Privacy-Constrained Video Streaming

Anonymous Author(s)
Affiliation
Address
email

Abstract: Many robots (e.g., iRobot's Roomba) operate based on visual observations from live video streams, and such observations may inadvertently include privacy-sensitive objects, such as personal identifiers. Existing approaches for preserving privacy rely on deep learning models, differential privacy, or cryptography. They lack guarantees for the complete concealment of all sensitive objects. Guaranteeing concealment requires post-processing techniques and thus is inadequate for real-time video streams. We develop a method for privacy-constrained video streaming, PCVS, that conceals sensitive objects within real-time video streams. PCVS takes a logical specification constraining the existence of privacy-sensitive objects, e.g., always person$\rightarrow \neg$face (never show faces when a person exists). It uses a detection model to evaluate the existence of these objects in each incoming frame. Then, it blurs out a subset of objects such that the existence of the remaining objects satisfies the specification. We then propose a conformal prediction approach to (i) establish a theoretical lower bound on the probability of the existence of these objects in a sequence of frames satisfying the specification and (ii) update the bound with the arrival of each subsequent frame. Quantitative evaluations show that PCVS achieves over 95 percent specification satisfaction rate in multiple datasets, significantly outperforming other methods. The satisfaction rate is consistently above the theoretical bounds across all datasets, indicating that the established bounds hold. Additionally, we deploy PCVS on robots in real-time operation and show that the robots operate normally without being compromised when PCVS conceals objects.

Keywords: Privacy, Data Generation, Conformal Prediction, Formal Methods

1 Introduction

While robots utilize visual observations from video streams during operational routines for decision-making purposes, recording and disseminating such videos potentially exposes private information [1]. A recent story highlighting a Roomba taking images of a person in a toilet room attests to the legitimacy of privacy concerns during robot operation [2].

Existing approaches protect privacy by concealing sensitive objects, but they either fail to guarantee complete concealment or cannot process video streams in real-time. For instance, Cangialosi et al. [3] developed a differential privacy mechanism to protect video privacy, and Rahman et al. [4] propose a cryptographic approach for video privacy. Other methods for concealing sensitive objects rely on deep learning models [5, 6, 7, 8]. However, all these methods fail to guarantee complete concealment, and sensitive objects may still be exposed after they are applied.

To this end, some approaches guarantee that a given complete video adheres to privacy concerns formulated as temporal logic specifications. Recent works [9, 10, 11, 12] construct a finite automaton representing the existence of objects across frame sequences and verify this automaton against temporal logic specifications. However, these works require post-processing techniques, which make them unusable for real-time video streams.

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

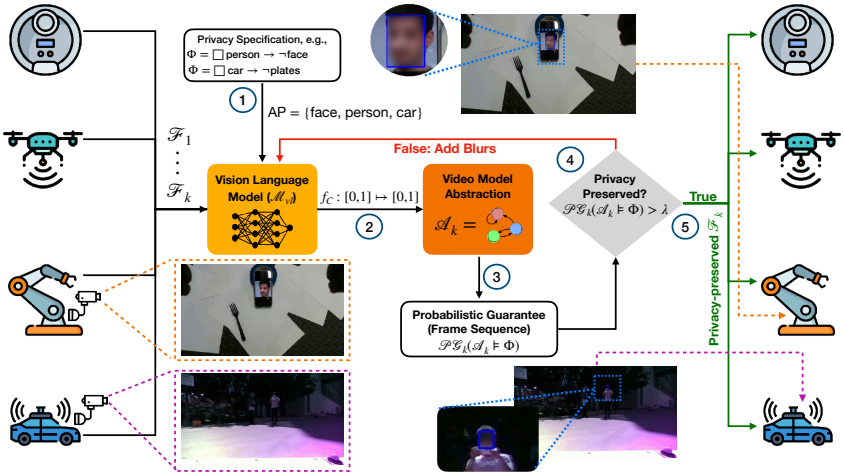


Figure 1: **Pipeline of Privacy-Constrained Video Streaming: (Step 1)** Given a privacy specification Φ, we define a set AP of atomic propositions describing privacy-sensitive objects. **(Step 2)** Given an incoming frame $\mathcal{F}_k$ from the video, the method uses a vision-language model (VLM) to detect sensitive objects in the frame. Each detection is associated with a confidence score from the VLM. The method calibrates a confidence score to a per-frame probability bound for correct detection via a calibration function f_C, as in Equation 1. **(Step 3)** The method builds an abstract model $\mathcal{A}_k$ representing object detections and their probability bounds in the frame sequence $\mathcal{F}_1, ..., \mathcal{F}_k$ via Algorithm 1. Then, it computes a theoretical bound for the probability of $\mathcal{A}_k$ satisfying Φ, i.e., a probabilistic guarantee $\mathcal{PG}_k(\mathcal{A}_k \models \Phi)$ using Equation 3. **(Step 4)** If $\mathcal{PG}_k(\mathcal{A}_k \models \Phi)$ is below a user-given privacy threshold λ, the method removes a subset of sensitive objects and goes back to Step 2 to recompute a guarantee. **(Step 5)** If $\mathcal{PG}_k(\mathcal{A}_k \models \Phi)$ is above λ, the method adds $\mathcal{F}_k$ back to the stream and proceeds to Step 1 with the next incoming frame. We number each step in blue.

