Han Wang, Shangyu Xie, and Yuan Hong*

VideoDP: A Flexible Platform for Video Analytics with Differential Privacy

Abstract: Massive amounts of videos are ubiquitously generated in personal devices and dedicated video recording facilities. Analyzing such data would be extremely beneficial in real world (e.g., urban traffic analysis). However, videos contain considerable sensitive information, such as human faces, identities and activities. Most of the existing video sanitization techniques simply obfuscate the video by detecting and blurring the region of interests (e.g., faces, vehicle plates, locations and timestamps). Unfortunately, privacy leakage in the blurred video cannot be effectively bounded, especially against unknown background knowledge. In this paper, to our best knowledge, we propose the first differentially private video analytics platform (VideoDP) which flexibly supports different video analyses with rigorous privacy guarantee. Given the input video, VideoDP randomly generates a utility-driven private video in which adding or removing any sensitive visual element (e.g., human, and object) does not significantly affect the output video. Then, different video analyses requested by untrusted video analysts can be flexibly performed over the sanitized video with differential privacy. Finally, we conduct experiments on real videos, and the experimental results demonstrate that VideoDP can generate accurate results for video analytics.

Keywords: Differential Privacy, Video Privacy

DOI 10.2478/popets-2020-0073
Received 2020-02-29; revised 2020-06-15; accepted 2020-06-16.

1 Introduction

Massive amounts of video data are ubiquitously generated everyday from many different sources such as personal cameras and smart phones, traffic monitoring and video surveillance facilities, and many other video recording devices. Analyzing such complex, unstructured and voluminous data [51] would be extremely beneficial in real world. For instance, traffic monitoring videos can be analyzed by traffic authorities, urban planning officials, and researchers [6] for learning urban traffic and pedestrian behavior. Videos recorded by surveillance devices might be analyzed for detecting anomalies or suspicious behavior.

However, directly releasing videos to the analysts may result in severe privacy concerns due to the considerable amount of sensitive information involved in videos, such as human faces, objects, identities and activities [5]. For instance, traffic monitoring cameras can capture all the vehicles which may involve the make, model and color of vehicles, moving speed and trajectories, and even the drivers' faces. Most of the existing privacy preserving video sanitization techniques (including the YouTube Blurring application [1]) obfuscate the video by *detecting* and then directly *blurring* the region of interests, e.g., faces, vehicle plates, and locations [28, 49]. Unfortunately, the *privacy leakage* in the blurred videos cannot be effectively bounded, especially against unknown background knowledge. Specifically, such approaches cannot quantify and bound the privacy leakage in the outputs (e.g., limiting the probability of identifying any individual from the sanitized video [19, 36, 52]). Although all the detected sensitive information can be blurred with fully black/white boxes to address the privacy leakage, the sanitized videos may result in very low utility (see Section 6).

To address such deficiency, we propose a novel platform (namely, VideoDP) that ensures differential privacy [19] for any video analysis requested from untrusted data analysts, including queries or query-based analyses over the input video. Notice that, as the state-of-the-art privacy model, differential privacy (DP) [19] can ensure indistinguishable analysis result derived from the input data with and without any single record (protecting any record against arbitrary background knowledge). In VideoDP, we define a novel DP notion in which adding or removing any sensitive visual element (e.g., human or object) into the input video does not significantly affect the analysis result. Thus, the privacy risks

Han Wang: Illinois Institute of Technology, E-mail: hwang185@hawk.iit.edu
Shangyu Xie: Illinois Institute of Technology, E-mail: sxie14@hawk.iit.edu
***Corresponding Author: Yuan Hong:** Illinois Institute of Technology, E-mail: yuan.hong@iit.edu

can be strictly bounded even if the adversaries possess arbitrary background knowledge (e.g., knowing the objects or humans). To our best knowledge, this is the first work proposed to provide DP video analysis. Specifically, in VideoDP, we address the following unique challenges (different from the existing DP schemes applied to other datasets, e.g., [13, 19, 25, 30, 41, 47]).

Differential Privacy. Recall that we consider the "identification of sensitive visual element" as the root cause of privacy leakage in videos, and then seek for the protection that the untrusted analyst cannot *distinguish if any sensitive visual element (e.g., an object or human) is included in the video or not*, even if the adversaries have arbitrary background knowledge about the visual elements. Then, we first address the challenge on accurately detecting and tagging all the sensitive visual elements in any video (by utilizing state-of-the-art computer vision techniques [56]). For instance, given a video recorded on the street, our objective is to protect sensitive objects (e.g., vehicles) and/or humans (e.g., pedestrians) which are pre-specified by the video owner.

Utility-driven Private Video. Given any input video for analysis, different from traditional differentially private mechanisms (e.g., injecting noise into queries or analyses), we propose a novel randomization scheme (via sampling) to generate a utility-driven private video while ensuring the defined differential privacy. Specifically, our VideoDP involves three phases. The first phase randomly samples pixels for the output video based on the visual elements and background scene in the vidoe. Since videos are extremely large scale and highly-dimensional (generally consisting of millions of pixels with very diverse RGBs [9]), it is extremely challenging to ensure good utility for video via pixel sampling (e.g., many RGBs/pixels cannot be sampled).

To further improve the output utility, after executing pixel sampling in VideoDP, the second phase generates a (random) *utility-driven private video* by interpolating the RGB values of unsampled pixels and integrates such "estimated pixels" into the missing pixels. Note that the addition of interpolation into the randomizaiton algorithm still ensures the same indistinguishability (regardless of adding or removing any visual element in the input video) since the interpolation can be considered as a post-processing procedure performed on differentially private outputs [20].

Flexible Video Analytics Platform. In the first two phases, VideoDP generates the (probabilistic) utility-driven private video with differential privacy. Therefore, in the third phase, different video analyses requested by untrusted data analysts (e.g., queries over the video for analytics) can be flexibly performed over the utility-driven private video, as analyzed in Section 4.2. VideoDP significantly outperforms the PINQ platform [40] in the context of video analytics with reduced perturbation and superior flexibility for different video analyses as validated in the experiments (Section 6).

Contributions. The major contributions of this paper are summarized as below:

- To the best of our knowledge, we define the first differential privacy notion with respect to protecting all the sensitive visual elements in any video.
- We propose a novel platform VideoDP which can flexibly perform any video analysis requested by the untrusted video analysts with differential privacy.
- VideoDP randomly generates a utility-driven private video by sampling pixels (Phase I) with differential privacy and interpolating unsampled pixels (Phase II) to boost the utility for video analytics. Then, it enables flexible private video analyses (Phase III) for untrusted analysts.
- We have conducted extensive experiments to validate the performance of VideoDP on real videos.

The remainder of this paper is organized as follows. Section 2 introduces some preliminaries. Section 3 illustrates the first phase of VideoDP and analyzes the privacy guarantee. Section 4 presents the second phase and third phase as well as the differential privacy guarantee. Section 5 discusses some relevant aspects of VideoDP. Section 6 demonstrates the experimental results. Section 7 and 8 present the literature and conclusions.

2 Preliminaries

In this section, we present some preliminaries required in this paper. For simplification of notations, we use "VE" to represent visual elements in this paper.

2.1 Video Processing

Referring to the RGB color model [9], video data includes *frame ID, pixel coordinates, red, green, blue* (we focus on visual information in this paper). Thus, we denote any pixel's frame ID as t, its coordinates as (a,b), and its RGB as a 3-dimensional vector $\theta(a,b,t) \in [0,255]^3$ (*16,581,375 distinct RGBs in the universe*).

VE Detection. The state-of-the-art computer vision algorithms can be utilized to accurately detect VEs (e.g., objects [24] and humans [15]) in videos. Specifically, all the VEs in a video (denoted as $\Upsilon_j, j \in [1, n]$) are detected using the tracking algorithm [56] in which the same human/object in different frames is assigned the same unique identifier (see Section 6 for details). Notice that, each VE from different angles will be considered as the same VE for protection if they can be tracked in multiple frames by the algorithm (in most cases). If they cannot be tracked in multiple frames, they are also protected separately in VideoDP. In addition, the detection/tracking accuracy can be as high as 90%+ on general videos [59] (which can be further improved by integrating multiple algorithms and repeated detection; and the accuracy is close to 100% in our experiments). These make our defined differential privacy (for VEs) strong enough for protecting the entire video.

Notice that different VEs may have different sizes, and the same VE Υ_j may also have different sizes and different RGB values (e.g., as a vehicle moves close to the camera, its size visually grows). Then, VideoDP aims at protecting all the RGBs of different VEs in all the frames. To break down the video into pixels with RGBs, we denote the set of distinct RGBs in VE Υ_j (in all the frames) as Ψ_j where the cardinality is written as $|\Psi_j|$ (the number of distinct RGBs in Υ_j). Table 3 in Appendix A shows the notations in this paper.

2.2 Privacy Model

To protect sensitive VEs in the video, we first consider two input videos V and V' that differ in any visual element Υ (in all the frames) as two neighboring inputs. Specifically, given a video V, after completely removing Υ in all the frames of V, we can obtain V' (or vice-versa). Note that V and V' have identical number of frames and background scene.

Then, VideoDP ensures that adding any VE into any number of frames in a video or completely removing any VE from the video would not result in significant privacy risks in video analytics, assuming that the adversary possesses arbitrary background knowledge on all the VEs. W.l.o.g., denoting $V = V' \cup \Upsilon$, we have:

Definition 1 (ϵ-Differential Privacy)**.** *A randomization algorithm $\mathcal{A}$ satisfies ϵ-differential privacy if for any two input videos V and V' that differ in any visual element (e.g., object or human) Υ, and for any output $O \in range(\mathcal{A})$, we have $e^{-\epsilon} \leq \frac{Pr[\mathcal{A}(V)=O]}{Pr[\mathcal{A}(V')=O]} \leq e^{\epsilon}$.*

Definition 1 protects all the sensitive VEs in the video (which are pre-defined and accurately detected by the video owner, as discussed in Section 2.1). If necessary, any part of the video can be specified as a sensitive VE for protection (including the background scene), as discussed in Section 5 (see the "Background Scene(s) as VE" mode in VideoDP).

Moreover, given two neighboring videos V and V', a possible output $O \in range(\mathcal{A})$ may make any of $Pr[\mathcal{A}(V) = O]$ and $Pr[\mathcal{A}(V') = O]$ equal to 0. For instance, in case that the extra VE Υ is included in V but not in V', an output O involving Υ cannot be generated from V' (simply due to $\Upsilon \cap V' = \emptyset$). At this time, for such output O, we have $Pr[\mathcal{A}(V) = O] > 0$ while $Pr[\mathcal{A}(V') = O] = 0$. In such cases, the multiplicative difference between $\frac{Pr[\mathcal{A}(V)=O]}{Pr[\mathcal{A}(V')=O]}$ and $\frac{Pr[\mathcal{A}(V')=O]}{Pr[\mathcal{A}(V)=O]}$ cannot be bounded by e^{ϵ} (due to the zero denominator). Thus, a relaxed privacy notion [25, 38] can be defined:

Definition 2 ((ϵ, δ)-Differential Privacy [25, 38])**.** *A randomization algorithm $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy if for all video V, we can divide the output space $range(\mathcal{A})$ into two sets Ω_1, Ω_2 such that (1) $Pr[\mathcal{A}(V) \in \Omega_1] \leq \delta$, and (2) for any of V's neighboring video V' and for all $O \in \Omega_2$: (2) $e^{-\epsilon} \leq \frac{Pr[\mathcal{A}(V)=O]}{Pr[\mathcal{A}(V')=O]} \leq e^{\epsilon}$.*

This definition guarantees that algorithm $\mathcal{A}$ achieves ϵ-DP with a high probability ($\geq 1 - \delta$) [25, 38]. The probability that generating the output with unbounded multiplicative difference for V and V' is bounded by δ.

2.3 VideoDP Framework

2.3.1 Limitations of PINQ-based Video Analytics

Privacy Integrated Queries (PINQ) [40] platform was proposed to facilitate data analytics by injecting Laplace noise into the queries. Similarly, PINQ can be simply extended to function video analytics. However, there are two major limitations of PINQ-based video analytics, which greatly limit the usability in practice.

Sensitivity. In PINQ-based video analytics, global sensitivity [19] can be defined for some queries with small sensitivities such as "the count of vehicles in the video" (sensitivity as 1). However, for queries with large sensitivities, the query result would be overly obfuscated (see Section 6). For instance, in the query "the average time each object stays in the video", since an object can stay in the video for the entire video or only 1 second (a few frames), global sensitivity would be too large and diffi-

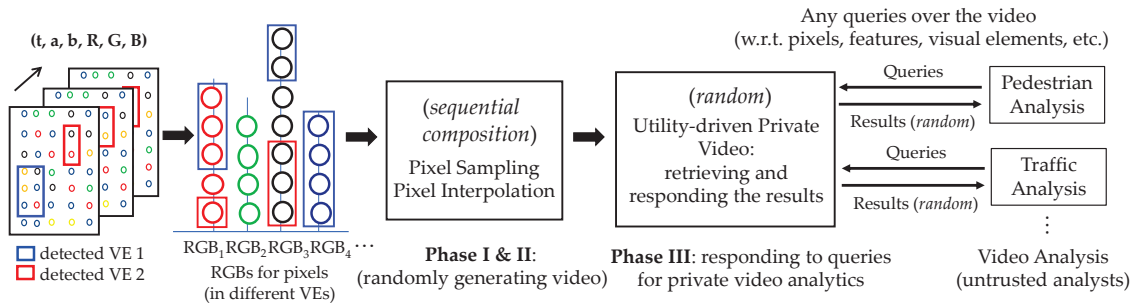


Fig. 1. VideoDP Framework: ϵ-differential privacy for Phase I–III (which can be relaxed to (ϵ, δ)-differential privacy)

cult to define. Meanwhile, it might be also impractical to achieve (smooth) local sensitivity [44] for all different queries in the analysis due to computational overheads.

