that an adversarial model can successfully detect a piece of private information in the video).

G4 (safe aggregations): Support safe queries in the aggregate over private entities with high accuracy.

We believe that an important use case for public video analytics is extracting *aggregate statistics* without threatening the privacy of any one individual. For example, a urban planner may wish to know the number of people crossing a street intersection in a window of time. Answering this query, and many others, relies on identifying private information (i.e., persons in the video), but necessarily aggregate the answer in a way that does not violate the privacy of those persons. An ideal system must allow answering such aggregate queries with a high accuracy.

3.2 Existing Approach: Input Denaturing

The vast majority of prior work on preserving privacy for videos (and images in general) take the approach of perturbing the input video before releasing them to analysts. We group these approaches under the common label of *input denaturing*. The basic idea is to detect all the private information in a video (e.g., human faces), obscure the aforementioned information to create a "clean" version of the video, and use the "clean" version of the video for querying by analysts. In principle, any analyst computation can be supported, meeting goal G1 (bring your own models). Prior work has explored a variety of techniques to remove the private information, but rely on one of the following general primitives:

- Object masking [57]: objects are blacked out using either pre-defined shapes or object-specific silhouettes (e.g., people in Fig. 2).
- Blurring and pixelization: the pixels representing an object are distorted to obfuscate the fine-grained properties of the object [12].
- Background inpainting [16]: the pixels representing an object are removed, then filled in with an estimate of the background (i.e., completing what the scene would have looked like if the object was not present).

Fundamentally, any denaturing mechanism cannot satisfy goal G2 (private-obliviousness). Denaturing private information requires detecting that private information. Further, even if a video owner had perfect knowledge of all the private information they choose to protect, denaturing that information from the video significantly hampers the ability of analysts to issue safe aggregate queries, failing goal G4 (safe aggregations)—denaturing prevents labeling and aggregation over objects in the image that are deemed private. As a workaround, such queries (e.g., counting the number of persons) could identify the denatured version of the objects (e.g., a black box) as a proxy; however, in our experience, this workaround makes it difficult to distinguish private objects that are next to each other in a small region of the video. The inability to predict how denaturing impacts the accuracy of query results makes it impossible to provide any guarantees on the accuracy of the query, failing goal G3 (quantifiable guarantees). In fact, denaturing does not guarantee privacy either, since the models used to label private objects may make mistakes or leave out auxiliary information (e.g., a bike, see Fig. 2) that can then be associated with an individual. While object detectors can be tuned to be more or less aggressive in obscuring information deemed to be private, such tuning (requiring a specific trade-off between false positives and false negativesmust be customized to each query, since the computation of the specific query (e.g., counting people, versus counting cars) determines how aggressively people need to be obscured from the video. We believe that such an approach would be impractical, given the untrusted nature of the queries (G1 (bring your own models)).

3.3 Privid's Approach

We apply the framework of *differential privacy (DP)* which provides the benefit of a quantifiable tradeoff (G3 (quantifiable guarantees)) between utility (expressed as an error in query results) and privacy (expressed as a probability of detection of private information). Our system, Privid, asks the analyst to submit their query to the compute provider, which uses DP to answer the query in a privacy-preserving manner (we will formalize our algorithm in §4.1).

Informally, DP ensures that the query output isn't noticeably changed if we add or remove information about any one individual in the input. This ensures that individuals cannot be identified within the dataset (here, video), and hence an individual suffers no adverse consequences from being included in the video. There are many methods to satisfy this requirement. One method is to quantify the maximum amount by which the query's output could change based on the presence of any one individual. This amount is called the *sensitivity* of the query. By adding noise to the query's result that is *proportional to the query's sensitivity*, the presence of any one individual in the data can be obscured. Further, the constant of proportionality can be used as a way to walk the line between introducing too much noise (degrading privacy) and leaking information about the individual (reducing privacy). In §3.4, we provide a formal explanation of DP and the basic mechanism to add noise.

Intuitively, it is easy to see that DP-based approaches satisfy goals G3 (quantifiable guarantees) and G4 (safe aggregations). The parameters used in adding noise determine a quantifiable trade-off that a video owner is willing to make between privacy and utility. Since queries that perform aggregation typically have low sensitivity (§4.1) relative to their outputs, aggregated results can be safely released with a high signal-to-noise ratio.

However, it is not straightforward to provide DP for queries that are unrestricted and untrusted (considering G1 (bring your own models)). Sensitivity is a property of an algorithm – how

must the video owner determine the sensitivity of an analyst query without knowing what (untrusted) computation is run by the analyst? At first glance, DP seems unable to satisfy G2 (private-obliviousness) as well. To determine how much any one piece of private information might impact a query's output, it appears to be necessary to tag which information is private in the first place. Unfortunately, video is unlike "regular" databases where one can cleanly associate a private entity with a set of database rows. At best, each piece of private information (e.g., a human face) can be associated with frames in which it appears through an error-prone model (§3.1). Hence, the combination of an untrusted (black-box) computation with the unstructured input data format (i.e., video) makes it challenging to compute sensitivity, and hence, apply DP, to public video analytics.

We make two key observations that enable us to successfully apply DP to video analytics.

1. Sensitivity is a threshold for privacy. We re-interpret sensitivity not as an algorithmic property about the maximum difference between outputs over data sets, but as a *threshold* which defines which private entities are protected by DP. If the presence of a private entity in a dataset makes the query's output differ by less than the threshold, then that entity is protected in a differentially-private sense.

2. Private entities persist for short periods in public videos. From our experience processing publicly available video feeds (§5.1), we find that private entities, such as people, tend to stay in view of the cameras for a short fraction of the time over which we run analyst queries.

In §4.1, we show how we leverage these two observations to enable a practical differentially-private scheme that satisfies all our design goals (§3.1). The guarantee that our scheme provides is that any private entity persisting in a window of video for less than ρ fraction of the time is protected in a differentially-private sense. Here, ρ, which we call the *protected persistence threshold,* is a parameter that a video owner sets *once for each camera.* The choice of ρ guides the trade-off between privacy and utility. Importantly, the parameter does not change according to the query. Setting a suitable ρ may be accomplished manually or automatically. For example, for a camera overlooking a traffic light that changes every 60 seconds, the video owner can be confident that a threshold slightly above that will protect all cars in the scene. It is also possible to set ρ automatically by analyzing historical video data from the camera, and using a summary statistic (e.g., 99^{th} percentile) on how long an individual lingers in front of the camera to set the protected persistence threshold. We describe such an approach in §5.1. While such automatic analyses do require identifying private information in the video, we believe it is strictly easier to determine a coarse-grained summary statistic accurately than to label private information correctly on each frame—such as in the denaturing-based approaches.

3.4 A Primer on ϵ-DP

ϵ-differential privacy [21, 22] formalizes the notion that the adversary should not be able to discern significant information about an individual from query outputs. To do so, it ensures that query outputs look "similar" for any pair of datasets D and D' that differ only on the data for one individual. If the adversary cannot distinguish D and D', it cannot determine any information about the changed individual. Formally, for any pair of D and D' that differ only on the data of one individual, a query algorithm Alg is called ϵ-differentially private if:

$$\mathbb{P}(\mathsf{Alg}(D)\in\mathcal{A}) \leq e^{\epsilon}\mathbb{P}(\mathsf{Alg}(D')\in\mathcal{A}) \quad \forall D\sim D', \forall \mathcal{A}\in \mathrm{im}\,\mathsf{Alg} \tag{3.1}$$

where the image of Alg, $\mathrm{im}\,\mathsf{Alg}$, is a measurable set and the probability is over the randomness in Alg. Smaller values of ϵ mean that Alg leaks less information about an individual. Setting $\epsilon=0$ would make the outputs from D and D' completely indistinguishable. Useful queries usually have $\epsilon>0$.

DP is often achieved by taking a non-private algorithm and adding noise to its result. The noise is proportional to the algorithm's *sensitivity* Δ; how much its output can change when one individual's data is changed. For instance, the sum of heights of a set of people can change at most by the maximum possible height of any one person (e.g. 10 ft). As long as we add enough noise to mask this difference, the result will be hard to distinguish. Laplace noise is often used for this purpose. The Laplace distribution has density $p(\eta)=\frac{1}{2\alpha}\exp(-\eta/\alpha)$ and for a scalar output with sensitivity Δ, setting $\alpha=\frac{\Delta}{\epsilon}$ guarantees that the noise can provide ϵ-DP. Larger ϵ (less privacy) makes the density have lower variance and larger Δ makes the density have higher variance (to hide larger differences).

Often, users of a DP query system need to make multiple queries on the same dataset. Answering n queries with parameters $\epsilon_1,...,\epsilon_n$ leaks information equal to answering one query with parameter $\sum_{i=1}^{n}\epsilon_i$. Hence a query system can set a privacy budget ϵ_{tot} that bounds the *total* amount of information leaked by all the queries made on the dataset. Once this budget is exhausted, the system may disallow any more queries on the same dataset. A larger ϵ_{tot} allows more queries or queries with smaller noise (better utility). Smaller ϵ_{tot} values provide better privacy. Hence, the ϵ parameter of DP provides the capability of quantifying the tradeoff between utility and privacy. Privid benefits from this capability.