We develop a method to conceal privacy-sensitive objects in real-time video streams from robot cameras. The method takes a logical specification constraining the existence of sensitive objects. The specifications allow users to describe complex privacy requirements with conjunctions, disjunctions, implications, etc. For each incoming frame from the stream, the method first uses a vision-language model to detect and locate all sensitive objects. Next, it removes a subset of objects (add Gaussian blurs or blackout) so that the existence of the remaining objects satisfies the specification.

We then establish a theoretical bound on the probability of complete concealment of sensitive objects in a video stream. As deep learning models are typically over-confident in detecting objects, we use conformal prediction to calibrate the model's confidence to a probability of correct detection. Next, we express the specification in a temporal logic formula and build a finite automaton representing the object detections in a sequence of frames and the probabilities of correct detections. Then, we compute the probability that the automaton satisfies the specification. We optimize the computing procedure to update the probability with the arrival of each subsequent frame from the video stream. Such probability helps users determine whether to use the video based on their privacy tolerance.

We evaluate the method over two large-scale datasets and present real-robot examples for real-time privacy protection. The method achieves a 95 percent satisfaction rate of specifications while preserving all non-sensitive information, significantly outperforming existing automated solutions. By seamlessly integrating concealment capabilities into the robot's visual perception system, we prevent potential privacy leakage from the robot. Simultaneously, this integration ensures the unhindered functionality of the robot's control policies, enabling it to operate normally without compromise.

2 Problem Formulation

A video $\mathcal{V}$ is a sequence of frames $\mathcal{F}_1, ..., \mathcal{F}_k$ where each $\mathcal{F}_k \in \mathbb{R}^{C \times W \times H}$ is an RGB image with C channels, W width, and H height. A video can be prerecorded or live-streamed from sources such as autonomous vehicles or security cameras.

We define a **privacy specification** Φ as a temporal logic formula [13] constraining the appearance of privacy-sensitive objects. Since we want to preserve privacy at all times, we express a privacy specification as $\Phi = \square(\tilde{\Phi})$, where $\square$ represents the "ALWAYS" temporal operation and $\tilde{\Phi}$ is a first-order logic formula [14]. The presence of privacy-sensitive objects is constrained by Φ.

We define a set of atomic propositions AP, where each proposition $p_i \in AP$ is a textual description of a privacy-sensitive object. Then, we use a vision-language model (VLM), $\mathcal{M}_{vl}$, to detect these objects. $\mathcal{M}_{vl} : \mathbb{R}^{C\times W\times H} \times AP \to [0, 1]$ takes a frame $\mathcal{F}_k \in \mathbb{R}^{C\times W\times H}$ and a proposition $p_i \in AP$ as inputs, and returns a confidence score $c \in [0, 1]$, denoted as $c = \mathcal{M}_{vl}(\mathcal{F}_k, p_i)$. However, deep learning models are often overconfident, and their detection accuracy cannot be guaranteed. Therefore, we calibrate the confidence using conformal prediction [15], which provides a lower bound for the probability of correctly detecting privacy-sensitive objects in every frame, considering the inherent uncertainty in deep learning model predictions.

However, traditional conformal prediction approaches focus on post-processing and do not account for temporal events. Therefore, we use calibrated confidence to detect and constrain privacy-sensitive objects over time and provide a probabilistic guarantee on a sequence of frames.

To achieve this, we develop an algorithm $\mathcal{VA}$ that takes a sequence of k frames and returns a formally verifiable video abstraction $\mathcal{A}_k$ encoding the object detection across the sequence: $\mathcal{VA}([\mathcal{F}_1, ..., \mathcal{F}_k]) = \mathcal{A}_k$. The **video abstraction** $\mathcal{A}_k$ is represented as a labeled Markov chain, detailed rigorously in Section 3 as it requires extensive background and mathematical notation. This provides a probabilistic guarantee on a frame sequence via formal verification [16].

Definition 1 (Probabilistic Guarantee on a Frame Sequence). Given a sequence of frames $\mathcal{F}_1, ..., \mathcal{F}_k$, a privacy specification Φ, and a video abstraction $\mathcal{A}_k$ at the k^{th} frame, a probabilistic guarantee $\mathcal{PG}_k(\mathcal{A}_k \models \Phi)$ on the frame sequence $\mathcal{F}_1, ..., \mathcal{F}_k$ represents the theoretical minimum probability that the presence of privacy-sensitive objects in the frame $\mathcal{F}_1$ through $\mathcal{F}_k$ adheres to Φ.