Flexibility. PINQ is inflexibly adapted for different video analyses. For each requested analysis, a specific DP scheme would be required for improving the utility of the private analysis. The algorithm (e.g., budget allocation, composition of queries [40]) has to be redesigned for any new analysis on the video.

Instead, we propose a novel flexible framework VideoDP for universally optimizing the utility of different video analysis, detailed as follows.

2.3.2 VideoDP for Video Analytics

Figure 1 shows that VideoDP consists of three major phases (after detecting all the sensitive VEs):

1. **Phase I**: video (including detected VEs) can be represented as pixels, which can be grouped by their RGBs (notice that, different from generating RGB histograms, each pixel still keeps its original coordinates and frame ID). Rather than injecting Laplace noise, this phase randomly samples a subset of pixels (with its original features) for each RGB, where privacy budgets are allocated for different RGBs (*sequential composition* [40]) to optimize the output utility. Phase I in VideoDP satisfies ϵ-DP, which can be relaxed to (ϵ, δ)-DP. See details in Section 3.
2. **Phase II**: after sampling all the pixels, the output video has numerous unsampled pixels (due to privacy constraints). This phase estimates the RGBs for unsampled pixels via interpolation. We show that Phase II does not leak any additional information (still ensuring *indistinguishability*). Thus, Phase II can boost the video utility without additional privacy loss. See details in Section 4.
3. **Phase III**: VideoDP applies the requested queries (for video analysis, e.g., traffic and pedestrian analysis [3, 42]) to the *random* utility-driven private video and directly returns the results (which are also random) to untrusted analysts, where differential privacy is also guaranteed (as analyzed in Section 4).

3 Phase I: Pixel Sampling

In this section, we present the sampling algorithms while ensuring differential privacy.

3.1 Pixel Sampling Mechanism

Recall that Section 2.3 has briefly discussed the pixel sampling. Since each sensitive VE involves a set of RGBs and the pixel sampling for distinct RGBs (in all the VEs) is expected to satisfy differential privacy, the privacy budget ϵ will be allocated for the individual pixel sampling w.r.t. distinct RGBs (which follows sequential composition [40]). Specifically, for each RGB θ_i in V, a number of x_i pixels with such RGB (out of the original c_i pixels in V) will be randomly selected (uniform distribution) to output with their original coordinates and frame ID. A privacy budget ϵ_i will be allocated for sampling pixels for RGB θ_i that is used to bound the probabilities for its differential privacy guarantee.

Since every video may involve millions of distinct RGBs, given a privacy budget ϵ for pixel sampling, it is nearly impossible to allocate an equal budget to every unique RGB (each share would be negligible). To address such challenge, we categorize all the RGBs $i \in [1, m], \theta_i$ for pixel sampling in different cases (some of which indeed do not consume any privacy budget) and explore the optimal budget allocation as well as the differential privacy guarantee in Section 3.2.

3.2 Privacy Budget Allocation

As the privacy budget ϵ is specified for pixel sampling, our goal is to optimize the allocated budgets for RGBs towards their count distributions in the original video. Given V and V' where $V = V' \cup \Upsilon$ (w.l.o.g.) and Υ can be any VE, we have three types of RGBs:

- Case (1): RGB $\theta_i \in \Upsilon \setminus V'$ (the RGB is included in the extra visual element Υ but not V').
- Case (2): RGB $\theta_i \in V' \setminus \Upsilon$ (the RGB is included in V' but not the extra visual element Υ).
- Case (3): RGB $\theta_i \in V' \cap \Upsilon$ (the RGB is included in both V' and the extra visual element Υ).

Then, we investigate the budget and the privacy guarantee for these three cases as below.

3.2.1 Case (1): RGB $\theta_i \in \Upsilon \setminus V'$

Pixels in this case is the reason why we need the relax in definition, which we will discuss this in the Section 5. Given x_i as the output count of θ_i and c_i is the input count in V, we let $x_i = 0$ (does not output pixels with such RGB θ_i) since θ_i cannot be found in V', if generating any pixel with RGB θ_i into the output video O (in Phase I). Extending it to an randomization algorithm $\mathcal{A}$ applied to V (with n VEs $\Upsilon_1, \ldots, \Upsilon_n$), w.l.o.g., considering V as the video with an arbitrary extra VE $\Upsilon \in \{\Upsilon_1, \ldots, \Upsilon_n\}$ (compared to V'), we thus have: $\forall j \in [1, n], \Upsilon_j$, *if RGB* $\theta_i \in \Upsilon_j \setminus (V - \Upsilon_j)$, *then* $x_i = 0$ *(do not sample pixels with such RGB).*

3.2.2 Case (2): RGB $\theta_i \in V' \setminus \Upsilon$

Since all the pixels with such RGB θ_i in V and V' are equivalent, we can let $x_i = c_i$ (retaining all the pixels with such RGB θ_i) without violating privacy. Then, for any $x_i > 0$ (can be maximized to c_i), sampling pixels for this RGB θ_i does not consume any privacy budget.

Similarly, extending it to the randomization algorithm $\mathcal{A}$ (applied to V), w.l.o.g., considering V as the video with an arbitrary extra VE $\Upsilon \in \{\Upsilon_1, \ldots, \Upsilon_n\}$ (compared to V'), since VideoDP should protect any arbitrary VE, we thus have: $\forall j \in [1, n], \Upsilon_j$, *if any RGB* $\theta_i \in V' \setminus \Upsilon_j$, *then* $x_i = c_i$ *(retaining all the pixels with such RGB in the utility-driven private video). This does not consume any privacy budget since such RGBs do not exist in any of the VEs.*

3.2.3 Case (3): RGB $\theta_i \in V' \cap \Upsilon$

The pixel sampling for each RGB in this case should satisfy ϵ-differential privacy where $e^{-\epsilon} \leq \frac{Pr[\mathcal{A}(V)=O]}{Pr[\mathcal{A}(V')=O]} \leq e^{\epsilon}$ holds. Thus, we should allocate privacy budgets for different RGBs in this case. However, due to the *sequential composition* [40], we cannot allocate a budget for every RGB in this category (otherwise, given any ϵ, for a large number of distinct RGBs, each share of the budget would be too extremely small). In other words, all the RGBs in this category may have to be suppressed (not sampled in the output video). To improve the output utility, our VideoDP has the following three procedures for budget allocation in pixel sampling (Phase I):

1. Determine the RGBs selection rule (*selecting the most representative RGBs in each VE for generating the utility-driven private video*).
2. Derive an optimal number of distinct RGBs within each VE (*maximizing the utility of the VEs in the utility-driven private video*).
3. Allocate appropriate budgets for selected RGBs (*per their RGB count distribution in the original video*).

1) RGBs Selection Rule. Denoting the number of distinct RGBs in $\Upsilon_j, j \in [1, n]$ (which receive privacy budgets to output after Phase I) as k_j, the remaining RGBs in Υ_j will be suppressed (not sampled) during pixel sampling. Thus, this procedure ensures that the selected k_j RGBs in Υ_j are most representative to reconstruct the object (without compromising privacy).

An intuitive rule is to select the top frequent k_j RGBs in Υ_j. However, it might be biased to specific regions with intensive counts of similar RGBs in a VE. To address such limitation, we adopt the multi-scale analysis [57] in computer vision to partition each VE Υ_j into k_j cells and select the top frequent RGB in each cell to allocate privacy budgets (as the "representative RGBs"). Then, the sampled RGBs can be effective to reconstruct the VE in the utility-driven private video.

2) Optimal k_j in Each VE. This procedure is designed to maximize the utility of the VEs in the utility-driven private video (after bilinear interpolation [17] in Phase II). If the number of distinct RGBs in Υ_j that receive privacy budgets k_j is large, more distinct RGBs can be sampled in the VE but the budget allocated for each RGB would be extremely small; if k_j is small, the budget allocated for each RGB would be large but less distinct RGBs can be sampled. We now seek for the optimal k_j for Υ_j that can minimize the MSE between the interpolated VE (after Phase II) and the original VE.

Specifically, since every pixel in Υ_j can be sampled (with the original RGB) or unsampled (with an estimated RGB), we minimize the expectation of MSE (referring to Equation 2) after the Phase II bilinear interpolation [17]. The expectation of each pixel's RGB is determined by the probabilities of "sampled" (denoted as $Pr(a,b,t)$) and "unsampled but interpolated by its neighboring pixels" (4 neighbors for non-border pixels, 3 neighbors for border-but-not-corner pixels, and 2 neighbors for corner pixels, as shown in Figure 4). Denoting pixel (a,b,t)'s RGB in the output as $\hat{\theta}(a,b,t)$, for simplicity of notations, we denote the RGBs of its neighboring pixels as $\hat{\theta}_N, \hat{\theta}_S, \hat{\theta}_W$ and $\hat{\theta}_E$, for pixels $(a-1,b,t)$, $(a+1,b,t)$, $(a,b-1,t)$ and $(a,b+1,t)$, respectively. For a non-border pixel (4 neighbors), the expectation of its RGB[1] can be derived as:

$$
\begin{aligned}
&E[\hat{\theta}(a,b,t)] = Pr(a,b,t) * \theta(a,b,t) + \sigma_0(a,b,t) * 0 \\
&+\frac{\sigma_1(a,b,t)[1-Pr(a,b,t)][E(\hat{\theta}_N)+E(\hat{\theta}_S)+E(\hat{\theta}_W)+E(\hat{\theta}_E)]}{4} \\
&+\frac{\sigma_2(a,b,t)[1-Pr(a,b,t)][3E(\hat{\theta}_N)+3E(\hat{\theta}_S)+3E(\hat{\theta}_W)+3E(\hat{\theta}_E)]]}{6*2} \\
&+\frac{\sigma_3(a,b,t)[1-Pr(a,b,t)][3E(\hat{\theta}_N)+3E(\hat{\theta}_S)+3E(\hat{\theta}_W)+3E(\hat{\theta}_E)]}{4*3} \\
&+\frac{\sigma_4(a,b,t)[1-Pr(a,b,t)][E(\hat{\theta}_N)+E(\hat{\theta}_S)+E(\hat{\theta}_W)+E(\hat{\theta}_E)]}{4}
\end{aligned} \tag{1}
$$

where $\theta(a,b,t)$ is the original RGB (a constant) and probability of "sampled" $Pr(a,b,t)$ is determined by k_j (given V and k_j, it is deterministic if the RGB selection rule is decided previously). Probabilities $\sigma_0(a,b,t), \sigma_1(a,b,t), \sigma_2(a,b,t), \sigma_3(a,b,t)$ and $\sigma_4(a,b,t)$ are probabilities that pixel (a,b,t) has 0 neighbor, 1 neighbor, 2 neighbors, 3 neighbors and 4 neighbors after sampling (which are also constants if V, k_j and sampling mechanism are determined; note that $\sigma_0(a,b,t)+\cdots+\sigma_4(a,b,t) = 1$). In the equation, $E[\hat{\theta}_N]$, $E[\hat{\theta}_S]$, $E[\hat{\theta}_W]$ and $E[\hat{\theta}_E]$ are the RGB expectation of its four neighbors in the same tth frame (*Equation 1 presents the relation among the RGB expectations of the five pixels*, which are detailed in Appendix B.1). Similarly, we can obtain two other equations for pixels with special coordinates (border-but-not-corner or corner pixels of the frame, please see Equation 8 and 9 in Appendix B.1).