4. PRIVID'S DESIGN

4.1 Differentially-Private Video Analytics

We develop a design for an analytics system that achieves the goals above using differential privacy. In this section, we make a few assumptions about the analytics queries for ease of illustration of our basic techniques. We will relax each of these assumptions later.

- Assumption A1 (stateless queries over frames): We assume that an untrusted, analyst-provided query runs over each frame of the video without sharing any state across frames. The computation may take the form of an object-recognition neural network or a computer-vision algorithm [38] or any post-processing of the results thereof. In §4.2, we will show how a model can execute and maintain state over short multi-frame video snippets.
- Assumption A2 (single scalar-valued query): We assume there is exactly one query operating over the video. Further, the result of that query is a single numerical value emitted for each frame. In §4.2 we show how we handle multiple queries (and in §5.3 we show how a query can emit vector-valued or non-numerical results).

We use the framework of differential privacy since it provides quantifiable trade-offs between accuracy and utility (goal G3 (quantifiable guarantees)). However, the key challenge is in applying the notion of private information held in "database rows" to private entities appearing in video frames.

Traditional differentially-private algorithms are modifications made to an algorithm to provide quantifiable guarantees

on the privacy and output error (accuracy) of the original algorithm [21]. Key to these modifications is the question:

What is the maximum change (Δ) in the output of an algorithm when a private entity x is removed from the input data set?

This quantity, termed the *sensitivity* of the algorithm, is the maximum amount by which the output of the algorithm may change between "neighboring databases", i.e., databases which only differ in the existence of a single private entity. The sensitivity determines the amount of noise we need to add to the output to obscure the presence of any private entity in the database. Specifically, to provide ϵ-DP for an algorithm with sensitivity Δ, we do:

$$R = \text{query(database)}$$
$$\text{noise } \eta \propto \text{Laplace}(\Delta/\epsilon)$$
$$R_{priv} = R + \eta$$

We can then release R_{priv}, which satisfies ϵ-DP. Unfortunately, when an analyst brings their own NN model to run over the video (goal G1 (bring your own models)), the sensitivity (Δ) is unbounded. For example, the analyst's computation may return a very large value when some private information is recognized, and 0 otherwise.

We resolve the question of how we add differentially-private noise through the following insights that restrict what queries can output, without significantly limiting the expressiveness of the queries that can be executed over video in practice.

Bound per-frame results. We restrict analyst queries to only output values in a finite range fixed by the analyst and enforced by the video owner, with lower bound l and upper bound u. The sensitivity Δ is then $u-l$, since a malicious analyst query can output, say, u upon identifying private information, and l otherwise, maximizing the difference between the outputs when the query detects sensitive private information and when it doesn't. Then, the results from each frame can be made ϵ-DP as follows:

$$R_{frame} = \text{query(frame)}$$
$$\text{noise } \eta_{frame} \propto \text{Laplace}\left(\frac{u-l}{\epsilon}\right)$$
$$R_{priv} = R_{frame} + \eta$$

Many analyst queries we investigate (§6) naturally have a range associated with the results that they emit. Unfortunately, however, the noise added above would significantly impact the accuracy of the query result. The noise is proportional to the entire output range, since the signal and the noise are of the same order of magnitude. Hence, a given observed noisy value R_{priv} could have resulted from perturbing *any* possible true result R_{frame} with high probability. This renders the result R_{priv} private but completely useless to the analyst.

Trusted aggregations over untrusted results. To provide a usable query result while preserving privacy, we restrict our system to only output results *aggregated over a fixed number of video frames* which we call *chunks*, rather than output a result for each frame of video. The analyst fixes the size of the chunk (e.g., 10 seconds of video), and also chooses from a menu of *safe* aggregation functions (implemented by Privid) [4]to apply over the per-frame results within each chunk.

[4]This work focuses on supporting functions that weight each input to the aggregation evenly, such as sum, mean, and variance. We discuss how other functions with known DP versions could be imported in Appendix E.

Notation	Meaning
x	A private entity, e.g., a person
$\|C\|$	The size of a chunk in frames
$\|C_x\|$	The number of frames x appears in a chunk
$\|W\|$	The size of a window in chunks
$\|W_x\|$	The number of chunks x appears in a window
η	Differentially-private noise
R	A query result
u	Upper bound for query results
l	Lower bound for query results
ϵ	Privacy parameter in ϵ-DP
R_{priv}	Result released to the analyst
$fr(x)$	Fraction of window x appears in (i.e., persistence)
ρ	Maximum persistence that is protected

Figure 3: *Table of Notations.*

- Assumption A3 (confined execution): We assume that the query executing over each frame is *confined* to only operate on its own frame, without any communication across query instances operating across different frames. Data from one frame does not directly influence the output from any other frame. We discuss sandboxing necessary to satisfy this assumption in Appendix B.

Suppose a private entity x appears in $|W_x|$ frames within a query window W which is $|W|$ frames long. Further suppose that the analyst chooses to *average* the frame-level outputs of her query over the duration of the window. Each frame is computed upon independently, hence the maximum difference between two executions of the query—one with and without a private entity x—is $(u-l)\cdot \max_x |W_x|/|W|$, where the maximization is over all private entities x. Then, the results from a chunk R_{priv} may be made ϵ-DP by doing:

$$R_{chunk,avg} = \frac{\sum R_{frame}}{|W|}$$
$$\text{noise } \eta_{avg} \propto \text{Laplace}\left(\frac{(u-l)\cdot \max_x |W_x|}{|W|\cdot\epsilon}\right)$$
$$R_{priv,avg} = R_{chunk,avg} + \eta_{avg}$$

In general, any aggregation function that equally weighs the outputs from each frame can be handled in an analogous way. Since $|W_x| \leq |W|$, aggregating query results across frames and adding the noise above can provide a higher query accuracy relative to adding noise per frame. Many useful queries already implement some form of aggregation over per-frame results (§6), since information at the level of every frame is too detailed. The idea of implementing safe aggregations over private inputs was employed by the Airavat system [48] in the context of Hadoop queries, by associating a private individual's data to the rows of a database. However, the notion of a database row is not well-defined in the case of video, much less the notion of how individuals are associated with database rows, making it challenging to apply that solution to video directly.

The differentially-private noise that we added above depends crucially on $\max_x |W_x|$, where x ranges over all the private entities in the video, e.g., people, cars, *etc.* However, it is infeasible to compute this value directly, given the challenges of describing and automatically recognizing private information, as discussed in §3.1. Further, in the worst case, a private entity x may appear in all frames of a chunk. Akin to a database where a single individual's presence affects all rows of the database, it is very challenging for *any* mechanism to simultaneously provide meaningful privacy guarantees for x and highly-accurate query outputs.

Leveraging limited object-persistence. Instead of attempting to provide privacy guarantees in the worst case, we instead

leverage a domain-specific empirical observation about video data, which we call *limited object-persistence*, which allows us to provide privacy guarantees in the "expected case":

- Assumption A4 (*Limited object-persistence*) Private entities are likely to persist for a small fraction of the overall time during which analyst queries are executed. We show empirically that this assumption holds for the video feeds we investigated (§5.1, §6) and expect that the assumption may hold more generally for individuals appearing in front of public-facing cameras. We also describe how privacy degrades when this assumption is violated in §4.2.

Suppose a private entity x appears in a fraction $fr(x)$ of the time duration over which query results are aggregated. We call $fr(x)$ the *persistence* of entity x. We say that a query satisfies **ρ-persistent privacy** if entity x is protected (in an ϵ-differentially-private sense) whenever $fr(x) < \rho$, i.e., the addition or removal of x does not noticeably change the query output as long as $fr(x) < \rho$. It is important to fix the value ρ according to the context of the video, an issue we address in §5.1. If we add to the query output

$$\text{noise } \eta_{avg} \propto \text{Laplace}\left(\frac{(u-l)\cdot\rho}{\epsilon}\right).$$

then the query satisfies ρ-persistent privacy.

4.2 Relaxing the assumptions

(A1) Running stateful queries over video frames. It is desirable for queries to maintain arbitrary state across sequential frames to compute a value, e.g., identifying unique objects and reporting aggregate properties of their motion. To support this, we allow analyst queries to *compute over a set of frames* in a chunk in one shot. There are two main differences in executing queries over chunks rather than frames (§4.1). First, a query is only allowed to produce one result per chunk, rather than one per frame. Second, we execute safe aggregations over the results of groups of contiguous chunks, which we call *windows*. The ϵ-DP guarantees from §4.1 translate straightforwardly to this setting, by adding noise to the query output as follows (illustrated for the `avg` aggregation function):

$$R_{chunk} = \text{query(all frames in chunk)}$$

$$R_{window} = \frac{\sum R_{chunk}}{|W|}$$

$$\text{noise } \eta_{avg} \propto \text{Laplace}\left(\frac{(u-l)\cdot\max_x|W_x|}{|W|\cdot\epsilon}\right)$$

$$R_{priv} = R_{window} + \eta_{avg}$$

where $|W_x|$ is the maximum number of chunks that a private entity x may be present within a window, and $|W|$ is the total number of chunks in a window.