Problem 1 (Real-Time Video Privacy Preservation). Given a frame sequence $\mathcal{F}_1, ..., \mathcal{F}_{k-1}$, an incoming frame $\mathcal{F}_k$ from a video stream, a privacy specification Φ, and an algorithm $\mathcal{VA}$ that builds a video abstraction from the frame sequence, we aim to remove privacy-sensitive objects in $\mathcal{F}_k$ such that $\mathcal{A}_k = \mathcal{VA}([\mathcal{F}_1, ..., \mathcal{F}_k])$ satisfies Φ with a probability at least $\mathcal{PG}_k(\mathcal{A}_k \models \Phi)$.

3 Privacy-Constrained Video Streaming

We develop privacy-constrained video streaming (PCVS), a method to enforce live video streams satisfy a user-given privacy specification with a probabilistic guarantee. The overall pipeline for PCVS is illustrated in Figure 1.

Real-Time Video Privacy Preservation Framework. We explain our framework with a running example in a real-time video stream from a real robot (see Figure 2). We aim to hide human faces so that no personal identity will be revealed in vision-based robot operations. Therefore, the privacy specification is $\Phi = \square\,\text{person} \to \neg\text{face}$, where $\to$ and $\neg$ mean "implies" and "not," respectively. We detect humans and faces at every frame via the VLM. Subsequently, we use conformal prediction to obtain a calibrated confidence score for the detection in the current frame. We build a video abstraction $\mathcal{A}_k$ to represent the detection results for humans and faces across a sequence of frames and utilize it to obtain a probabilistic guarantee on Φ being satisfied. We then verify if the guarantee is above the user-given privacy threshold $\lambda \in [0, 1]$. If this threshold is not met, we iteratively remove the detected faces and update the guarantee $\mathcal{PG}_k(\mathcal{A}_k \models \Phi)$ until the threshold is met.

3.1 Probabilistic Guarantee on Video Privacy

Given a sequence of k frames and a privacy specification Φ, we compute a probabilistic guarantee $\mathcal{PG}_k(\mathcal{A}_k \models \Phi)$. This guarantee is updated at each incoming frame. We use techniques from formal methods to prove that the guarantee holds.

Confidence Calibration via Conformal Prediction. Recall that a VLM $\mathcal{M}_{vl}(x_i, y_i) = c$ receives an image x_i and a textual object label y_i as a prompt and returns a confidence score $c \in [0, 1]$. Given the VLM $\mathcal{M}_{vl}$ and a labeled calibration dataset that is distributed identically to the task domain, using conformal prediction [15], we learn a calibration function $f_C : [0, 1] \mapsto [0, 1]$ that maps a confidence score, $c \in [0, 1]$ to a lower bound for the probability of correct detection.

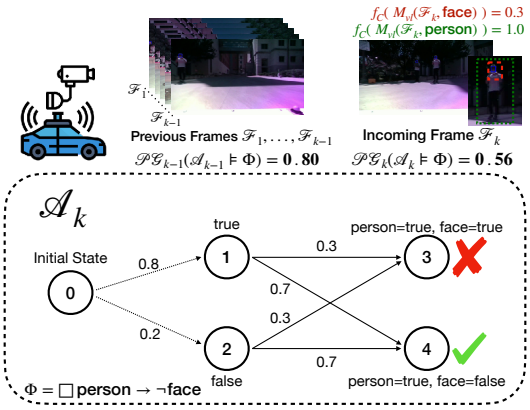


Figure 2: A running example on how to compute the probabilistic guarantee via video abstraction.

We first collect a calibration set $\{(x_i, y_i)\}_{i=1}^m$ consisting of m (image, ground truth text label) tuples. Then, we apply $\mathcal{M}_{vl}$ to detect the privacy-sensitive objects in the images $\{x_i\}_{i=1}^m$ and get a set of *nonconformity scores*: $\{1 - \mathcal{M}_{vl}(x_i, y_i)\}_{i=1}^m$. A nonconformity score is the sum of confidence scores of wrong detections. Next, we estimate a probability density function of these scores, denoted as $f_{nc}(z)$, where z is a nonconformity score. Then, we use Theorem 1 to establish a theoretical lower bound for the probability of the correct detection.

Theorem 1. Let $\epsilon \in [0, 1]$ be a pre-defined error bound and x_n be an image outside the calibration set. We define a *prediction band* as $\hat{C}(x_n) = \{p_i : \mathcal{M}_{vl}(x_n, p_i) \geq 1 - c^*, p_i \in AP\}$. Then, according to conformal prediction, there exists a confidence c^* such that $\epsilon = 1 - \int_0^{c^*} f_{nc}(z)dz$ satisfies $\mathbb{P}\left[y_n \in \hat{C}(x_n)\right] \geq 1 - \epsilon$, where y_n is the ground truth label for x_n. Proof in [15].