Thus, for each pixel in VE Υ_j (in all the frames), there exists exactly one equation out of three cases in Equation 1, 8 and 9 (latter two are in Appendix B.1). As k_j is determined, $\theta(a,b,t)$ and $Pr(a,b,t)$ are constants, then we can solve all the equations to obtain $\forall(a,b,t) \in \Upsilon_j, E[\hat{\theta}(a,b,t)]$. Thus, each k_j value corresponds to the solved $\forall(a,b,t) \in \Upsilon_j, E[\hat{\theta}(a,b,t)]$, and then we can efficiently derive the optimal k_j for Υ_j as:

$$
\arg\min_{k_j} \frac{1}{|\Upsilon_j|} \sum_{\forall(a,b,t)\in\Upsilon_j} \left(E[\theta(a,b,t)] - E[\hat{\theta}(a,b,t)]\right)^2 \tag{2}
$$

where $|\Upsilon_j|$ denotes the total number of pixels in Υ_j. Solving the above problem requires complexity $O(n^3\log(n))$, which is much faster than executing pixel sampling for all the possible k_j and then comparing all the MSE results to get the optimal k_j (*since iteratively sampling all the pixels is expensive*). Details of the solver is given in Appendix B.2. Notice that,

- **Range for k_j.** The optimal k_j is derived from a specified range of k_j. It is unnecessary to traverse k_j to a extremely large number (otherwise, the allocated budget for each RGB would be extremely small). The larger k_j, more diverse RGBs can be allocated with a privacy budget; the smaller k_j, each RGB will be allocated with a larger privacy budget. Thus, the lower/upper bounds for k_j can be selected according the requested diversity of RGBs in the visual elements in practice ($k_j \leq 20$ can give good utility in our experiments).
- **Approximation.** As discussed before, since Υ_j in different frames may have different sizes and different sets of RGBs (though the difference can be minor), the most accurate k_j can be obtained by solving the equations for all the pixels of Υ_j in all the frames (*with complexity $O(n^3\log(n))$*, as proven in Appendix B). If more efficient solvers are desirable, we can randomly select a frame (including Υ_j) to solve the equations to obtain an approximated k_j for Υ_j by assuming the VE does not change much in the video. Another alternative solution is to solve the optimal k_j for each frame and average them (which is more efficient but less accurate).

Thus, we can repeat the above procedure for all the VEs such that the optimal $k_j, j \in [1,n]$ can be obtained to minimize the MSE of the VEs in the output video.

3) Budget Allocation. As the optimal k_j for each visual element $\Upsilon_j, j \in [1,n]$ is derived, we denote the set of RGBs in $\Upsilon_j, j \in [1,n]$ to allocate budgets as Ψ_j with the cardinality $|\Psi_j| = k_j$. Then, we have the total number of RGBs to sample in V (Case (3)) as the cardinality $|\Psi|$ of the union $\Psi = \bigcup_{j=1}^{n} \Psi_j$. We then present

1 Although the RGBs of all the pixels in Υ_j are random (due to the sampling in Phase I), the expectations of RGBs for its neighboring pixels in Υ_j always satisfy a condition (ensured by bilinear interpolation [17]), e.g., Equation 1.

how to allocate privacy budget ϵ in Phase I for $|\Psi|$ different RGBs. The criterion for allocating budget is to allocate the privacy budgets based on the count distributions of RGBs in different VEs while fully utilizing the privacy budget ϵ. For each VE Υ_j, all the RGBs in Ψ_j can fully enjoy the budget ϵ (since Ψ_j includes all the RGBs that could generate visual element Υ_j in all the frames, and other RGBs would not be sampled into the visual element Υ_j).[2]

Then, we denote the ith RGB in Ψ_j as $\widetilde{\theta}_{ij}$ where $i \in [1, k_j]$, and the count of $\widetilde{\theta}_{ij}$ in Υ_j as $d_j(\widetilde{\theta}_{ij})$ and the overall pixel count in Υ_j (in all frames) as d_j. Apparently, we can allocate $\frac{d_j(\widetilde{\theta}_{ij})\epsilon}{d_j}$ to RGB $\widetilde{\theta}_j(i), i \in [1, k_j]$ and apply this criterion to all the VEs. However, if any RGB $\widetilde{\theta}$ is included in multiple VEs (the intersections among the sets $\forall j \in [1, n], \Psi_j$), $\widetilde{\theta}$ will receive privacy budgets from different VEs (and should satisfy differential privacy for all of them). At this time, its budget should be allocated as the *minimum* one out of all (otherwise, not all the VEs in pixel sampling can be protected with ϵ-differential privacy since the budget for some VEs may exceed ϵ).

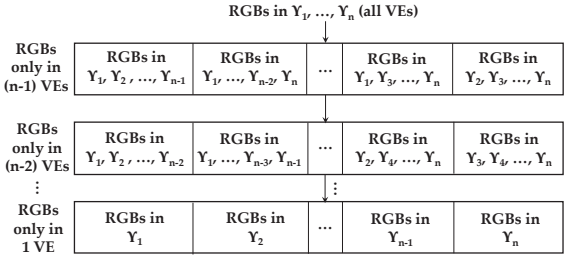


Fig. 2. Prioritizing RGBs (for allocating budgets)

Nevertheless, if the minimum budget is adopted as above, some VEs cannot fully enjoy ϵ (the gap between $\widetilde{\theta}$'s original budget in a specific VE and its minimum budget among all the VEs would be wasted). To fully utilize the privacy budgets, we propose a *budget allocation algorithm* for all the $|\Psi|$ distinct RGBs by *prioritizing* them in the RGB set $\Psi = \bigcup_{j=1}^{n} \Psi_j$.

Specifically, we prioritize $|\Psi|$ different RGBs into n disjoint partitions: as shown in Figure 2 (from top to down), RGBs in the first partition are included in all the VEs, RGBs in the second partition are included in $(n-1)$ VEs, ..., RGBs in the nth partition are only included in a single VE. Then, our algorithm iteratively allocates budgets for RGBs in n partitions (*allocating budgets for all the RGBs in a partition in each iteration*).

Since all the RGBs within each VE follow *sequential composition* [40], after allocating the budgets for all the RGBs in the ℓth partition, the allocation in the $(\ell+1)$th partition will be based on the remaining budget for every VE. In the ℓth iteration (for the ℓth partition), the budget for each RGB $\widetilde{\theta}$ is allocated based on its count distribution out of the remaining RGBs in each of the $(n-\ell+1)$ VEs (which include $\widetilde{\theta}$). Then, the *minimum* budget derived from all the VEs is allocated to $\widetilde{\theta}$.

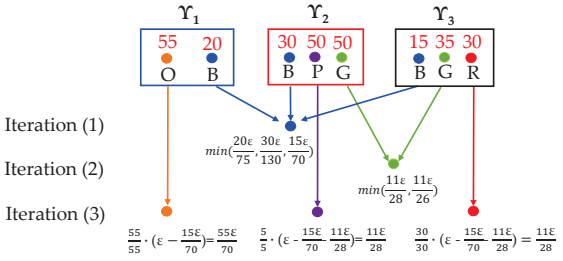


Fig. 3. Example of Budget Allocation

Example 1. Figure 3 shows three VEs (Υ_1, Υ_2 and Υ_3). Blue exists in all the VEs $\Upsilon_1, \Upsilon_2, \Upsilon_3$ with counts $20, 30, 15$. Green exists in Υ_2 and Υ_3 with counts 50 and 35. All the remaining RGBs only exist in only one VE (and the non-VE part of the video): Orange in Υ_1 with count 55, Purple in Υ_2 with count 5, and Red in Υ_3 with count 30. Thus, five RGBs are prioritized to (three partitions):{B}, {G}, and {O, P, R}.

In the 1st iteration (partition), Blue is first allocated with a privacy budget as the $\min\{\frac{20\epsilon}{75}, \frac{30\epsilon}{130}, \frac{15\epsilon}{70}\}$ (the minimum budget from three different VEs). The remaining budget for all the VEs is $\frac{55\epsilon}{70}$. In the 2nd iteration, Green is allocated with a privacy budget $\frac{55\epsilon}{70} \cdot \min\{\frac{50}{100}, \frac{35}{65}\} = \min\{\frac{11\epsilon}{28}, \frac{11\epsilon}{26}\}$. In the 3rd iteration, Orange is allocated with budget $\frac{55}{55} \cdot (\epsilon - \frac{15\epsilon}{70}) = \frac{55\epsilon}{70}$, Purple is allocated with budget $\frac{5}{5} \cdot (\epsilon - \frac{15\epsilon}{70} - \frac{11\epsilon}{28}) = \frac{11\epsilon}{28}$, and Red is allocated with budget $\frac{30}{30} \cdot (\epsilon - \frac{15\epsilon}{70} - \frac{11\epsilon}{28}) = \frac{11\epsilon}{28}$.

Since almost all the VEs have RGBs in the last partition (every VE in real videos include numerous RGBs that are not included in other VEs), the budget can be fully allocated for all the RGBs. Thus, *the budget sum of all the RGBs* in any VE equals ϵ, and the size of VEs does not result in additional leakage. Algorithm 2 in Appendix C presents the details of budget allocation.

2 Any two VEs do not share pixels in the video since the front VE blocks a part of the back VE if they overlap in any frame. In such complex scenario, both VEs can be accurately detected in our experiments. The front VE includes all the pixels while the back VE will be all the pixels that cameras can capture.

3.3 Pixel Sampling Algorithm

To illustrate the algorithm for Phase I, we again discuss the pixel sampling for three different cases of RGBs.

Recall that in Case (1), for all the RGBs $\theta_i \in \Upsilon \setminus V'$, all the pixels with such RGBs will not be sampled (ensuring that $\delta = 0$). In Case (2), for all the RGBs $\theta_i \in V' \setminus \Upsilon$, all the pixels with such RGBs will be sampled (with the original coordinates and frame). Sampling pixels for all the RGBs in Case (2) satisfy 0-DP.

In Case (3), for all the RGBs $\theta_i \in V' \cap \Upsilon$, as discussed in Section 3.2, we sample pixels for $|\Psi|$ distinct RGBs where $|\Psi| \leq \sum_{j=1}^{n} k_j$ (since different VEs may have common RGBs). We denote the set $\Psi = \bigcup_{j=1}^{n} \Psi_j = \{\widetilde{\theta}_1, \ldots, \widetilde{\theta}_{|\Psi|}\}$ (the set of RGBs which request privacy budgets), and its set of budgets $\{\epsilon(\widetilde{\theta}_1), \ldots, \epsilon(\widetilde{\theta}_{|\Psi|})\}$. It is straightforward to show the *sequential composition* [40] of allocated privacy budgets (by Algorithm 2) for all the RGBs:

$$\sum_{\forall \widetilde{\theta}_i \in \Psi_j} \epsilon(\widetilde{\theta}_i) = \epsilon \tag{3}$$

where $\widetilde{\theta}_i$ is denoted as the ith RGB in Ψ. Then, for any V and V' differing in an arbitrary VE $\Upsilon_j, j \in [1, n]$,

$$\forall \widetilde{\theta}_i \in \Psi_j, e^{-\epsilon(\widetilde{\theta}_i)} \leq \frac{Pr[\mathcal{A}(V(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)]}{Pr[\mathcal{A}(V'(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)]} \leq e^{\epsilon(\widetilde{\theta}_i)} \tag{4}$$

where $V(\widetilde{\theta}_i)$ and $V'(\widetilde{\theta}_i)$ are the pixels with RGB $\widetilde{\theta}_i$ in V and V'. Deriving the probability for randomly picking $\widetilde{x}_i$ out of $\widetilde{c}_i$ pixels with RGB $\widetilde{\theta}_i$ (pixel sampling using input V and V', differing in Υ_j), we have:

$$\forall i \in [1, |\Psi|],\ Pr[\mathcal{A}(V(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)] = 1 / \binom{\widetilde{c}_i}{\widetilde{x}_i}$$

$$Pr[\mathcal{A}(V'(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)] = 1 / \binom{\widetilde{c}_i - \widetilde{c}_i^j}{\widetilde{x}_i}$$

$$\Longrightarrow e^{-\epsilon(\widetilde{\theta}_i)} \leq \binom{\widetilde{c}_i}{\widetilde{x}_i} \Big/ \binom{\widetilde{c}_i - \widetilde{c}_i^j}{\widetilde{x}_i} \leq e^{\epsilon(\widetilde{\theta}_i)} \tag{5}$$

where $\widetilde{c}_i$ and $\widetilde{x}_i$ are the input and output counts of RGB $\widetilde{\theta}_i$ while $\widetilde{c}_i^j$ denotes the count of $\widetilde{\theta}_i$ in VE Υ_j.

Thus, we can derive a maximum output count for sampling pixels for each RGB $\widetilde{\theta}_i, i \in [1, |\Psi|]$ and the maximum $\widetilde{x}_i$ can be efficiently computed as below (the only variable): $\forall i \in [1, |\Psi|]$,

$$\max\{\widetilde{x}_i | \forall j \in [1, n], \binom{\widetilde{c}_i}{\widetilde{x}_i} \Big/ \binom{\widetilde{c}_i - \widetilde{c}_i^j}{\widetilde{x}_i} \leq e^{\epsilon(\widetilde{\theta}_i)}\} \tag{6}$$

The maximum output count of the ith RGB $\widetilde{x}_i, i \in [1, |\Psi|]$ can be efficiently computed from Equation 6 (e.g., via binary search) since the left-side of the inequality is monotonic on $\widetilde{x}_i$. To sum up, Algorihtm 1 presents the details of Phase I.