However, increasing the size of a chunk decreases the denominator in $fr(x)$ and thus increases the fraction of an aggregation window that an individual may impact. To account for this, in order to actually satisfy ρ-persistent privacy for a given chunk size $|C|$, we must use ρ' when adding noise, adjusted as follows:

$$\text{noise}\eta_{avg} \propto \text{Laplace}\left(\lceil\frac{\rho}{|C|}\rceil + 1\right) \tag{4.1}$$

(A2) Enabling multiple vector-valued queries. We wish to support multiple queries over the same video data Clearly, each value emitted from a query or a series of queries leaks private information, and simply implementing ρ-persistent privacy for each query in isolation is insufficient, as an adversary can combine information across emitted values.

If a video owner wishes to ensure ρ-persistent ϵ-DP for a set of queries, they must each satisfy ρ_i-persistent $\epsilon - DP$ $i = 1,...,k$, such that: $\frac{1}{\sum_{i=1}^{n}\frac{1}{\rho_i}} \leq \rho$

This can also be applied to other contexts to protect the privacy of individuals appearing in multiple windows of execution. For example, if the video owner wants to ensure ρ-persistent privacy for an entity x reappearing in up to n windows, the video owner can enforce a threshold of $\rho' = n\cdot\rho$. Or, to ensure ρ-persistent privacy for an entity which might be visible to c cameras (either simultaneously or sequentially), the video owner can use $\rho' = c\cdot\rho$.

(A3) Implementing confined query execution. The independence of query execution over each chunk is *necessary* to provide the differential-privacy guarantees, i.e., the value(s) emitted by a query over each chunk, before aggregation, must reflect only the content of that chunk, and not other chunks. If this is not guaranteed, a malicious query can leak and propagate private information across chunks within a window, making aggregation across the chunks of a window ineffective in protecting the privacy of an entity, even if the entity only appears in one chunk. Formally, each chunk is processed by a Turing machine with access to an input tape with data from the chunk and optionally, a random tape. The output of this should lie in the range specified by the analyst. If the machine doesn't terminate within a fixed time budget, crashes or produces invalid output, Privid picks a default value from the specified range. The analyst can specify this default value (e.g. 0). Specifying a time budget is important to prevent the analyst from using query execution time to leak information. Even if the query completes earlier, the answer is only returned after the budget expires.

In [34], Lampson enumerates 6 general sources of information leaks across processes, such as shared memory, filesystem changes, communication over a network, access to pseudo-random number generators, resource usage, and process execution time. This list serves as a useful guide for implementing confinement across query executions. We detail and address these sources of information leakage in Appendix B. With perfect confinement, the analyst observes only two pieces of information from the system: whether or not a query was accepted (which depends only on the total number of queries and the video owner's budget, both of which constitute public information), and the result of the query (if it there is sufficient privacy budget remaining). We have already shown that the query result itself is safe to release.

(A4) Graceful degradation of privacy for entities with large persistence. ρ-persistent privacy does not protect entities x in an ϵ-DP manner whenever their persistence, $fr(x)$ over the videos, is larger than ρ. However, a nice property of differentially-private algorithms [21] is that privacy degrades "gracefully," with an adversary only able to do slightly better than random guessing as $fr(x)$ increases slightly beyond ρ. As $fr(x)$ increases well beyond ρ, an adversary can determine the presence of x with increasingly higher confidence. In Appendix C, we show that the probability that an adversary detects x with a false-positive probability at most α is at most:

$$\min\{e^{\epsilon}\alpha, e^{-\epsilon}(\alpha - (1-e^{\epsilon}))\} \tag{4.2}$$

This probability is shown in Fig. 4 for a 4 example α. It is worth noting that this analysis quantifies the *potential* loss of privacy in the worst case, when an adversary has implemented

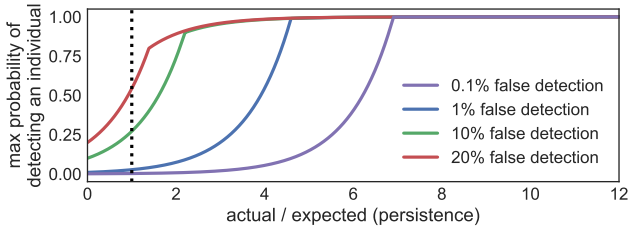


Figure 4: *Privacy Degradation. As an individual persists past the privacy parameter, their presence is not immediately revealed, but rather they provide more and more evidence to a hypothetical adversary, who can determine with more and more confidence that the individual was present. We show how privacy degrades for 4 example adversary confidence thresholds.*

an optimal query that identifies the presence of x. If the analyst isn't explicitly identifying x, its privacy isn't compromised regardless of its persistence. In contrast, if a denaturing-based solution misses x during the denaturing process, the presence of x is readily available to be detected by any subsequent query, regardless of the persistence of x.

Putting all of the steps above together, the Privid query execution process is shown in Algorithm 1. We provide an illustration in Appendix D.

Algorithm 1: Privid Query Execution

Input: Video, persistence ρ, DP parameter ϵ, analyst query ={window size $|W|$, chunk size $|C|$, output range (l,u)} ,

Output: ρ-persistent-private query result

1 Split window into $|W|$ consecutive chunks, each containing $|C|$ consecutive frames.

2 Feed each chunk into a *confined* execution instance of the analyst query. Chunk i produces output o_i.

3 Query outputs over chunks within a window are fed into a trusted aggregation function f (e.g., avg or $\sum$). The result is $R_{window}=f(o_1,o_2,...,o_{|W|})$.

4 Sample noise η corresponding to the sensitivity, depending on the aggregation function used. For example, with the avg aggregation function $\eta_{avg}=\text{Laplace}\left(\frac{(u-l)\cdot\rho}{\epsilon}\right)$.

5 Return result $R_{priv}=R_{window}+\eta$.

5. ENHANCING PRACTICALITY

In §4 we exploited limited persistence in the temporal dimension of video. Here, we leverage spatial properties of video content (§5.1 lowers persistence and §5.2 reduces output range) and expand the query output interface (§5.3) to improve accuracy without increasing the leakage of private information.

5.1 Spatial Masking

Although most individuals tend to persist in a camera's view for a short amount of time (§4.1), the persistence values can be heavy-tailed in practice, i.e., a few individuals may linger for long periods of time. This poses a tricky tradeoff: considering the maximum persistence values will greatly degrade query accuracy, while targeting anything less will sacrifice privacy for certain individuals.

To address this, we leverage our empirical observation that the individuals that persist in a scene for a long duration (relative to the majority) tend to spend the majority of their time in one of a few fixed regions in the scene. These regions are typically scene-specific and stable over time, and can thus be determined by the video owner a priori, e.g., based on historical data. For example, at a crosswalk, people may spend a long time sitting on a bench or waiting to cross the street, but they will spend a relatively short amount of time walking to and from the bench or crosswalk. Similarly, parked cars will spend most of their time in the parking spot, and a relatively short amount of time entering/exiting the spot.

This empirical observation motivates a simple (yet effective) technique to reducing maximum persistence values: if the regions housing individuals with high persistence values are masked (e.g., replaced with black pixels) prior to the query execution, the observable maximum persistence will be drastically reduced. This, in turn, will enable Privid to safely lower the noise value and improve query accuracy for the analyst, without harming the privacy of any individuals.

Automatically detecting high-persistence regions. In order to understand the efficacy of this approach on reducing persistence values, we analyzed 3 videos in our dataset (§6.1), considering 24 hours from each. In each video, we first identified high-persistence regions by generating a spatial distribution of persistence. This distribution shows the amount of time that objects (or individuals) spend in each region of the frame, which in turn can highlight regions with the highest persistence values. To do this, we run object detection and tracking (for private objects) over each video to obtain the bounding boxes for each individual x in each frame f. Then, for each pixel (i,j), we assign a value equal to the maximum number of frames that any one individual overlapped with that pixel.

The top row in Fig. 5 illustrates each distribution as a heatmap overlaid on a screenshot of the video. In the figure, brighter (yellow) values represent areas of higher persistence, while darker (purple and grey) areas represent areas where no objects persisted for long times. Using these distributions, in each video we then identified the minimum-sized region that maximized the coverage of high-persistence coordinates. These regions are shown as masked in the bottom row of Fig. 5, and are somewhat intuitive based on the context of each scene. For example, for `campus` and `urban`, the mask predominantly covers the area where pedestrians wait to cross the street, while for `highway` the mask covers an area where a car is typically parked.

An important observation is that some regions naturally have 0 persistence, meaning across all historical data no individuals ever overlapped those pixels. This provides opportunity for some queries over non-private objects to use extensive masks for very high-accuracy low-granularity queries. For example in §6.4 we leverage this to monitor the duration of traffic lights.