Note that $\mathcal{M}_{vl}(x_i, p_i)$ returns a single confidence score indicating whether p_i exists in x_i. By the theory of conformal prediction, $1 - \epsilon$ is a theoretical lower bound for the probability of the ground truth label belonging to the prediction band $\hat{C}(x_n)$. If $\mathcal{M}_{vl}(x_i, p_i) > 0.5$, we provide a lower bound for the probability of the existence of p_i. Otherwise, if $\mathcal{M}_{vl}(x_i, p_i) \leq 0.5$, we bound the probability of non-existence. Hence, we get a calibration function

$$f_C(c) = \begin{cases} \int_0^c f_{nc}(z)dz, \text{ if } c > 0.5 \\ \int_0^{1-c} f_{nc}(z)dz, \text{ otherwise.} \end{cases} \tag{1}$$

Video Abstraction. For verifying a *real-time* video stream against the privacy specification Φ, a key challenge is to verify the temporal behaviors of *all* the previously received frames plus the current frame. This makes verification space- and time-inefficient because we must repeatedly verify previous frames for each new incoming frame. To overcome this challenge, we build an abstraction for the video stream, which enables real-time verification.

Definition 2 (Video Abstraction). A video abstraction is a labeled Markov chain (S, s_0, P, L), where S is a set of states, each state corresponds to a conjunction of atomic propositions, $s_0 \in S$ is the initial state, $P : S \times S \to [0, 1]$ is a transition function. $P(s, s')$ represents the probability of transition from a state s to a state s' and $\sum_{s' \in S} P(s, s') = 1$. $L : S \to 2^{AP}$ is a labelling function.

We propose Algorithm 1 to build video abstractions. We demonstrate it through an example in Figure 2. First, we add an initial state (State 0 in Figure 2), a state representing the event that all previous frames (if they exist) satisfy Φ (State 1), and a state representing the event that previous frames fail Φ (State 2), as in lines 1-3. Next, we add transitions from State 0 to State 1 and to State 2 with the probability of previous frames satisfying Φ as in line 4. Then, we detect objects in the incoming frame $\mathcal{F}_k$ and get the probability bound for correct detection. For each conjunction of propositions (e.g., person=true and face=false), we build a state and add transitions to this state with the probability bound of correctly detecting objects described in this conjunction, as in lines 6-9. Hence, we obtain the video abstraction $\mathcal{A}_k$ as presented in Figure 2.

Algorithm 1: Real-Time Video Abstraction

Require: vision-language model $\mathcal{M}_{vl}$, calibration function f_C, set of propositions AP, specification Φ, probability p_{k-1} of previous frames satisying Φ, incoming frame $\mathcal{F}_k$

S_{obs}, P, L = {}, {}, {} ▷ Initialize the abstraction
S_{obs}.add(0), S_{obs}.add(1), S_{obs}.add(2) ▷ We represent each state with an Arabic numeral
$L(1) = \text{false}$, $L(2) = \text{true}$
$P(0,1) = 1 - p_{k-1}$, $P(0,2) = p_{k-1}$ ▷ Add transitions to indicate the probability of previous frames satisfying Φ
$i = 3$ ▷ Initialize a indexer representing states
for σ in 2^{AP} **do** ▷ σ is a conjunction of atomic propositions
    prob = $\prod_{p \in \sigma} f_C(\mathcal{M}_{vl}(\mathcal{F}_k, p))$ ▷ Get a lower bound for a detection result
    if prob > 0 **then** ▷ Add a state to represent the detection with the lower bound
        S_{obs}.add(i), $L(i) = \sigma$, $P(1,i) =$ prob, $P(2,i) =$ prob, $i = i + 1$
    end if
end for
return $S_{\text{obs}}, s_0 = 0, P, L$

Following Algorithm 1, we incrementally add new states to the abstraction (rather than build a new one) with the arrival of each new incoming frame and check it against Φ. Hence, this abstraction can be used to check video streams efficiently. Then, we theoretically prove that the probabilistic guarantee obtained through this abstraction holds.

Probabilistic Guarantees for Frame Sequence. Given a video abstraction $\mathcal{A} = (S, s_0, P, L)$, we define a *path* π as a sequence of states starting from s_0. The states evolve according to the transition function P. A *prefix* is a finite path fragment starting from s_0. We define a *trace* as $\psi = \text{trace}(\pi) = L(s_0)L(s_1)L(s_2)\ldots$, where $s_0, s_1, s_2, ... \in \pi$. Traces($\mathcal{A}$) denotes the set of all traces from $\mathcal{A}$. Each trace $\psi = L(s_0)L(s_1)L(s_2)\ldots$ is associated with a probability $\mathbb{P}(\psi) = P(s_0, s_1) \times P(s_1, s_2) \times ...$

As mentioned, the privacy specification is in the form of $\square\,\tilde{\Phi}$. Hence, a privacy specification describes a *safety property* [17].