Input : input video V, privacy budget ϵ
Output: sampled video O (pixels)
1 detect all the visual elements $(\Upsilon_1, \ldots, \Upsilon_n)$ in V
// Case (1)
2 **foreach** $\Upsilon_j, j \in [1, n]$ **do**
3 **foreach** *RGB* $\theta_i \in \Upsilon_j$ *but* $\notin (V \setminus \Upsilon_j)$ **do**
4 suppress all c_i pixels with RGB θ_i in V
// Case (2)
5 **foreach** *RGB* $\theta_i \in V \setminus \bigcup_{j=1}^{n} \Upsilon_j$ **do**
6 output all c_i pixels with RGB θ_i in V (original coordinates and frame)
// Case (3)
7 **foreach** $\Upsilon_j, j \in [1, n]$ **do**
8 compute the optimal number of distinct RGBs to sample in Υ_j (minimum expectation of MSE): k_j
9 execute Algorithm 2 (in Appendix C) to allocate budgets for all the RGBs in $\Psi = \{\widetilde{\theta}_1, \ldots, \widetilde{\theta}_{|\Psi|}\}$
10 **foreach** $\widetilde{\theta}_i, i \in [1, |\Psi|]$ **do**
11 compute the maximum $\widetilde{x}_i$:
$\max\{\widetilde{x}_i | \forall j \in [1, n], \binom{\widetilde{c}_i}{\widetilde{x}_i} \Big/ \binom{\widetilde{c}_i - \widetilde{c}_i^j}{\widetilde{x}_i} \leq e^{\epsilon(\widetilde{\theta}_i)}\}$
12 randomly pick $\widetilde{x}_i$ pixels with RGB $\widetilde{\theta}_i$ in V to output (original coordinates and frame)

Algorithm 1: Pixel Sampling (ϵ-DP)

Theorem 1. *The pixels sampling in VideoDP (Phase I) satisfies ϵ-differential privacy.*

Proof. We can prove the differential privacy guarantee for three cases of pixel sampling in the algorithm.

In Case (1), since all the pixels with such RGBs are suppressed, $\delta = 0$ always holds with Line 2-4 in Algorithm 1. In Case (2), since $\forall \theta_i$, $\frac{Pr[\mathcal{A}(V(\theta_i)) = O(\theta_i)]}{Pr[\mathcal{A}(V'(\theta_i)) = O(\theta_i)]}$ always equals 1, Line 5-6 in Algorithm 1 does not result in privacy loss. In Line 7-12 of the algorithm (Case (3)), we have $\forall i \in [1, |\Psi|]$, $e^{-\epsilon(\widetilde{\theta}_i)} \leq \frac{Pr[\mathcal{A}(V(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)]}{Pr[\mathcal{A}(V'(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)]} \leq e^{\epsilon(\widetilde{\theta}_i)}$ holds. Per the *sequential composition* of differential privacy [40], for all V and V' differing in any VE $\Upsilon_j, j \in [1, n]$, we have:

$$\prod_{\forall \widetilde{\theta}_i \in \Psi_j} \frac{Pr[\mathcal{A}(V(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)]}{Pr[\mathcal{A}(V'(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)]} \leq exp[\sum_{\forall \widetilde{\theta}_i \in \Psi_j} \epsilon(\widetilde{\theta}_i)]$$

$$\prod_{\forall \widetilde{\theta}_i \in \Psi_j} \frac{Pr[\mathcal{A}(V(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)]}{Pr[\mathcal{A}(V'(\widetilde{\theta}_i)) = O(\widetilde{\theta}_i)]} \geq exp[-\sum_{\forall \widetilde{\theta}_i \in \Psi_j} \epsilon(\widetilde{\theta}_i)]$$

$$\Longrightarrow e^{-\epsilon} \leq \frac{Pr[\mathcal{A}(V) = O]}{Pr[\mathcal{A}(V') = O]} \leq e^{\epsilon} \tag{7}$$

Thus, this completes the proof. □

Note that composing the sampled pixels would not result in additional leakage. First, composing pixels with the same RGB is done within each individual sampling (that satisfies differential privacy with the allocated budget for the RGB). Second, composing pixels with different RGBs follows sequential composition. Thus, the sum of the allocated budgets would be the privacy bound (total leakage), and there is no additional leakage. Furthermore, in case of $V' = V \cup \Upsilon$, adding an arbitrary VE Υ to V to generate V'. Similarly, for all $\widetilde{\theta}_i$, $\widetilde{x}_i$ can also be derived from V' and V to ensure differential privacy for pixel sampling.

4 Phase II and Phase III

After sampling pixels in Phase I, the suppressed pixels in Case (1) and unsampled pixels in Case (3) do not have any RGB (see Figure 4). Then, Phase II generates the utility-driven private video by estimating the RGBs for the missing pixels, and Phase III responds to the queries (over the private video) for video analysis.

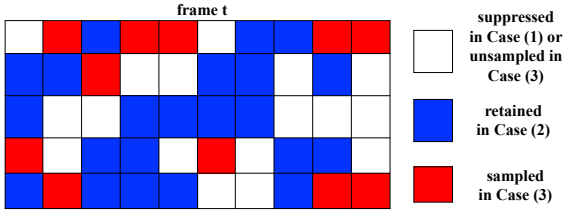


Fig. 4. Pixels after Sampling (Phase I)

4.1 Phase II: Video Generation

For all the coordinates with a RGB value after sampling, the RGBs of such pixels can be estimated using bilinear interpolation [17]. As discussed in Section 3.2, the allocated privacy budgets have been shown to optimize the utility of both sampling and bilinear interpolation, e.g., the optimal number of RGBs selected in each VE for sampling k_j tends to minimize the expectation of MSE between the utility-driven private video (after interpolation) and the original video. Thus, Phase II can directly apply bilinear interpolation. For simplicity of notations, we consider both retained pixels and sampled pixels as "sampled pixels", and both suppressed pixels and unsampled pixels as "unsampled pixels". Specifically,

First, in the output video of Phase I, pixels (not on the border) have at most 4 neighbors in each frame; the pixels on the border of each frame (not corner) have at most 3 neighbors; the pixels at the corner of each frame have at most 2 neighbors. Second, the algorithm interpolates pixels in visual elements and the remaining pixels (background), separately. For each interpolation, it traverses all the unsampled pixels in all the frames (e.g., a specific visual element). If any unsampled pixel has any sampled neighbor(s), the RGB for current unsampled pixel is estimated as the *mean* of all its sampled neighbors. Third, if any unsampled pixel's all the neighbors are also unsampled, the algorithm skips such unsampled pixel in the current traversal. The algorithm iteratively traverses all the skipped unsampled pixels. The algorithm terminates until every unsampled pixel is assigned with an interpolated RGB. In our experiments, the interpolation terminates very quickly since the RGB of any pixel can be readily estimated as long as it has at least one neighbor which is sampled or previously interpolated. Finally, if any VE does not have a sampled pixel in any frame, the interpolation of the pixels for the visual element in such frame will be executed with the remaining pixels (background) $V \setminus \bigcup_{j=1}^{n} \Upsilon_j$.

4.2 Video Analytics and Privacy Analysis

Similar to the framework of PINQ for data analytics [40], VideoDP can also function most of the analyses performed on videos. If breaking down any video analysis into queries, VideoDP (Phase III) directly applies the queries to the *utility-driven private video* (which is randomly generated in Phase I and II) and return the results to untrusted analysts. For any query created at the pixel, feature or visual element level [12, 16, 29], VideoDP (Phase III) could efficiently respond the results with differential privacy guarantee.

Theorem 2. *VideoDP satisfies ϵ-differential privacy.*

Proof. Recall that we have proven Phase I satisfies ϵ-differential privacy in Theorem 1. We now prove that Phase II and III do not result in additional privacy risks.

Since Phase I in VideoDP satisfies ϵ-DP, for any pair of neighboring videos V and V', we have $e^{-\epsilon} \leq \frac{Pr[\mathcal{A}(V)=O]}{Pr[\mathcal{A}(V)=O]} \leq e^{\epsilon}$. Such differential privacy satisfies ϵ-probabilistic differential privacy [25, 38], which also satisfies ϵ-indistinguishability differential privacy [18, 19] (bounding $Pr[\mathcal{A}(V) \in S]$ and $Pr[\mathcal{A}(V') \in S]$ where S is any set of possible outputs), as proven in [25, 38].

Then, after applying VideoDP to inputs V and V', the outputs of Phase I are ϵ-indistinguishable. Since the

pixel interpolation (Phase II) and video queries/analysis (Phase III) are deterministic procedures applied to the output of Phase I (which can be considered as *post-processing* differentially private results), the output $\mathbb{O}$ of Phase II and the analysis/query results of Phase III derived from V and V' are also ϵ-indistinguishable ("*Differential privacy is immune to post-processing*" was proven in [20]). Thus, VideoDP also satisfies ϵ-DP. □

The procedures and privacy guarantee in VideoDP can be interpreted as follows. Given any two videos V and V' that differ in any VE (e.g., a pedestrian), while applying a randomization algorithm (i.e., Phase I-III in VideoDP) to V and V', respectively, the possible outputs of sampling/obfuscating pixels (and post-processing) from V and V' are guaranteed to be indistinguishable. Then, the adversaries cannot identify if any VE (e.g., the pedestrian) is included in the input video or not (since "including" or "not including" such VE does not result in significant difference in the output). Such protection applies to any VE for any two neighboring videos V and V' (differing in a VE). Thus, all the sensitive VEs in any video can be protected by the randomization (obfuscating the pixels in the video).

In the meanwhile, the utility-driven private video can maintain good utility to allow useful computer vision algorithms to execute for the following reasons. First, the sampling randomly generates a subset of pixels with the original coordinates and RGBs in the output video, which are utilized for interpolating a video frame by frame. Second, the pixels in the background scene but not in the sensitive VEs are retained in the output video. Third, privacy budgets are allocated for different RGBs to maximally preserve the utility in the output. For instance, VideoDP only allocates budgets for the most representative RGBs in the VEs (given the privacy bound). Then, computer vision algorithms may still recognize some objects (*but not the specific objects due to uncertainty*) from the features extracted from the retained pixels and interpolated pixels.

5 Discussion

Relaxed Differential Privacy. Theorem 2 ensures that the analysis satisfies ϵ-DP. Thus, similar to PINQ [40], all the aggregation-based queries (w.r.t. more than one VE) could be protected with ϵ-DP in VideoDP. However, if querying on a specific VE (e.g., a unique red car or license plate) which is not included in one of the two neighboring videos, the protection requires another privacy bound δ to ensure (ϵ, δ)-DP (Definition 2). Such additional privacy bound is also required in other contexts (e.g., [25]). We will leave it for the future work.

Background Scene(s) as VE(s). If necessary, any part of the video can be specified as a sensitive VE for protection, including the background scene(s). In VideoDP, the failure of detection/tracking algorithms may occur (though the state-of-the-art techniques could minimize such risks [27]). To avoid such risks, we can consider the background scene as a VE (the "Background Scene(s) as VE" mode in VideoDP). Specifically, in Phase I, the same sampling algorithm will be applied to all the VEs by adding background VEs to Case (1), (2) and (3). Unique RGBs in background VE(s) will be suppressed (same as other VEs), and budgets are also allocated for non-unique RGBs in the background VE(s) using the same Algorithm 1 (same as other VEs). In Phase II and III, the same bilinear interpolation and queries are applied. Thus, DP can be ensured for all the pixels in the video. We have experimentally evaluated the performance of such strong protection in Section 6.

Defense against Correlations. Videos include a large number of sequential frames, if protecting specific VEs in only one frame, the correlations in sequential frames may also leak information to adversaries [11, 50]. Our VideoDP can address such vulnerabilities since all the VEs in all the frames are protected using our privacy notion – adding or removing any VE in any number of frames would not result in significant risks. From this perspective, the privacy notion is defined for the entire period of the video, rather than a specific time. Thus, possible privacy leakage resulted from correlations among multiple frames can be tackled.

System Usability. Similar to many smartphone Apps with face/object detection, the detection/tracking algorithms for different types of VEs can be simply integrated into VideoDP (in the preprocessing), and upgraded with newer algorithms when necessary. Thus, both the video owner and the video analyst are not required to be experts of computer vision. The video owner only needs to specify that what types of visual elements (e.g., humans) should be protected. Then, the pre-processed video can be sampled and interpolated (Phase I and II). After that, the utility-driven private video will be generated and stored by the trusted server for external video analyses (Phase III). The video analysts only need to submit the video name and query (e.g.,<*VEH, total vehicle#*>), which may include additional parameters. The trusted server will respond the query result with differential privacy.

6 Experiments

In VideoDP, we implement the VE detection/tracking algorithm in [56] throughout the entire video. It first detects all the VEs in each frame, and then utilizes the tensorflow training database to tag all the humans/objects, which are considered as sensitive VEs. Each detected VE can be tracked with the same ID if their overlap in multiple frames has exceeded a threshold. This method ensures a high detection/tracking accuracy [56]. We conduct our experiments on three video datasets in which different VEs (with different sizes) are protected. Table 1 shows the characteristics of the videos.