Microbenchmark: gains from applying masking. Finally, to determine the impact on maximum persistence values (and query accuracy), we ran object tracking on each video both before and after applying the mask, and compared the distribution of persistence values. Our results are promising: in `campus`, the mask decreases the maximum persistence by 60% (250 to 100 seconds), and the 99th percentile by 80% (120 to 25 seconds); for `highway`, the maximum persistence decreased by 94% (3600 to 200 seconds); for `urban`, maximum persistence decreased by 76% (340 to 80 seconds). We show how these improvements translate to a significant increase in query accuracy in §6.

How does masking differ from denaturing? It is important to note that, although these masks represent a form of denaturing, they do not suffer from the problems described in §3.2. The reason is

(a) `campus` (Auburn) (b) `highway` (Hampton) (c) `urban` (Shibuya)

Figure 5: *Heatmaps (yellow/blue indicates max/min persistence) and resulting masks for each video in our dataset; persistence range is normalized per video.*

Video	Max(frame)	Max(piece)	Reduction
`campus`	3	6	2.00×
`highway`	40	23	1.74×
`urban`	37	16	2.25×

Table 1: *Potential improvement by splitting each of our 3 videos into distinct regions. Reduction shows the factor by which the noise could be reduced. 2× cuts the necessary privacy level in half.*

that the masks here cover fixed locations in a scene, as opposed to dynamic locations that cover moving objects across frames. As a result, these masks will not dynamically obfuscate other objects and confuse an object detector (leading to unpredictable effects on query accuracy). In addition, these masks can be be safely released to the analyst ahead of time, so that the analyst can account for them in their query implementation (e.g., ignore any detected "objects" whose coordinates fall within the masked region, because they are likely to be false positives). In many cases, this can be done without negatively impacting query accuracy. For example, a query counting the total number of people can count people before they enter or after they leave the masked region–their behavior within the masked region does not impact the result over even short time windows.

Incorporating Masking. Privid always aims to ensure the desired privacy level of the video owner. Masking provides an opportunity for the analyst to decrease the amount of noise necessary to meet that level of privacy (and thus increase the accuracy of their query) at the cost of losing some information from the raw video. Depending on the goal of their query, the analyst can use persistence information provided by the video owner to construct a mask that will minimize persistence without masking a region they care about. Admittedly, masking may negatively impact some queries, such as an estimation of the average density of people in a location. In these cases, the analyst can choose to forgo masking entirely and accept the higher privacy level. The choice of mask is irrelevant to the video owner, as Privid will continue to protect the privacy of all non-masked individuals.

5.2 Spatial Splitting

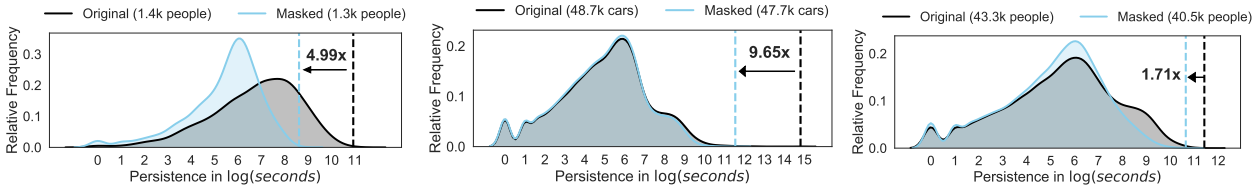
The techniques in §5.1 aim to reduce the persistence values for a given video in order to reduce the amount of noise with Privid. Here, we focus on reducing the other noise-determining factor: the maximum output value for the query under consideration.

Our high-level insight is that, at any point in time, an individual typically only occupies a relatively small portion of a video frame. Building off of this, if a query can be implemented properly by analyzing subsets of each frame *independently*, then Privid can provide a more desirable (i.e., lower) bound on the amount of noise that must be added to preserve privacy. Intuitively, we seek to split each frame into smaller *pieces*, such that the maximum query output value for each piece is lower than that of the total frame, e.g., the person count in a given piece will be $\leq$ to that for the entire frame[5]. Such spatial splitting enables us to limit the amount of the query output that a single individual can impact. More formally, if a given frame is split into n pieces, and we know that any individual can be in at most s of those pieces, then we can reduce the privacy parameter by up to $\frac{s}{n}$.

Unfortunately, leveraging this insight in a way that yields noise benefits is challenging for two reasons that both relate to an individual affecting multiple pieces of a frame (precluding noise reductions). First, Privid applies noise at the granularity of video chunks, i.e., multiple frames. The problem is that an individual may move between pieces during a chunk, affecting the query output values for each piece that they appear in. The precise impact depends on the chunk size, the distance of the individual from the camera, and the speed and direction of their movement, all of which are difficult to bound tightly. To address this, when spatial splitting is applied, we mandate that the chunk size must be a single frame. Since an object can only be in one place at any given time, this approach ensures that an object cannot move between pieces within a chunk, but only applies for queries that do not need temporal context, such as density queries.

Second, once a frame is split into pieces, it is feasible for an individual to be detectable in multiple pieces, e.g., if the individual is at the intersection of pieces or is larger than a the size of a piece. Privid offers two ways of handling this. The video owner can automatically compute the maximum size of a private object (using historical data from the scene), and then select the number and size of pieces in a way that bounds the number of pieces that an individual can overlap with at one time. Alternatively, the video owner could leverage scene-specific insights to manually select reasonable splits where they have confidence that an individual can impact only one

[5]In certain settings, splitting in this manner could also yield per-piece persistence reductions, as compared to the entire frame.



(a) `campus` (b) `highway` (c) `urban`

Figure 6: *The distribution of persistence of private objects is heavy tailed. Applying the heatmap-driven mask from Fig. 5 significantly lowers the maximum persistence value (and thus required noise level), while still allowing most private objects to be detected. The key denotes the total number of private objects detectable before and after applying the mask. The dotted lines highlight the maximum persistence, and the arrow text denotes the relative reduction in max persistence.*

at any time. For instance, a video owner may choose to split a frame based on highway lanes or parking spots (since a car should be in only one at a given time), or crosswalks at an intersection (since a person will only cross one at a time).

Microbenchmark: gains from spatial splitting. To evaluate the ability of this technique to reduce maximum query output values, we applied manually-defined spatial splits to each of the 3 videos in our dataset (described in §6.1); splitting only leveraged the intuitive insights described above, such as separating distinct road lanes into separate pieces. Table 1 compares the maximum output value for car and people counting queries both with and without spatial splitting. As shown, for each video, spatial splitting has the potential to cut the privacy parameter (and thus noise) in half. We show how these improvements translate to substantial increases in query accuracy in §6.

5.3 Expanding Query Interface

The query interface described in §4 limited the output for simplicity of explanation. We briefly describe possible extensions to illustrate Privid's flexibility (their proofs and more extensions in Appendix E).

ArgMax Consider a query wishing to monitor N values simultaneously (such as the number of visitors to each of a few nearby stores) and just find the largest one. Since each query can only output a single value, this would require N queries each with threshold $n{\cdot}\rho$. Instead, each chunk can output a value per store (N fixed ahead of time), independently sum the set of values for a given store, and output the `ArgMax` across these sums.

Set Intersection Denaturing fundamentally cannot support queries that refer identities across time or cameras (e.g., such as the number of cars that passed by one camera and then another). However, Privid can support this by allowing each chunk to output a *set* of unique objects (e.g., set of license plate strings) At the end of the window, we can simply apply a union operator across all aggregate $a \in A$ with noise proportional to Δ^a (which is the same as normal except that it must be further multiplied by the max length of any chunk output). Finally, we can apply a set intersection between different O^a (from the same camera or different cameras); then it is safe to output the intersection set size with sensitivity $max_{a \in A} \Delta^a$.

6. EVALUATION

The evaluation highlights of Privid are as follows:

1. Privid's spatial and temporal splitting and masking allow it to significantly reduce (597×-3093×) the amount of noise necessary to achieve differentially private output without hampering the query's accuracy, compared to a straightforward application of DP in video analytics (§6.2).
2. By exposing parameters that bear clear impact on accuracy (utility) and temporal granularity of queries, Privid enables video owners and analysts to flexibly and formally trade utility loss and query granularity while preserving the same DP guarantees (§6.3).
3. Privid supports a diverse range of video analytics queries, including object counting, duration queries, and composite queries; for each, Privid ensures privacy with low utility cost through leveraging query-specific opportunities with the same design (§6.4).

6.1 Evaluation Setup

Dataset. To test Privid under diverse scenarios and temporal patterns, we use three long videos from the following public video scenarios (screenshots in Fig. 5), each collected from YouTube over a period of 12 hours continuously (6am-6pm EST). `campus` overlooks a street corner with moderate activity. It represents a suburban area where there is a variable number of people and cars throughout the day, but low density at any given time. `highway` overlooks a highway in Southampton, NY. There are few people, if any, and the main private objects are cars (which could be used to identify the owner). Cars move at varying speeds and the traffic density varies throughout the day. `urban` overlooks a busy intersection in Shibuya, Japan, which is typical in a high-density urban area. Further, these scenarios tend to have much higher persistence (people often wait, loiter, and chat) making it challenging to ensure privacy (which requires more noise) and good utility.