Definition 3 (Safety Property). A safety property P_{safe} is a set of traces in $(2^{AP})^\omega$ (ω indicates infinite repetitions) such that for all traces $\psi \in (2^{AP})^\omega \backslash P_{\text{safe}}$, there exists a finite prefix $\hat{\psi}$ of ψ such that $P_{\text{safe}} \cap \{\psi' \in (2^{AP})^\omega \,|\, \hat{\psi} \text{ is a prefix of } \psi'\} = \Phi$. $\hat{\psi}$ is a *bad prefix* and $\text{BadPref}(P_{\text{safe}})$ is the set of all bad prefixes with respect to P_{safe}.

A video satisfies the privacy specification if its abstract representation $\mathcal{A}$ satisfies the safety property P_{safe}, i.e., Traces($\mathcal{A}$) $\subseteq P_{\text{safe}}$. The probability that a video satisfies the specification is

$$\mathbb{P}[\mathcal{A} \text{ is safe}] = \mathbb{P}[\pi \in \text{path}(s_0) \,|\, \text{trace}(\pi) \in P_{\text{safe}}] = \sum_{\psi \in \text{Traces}(\mathcal{A}) \cap P_{\text{safe}}} \mathbb{P}(\psi). \tag{2}$$

Note that this probability is a probabilistic guarantee on a sequence of frames. According to the definition of safety property, we derive the following theorem (proof in Appendix):

Theorem 2. Consider a set of prefixes $\hat{\Psi}$ such that $\mathbb{P}\{\hat{\psi} \in \hat{\Psi} \,|\, \hat{\psi} \notin \text{BadPref}(P_{\text{safe}})\} \geq \alpha$. Let $\bar{S} \subset S$ be a subset of states in $\mathcal{A}$ such that $\mathbb{P}\{\hat{\psi}L(s) \notin \text{BadPref}(P_{\text{safe}}) \,|\, \hat{\psi} \notin \text{BadPref}(P_{\text{safe}}) \text{ and } s \in \bar{S}\} \geq \beta$. Then, $\mathbb{P}\{\hat{\psi}L(s) \notin \text{BadPref}(P_{\text{safe}}) \,|\, s \in \bar{S} \text{ and } \hat{\psi} \in \hat{\Psi}\} \geq \alpha\beta$.

From Theorem 2, we can compute a new probabilistic guarantee on a sequence of frames after each incoming frame. However, the length of the abstraction's prefixes increases as the stream continues, leading to high complexity. Therefore, we derive the following proposition to show that Theorem 2 holds even if we fix the length of the prefixes (proof of the proposition is in the Appendix):

Proposition 1. Let $\hat{\psi}_T$ and $\hat{\psi}_F$ be single element prefixes whose corresponding paths only consist of one state such that $\hat{\psi}_T \notin \text{BadPref}(P_{\text{safe}})$ and $\hat{\psi}_F \in \text{BadPref}(P_{\text{safe}})$. Let $\mathbb{P}(\hat{\psi}_T) = \alpha$, $\mathbb{P}(\hat{\psi}_F) = 1 - \alpha$, $\Psi' = \{\psi_T, \psi_F\}$, then if we replace $\hat{\Psi}$ with Ψ' in Theorem 2, the Theorem still holds.