Table 1. Characteristics of Experimental Datasets

Datasets	Avg. Resolution	Video #	Avg. Frame #
MOT	1920×1080	**15**	**846**
UAD	740×480	**24**	**180**
BVD	2464×2056	**5**	**1200**

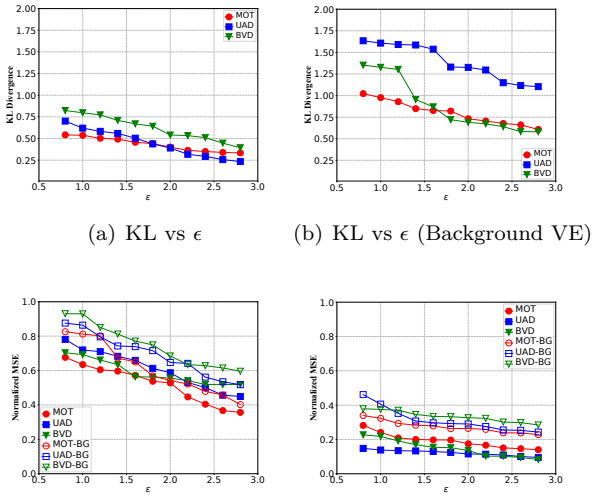
1. MOT [42]: 15 videos with different scenes. Sensitive VEs in these videos are pedestrians and vehicles. We denote this dataset as "MOT".
2. UAD [10]: the UCSD anomaly detection dataset includes crowded pedestrians as sensitive VEs. 24 different videos are captured at 2 different scenes. We denote this dataset as "UAD".
3. BVD [2, 3]: the Boxy [2] vehicle detection dataset includes over 200,000 sequential images at 5 different scenes such as sunny, rainy, and nighttime drive. We take such sequential images (as videos) and a "highway" video [3]. We denote them as "BVD".

All the programs were implemented in Python 3.6.4 with OpenCV 3.4.0 library [4] and tested on an HP PC with Intel Core i7-7700 CPU 3.60GHz and 32G RAM.

6.1 Evaluating Utility-driven Private Video

Pixel Level Evaluation. We consider the RGB color model [9] by breaking down the videos into pixels with RGBs at different coordinates (a, b) and frame t, and then measure the differences between input V and output O. Specifically, we evaluate two types of utility: (1) the difference between the count distributions of all the RGBs in V and O, and (2) the difference between RGB values of all the pixels in V and O.

First, considering the distributions of all the RGB counts $\forall c_i$ and $\forall x_i$ in the input/output, we can measure the utility loss using their KL divergence. If the distribution of RGBs lie closes in the input and output, the performance of pixel interpolation (estimating RGBs for unsampled pixels based on the RGBs of sampled pixels) can be greatly improved [17]. For other measures, e.g., L_1 norm, the output counts of different RGBs might be biased towards certain RGBs with high counts such that the interpolated RGBs might be significantly deviated.



(a) KL vs ϵ (b) KL vs ϵ (Background VE)

(c) MSE vs ϵ (after Phase I) (d) MSE vs ϵ (after Phase II)

Fig. 5. Pixel Level Utility Evaluation (three video datasets MOT, UAD and BVD) – BG refers to "Background Scene(s) as VE(s)"

Second, after interpolating all the pixels in the Phase II of VideoDP, we measure the difference between all the pixel RGBs in V and O using the expectation of mean squared error (MSE). The 3-dimensional RGBs are generally converted to gray for measuring the MSE [32], which are normalized to values in $[0, 1]$.

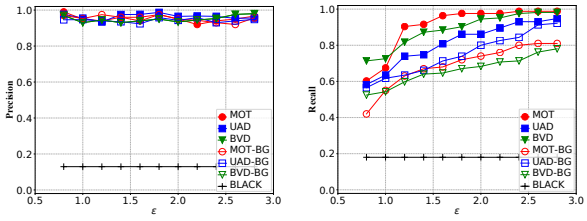
Note that we will demonstrate the average of KL-divergence and MSE values in different datasets, each of which includes multiple videos. Specifically, we conducted two groups of experiments to test how ϵ influences the utility (ϵ=0.8,..., 2.8). As discussed earlier, if necessary, VideoDP can define any part of the video including the background scenes (pixel-level protection) as sensitive VEs. Then, we conduct experiments for both cases (background scene(s) as sensitive VE(s) or not). Figure 5(a) and 5(b) present the KL divergence values for two cases, respectively. In all the datasets, the results monotonically decrease while ϵ increases, and the results of "background scene(s) as VE(s)" are larger than "background scene(s) not as VE(s)" since more pixels can be preserved within background in the latter case.

In addition, we also evaluated the MSE of the output videos (after Phase I, and after Phase II). Figure 5(c) and 5(d) show that the MSE (of the entire video) declines as ϵ increases. This matches the fact that larger ϵ (with weaker privacy protection) trades off less utility. Also, the MSE has been greatly reduced after Phase II (comparing the results in Figure 5(c) and 5(d)), which greatly improves the query accuracy for video analyses. We can also observe that the MSEs of "background scene(s) as VE(s)" are larger since pixels in the background scenes are sampled rather than fully retained.

Video Utility Evaluation. Detection and tracking accuracy (e.g., precision and recall) is an important measure for utility evaluation. Considering the results obtained from three original video datasets (MOT, UAD and BVD) as the benchmarks, we test the *precision* and *recall* of detecting and tracking VEs in different outputs. Precision returns the percent of true VEs out of all the detected/tracked results in the videos. Recall returns the percent of detected/tracked true VEs out of all the true VEs (the benchmarking results).

Specifically, we compare VideoDP with the method of blacking the detected VEs in the entire video (denoted as "Black") in which the contours of VEs are detected and pixels within the contours are assigned the black RGB ("000000"). Since the classifiers in common detection algorithms (e.g., HOG [15], SIFT [46] and CNN [27]) primarily rely on the features rather than the contours, the detection accuracy is quite close to 0. Then, we use the recent contour detection algorithm [58] in the experiments instead, which can maintain a relatively good detection accuracy (i.e., around 80%). However, the accuracy of tracking black contours across multiple frames is still quite low (less than 20% of precision and recall in all the three video datasets MOT, UAD and BVD) since the tracking algorithm cannot distinguish the VEs in multiple frames (which are similar contours with black pixels). Figure 6 demonstrates the precision and recall on a varying privacy budget ϵ (vs the low accuracy of "Black"). The precision can always be high (close to 1), and the recall grows quickly as ϵ increases (since a larger ϵ can generate more accurate random videos for analysis).

Based on the detection/tracking, we also empirically evaluate the utility of queries over the VEs by benchmarking with Black and PINQ [40] in which the sensitivity might be extremely large (e.g., queries involving the frames). The example queries are set as "the number of frames with more than 15 pedestrians in each video of MOT, 10 pedestrians in each video of UAD, and 10 vehicles in each video of BVD, respectively" where the results are averaged in each video dataset (similar performance can be derived from other similar queries).



(a) Precision vs ϵ (b) Recall vs ϵ

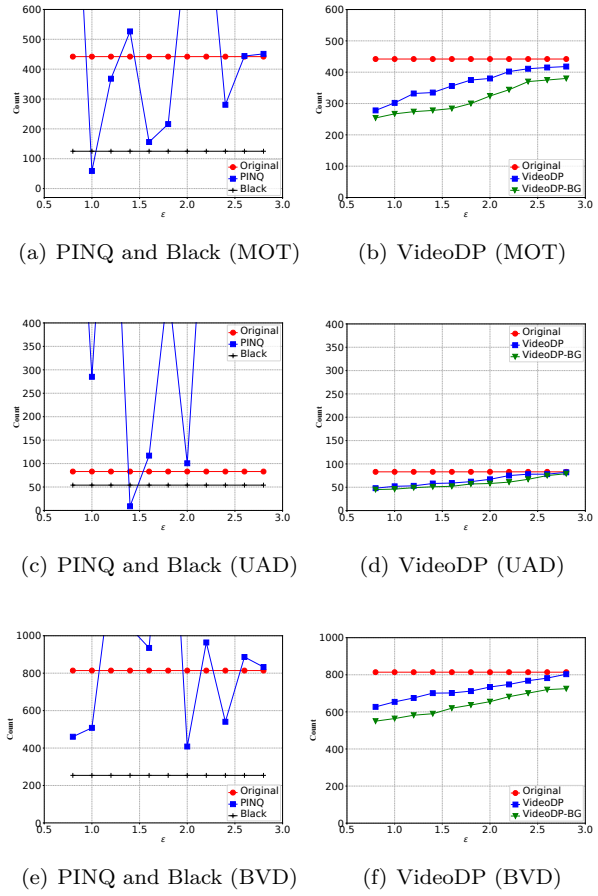
Fig. 6. Visual Elements Detection and Tracking

Figure 7 demonstrates the average counts of frames with 15+ pedestrians in the MOT videos, 10+ vehicles in the BVD videos, and 10+ pedestrians in the UAD videos, including the PINQ results, Black results, VideoDP results and original results, respectively (different privacy budgets ϵ for PINQ and VideoDP). We can observe that VideoDP returns more accurate results (also random) than PINQ, and more accurate results than Black in general (only except the very small ϵ cases). Also, in the Black results, the accuracy of counting the contours is highly reduced in the videos in which VEs frequently overlap or there are more than one type of VEs (e.g., both pedestrians and vehicles are included in some videos in the MOT dataset).

6.2 Case Study: Video Queries/Analysis

The videos randomly generated in VideoDP can function a wide variety of analyses (aggregation-based queries), such as head counting, crowd density and traffic flow analysis [3, 26, 42]. We empirically evaluate some representative queries for such analysis by benchmarking with the PINQ platform [40] in specific videos (e.g., results in different frame) since different VEs cannot be accurately tracked by the Black method these applications. We choose three empirical videos ("MOT16-04" "MOT16-14" and "highway"[3]) from the MOT and BVD datasets, with pedestrians, vehicles, and both. Then, the three videos are denoted as "PED" "VEH" and "PV", respectively. Note that all the queries satisfy ϵ-DP in the following case study.

(1) VE Stay Time (Large Sensitivity). Besides the queries on counting, VideoDP can also privately return query results based on detected/tracked VEs in different applications. For instance, a query returns "how long



(a) PINQ and Black (MOT) (b) VideoDP (MOT)

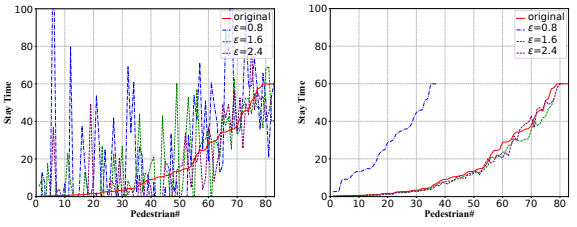
(c) PINQ and Black (UAD) (d) VideoDP (UAD)

(e) PINQ and Black (BVD) (f) VideoDP (BVD)

Fig. 7. Average Frame Counts with 15+ VEs in MOT Videos, 10+ VEs in UAD Videos, and 10+ VEs in BVD Videos

each pedestrian/vehicle stays in the video" (namely, stay time) which can be measured by the number of frames involving each VE. Then, pedestrians/vehicles are detected/tracked in all the frames, and then query results can be computed and returned for private analysis. Since too many groups of fine-grained empirical results may mess up the plots (e.g., in Figure 8), we only show three groups of results for $\epsilon = 0.8, 1.6$ and 2.4, which represent small, medium and large ϵ, respectively. Other groups of results lie between them.

– **Pedestrians**. In PED, 83 pedestrians are walking on the street. How long each pedestrian stays in the video can be utilized to learn the human behavior. Figure 8 presents the original results, PINQ results and VideoDP results for the PED. The 83 pedestrians in the PED (marked on the x axis), and the stay time is ranked from short to long (see the red curve in two subfigures). In PINQ (Figure 11(c)), the stay times of all the pedestrians are overly obfuscated even if ϵ is large since sensitivity Δ should be set as 60 (for even longer videos, Δ should be larger). Nevertheless, VideoDP significantly outperforms PINQ. As shown in Figure 11(d), in case of $\epsilon = 0.8$ (small privacy budget), approximately 40 distinct pedestrians are detected in the result. Although not all the pedestrians are sampled in VideoDP, the distribution of all the stay times (of the pedestrians) still lies close to the original result. The results of $\epsilon = 0.8$ show less pedestrians in the x-axis than other two groups of results ($\epsilon = 1.6$ and 2.4) since many pedestrian cannot be detected for small ϵ. As ϵ increases to 1.6, VideoDP results are close to the original results (however, PINQ results are still fluctuated).