Testbed. We implemented Privid in 4k lines of Python. We used `ffmpeg` to split videos into chunks and decode individual frames, and used LXC (Linux containers) to provide a secure sandbox for query execution. All queries were implemented using Faster-RCNN [46] model within the Detectron-v2 [60] framework for object detection, and DeepSORT [59] for object tracking. We chose the hyperparameters for detection and tracking on a per-video basis. More details are presented in Appendix A.

Metrics. For all queries, Privid seeks to reduce the amount of noise while providing ρ-persistent privacy (with $\epsilon = 1$). The ρ threshold used for each video is listed in Table 2 and directly results from the masks we chose in §5.1. We assume the query for each video segment is able to use the entire privacy budget for that segment.

Parameter settings. By default, we consider a representative query (we will discuss more queries in §6.4) to examine how much noise Privid's techniques reduce to achieve DP compared to a *naive* application of DP. The naive DP assumes the worst-case sensitivity to the presence of an individual, so the noise is equivalent to the entire output range. Notice that this is the only way to ensure differentially private output without leveraging Privid's observation of persistence. We describe the queries we chose and their parameters in Table 2.[6]

[6] We chose the best parameters (shown by "X" in Fig. 9) for each query based on the same information an analyst would

Video	Query	$\|C\|$	Adjust	ρ	Range
`campus`	Count Unique People	30 sec	Mask	49 sec	(0,6)
`highway`	Count Unique Cars	2.0 min	Mask & Split	2.0 min	(0,100)
`urban`	Count Unique People (in crosswalks)	30 sec	Mask & Split	3.3 min	(0,23)

Table 2: *Representative queries used for §6.2 and §6.3. Queries use a window size of 1 hr.*

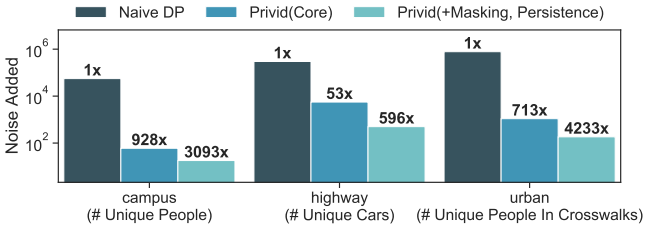


Figure 7: Privid *significantly reduces the amount of noise necessary to satisfy DP compared to naively applying it (Naive DP) for our three video/query pairs. Privid(Core) incorporates the persistence threshold, but not masking or splitting (§5). Privid(+) adds them. The y-axis is the proportion of noise that would be added to the aggregate query result. For example, Naive would add noise proportional to 1k unique people for* `campus`*. The number above each bar shows the relative* decrease *in noise compared to Naive.*

6.2 Cost of Privacy

Noise reduction to satisfy same DP. We begin with two key questions: to achieve DP, how much noise does the basic Privid framework reduce and further, how much noise do the video-specific optimizations (masking and splitting) reduce? Fig. 7 shows that, e.g., for `campus`, utilizing Privid's query execution framework developed in §4 results in a three orders of magnitude reduction (928×) in the amount of noise necessary to satisfy DP compared to Naive. Incorporating masking and splitting further improves the reduction 3,093× relative to Naive. In practice, this translates to adding or removing roughly 18 people by Privid to the total aggregate count as noise to achieve the same worse-case differentially private output that Naive needs to add/remove 1k people to achieve.

Minimal utility loss. Privid introduces two forms of inaccuracy to a query result: (1) intentional noise to satisfy DP, (2) unintentional inaccuracies caused by the impact of temporally and spatially splitting and masking videos on the underlying the query. In Fig. 8, we separate these two to show that Privid's execution framework does not significantly hamper the accuracy of the query, and that the amount of noise added by Privid allows the final result to preserve the overall trend of the original.

6.3 Impact of Parameters

Privid provides a lot of flexibility for the analyst to choose how their query is executed over the video, allowing the analyst to balance coarseness and accuracy of results, while ensuring the same ρ-persistence privacy. In particular, given a query implementation and aggregation function they wish to run, they can choose the window size ($|W|$, how often aggregates are computed), the chunk size ($|C|$, sequential frames each independent query execution can view at a time), and the range of values each chunk can output. To showcase this flexibility, we re-execute the three video/query pairs from §6.2 and jointly sweep over a range of chunk size and output range (Fig. 9) and over a range of window sizes (Fig. 10).

Fig. 9 shows that as we increase the chunk size for a given output range, the average error decreases (due to better raw query accuracy), but the size of the error bar increases (due to additional noise). This seems to contradict the intuition that, as one increases the chunk size, the granularity over which we can restrict the impact of an individual becomes coarser, meaning Privid has to be more conservative and increase the noise to meet the same ρ-persistent privacy guarantee (this follows from Eq. 4.1). However, at the same time, this additional context allows the query to more accurately compute the number of unique people, and thus the raw accuracy increases. That said, for relatively small chunk sizes (less than the persistence), the decrease in error from having more context outweighs the increase in error from slightly larger noise.

Fig. 10 keeps the chunk size and output range constant (at the "X" values in Fig. 9) and shows that as the window size increases, the number of chunks an individual could influence remains constant, while the total number of chunks included in the aggregate result grows (i.e., $fr(x)$ decreases, see §4.2-A1). Consequently, the proportional noise required to hide an individual decreases, and thus Privid's average error decreases.

6.4 Applying Privid to Other Queries

Privid's interface enables a diverse set of queries. Given our dataset, we chose three more queries to highlight a few different opportunities Privid enables (results are summarized in Table 3).

Case study 1 (Counting): Non-private objects that change infrequently. To exemplify this, we measure the fraction of trees that are still in bloom (this property changes roughly twice a year). Because the objects are relatively static within a window, the query can roughly output the exact same value for every single chunk, allowing for a high utility. The window size could be greatly increased to further reduce the noise, but even 12 hours gives us very high accuracy. Such queries would be most useful when executed over an entire network of cameras, e.g., to identify a relatively small subset of cameras where trees are still in bloom in the fall.

Case study 2 (Duration query): Using restrictive masking to enable fine-grained results. To exemplify this, we measure the average amount of time a traffic signal was "red". By masking the entire video except the traffic lights (shown in Appendix **??**), Privid can offer zero persistence, because no private objects impact the traffic light pixels. This allows for perfectly accurate results (compared to not using Privid).

Case study 3 (Composite query): Stateful filters. To exemplify this, we filter people in `campus` to only count those walking towards campus. This exemplifies Privid's ability to enable complex queries over the trajectories of objects. The majority of the error is a result of our computer vision model failing to re-identify people that cross the masked region as the same individual. This is not fundamental to Privid and could be improved with a better tracking algorithm.

7. RELATED WORK

Denaturing. The wide applications of computer vision have inspired much effort to treat a balance between privacy and utility of these applications. As detailed in §3.2, current solutions commonly use various video-denaturing techniques (e.g., [12, 16, 43, 55–57]), and thus must achieve two goals which are both hard and sometimes conflicting [40]: blocking all private information in source videos and not adversely impacting accuracy of computer vision algorithms. In contrast, Privid treats the video analytics pipelines as a whole and perturbs the output by the concepts of differential privacy (DP) to ensure privacy of each individual's presence.

have (a screenshot and persistence heatmap). These parameters are not necessarily optimal, but enable a fair comparison.

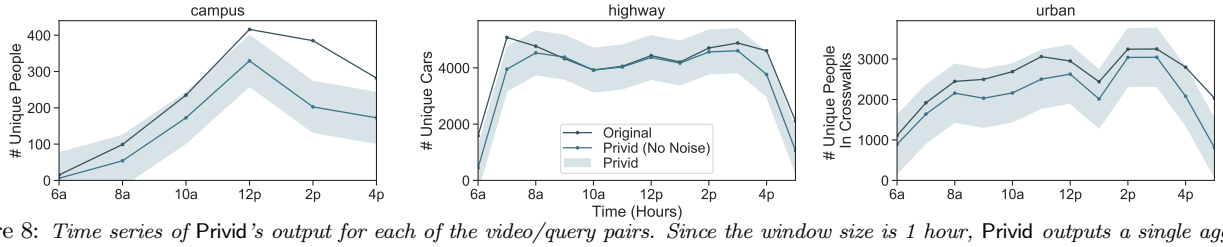


Figure 8: *Time series of* Privid*'s output for each of the video/query pairs. Since the window size is 1 hour,* Privid *outputs a single aggregate count per hour. The "Original" line shows what the per-hour query output would be without* Privid*, analyzing the video one hour at a time. "Privid (No Noise)" shows what* Privid *would output if it did not add noise. Both Original and* Privid *use the same exact query implementation (detection and tracking algorithms), though* Privid *involves splitting and masking in the execution. The light blue ribbon around the Privid line shows where noisy values will fall relative to the raw output 99% of the time.*