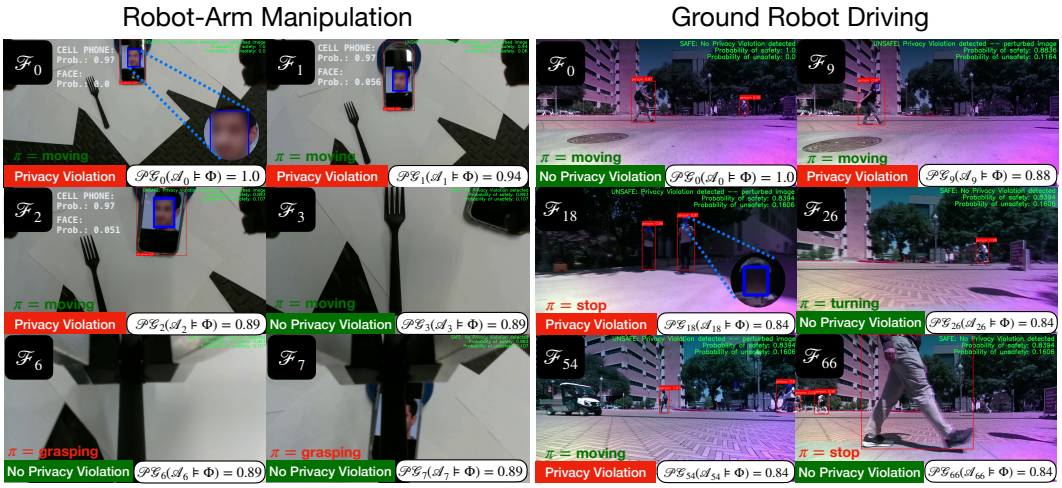


Figure 3: **Robot-Arm Manipulation**: We demonstrate that a Franka robot arm is capable of grasping a target object, a fork, using a privacy-constrained video with the privacy specification $\Phi = \Box(\neg\text{face})$. The operation's probabilistic guarantee of privacy preservation is 0.89. **Ground Robot Driving**: Our ground robot can drive autonomously based on visual observations while adhering to the privacy specification $\Phi = \Box(\text{person} \rightarrow \neg\text{face})$. It stops driving when a person is detected at a short distance. The driving operation has a probabilistic guarantee of privacy preservation at 0.84. Both demonstrations effectively maintain privacy above the user-given privacy threshold of 0.80, denoted as $\mathcal{PG}_k(\Phi) > 0.80$. In addition, non-private visual features such as "cell phone" and "person" are preserved and detected by $\mathcal{M}_{vl}$.

From Theorem 2 and Proposition 1, we can compute $\mathcal{PG}_k(\mathcal{A}_k \models \Phi)$ as follows:

$$\mathcal{PG}_k(\mathcal{A}_k \models \Phi) = \mathcal{PG}_{k-1}(\mathcal{A}_{k-1} \models \Phi) \times \left(\sum_{\sigma \models \tilde{\Phi}} \prod_{p \in \sigma} f_C(\mathcal{M}_{vl}(\mathcal{F}_k, p)) \right) \tag{3}$$

The video abstraction captures all previous frames in only two states (States 1 and 2 in Figure 2) instead of accumulating states for every frame in the sequence. Thus, we can efficiently update the guarantee through a single computation. In Figure 2, $\mathcal{PG}_k(\mathcal{A}_k \models \Phi) = 0.8 \times 0.7 = \mathbf{0.56}$.

4 Real Robot Demonstration

We demonstrate our approach on two real robots: a ground robot (Clearpath Jackal) for autonomous driving and a robot-arm (Franka) for manipulation (see Figure 3). Given video streams from robot cameras, we aim to execute actions based on the control policy (π). Our approach effectively preserves privacy with formal guarantees without compromising performance in the real-time robot operation. We use YOLOv9 [18] in our method for both demonstrations. Additional demonstrations on real robots and simulations with complex privacy specifications such as $\Box((\text{bicycle} \rightarrow \neg\text{person}) \wedge (\text{car} \vee \text{bus} \rightarrow \neg\text{person}))$ are presented in the Appendix.

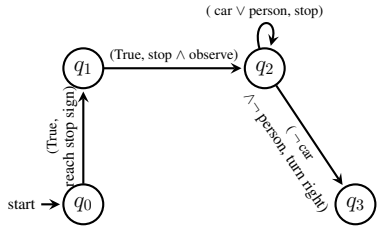
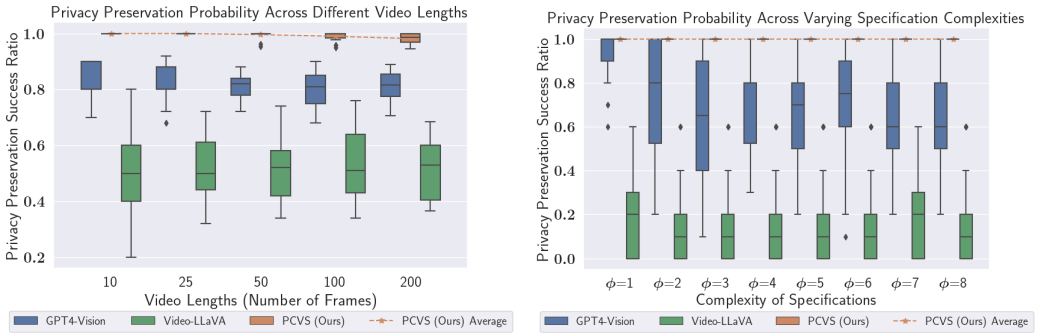


Figure 4: **A sample control policy for the ground robot.** Each transition is associated with an (input, output) tuple.

Robot-Arm Manipulation. The policy requires vision-based observation from a camera sensor and is designed to grasp a specific object. The privacy specification in this demonstration is $\Phi = \Box(\neg\text{face})$. PCVS efficiently ensures real-time privacy preservation by effectively removing private information such as faces. The robot arm's control policy utilizes this constrained-visual observation to grasp the object as presented in Figure 3.