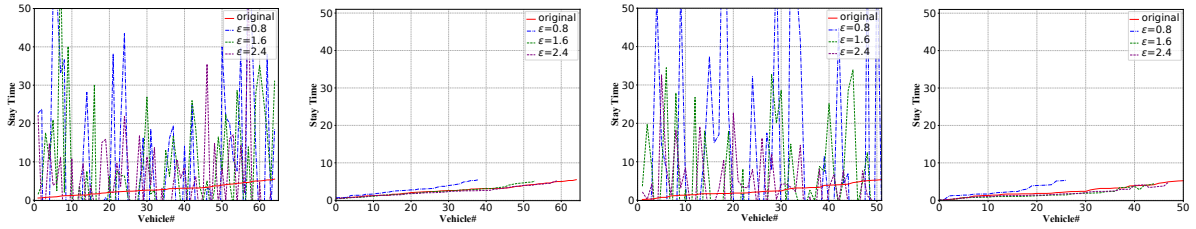


(a) Pedestrian (PINQ) (b) Pedestrian (VideoDP)

Fig. 8. Pedestrian Stay Time in PED

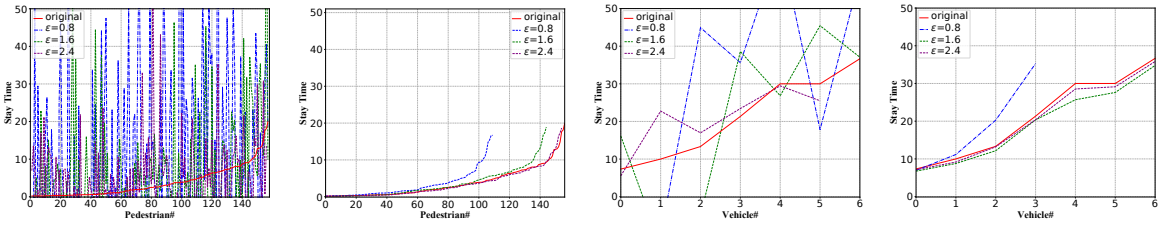
– **Vehicles**. In the VEH, there are 115 distinct vehicles driving on the highway. We define the two-way moving directions as "upstream" and "downstream". Figure 9 demonstrates the length of time the vehicles stay in the video (upstream and downstream), which can be utilized to estimate the moving speed of vehicles, queue length estimation, etc. We can draw similar observations for the stay times of vehicles for both moving downstream and upstream directions in the VEH. VideoDP also significantly outperforms PINQ.
– **Pedestrians and Vehicles**. In the PV, there are 157 distinct pedestrians and 7 vehicles. Figure 10 demonstrates the length of time the pedestrians and vehicles stay in the video. It presents similar trends.

(2) VE Density (Small Sensitivity). We also conduct empirical studies to compare VideoDP and PINQ on queries with a smaller sensitivity. For instance, the vehicle density query returns the vehicle count in each frame of the video (sensitivity $\Delta = 1$), which can also facilitate the analyst to learn the traffic flow. Figure 11 shows the count of vehicles in each frame of VEH and pedestrians of PV, including the original results, PINQ results and VideoDP results (where $\epsilon = 0.8, 1.6$



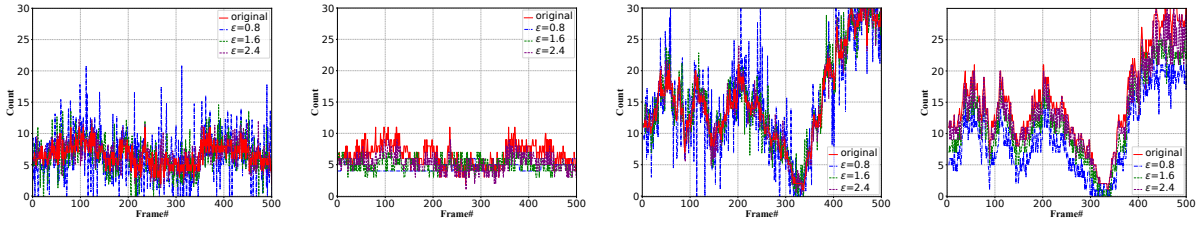
(a) Upstream (PINQ) (b) Upstream (VideoDP) (c) Downstream (PINQ) (d) Downstream (VideoDP)

Fig. 9. Vehicle Stay Time in VEH



(a) Pedestrian (PINQ) (b) Pedestrian (VideoDP) (c) Vehicle (PINQ) (d) Vehicle (VideoDP)

Fig. 10. Pedestrian and Vehicle Stay Time in PV



(a) Vehicle (PINQ) (b) Vehicle (VideoDP) (c) Pedestrian (PINQ) (d) Pedestrian (VideoDP)

Fig. 11. VE Count in Each Frame

and 2.4). Note that every vehicle only appears in a few frames of the video in VEH (see Figure 11(a) and 11(b)). The noisy results are both acceptable in PINQ ($\Delta = 1$) and VideoDP. However, the counts of vehicles are more fluctuated in PINQ as ϵ is small. We can draw similar observations in Figure 11(c) and 11(d).

6.3 Deep Learning Attack

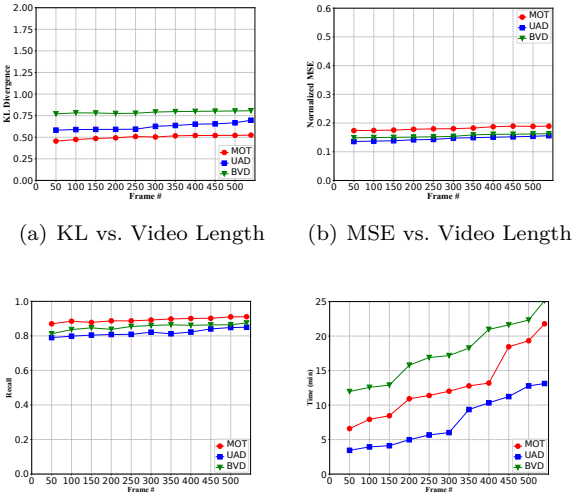
We also perform the CNN based attacks [39] to demonstrate VideoDP's protection against deep learning, though VideoDP does not directly reveal videos/frames to analysts (DP algorithms reveal the query results in general). Assuming that the adversary has known everything about specific VE(s), and tries to re-identify such VE(s) in the output videos (notice that, for each output video of VideoDP which is random, we generate 20 videos). We compare the performance of VideoDP with the Mosaic blurring method against the CNN attack [39]. Mosaic blurring considers each square of pixels (a.k.a., "pixel box") as the mosaic window, computes the average color of every pixel in each square, and sets the entire square as that color. In the experiments, we set privacy budget ϵ as 0.8, 1.2, 1.6 and 2.4 in VideoDP, and the sizes of pixel boxes as 2×2, 4×4, 8×8 and 16×16. Table 2 shows the average accuracy of successfully identifying such VE(s) from the random outputs of VideoDP and the videos sanitized by Mosaic blurring. For all different ϵ, the VE(s) cannot be identified with high confidence, compared to Mosaic blurring.

6.4 Scalability

We also evaluate how video length affects the performance of VideoDP (frames in UAD videos are repeated to synthesize longer videos). First, in Figure 12(a) and

Table 2. Accuracy of the CNN Attack (%).

Video Datasets	Mosaic (Pixel Box Sizes for Blurring)			
	2×2	4×4	8×8	16×16
MOT	97.45	91.33	89.75	64.75
UAD	99.23	95.47	92.25	76.25
BVD	85.78	75.62	63.14	44.39
Video Datasets	**VideoDP (ϵ)**			
	0.8	1.2	1.6	2.4
MOT	3.62	5.96	12.96	16.47
UAD	4.93	11.27	16.22	19.78
BVD	5.34	7.54	18.77	21.41



(a) KL vs. Video Length (b) MSE vs. Video Length

(c) Recall vs. Video Length (d) Runtime vs. Video Length

Fig. 12. Performance vs. Video Length ($\epsilon = 1.6$)

12(b) ($\epsilon = 1.6$), the KL and MSE values slightly change as the length of three sets of videos increases. Second, as the number of frames increases, the detection accuracy (recall) slightly increases for all the three videos (see Figure 12(c)). Third, we have also evaluated the runtime of VideoDP. Figure 12(d) shows a linear runtime trend on the video length, which provides sufficient efficiency for randomly generating longer high-resolution videos (e.g., 1920×1080). For longer videos, we can split the input video into multiple fragments (e.g., 1 minute per fragment). Then, we can still apply VideoDP to efficiently sanitize all the fragments which are integrated later. In many videos (e.g., traffic monitoring videos), VEs move rapidly and appear in the video for a few seconds. Then, fragmentation, generation and integration would not affect the privacy. Finally, in each video, all the VEs may have different sizes. While VEs are moving, the size of the same VE may also vary in different frames. VideoDP protects all the VEs (including all the pixels of each VE in all the frames). VideoDP generates good utility for all the videos with different VE sizes.

7 Related Work

Dwork [18] first proposed the notion of differential privacy that provides rigorous privacy guarantee for statistical databases [19] against arbitrary background knowledge possessed by adversaries. Such privacy notion has been extended to sanitize and release data for different applications, such as classification [53], histograms [13], search logs [31], locations [47], trajectories [37], and data synthesis [8, 35]. To our best knowledge, we first address the deficiency in differentially private video analysis.

Since VideoDP locally perturbs the input video by the video owner (and then flexibly offers queries/analysis to untrusted analysts), the emerging local differential privacy (LDP) models [14, 21] are also relevant to this work. The state-of-the-art LDP techniques perturb local data by the owners to generate statistics for histograms [7], reconstruct social graphs [48] and videos [54], and function frequent itemset mining [55].

Most of the existing video sanitization techniques use straightforward measures for quantifying the privacy loss in videos. For instance, in [23, 43], if faces are present, then it is considered as complete privacy loss, otherwise no privacy loss. Besides only considering the faces as privacy leakage, recent works also investigated the privacy risks in the activities, places and other implicit channels (e.g., when and where to record the video) [49]. Fan [22] applied Laplace noise to obfuscate pixels in an image to ensure DP for protecting specific regions of an image. However, the image quality has been significantly reduced (protection for each single pixel without composition may not work as well [34]). Neither the privacy notion or the Laplace noise (generated with high sensitivity) can be effectively applied to all the pixels for sanitizing full videos.

Moreover, most existing techniques directly adopted computer vision techniques [23, 33] to first detect faces and/or other sensitive regions in the video frames and then blur them. However, such detect-and-protect solutions have some limitations. For instance, they cannot formally quantify and bound the privacy leakage. Thus, how much risk any VE can be identified from the video is unknown. In addition, blurred regions might still be reconstructed by deep learning methods [39, 45]. VideoDP can address these limitations with strong privacy protection against arbitrary prior knowledge.

8 Conclusion

In this paper, to our best knowledge, we take the first step to study the problem of video analysis with *differential privacy guarantee*. Specifically, we have proposed a new sampling based differentially private mechanism to generate utility-driven private videos for any private analysis. The proposed VideoDP has also provided a flexible platform for untrusted analysts to privately conduct any kind of query/analysis over the randomly generated utility-driven private video. We have proven the differnetial privacy guarantee, and conducted extensive experiments to validate the performance of VideoDP by benchmarking the results with the PINQ-based video analyses. The experimental results have demonstrated superior utility in different analyses.

9 Acknowledgements

This work is partially supported by the National Science Foundation (NSF) under the Grant No. CNS-1745894. The authors would like to thank Li Xu and Yu Kong for their support on implementing and benchmarking the computer vision algorithms. The authors also thank the anonymous reviewers for their constructive comments.

References

[1] YouTube Official Blog 2012. (02/2020)

[2] https://boxy-dataset.com/boxy/ (02/2020)

[3] https://drive.google.com/file/d/1hYa5s7fjvQc1S1wRY6GcRqOPwL0Hy_aE/view. (02/2020)

[4] https://opencv.org/. (02/2020)

[5] 2012. H.R. 6671 (112th): Video Privacy Protection Act.

[6] B. Abreu, L. Botelho, A. Cavallaro, D. Douxchamps, T. Ebrahimi, P. Figueiredo, B. Macq, B. Mory, L. Nunes, J. Orri, M. Trigueiros, and A. Violante. Video-based multi-agent traffic surveillance system. *Intelligent Vehicles Symposium*, 2000.

[7] R. Bassily and A. Smith. Local, Private, Efficient Protocols for Succinct Histograms. *STOC*, pages 127–135, 2015.

[8] R. Bild, K. A. Kuhn, and F. Prasser. Safepub: A truthful data anonymization algorithm with strong privacy guarantees. *PoPETs*, 2018(1):67–87, 2018.

[9] T. Acharya and A. K. Ray. *Image Processing - Principles and Applications*. Wiley-Interscience, 2005.

[10] A.B. Chan, Z.S.J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. *CVPR*, pages 1–7, 2008.

[11] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong. Quantifying Differential Privacy under Temporal Correlations. *ICDE*, pages 821–832, 2017.

[12] O. Chapelle, H. Patrick, and V. N. Vapnik. Support vector machines for histogram-based image classification. *Neural Networks*, pages 1055–1064, 1999.

[13] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu. Differentially private spatial decompositions. In *PVLDB*, pages 20–31, 2012.

[14] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang. Privacy at Scale: Local Differential Privacy in Practice. In *SIGMOD*, pages 1655–1658, 2018

[15] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[16] P. Dollár, V. abaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VSPETS*, 2005.

[17] D. Doma. Comparison of different image interpolation algorithms. *West Virginia University*, 2008.

[18] C. Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[19] C. Dwork, F. McSherry, K. Nissim and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[20] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. In *Foundations and Trends in Theoretical Computer Science 9(3-4)*, pages 265–284, 2014.

[21] U. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving orindal response. In *CCS*, pages 1054–1067, 2014.

[22] L. Fan. Image Pixelization with Differential Privacy. In *DBSec*, pages 148–162, 2018.

[23] D. Fidaleo, H. Nguyen and M. Trivedi. The networked sensor tapestry (nest): a privacy enhanced software architecture for interactive analysis of data in video-sensor networks. In *ACM MM Workshops*, pages 46–53, 2004.

[24] R. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015.

[25] M. Götz, A. Machanavajjhala, G. Wang, X. Xiao and J. Gehrke. Publishing search logs - a comparative study of privacy guarantees. *TKDE*, 24(3):520–532, 2012.