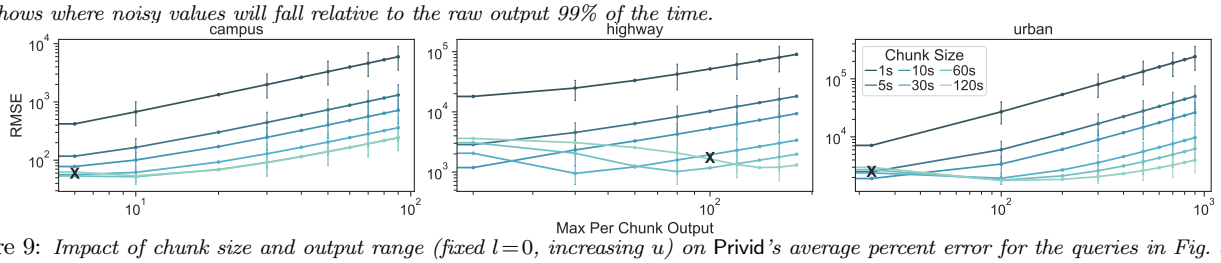


Figure 9: *Impact of chunk size and output range (fixed $l=0$, increasing u) on* Privid*'s average percent error for the queries in Fig. 8. The reference value is the same as Fig. 8, namely the "Original" line. Error bars computed over 100 samples of noisy outputs from* Privid*. The "X" represents the exact pair of parameters we chose for each video in Fig. 8.*

Query	Parameters	Video	Persistence	Query Output	Accuracy
Fraction of trees with leaves (%)	$\|W\|$=12 hrs, $\|C\|$=1 frame, Agg = `mean`, l,u=(0,100)	`campus`	48.89 sec	15/15 = 1.00	99.90%±0.11%
		`highway`	6.21 min	3/7 = 0.43	98.24%±1.90%
		`urban`	3.34 min	4/6 = 0.67	99.39%±0.66%
Duration of Red Light (seconds)	$\|W\|$=12 hrs, $\|C\|$=10 min Agg = `mean`, l,u=(0,300)	`campus`	0 sec	75	100.00%
		`highway`	0 sec	50	100.00%
		`urban`	0 sec	100	100.00%
# Unique People (Filter: trajectory moving towards campus)	$\|W\|$=12 hrs, $\|C\|$=10 min Agg = `sum`, l,u=(0,25)	`campus`	49 sec	576	78.21%±10.56%

Table 3: *Summary of additional query results. For a given query, we use the same parameters for each video. We use the same mask, so the persistence values are the same as in the previous section. Accuracy values are the mean ± 1 std. dev. compared to a ground truth running the same query over the entire video, but without using* Privid*.*

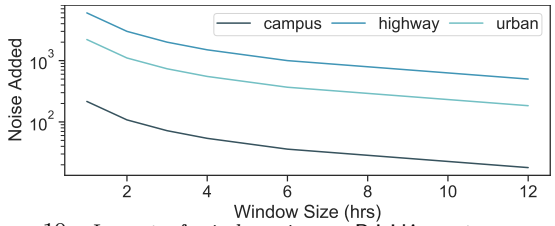


Figure 10: *Impact of window size on* Privid*'s root mean square error (RMSE) for the queries in Fig. 8. As window size increases,* Privid *can add less noise to meet the same ρ-persistent guarantee.*

Traditional DP. While DP has made remarkable successes in problems where queries are transparent and data are tabular-structured with well-defined private records, such as generic databases (e.g., [30,39,41,62]), network data (e.g., [17]), time-series (e.g., [18]) and geospatial data (e.g., [13,33]), none is true in video analytics: neither DNN-based queries nor video data are well-structured. The closest work to Privid tries to support arbitrary queries (e.g., MapReduce [48]), but it still relies on tabular data and pre-annotated private rows. Privid makes DP practical in video analytics by relaxing these assumptions while leveraging video-specific properties (e.g., spatially uneven distribution of private information in public videos) to rein in excessive noise.

DP for video/training. PixelDP [35] leverages DP to develop models robust to adversarial examples, but does not use it to provide *privacy* for individuals in video content, which Privid does.

Video Analytics Pipelines. Recent work develops query pipelines and abstraction for public video that incorporates performance optimizations, but does not explicitly try to ensure privacy (e.g., [24,25,28,29,31,38,45,63]). As Privid is agnostic to the underlying video analytics pipelines (though it limits query output), we expect Privid can be expanded to combine these approaches to create a private, efficient, and high-level query language.

8. CONCLUSION AND ETHICS

This paper presents Privid, the first system that extends formal differential privacy techniques to video analytics pipelines. To ensure practicality despite the opaqueness in data structure and query operation, Privid introduces several video-specific optimizations that enable the formal DP guarantee for individuals in a video, while providing a sensible bound on how the query result is impacted.

In building Privid, we do not advocate for the increase of public video surveillance and analysis. Instead, we observe that this is prevalent already, and is driven by strong economic and public safety incentives. Consequently, it is undeniable that the analysis of public video will continue, and thus, it is paramount that we provide tools to improve the privacy landscape for such analytics. We seek to encourage video owners that it is indeed possible to have privacy as a first-class citizen, while still enabling useful queries. Further, we do anticipate new legislation that restricts video collection and analysis; privacy-preserving video analytics systems (like Privid) will become even more crucial to enable critical applications while complying with laws.

9. REFERENCES

[1] Absolutely everywhere in beijing is now covered by police video surveillance. https://qz.com/518874/.

[2] Are we ready for ai-powered security cameras? https://thenewstack.io/are-we-ready-for-ai-powered-security-cameras/.

[3] British transport police: Cctv. http://www.btp.police.uk/advice_and_information/safety_on_and_near_the_railway/cctv.aspx.

[4] Can 30,000 cameras help solve chicago's crime problem? https://www.nytimes.com/2018/05/26/us/chicago-police-surveillance.html.

[5] Data generated by new surveillance cameras to increase exponentially in the coming years. http://www.securityinfowatch.com/news/12160483/.

[6] Oakland bans use of facial recognition. https://www.sfchronicle.com/bayarea/article/Oakland-bans-use-of-facial-recognition-14101253.php.

[7] Paris hospitals to get 1,500 cctv cameras to combat violence against staff. https://bit.ly/2OYiBz2.

[8] Powering the edge with ai in an iot world. https://www.forbes.com/sites/forbestechcouncil/2020/04/06/powering-the-edge-with-ai-in-an-iot-world/.

[9] San francisco is first us city to ban facial recognition. https://www.bbc.com/news/technology-48276660.

[10] Video analytics applications in retail - beyond security. https://www.securityinformed.com/insights/co-2603-ga-co-2214-ga-co-1880-ga.16620.html/.

[11] The vision zero initiative. http://www.visionzeroinitiative.com/.

[12] P. Aditya, R. Sen, P. Druschel, S. Joon Oh, R. Benenson, M. Fritz, B. Schiele, B. Bhattacharjee, and T. T. Wu. I-pic: A platform for privacy-compliant image capture. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys 16, page 235248, New York, NY, USA, 2016. Association for Computing Machinery.

[13] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914, 2013.

[14] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[15] Z. Cai, M. Saberian, and N. Vasconcelos. Learning complexity-aware cascades for deep pedestrian detection. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 3361–3369, Washington, DC, USA, 2015. IEEE Computer Society.

[16] A. Chattopadhyay and T. E. Boult. Privacycam: a privacy preserving camera using uclinux on the blackfin dsp. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[17] R. Chen, B. C. Fung, S. Y. Philip, and B. C. Desai. Correlated network data publication via differential privacy. *The VLDB Journal*, 23(4):653–676, 2014.

[18] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau. Pegasus: Data-adaptive differentially private stream processing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1375–1388, 2017.

[19] S. Chinoy. We built an unbelievable (but legal) facial recognition machine. https://www.nytimes.com/interactive/2019/04/16/opinion/facial-recognition-new-york-city.html, 2019.

[20] C. Dwork. Differential Privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, July 2006. Edition: 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006).

[21] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284, Berlin, Heidelberg, Mar. 2006. Springer.

[22] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, Aug. 2014.

[23] Z. Gu, H. Huang, J. Zhang, D. Su, H. Jamjoom, A. Lamba, D. Pendarakis, and I. Molloy. Yerbabuena: Securing deep learning inference data via enclave-based ternary model partitioning. *arXiv preprint arXiv:1807.00969*, 2018.

[24] B. Haynes, A. Mazumdar, A. Alaghi, M. Balazinska, L. Ceze, and A. Cheung. Lightdb: A dbms for virtual reality video. *Proceedings of the VLDB Endowment*, 11(10):1192–1205, 2018.

[25] K. Hsieh, G. Ananthanarayanan, P. Bodik, S. Venkataraman, P. Bahl, M. Philipose, P. B. Gibbons, and O. Mutlu. Focus: Querying large video datasets with low latency and low cost. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 269–286, 2018.

[26] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. {GAZELLE}: A low latency framework for secure neural network inference. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1651–1669, 2018.

[27] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63(6):4037–4049, 2017.

[28] D. Kang, P. Bailis, and M. Zaharia. Blazeit: optimizing declarative aggregation and limit queries for neural network-based video analytics. *Proceedings of the VLDB Endowment*, 13(4):533–546, 2019.