(a) (b)

Figure 5: PCVS **effectively maintains privacy in long-horizon videos and complex privacy specifications**. In Figure 5a, PCVS consistently preserves privacy, achieving an average Privacy Preservation Success Ratio of 0.97 across various video lengths. In Figure 5b, we show that PCVS consistently upholds privacy regardless of the complexity of specifications with an average Privacy Preservation Success Ratio of 0.94.

Ground Robot Driving. We deploy the control policy on the ground robot for five driving tasks, such as turning right at a stop sign (as presented in Figure 4). We embed PCVS in the robot's camera to conceal sensitive objects during real-time operation.

5 Quantitative Analyses

We present quantitative analyses in two areas: preserving privacy and preserving non-private visual features. We use YOLOv9 [18] and large-scale image datasets, ImageNet [19] and MS COCO [20]. Our analyses show PCVS can preserve privacy even for long-horizon videos with complex privacy specifications. We define the **complexity of specifications**, ϕ, as the number of propositions in Φ. For instance, the complexity of a specification $\Phi = \square(\neg p_1 \wedge \neg p_2 \vee \neg p_3)$ with propositions $AP = \{p_1, p_2, p_3\}$ is $\phi = 3$.

Evaluation Dataset I (ED1): We focus on the presence of "person" in videos. We select images of a person from the ImageNet dataset and randomly insert these images at various positions for each video duration, filling any remaining slots with random images. We produce five different video lengths: 10, 25, 50, 100, and 200, with 25 video samples for each duration, resulting in 125 video samples overall.

Evaluation Dataset II (ED2): We use the MS COCO dataset to evaluate our method at different complexities of specifications because it has multiple labels per image. We randomly select images based on the complexity of the specifications. For example, if $\phi = 3$, the privacy specification for the dataset is $\Phi = \neg p_1 \wedge \neg p_2 \wedge \neg p_3$, where p_1, p_2, and p_3 are the ground truth labels of the selected image. These images are then randomly placed within the dataset, and the remaining slots are filled with random images.

Benchmarks: We compare our method to the benchmark for privacy preservation by assessing their ability to detect privacy violations based on a given privacy specification. We use GPT4-Vision [21] and Video LLaVA [22] because they can process a sequence of images from a video alongside a privacy specification. We present the prompts used for our experiments in the Appendix.

Privacy Preservation. We quantitatively evaluate the performance of privacy preservation across varying video lengths and specification complexities. For this evaluation, we define a metric representing the success ratio of privacy preservation as follows:

$$\text{Privacy Preservation Success Ratio} = \frac{\text{Number of } p_i \in AP \text{ detected or blurred}}{\text{Total number of labeled } p_i \in AP \text{ within } \mathcal{V}}.$$

Comparison by video length: Our method ensures privacy preservation in live video streams, which means the video length can be infinite. Hence, it is crucial to preserve privacy longer in the streams.

We test our method on various video with various lengths from ED1. Our method consistently maintains performance in preserving privacy, in contrast to benchmark methods that exhibit degraded performance as the length of videos increases (See Figure 5a).

Comparison by specification complexities: Next, we assess PCVS based on the complexity of specifications. This comparison is important because a privacy specification can be intricate, involving more than just two or three propositions. For example, a specification might require the detection and concealment of multiple privacy-sensitive objects within the same video, such as faces, license plates, and specific types of clothing. Our method significantly outperforms benchmark methods (See Figure 5b) regardless of the complexity of specifications. We demonstrate that PCVS can effectively handle highly complex privacy compositions in real-time video streams, ensuring robust privacy protection.

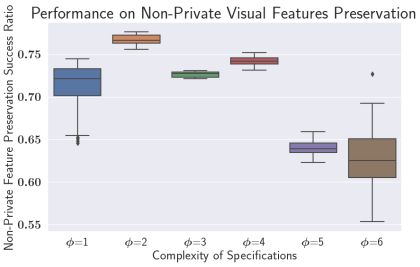


Figure 6: **Preserving non-private visual features for vision-based robot operation.** Our method can detect non-private objects after concealing private objects specified in Φ. However, performance degrades from $\phi = 5$ because more private objects get concealed.

Non-private Visual Feature Preservation. Preserving non-private visual features is crucial for vision-based robot operation, as it relies on visual observation for control policies. In our demonstration (as presented in Figure 3), the robot arm must recognize a fork from privacy-constrained video streams. Similarly, the ground robot must be capable of identifying people from privacy-constrained video footage to make appropriate decisions, such as stopping. We analyze how our method preserves non-private visual features using ED2 in Figure 6. We define a metric that represents the success ratio of preserving non-private visual features as follows:

$$\text{Non-Private Feature Preservation Success Ratio} = \frac{\text{Number of } \chi \text{ detected after blurring } p_i \in AP}{\text{Total number of labeled } \chi \text{ within } \mathcal{V}},$$

where χ is a non-private target object for detection. In our evaluation, non-private visual features remain preserved and detectable even after the concealment of privacy sensitive objects as defined in the privacy specifications. However, the success ratio of non-private preservation decreases as the complexity of these specifications increases. This is because PCVS conceals a larger image area as the number of privacy-sensitive objects increase.