[26] M. Handte, M.U. Iqbal, S. Wagner, W. Apolinarski, P.J. Marrón, E.M.M. Navarro, S. Martinez, S.I. Barthelemy and M.G. Fernández. Crowd Density Estimation for Public Transport Vehicles. In *EDBT/ICDT Workshops*, pages 315-322, 2014.

[27] Z.L. He, J. Zhang, M. Kan, S. Shan and X. Chen. Robust fec-cnn: A high accuracy facial landmark detection system. In *CVPR Workshops*, pages 98–104, 2017.

[28] S. Hill, Z. Zhou, L. Saul, and H. Shacham. On the (in) effectiveness of mosaicing and blurring as tools for document redaction. *PoPETs*, 2016(4):403–417, 2016.

[29] Z. Hong. Algebraic feature extraction of image for recognition. *Pattern recognition*, 24(3):211–219, 1991.

[30] Y. Hong, J. Vaidya, H. Lu, and M. Wu. Differentially private search log sanitization with optimal output utility. In *Proceedings of the EDBT*, pages 50–61. ACM, 2012.

[31] Y. Hong, J. Vaidya, H. Lu, P. Karras, and S. Goel. Collaborative search log sanitization: Toward differential privacy and boosted utility. *IEEE TDSC*, 12(5):504–518, 2015.

[32] A. Hore, D. Ziou, L. Saul, and H. Shacham. Image Quality Metrics: PSNR vs. SSIM document redaction. *20th ICPR*, pages 2366–2369, 2010.

[33] T. Koshimizu, T. Toriyama, and N. Babaguchi. Factors on the sense of privacy in video surveillance. In *ACM MM Workshops*, pages 35–44, 2006.
[34] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy*, pages 656–672, 2019.
[35] D. Leoni. Non-interactive differential privacy: a survey. In *WOD*, pages 40–52, 2012.
[36] N. Li, W. H. Qardaji, and D. Su. On sampling, anonymization, and differential privacy or, *k*-anonymization meets differential privacy. In *ASIACCS*, pages 32–33, 2012.
[37] B. Liu, S. Xie, H. Wang, Y. Hong, X. Ban and M. Mohammady. VTDP: Privately Sanitizing Fine-grained Vehicle Trajectory Data with Boosted Utility. In *TDSC*, 2019.
[38] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *ICDE*, pages 277–286, 2008.
[39] R. McPherson, R. Shokri, and V. Shmatikov. Defeating image obfuscation with deep learning. In *arXiv preprint arXiv:1609.00408*, 2016.
[40] F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD*, pages 19–30, 2009.
[41] F. McSherry. Mechanism Design via Differential Privacy. In *FOCS*, pages 94–103, 2007.
[42] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *CoRR*, 2016.
[43] S. Moncrieff, S. Venkatesh, and G. West. Dynamic privacy assessment in a smart house environment using multimodal sensing. *TOMMCAP*, 2008.
[44] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Theory of Computing*, pages 75–85, 2007.
[45] S.J. Oh, R. Benenson, M. Fritz, and B. Schiele. Faceless person recognition: Privacy implications in social media. In *ECCV*, pages 19–35, 2016.
[46] P. Piccinini, A. Prati,and R. Cucchiara. Real-time object detection and localization with SIFT-based clustering. In *Image and Vision Computing*, pages 573–587, 2012.
[47] W. Qardaji, W. Yang, and N. Li. Differentially private grids for geospatial data. In *ICDE*, pages 757–768, 2013.
[48] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao and K. Ren. Generating Synthetic Decentralized Social Graphs with Local Differential Privacy. In *CCS*, pages 425–438, 2017.
[49] M. Saini, P. Atrey, S. Mehrotra, and M. Kankanhalli. W3-privacy: understanding what, when, and where inference channels in multi-camera surveillance video. *Multimedia Tools and Applications*, 68:135–158, 2014.
[50] S. Song, Y. Wang, and K. Chaudhuri. Pufferfish Privacy Mechanism for Correlated Data. *SIGMOD*, pages 1291–1306, 2017.
[51] P. N. Sridharan and S. Raman. Characteristics of video data for signal analysis. In *ICSP*, pages 1254–1257, 1996.
[52] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. In International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, pages 571–588, 2002.
[53] J. Vaidya, B. Shafiq, A. Basu, and Y. Hong. Differentially private naive bayes classification. In *Web Intelligence*, pages 571–576, 2013.
[54] H. Wang, Y. Hong, Y. Kong, and J. Vaidya. Publishing Video Data with Indistinguishable Objects. In *EDBT*, pages 323–334, 2020.
[55] T. Wang, N. Li, and S. Jha. Locally Differentially Private Frequent Itemset Mining. In *SP*, pages 127–143, 2018.
[56] N. Wojke, A. Bewley, and D. Paulus. Simple Online and Realtime Tracking with a Deep Association Metric. In *ICIP*, pages 3645–3649, 2017.
[57] Y. Yang, J. Liu, and M. Shah. Video Scene Understanding Using Multi-scale Analysis. *ICCV*, pages 1669–1676, 2008.
[58] J. Yang, B. Price, S. Cohen, H. Lee and M.H. Yang. Object contour detection with a fully convolutional encoder-decoder network. *ICCV*, pages 193–202, 2016. Yang, , Price, B.,S. Cohen, , H. Lee and M. H. Yang. (2016).
[59] S. Zhang, L. Wen, X. Bian, Z. Li and S.Z. Li. Occlusion-aware R-CNN: detecting pedestrians in a crowd. In ECCV, pages 637–653, 2018.

Appendix

A The Notation Table

Table 3. Frequently Used Notations

VE	**visual element (e.g., object, human)**
V, O	**orignal video and output synthetic video**
$\|V\|, \|O\|$	**total pixel counts in V, O**
m	**the number of distinct RGBs in V**
θ_i	**the ith RGB in V where $i \in [1, m]$**
n	**the number of distinct VEs in V**
Υ_j	**the jth VE in V (all the frames), $j \in [1, n]$**
$\|\Upsilon_j\|$	**total number of pixels in $\|\Upsilon_j\|$**
Ψ_j	**set of RGBs in Υ_j with budgets**
$\|\Psi_j\|$	**cardinality of Ψ_j**
d_j	**total pixel count in Υ_j**
$\widetilde{\theta}_{ij}$	**the ith RGB in Ψ_j**
k_j	**(optimal) number of distinct RGBs in Υ_j**
Ψ, $\|\Psi\|$	**$\bigcup_{j=1}^{n} \Psi_j$, cardinality of Ψ**
$\widetilde{\theta}_i, \theta_i$	**the ith RGB in Ψ, the ith RGB in V**
$\widetilde{c}_i$ **(or** c_i**)**, $\widetilde{c}_i^j$	**total pixel count for RGB $\widetilde{\theta}_i$ (or θ_i) in V, Υ_j**
$\widetilde{x}_i$ **(or** x_i**)**	**total pixel count for RGB $\widetilde{\theta}_i$ (or θ_i) in O**
(a, b, t)	**the pixel with coordinates (a, b) and frame t**
$\theta(a, b, t)$	**the RGB of pixel (a, b, t) in V**
$\hat{\theta}(a, b, t)$	**the RGB of pixel (a, b, t) in O**
$Pr(a, b, t)$	**probability that pixel (a, b, t) is sampled**
$\sigma_0, \ldots, \sigma_4$	**probabilities that pixel (a, b, t) has $0, 1, \ldots, 4$ neighboring pixels after Phase I (sampling)**
$\hat{\theta}_N$	**simplified notation for $\hat{\theta}(a-1, b, t)$**
$\hat{\theta}_S$	**simplified notation for $\hat{\theta}(a+1, b, t)$**
$\hat{\theta}_W$	**simplified notation for $\hat{\theta}(a, b-1, t)$**
$\hat{\theta}_E$	**simplified notation for $\hat{\theta}(a, b+1, t)$**

B Optimal k_j for VE Υ_j

B.1 Equations for Different Pixels

If pixel (a,b,t) is a non-border pixel, we have Equation 1 to represent the relation between the RGB expectation of any pixel (a,b,t) and the RGB expectation of its four neighbors (denoted as $\hat{\theta}_N, \hat{\theta}_S, \hat{\theta}_W$ and $\hat{\theta}_E$). We now briefly discuss how to derive such relation.

First, if pixel (a,b,t) is sampled, then the RGB expectation equals $Pr(a,b,t) * \theta(a,b,t)$ where $Pr(a,b,t)$ is the probability of sampling (a,b,t) and $\theta(a,b,t)$ denotes its RGB in the original video V.

Second, if pixel (a,b,t) is not sampled, then it will be interpolated based on the RGBs of its neighbors. There are five subcases (denoting the probabilities that (a,b,t) has 0, 1, 2, 3 and 4 neighbors before interpolation as $\sigma_0(a,b,t), \sigma_1(a,b,t)$, $\sigma_2(a,b,t)$, $\sigma_3(a,b,t), \sigma_4(a,b,t)$):

1. 0 neighbor: all its neighbors are not sampled in Phase I. Then, the probability share is $\sigma_0(a,b,t) * 0$.
2. 1 neighbor: 3 of its neighbors are not sampled in Phase I. Then, the probability share is:

   $$\sigma_1(a,b,t) * [1 - Pr(a,b,t)] * \frac{E(\hat{\theta}_N) + E(\hat{\theta}_S) + E(\hat{\theta}_W) + E(\hat{\theta}_E)}{4}$$

   where all 4 neighbors can be used for interpolation.
3. 2 neighbors: 2 of its neighbors not sampled in Phase I. Then, the probability share is:

   $$\sigma_2(a,b,t) * [1 - Pr(a,b,t)] * \frac{3E(\hat{\theta}_N) + 3E(\hat{\theta}_S) + 3E(\hat{\theta}_W) + 3E(\hat{\theta}_E)}{6 * 2}$$

   where 6 different combinations of two neighbors can be used for interpolation and the interpolated RGB is the average of two neighbors' RGBs.
4. 3 neighbors: 1 of its neighbors is not sampled in Phase I. Then, the probability share is:

   $$\sigma_3(a,b,t) * [1 - Pr(a,b,t)] * \frac{3E(\hat{\theta}_N) + 3E(\hat{\theta}_S) + 3E(\hat{\theta}_W) + 3E(\hat{\theta}_E)}{4 * 2}$$

   where 4 different combinations of two neighbors can be used for interpolation and the interpolated RGB is the average of three neighbors' RGBs.
5. 4 neighbors: no neighbor is suppressed in sampling. Then, the probability share is:

   $$\sigma_4(a,b,t) * [1 - Pr(a,b,t)] * \frac{E(\hat{\theta}_N) + E(\hat{\theta}_S) + E(\hat{\theta}_W) + E(\hat{\theta}_E)}{4}$$

   where only 1 combination of 4 neighbors can be used for interpolation and the interpolated RGB is the average of 4 neighbors' RGBs.

Similarly, if pixel (a,b,t) is on the border but not at the corner (w.l.o.g., the left border), then we have:

$$\begin{aligned} &E[\hat{\theta}(a,b,t)] = Pr(a,b,t) * \theta(a,b,t) + \sigma_0(a,b,t) * 0 \\ &+ \frac{\sigma_1(a,b,t) * [1 - Pr(a,b,t)] * [E(\hat{\theta}_N) + E(\hat{\theta}_S) + E(\hat{\theta}_E)]}{3} \\ &+ \frac{\sigma_2(a,b,t) * [1 - Pr(a,b,t)] * [2E(\hat{\theta}_N) + 2E(\hat{\theta}_S) + 2E(\hat{\theta}_E)]]}{3 * 2} \\ &+ \frac{\sigma_3(a,b,t) * [1 - Pr(a,b,t)] * [E(\hat{\theta}_N) + E(\hat{\theta}_S) + E(\hat{\theta}_E)]}{3} \end{aligned} \tag{8}$$

If pixel (a,b,t) is located at the corner of the tth frame (w.l.o.g., the upper-left corner), then we have:

$$\begin{aligned} &E[\hat{\theta}(a,b,t)] = Pr(a,b,t) * \theta(a,b,t) + \sigma_0(a,b,t) * 0 \\ &+ \frac{\sigma_1(a,b,t) * [1 - Pr(a,b,t)] * [E(\hat{\theta}_S) + E(\hat{\theta}_E)]}{2} \\ &+ \frac{\sigma_2(a,b,t) * [1 - Pr(a,b,t)] * [E(\hat{\theta}_S) + E(\hat{\theta}_E)]]}{2} \end{aligned} \tag{9}$$

B.2 Solving Algorithm

The optimal number of distinct RGBs k_j (to allocate privacy budget) is computed based on minimizing the MSE expectation of visual element Υ_j (averaged by the number of pixels). Thus, we solve the following optimization (which is equivalent to Equation 2):

$$\arg\min_{k_j} \sum_{\forall (a,b,t) \in \Upsilon_j} \left(E[\theta(a,b,t)] - E[\hat{\theta}(a,b,t)] \right)^2$$

$$\begin{cases} E[\hat{\theta}(1,1)] = Pr(1,1)) * \theta(1,1) + [1 - Pr(1,1)] * \frac{(\sigma_1(1,1) + \sigma_2(1,1)) * (E[\hat{\theta}(1,2)] + E[\hat{\theta}(2,1)])}{2} \\ E[\hat{\theta}(1,2)] = Pr(1,2) * \theta(1,2) + [1 - Pr(1,2)] * \frac{(\sigma_1(1,2) + \sigma_2(1,2) + \sigma_3(1,2)) * (E[\hat{\theta}(1,1)] + E[\hat{\theta}(2,2)] + E[\hat{\theta}(1,3)])}{3} \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \forall a \in (1,A), \forall b \in (1,B) \\ E(\hat{\theta}(a,b)) = Pr(a,b) * \theta(a,b) + [1 - Pr(a,b)] * \frac{(\sigma_1(a,b) + \cdots + \sigma_4(a,b)) * (E[\hat{\theta}(a-1,b)] + \cdots + E[\hat{\theta}(a,b+1)])}{4} \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ E(\hat{\theta}(A,B)) = Pr(A,B) * \theta(A,B) + [1 - Pr(A,B)] * \frac{(\sigma_1(A,B) + \sigma_2(A,B)) * (E[\hat{\theta}(A-1,b)] + E[\hat{\theta}(A,B-1)])}{2} \end{cases}$$

Note that the above equations can be simply extended to all the pixels in Υ_j in all the frames (incorporating frame t). We use the inverse matrix to solve these equations where the coefficients of all the above equations can be represented as a $|\Upsilon_j| \times |\Upsilon_j|$ matrix (denoted as M). To ensure that the inverse matrix can solve the equations, M should have a full rank $|\Upsilon_j|$. In case that M is not a full rank matrix (indeed, the

rank of M is very high since $\forall(a,b,t) \in \Upsilon_j, \sigma_1(a,b,t), \sigma_2(a,b,t), \sigma_3(a,b,t), \sigma_4(a,b,t)$ are somewhat random), we can add a tiny random noise to the non-zero entries in M (in which the deviation is negligible).