[29] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, and M. Zaharia. Noscope: optimizing neural network queries over video at scale. *Proceedings of the VLDB Endowment*, 10(11):1586–1597, 2017.

[30] I. Kotsogiannis, Y. Tao, X. He, M. Fanaeepour, A. Machanavajjhala, M. Hay, and G. Miklau. Privatesql: A differentially private sql query engine. *Proc. VLDB Endow.*, 12(11):13711384, July 2019.

[31] S. Krishnan, A. Dziedzic, and A. J. Elmore. Deeplens: Towards a visual data management system. *arXiv preprint arXiv:1812.07607*, 2018.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[33] T. Kulkarni. Answering range queries under local differential privacy. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1832–1834, 2019.

[34] B. W. Lampson. A note on the confinement problem. *Commun. ACM*, 16(10):613615, Oct. 1973.

[35] M. Lécuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 656–672. IEEE, 2019.

[36] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5325–5334, June 2015.

[37] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, July 2017.

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

[39] A. Machanavajjhala, X. He, and M. Hay. Differential privacy in the wild: A tutorial on current practices & open challenges. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1727–1730, 2017.

[40] R. McPherson, R. Shokri, and V. Shmatikov. Defeating image obfuscation with deep learning. *arXiv preprint arXiv:1609.00408*, 2016.

[41] F. D. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD 09, page 1930, New York, NY, USA, 2009. Association for Computing Machinery.

[42] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[43] S. J. Oh, M. Fritz, and B. Schiele. Adversarial image perturbation for privacy protection a game theory perspective. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1491–1500. IEEE, 2017.

[44] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta. Visual privacy protection methods: A survey. *Expert Systems with Applications*, 42(9):4177–4195, 2015.

[45] A. Poms, W. Crichton, P. Hanrahan, and K. Fatahalian. Scanner: Efficient video analysis at scale. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018.

[46] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[47] M. S. Riazi, M. Samragh, H. Chen, K. Laine, K. Lauter, and F. Koushanfar. {XONN}: Xnor-based oblivious deep neural network inference. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1501–1518, 2019.

[48] I. Roy, S. T. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: Security and privacy for mapreduce. In *NSDI*, volume 10, pages 297–312, 2010.

[49] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS 15, page 13101321, New York, NY, USA, 2015. Association for Computing Machinery.

[50] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 3476–3483, Washington, DC, USA, 2013. IEEE Computer Society.

[51] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[52] S. Tople, K. Grover, S. Shinde, R. Bhagwan, and R. Ramjee. Privado: Practical and secure dnn inference. *arXiv preprint arXiv:1810.00602*, 2018.

[53] F. Tramer and D. Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. *arXiv preprint arXiv:1806.03287*, 2018.

[54] T. van Elsloo, G. Patrini, and H. Ivey-Law. Sealion: a framework for neural network inference on encrypted data. *arXiv preprint arXiv:1904.12840*, 2019.

[55] N. Vishwamitra, B. Knijnenburg, H. Hu, Y. P. Kelly Caine, et al. Blur vs. block: Investigating the effectiveness of privacy-enhancing obfuscation for images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 39–47, 2017.

[56] H. Wang, S. Xie, and Y. Hong. Videodp: A universal platform for video analytics with differential privacy. *arXiv preprint arXiv:1909.08729*, 2019.

[57] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan. A scalable and privacy-aware iot service for live video analytics. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 38–49. ACM, 2017.

[58] X. Wang. Intelligent multi-camera video surveillance: A review. *Pattern recognition letters*, 34(1):3–19, 2013.

[59] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[60] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[61] L.-Q. Xu, J. L. Landabaso, and B. Lei. *Segmentation and Tracking of Multiple Moving Objects for Intelligent Video Analysis*, pages 239–255. Springer London, London, 2006.

[62] M. Xu, T. Wang, B. Ding, J. Zhou, C. Hong, and Z. Huang. Dpsaas: multi-dimensional data sharing and analytics as services under local differential privacy. *Proceedings of the VLDB Endowment*, 12(12):1862–1865, 2019.

[63] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman. Live video analytics at scale with approximation and delay-tolerance. In *NSDI*, volume 9, page 1, 2017.

APPENDIX

A. MEASURING PERSISTENCE

Computing persistence values for a given scene requires the ability to track individuals in that scene. Unfortunately, even state-of-the-art vision techniques for object tracking are riddled with inaccuracies that stem from occlusion (i.e., line of sight to an object is blocked), illumination, and poor video quality; these challenges are exacerbated in low-quality public surveillance videos. Manual annotation of individuals in video can overcome these challenges but is far from scalable and is difficult to use for real-time video analysis.

We observe that, even though the aforementioned challenges preclude off-the-shelf algorithms from perfectly tracking every individual, their hyperparameters can be trained in a way that generates a reasonably accurate distribution of persistence values, which is sufficient for Privid to provide meaningful privacy guarantees.

For each video in our dataset (described in the paper), we first ran object detection using Facebook's Detectron2 [60] library with the included Faster-RCNN model [46]. Using these object detection results, we then manually annotated a subset of video for each camera, producing a ground truth dataset of persistence values. Annotation for a video involved recording the exact time each unique individual entered and exited the scene at the second granularity. Individuals may reappear and thus have multiple enter and exit times. We calculated the "persistence" of each individual in the ground truth by summing the difference between all their enter and exit times. For example, if an individual entered at $t=1m$, exited at $t=2m$, entered again at $t=8m$, and then exited at $t=9m$, their persistence was recorded as $(2-1)+(9-8)=2$ minutes.

Using our ground truth dataset, we then tuned the hyperparameters of a state-of-the-art tracking algorithm called DeepSORT [59] for each camera's video. Our goal was to find the configuration of parameters that produced the persistence distribution which most closely matched that of the annotated ground truth data. To do this, we ran DeepSORT with all possible combinations of the hyperparameters listed in Table 4. For each configuration, we computed the distribution of persistence values over the idividuals the algorithm identified, and compared it to our ground truth distribution.

In `highway`, we consider cars as the private object rather than people because no people are visible in the video, but a car's license plate, or their combination of make, model and color may be enough to identify an individual. As DeepSORT is specific to tracking people, we used SORT [14] instead. Table 5 lists the set of hyperparameters we considered and chose for tuning SORT. In practice, if a video contains both people and cars, the persistence distribution should account for both.

B. SANDBOX REQUIREMENTS

In order for Privid's privacy guarantees to hold, it must execute the query over each chunk of the video in a separate sandbox environment. Abstractly, it must ensure:

1. The query output for a chunk is based solely on information in that chunk, and not information in any other chunk.
2. The query output is the only information the analyst can observe (i.e., there are no side channels that can leak information beyond what Privid is expecting)

In practice, this requires accounting for a number of possible subtle mechanisms that could be used to communicate between executions of a sandbox or outside of the sandbox entirely. The following is a list of requirements on the execution process to ensure the above:

- The process must not be able to read or write from the network or any IPC mechanisms.
- The process must not be able to access or create files that are visible to another process.
- Access to a PRNG (e.g., `/dev/urandom`) must be cryptographically secure. Otherwise, if the sequence of bits were predictable, a writer execution could read from the generator until the desired bit is ready and then stop. If the reader can read after the writer has finished, it will see the writer's bit.
- If the process' resource usage is monitored (e.g., for billing purposes), it must not be made available to the analyst. It is reasonable to assume that, since the video owner is running the computation on behalf of the analyst, they may wish for the analyst to pay or the computation. In that case resource usage and cost must be determined entirely a priori and cannot depend on the actual execution itself, otherwise the precise resource usage could be a side channel.
- The process must not be able to vary its own execution time in a way that is visible to the analyst. For example, a malicious query could exit immediately if x is not present, but spin the processor for a long time if they are. This would cause a noticeable increase in total execution time compared to a segment of video where x was not present, leaking x's presence.

C. DEGRADATION OF PRIVACY WITH PERSISTENCE

A nice property of differentially-private algorithms is that privacy "degrades gracefully": coming close to satisfying the definition of privacy, but not all the way, still provides some level of privacy due to the randomness of the noise component. With Privid, ρ is the point at which the adversary can begin to do better than random guessing to determine the presence of x. If x persists for less than ρ, the adversary can do no better than random guessing. As x persists for longer and longer past ρ, the adversary can determine x's presence with greater and greater confidence.