6 Conclusion

We propose PCVS, a method to protect privacy in a live video stream that is either generated from robotic tasks or fed to robot learning algorithms with a probabilistic guarantee of the privacy specification being satisfied. Our method significantly outperforms state-of-the-art methods for short and long video sequences, achieving a 95% specification satisfaction rate. Further, we demonstrate the real-time capabilities of our algorithm on two real robots.

Limitations and Future Work. Our framework is limited by the capabilities of the VLM that we use. We are currently unable to process specifications that are action-based in nature (for example, remove humans who are seen eating in a video) because of the limited performance of the VLM (in the experiment) in detecting actions. In the future, we plan to address this limitation by integrating models that process multiple frames at a time to detect whether an action has occurred. Another future direction is extending our privacy specifications to generic specifications that can describe more properties besides safety, such as liveness and fairness.

References

[1] H. Kuehne, H. Jhuang, E. Garrote, T. A. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. V. Gool, editors, *IEEE International Conference on Computer Vision*, pages 2556–2563. IEEE Computer Society, 2011.

[2] E. Guo. A roomba recorded a woman on the toilet. how did screenshots end up on facebook?, Mar 2024. URL https://www.technologyreview.com/2022/12/19/1065306/roomba-irobot-robot-vacuums-artificial-intelligence-training-data-privacy/.

[3] F. Cangialosi, N. Agarwal, V. Arun, S. Narayana, A. Sarwate, and R. Netravali. Privid: practical,{Privacy-Preserving} video analytics queries. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 209–228, 2022.

[4] S. M. M. Rahman, M. A. Hossain, H. Mouftah, A. El Saddik, and E. Okamoto. Chaos-cryptography based privacy preservation technique for video surveillance. *Multimedia systems*, 18:145–155, 2012.

[5] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar. Efficient privacy preserving video surveillance. In *2009 IEEE 12th international conference on computer vision*, pages 1639–1646. IEEE, 2009.

[6] A. Padmanabhan, N. Agarwal, A. Iyer, G. Ananthanarayanan, Y. Shu, N. Karianakis, G. H. Xu, and R. Netravali. Gemel: Model merging for {Memory-Efficient},{Real-Time} video analytics at the edge. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 973–994, 2023.

[7] N. Sugianto, D. Tjondronegoro, R. Stockdale, and E. I. Yuwono. Privacy-preserving ai-enabled video surveillance for social distancing: Responsible design and deployment for public spaces. *Information Technology & People*, 37(2):998–1022, 2024.

[8] D. Kagan, G. F. Alpert, and M. Fire. Zooming into video conferencing privacy. *IEEE Transactions on Computational Social Systems*, 2023.

[9] E. Umili, R. Capobianco, G. De Giacomo, et al. Grounding ltlf specifications in images. In *Proceedings of the 16th International Workshop on Neural-Symbolic Learning and Reasoning*, pages 45–63, 2022.

[10] Y. Yang, J.-R. Gaglione, S. Chinchali, and U. Topcu. Specification-driven video search via foundation models and formal verification. *arXiv preprint arXiv:2309.10171*, 2023.

[11] M. Choi, H. Goel, M. Omama, Y. Yang, S. Shah, and S. Chinchali. Neuro-symbolic video search. *arXiv preprint arXiv:2403.11021*, 2024.

[12] Y. Cheng, Q. Fan, S. Pankanti, and A. Choudhary. Temporal sequence modeling for video event detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2227–2234, 2014.

[13] N. Rescher and A. Urquhart. *Temporal logic*, volume 3. Springer Science & Business Media, 2012.

[14] J. Barwise. An introduction to first-order logic. In *Studies in Logic and the Foundations of Mathematics*, volume 90, pages 5–46. Elsevier, 1977.

[15] G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[16] J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald. Formal methods: Practice and experience. *ACM Comput. Surv.*, 41(4), oct 2009. ISSN 0360-0300. doi:10.1145/1592434.1592436. URL https://doi.org/10.1145/1592434.1592436.

[17] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT press, 2008.

[18] C.-Y. Wang and H.-Y. M. Liao. YOLOv9: Learning what you want to learn using programmable gradient information. 2024.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll'a r, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

[21] OpenAI. Gpt-4 vision system card. https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023. Accessed: [insert date of access].

[22] B. Lin, Y. Ye, B. Zhu, J. Cui, M. Ning, P. Jin, and L. Yuan. Video-llava: Learning united visual representation by alignment before projection, 2023.