Specifically, denoting the expectation of the sth pixel in Υ_j as $E[\hat{\theta}(s)]$ where $s \in [1, AB]$. Then, we have

$$E[\hat{\theta}(s)] = \frac{1}{|M|} * \sum_{i=1}^{AB} [(-1)^{i+s} * M_{is}^{(AB-1)} * b_i] \quad (10)$$

where $|M|$ is the determinant of M, $M_{is}^{(AB-1)}$ denotes the sth cofactor (corresponding the sth pixel; including $(AB-1) \times (AB-1)$ entries) and b_i is the ith constant in the equation (in last column of M). Thus, $M_{is}^{(AB-1)}$ can be recursively represented:

$$M_{is}^{(AB-1)} = \sum_{i=1}^{AB} [(-1)^{i+s} * \mathbb{R}_i * M_{is}^{(AB-2)}] \quad (11)$$

where $M^{(AB-2)}$ represents the cofactor matrix of M^{AB-1} and $\mathbb{R}_i$ is a random constant (for ensuring full rank for M) which is close to $-\frac{[1-Pr(a,b,t)](\sigma_1(a,b,t)+\sigma_2(a,b,t))}{2}$ for corner pixels, $-\frac{[1-Pr(a,b,t)][\sigma_1(a,b,t)+\sigma_2(a,b,t)+\sigma_3(a,b,t)]}{3}$ for border pixels, and $-\frac{[1-Pr(a,b,t)][\sigma_1(a,b,t)+\sigma_2(a,b,t)+\sigma_3(a,b,t)+\sigma_4(a,b,t)]}{4}$ for non-border pixels. Then, Equation 11 can be:

$$M_{is}^{(AB-1)} = \sum_{i=1}^{AB} [(-1)^{i+s} * (\prod_{i=1}^{AB-3} \mathbb{R}_i) * M_{is}^{(2)}] \quad (12)$$

Since each row of M only has at most 5 non-zero entries (corresponding to the variables of the current pixel and its four/three/two neighbors), we have:

$$E[\hat{\theta}(s)] \approx -\frac{5^{AB-3} * AB}{|M|} * \max_{\forall i \in [1,AB]} \{|\mathbb{R}_i|^{AB-3} * M_{is}^{(2)} * b_i\} \quad (13)$$

Thus, we have the MSE expectation in VE Υ_j:

$$\sum_{i=1}^{AB} [\theta(a,b,t) + \frac{5^{AB-3} * AB}{|M|} * \max_{\forall i \in [1,AB]} \{|\mathbb{R}_i|^{AB-3} * M_{is}^{(2)} * b_i\}]^2$$

For each k_j, the corresponding MSE expectation can be computed using the above equation. Then, the optimal k_j can be obtained by traversing k_j in any range. In addition, it is straightforward to prove that the complexity of the inverse matrix based solver is $O(n^3 \log(n))$. Note that we assume that the optimal k_j is computed for minimum MSE based on the first traversal in the interpolation of each visual element. The deviation is very minor since most pixels are interpolated in the first traversal in our experiments. Moreover, the optimal k_j (derived as above) is also experimentally validated (see Figure 15(d)).

C Budget Allocation Algorithm

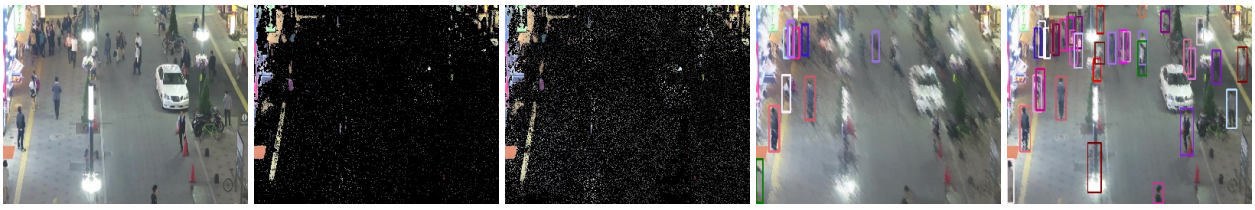
Input : n sets of RGBs $\forall j \in [1,n], \Psi_j = \{i \in [1,k_j], \widetilde{\theta}_{ij}\}$ privacy budget ϵ for Phase I of VideoDP
Output: privacy budget for each unique RGB in n sets

initialize the set of unique RGBs: $\Psi \leftarrow \bigcup_{j=1}^{n} \Psi_j$
foreach $j \in [1,n]$ **do**
  initialize the overall budget for set Ψ_j : $\epsilon(\Psi_j) \leftarrow \epsilon$
foreach $\ell \in [1,n]$ **do**
  foreach $\widetilde{\theta} \in \Psi$ **do**
    if $count(\widetilde{\theta} \in \{\Psi_1, \ldots, \Psi_n\}) = (n-\ell+1)$ **then**
      // w.l.o.g., $\widetilde{\theta} \in \Psi_1, \ldots, \Psi_{n-\ell+1}$
      initialize budget for RGB $\widetilde{\theta}$: $\epsilon(\widetilde{\theta})$
      $\epsilon(\widetilde{\theta}) \leftarrow \min_{\forall j \in [1,(n-\ell+1)]} [\frac{d_j(\widetilde{\theta})}{d_j} * \epsilon(\Psi_j)]$
      // $\frac{d_j(\widetilde{\theta})}{d_j}$ denotes the ratio of pixels with RGB $\widetilde{\theta}$ in Ψ_j
      foreach $j \in [1,(n-\ell+1)], \Psi_j$ **do**
        update budget: $\epsilon(\Psi_j) \leftarrow \epsilon(\Psi_j) - \epsilon(\widetilde{\theta})$
        update total pixel count: $d_j \leftarrow d_j - d_j(\widetilde{\theta})$
        // $\epsilon(\Psi_j) > \epsilon(\widetilde{\theta})$ for the first $(n-1)$ partitions; $\epsilon(\Psi_j) = \epsilon(\widetilde{\theta})$ for the last partition (0 budget left)
      return budget $\epsilon(\widetilde{\theta})$ for RGB $\widetilde{\theta}$
      $\Psi \leftarrow \Psi \setminus \widetilde{\theta}$

Algorithm 2: Budget Allocation

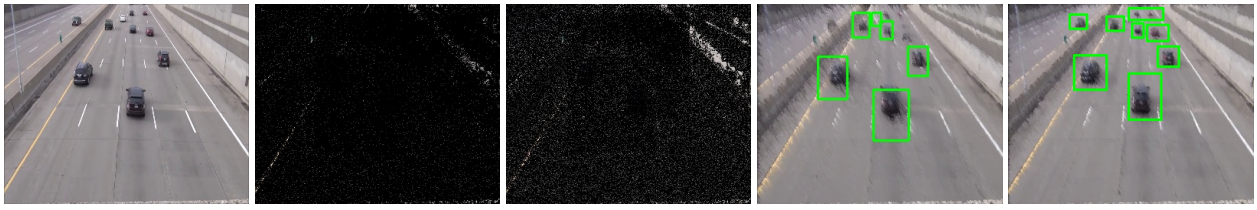
D Additional Results

While evaluating the utility of the sanitized videos in three datasets using two utility measures (KL divergence and MSE), we also fix ϵ and traverse different k for all the visual elements (assigning the same $k \in [4,30]$). Since optimal k may be different, we use specific videos to see how it affects result. Figure 15(a) and 15(b) present the KL divergence values for all the sampled pixels (where privacy budget ϵ is fixed as 0.8 and 1.6, respectively). We can observe that the KL value increases as k increases (if the same number of distinct RGBs in all the visual elements are selected to assign privacy budgets). This is true for the following reason: smaller k samples pixels with less diverse RGBs, but it can allocate a larger privacy budget to each RGB. Then, the



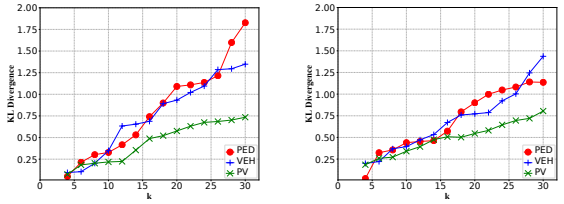
(a) Original (b) $\epsilon = 0.8$ (Phase I) (c) $\epsilon = 1.6$ (Phase I) (d) $\epsilon = 0.8$ (Phase II) (e) $\epsilon = 1.6$ (Phase II)

Fig. 13. Representative Frames in the Random Output Video of PED (available for differentially private queries/analysis)

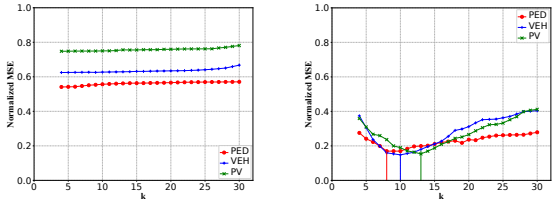


(a) Original (b) $\epsilon = 0.8$ (Phase I) (c) $\epsilon = 1.6$ (Phase I) (d) $\epsilon = 0.8$ (Phase II) (e) $\epsilon = 1.6$ (Phase II)

Fig. 14. Representative Frames in the Random Output Video of VEH (available for differentially private queries/analysis)



(a) KL vs k ($\epsilon = 0.8$) (b) KL vs k ($\epsilon = 1.6$)



(c) MSE vs k_j (after Phase I)(d) MSE vs k_j (after Phase II)

Fig. 15. Pixel Level Utility Evaluation with k

generated results can have better count distributions for all the sampled RGBs.

We also examine the *optimal number of selected RGBs to assign privacy budgets* k_j in visual elements. We select the visual element with most pixels in all the videos (PED , VEH and PV). Since the optimal values are derived based on MSEs, we plot the normalized MSEs for *all the pixels in the visual element* for two videos in Figure 15(c) (after Phase I) and Figure 15(d) (after Phase II), respectively. The normalized MSE does not change much (after Phase I) as k increases since the MSE expectation is optimized for Phase II. Instead, Figure 15(d) clearly shows that k_j goes *optimal* in the range (which equals the optimal k_j after solving Equation 2 detailed in Appendix B) in both videos for all possible values in the specified range. As k_j increases, the normalized MSE of the VE first decreases and then increases. This reflects that the best k_j with respect to the optimal MSE is neither too small nor too large for different VE in all the three videos.

Finally, we present some representative frames of the PED and VEH to show the effectiveness of pixel sampling (Phase I) and utility-driven private video generation (Phase II) in VideoDP. Specifically, we randomly select a frame in video PED and VEH. Figure 13 and 14 demonstrate such frames in the input videos and the output videos (after Phase I and II). Figure 13(b), 13(c),14(b) and 14(c) demonstrate that more pixels are sampled as private budget ϵ is larger ($\epsilon = 1.6$). Although the portion of the total sampled pixels is not high (after Phase I), the pixel interpolation (Phase II) can reconstruct the video with good quality as shown in Figure 13(d), 13(e), 14(d) and 14(e). We can observe that the pedestrian/vehicles are randomly generated in the frame (which are not directly revealed to the analysts). More pedestrians/vehicles can be detected as $\epsilon = 1.6$. Then, disclosing the any query/analysis result on such (random) video to analysts satisfies differential privacy.