We can formalize this using the framework of binary hypothesis testing. Consider an adversary who wishes to determine the presence of x in a given window of video, W. They submit a query to the system, and observe only $R_{priv}=R_W+\eta$. Based on this value, they must distinguish between the two hypotheses:

$$\mathcal{H}_0\colon\ x \text{ does not appear in } W$$
$$\mathcal{H}_1\colon\ x \text{ appears in } W$$

We write the false positive P_{FP} and false negative P_{FN} probabilities as:

$$P_{FP}=\mathbb{P}(x\in W|\mathcal{H}_0)$$
$$P_{FN}=\mathbb{P}(x\notin W|\mathcal{H}_1)$$

From Kairouz [27, Theorem 2.1], if an algorithm guarantees ϵ-differential privacy ($\delta=0$), then these probabilities are related as follows:

$$P_{FP}+e^{\epsilon}P_{FN}\geq 1 \tag{C.1}$$
$$P_{FN}+e^{\epsilon}P_{FP}\geq 1 \tag{C.2}$$

Suppose the adversary is willing to accept a false positive threshold of $P_{FP}\leq\alpha$. In ther words, they will only accept $\mathcal{H}_1$

video	cos	iou	age	n_init
campus (0.8)	0.1, 0.3, **0.5**, 0.7, 0.9	0.1, 0.3, 0.5, **0.7**, 0.9	16, 32, 48, 64, 80, **96**, 112	2, 3, 5, 7, **9**
urban (0.6)	**0.1**, 0.3, 0.5, 0.7, 0.9	0.1, 0.3, **0.5**, 0.7, 0.9	8, 16, 32, 48, 64, 80, **96**	2, 3, **5**, 7, 9

Table 4: Set of hyperparameters used for tuning DeepSORT for the `campus` and `urban` videos. The set of parameters that we ultimately used for our experiments are bolded.

video	max_age	min_hits	iou_dist
highway (0.2)	240, 480, **720**	3, 5, 7, **9**	**0.1**, 0.3, 0.5, 0.7

Table 5: Set of hyperparameters used for tuning SORT for the `highway` video. The set of parameters that we ultimately used for our experiments are bolded.

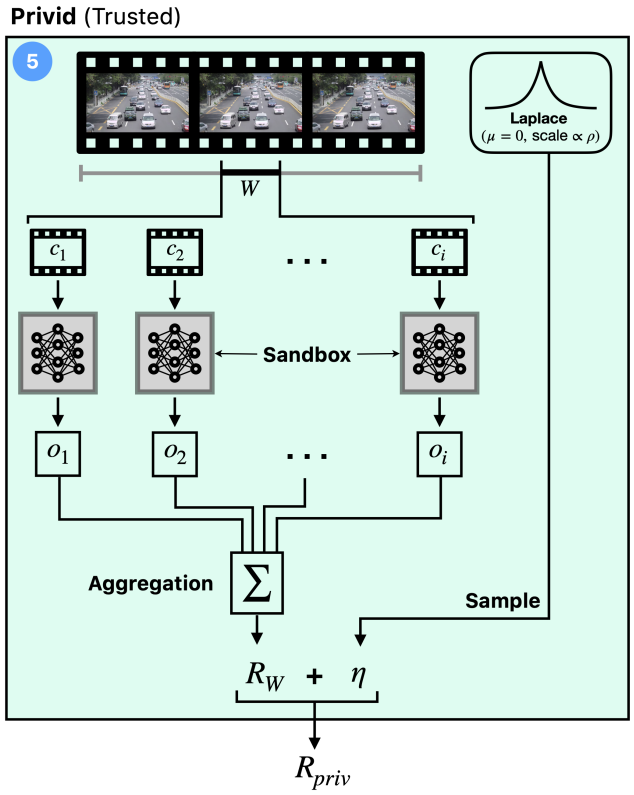


Figure 11: Overview of how Privid executes a query over a given window (W) of video. It takes as input the raw video and outputs a noisy R_{priv} which is safe to return to the untrusted analyst. $\sum$ is used in the diagram to represent the aggregation function, but other functions could be used.

(x is present) if there is less than α probability that x is not actually present.

Rearranging equations C.1 and C.2 in terms of the probability of correctly detetecting x is present ($1-P_{FN}$), we have:

$$\begin{aligned} 1-P_{FN} &\leq & e^{\epsilon}P_{FP} &\leq & e^{\epsilon}\alpha \\ 1-P_{FN} &\leq & e^{-\epsilon}(P_{FP}-(1-e^{\epsilon})) &\leq & e^{-\epsilon}(\alpha-(1-e^{\epsilon})) \end{aligned}$$

The probability that the adversary correctly decides x is present is then at most the minimum of these:

$$\mathbb{P}(x\in W|\mathcal{H}_1) \leq \min\{e^{\epsilon}\alpha, e^{-\epsilon}(\alpha-(1-e^{\epsilon}))\} \qquad \text{(C.3)}$$

In Fig. 4, we visualize C.3 as a function of an individual's persistence past ρ, for 4 different adversarial confidence levels (α=0.1%,1%,10%,20%).

D. QUERY EXECUTION ILLUSTRATION

Fig. 11 is an expansion of Step 5 from Fig. 1, which illustrates the overall query execution process as described in Algorithm 1.

E. EXTENDING QUERY INTERFACE

The initial query interface described in §?? limited the output to a single value and the aggregation function to sum or mean. In this section we describe how to expand the framework to incorporate other output values and aggregation functions.

This list is not exhaustive. Rather it serves to (a) show that the Privid interface is extensible, and (b) illustrate the thought process necessary to develop other extensions.

E.1 ArgMax

Consider a camera overlooking a popular shopping district and an analyst who is interested in which of N stores receives the most visitors at each time of day. In order to express this in the initial scheme, the analyst would be required to submit a separate query for *each* store that counts the number of visitors per chunk of frames and then compare the results manually. Since each query would operate over the same chunks of video, the analyst would need to use up N queries' worth of privacy budget. If the analyst is only interested in the most popular one, this is a waste.

Instead, we can offer an alternate output structure and aggregation function to provide an answer in just one query. First, we allow Q to output N (instead of 1) integers per chunk $o_{i,1},\ldots,o_{i,N}$ (where N is decided by the analyst ahead of time and submitted as part of the request). In this case, the analyst would assign each store to an output position and track the visitors to all of the stores in parallel. At the end of each window, we run the aggregation function independently on each of the N outputs, and then add noise to each aggregate value. At this point we essentially have N independent query results $R_{priv,j}$, whose value each satisfies ρ-DP just as in §4.

$$R_{priv,j} = (\mathsf{Agg}(\{o_{i,j},\ldots,o_{i+w,j}\}) + \eta_j) : 1 < j < N \qquad \text{(E.1)}$$

where each η_j is an independent sample of Laplace noise with the same scale as in the initial scheme. Releasing all of these would use up $N\epsilon$ privacy budget. However, instead we can figure out the maximum aggregate output position from this window $\hat{j}$:

$$\hat{j} = \mathsf{ArgMax}(W,Q) = \underset{1<j<N}{\operatorname{argmax}} R_{priv,j} \tag{E.2}$$

and then release both the output position (i.e., the most popular store) and its corresponding value R_{priv} (i.e., the number of visitors to that store), using ϵ instead of $N\epsilon$. The implementation of ArgMin is identical.

E.2 Counting Unique Objects

The initial Privid scheme does not permit B to maintain state *between* chunks of frames. Since the chunk size must be much smaller than the window size, this precludes queries such as counting all unique objects over an entire window (i.e., it would not be possible to correlate the same object appearing in multiple chunks and thus they would be counted once for each chunk in which they appear).

In order to get a robust estimate of the unique count, we introduce another alternate output structure and aggregation function. We allow B to output a *set* of unique objects O_i (where all objects are of the same fixed-size data type, not necessarily integers) for each chunk. For example, if the query is counting cars it might output a license plate string for each car identified, or if counting known people it might output a unique identifier for each person or an encoding of their face. At the end of each window, we can simply apply a union operator across all of the sets to get the full set of unique objects identified over the course of the window. We can then compute the size of this set and finally apply noise:

$$\mathsf{CountUnique}(W,Q) = |\bigcup_{i \in W} Q(f_i)| + \eta = |\bigcup_{i \in W} O_i| + \eta \tag{E.3}$$

However, the sensitivity is slightly different here. Let $\bar{O}$ be the maximum number of objects output for any one chunk in a window:

$$\bar{O} = max_{i \in W} |O_i|$$

In the worst case, a malicious analyst could output many "unique" objects to correspond to the same individual. For example, for `"A"`, they could output 100 strings `"A1"`,...,`"A100"`.

Now a single individual can show up not only in at most ρ sets, but also as at most $\bar{O}$ "unique" objects in any of those ρ sets:

$$\Delta_{\mathsf{CountUnique}} \leq \bar{O} \cdot \rho \tag{E.4}$$

Thus we must sample η proportional to $\Delta_{\mathsf{CountUnique}}$ rather than ρ alone.

E.3 Set Intersection Size

We can extend the idea of counting unique objects in a single video to counting unique objects across multiple videos, either from different windows of the same camera, or different cameras. This would enable queries such as "how many cars appearing in video A also appeared in video B?" or "how many cars are commuters (use the road every day) as opposed to visitors (use the road once)?".

Consider a query over a set of videos $V_1,\cdot,V_n$. The maximum number of set objects any single object could influence is at most the number of objects they can influence in any one of the videos alone. Thus, the sensitivity of the size of the intersection ($\Delta_{\mathsf{IntersectionSize}}$) is bounded by the maximum sensitivity of any of the individual sets:

$$\Delta_{\mathsf{IntersectionSize}} \leq \max_{0 \leq i \leq n} \Delta_{\mathsf{CountUnique}}(V_i) \tag{E.5}$$