LDP-FEAT: Image Features with Local Differential Privacy

Francesco Pittaluga
francescopittaluga@nec-labs.com

Bingbing Zhuang
bzhuang@nec-labs.com

NEC Labs America

Abstract

Modern computer vision services often require users to share raw feature descriptors with an untrusted server. This presents an inherent privacy risk, as raw descriptors may be used to recover the source images from which they were extracted. To address this issue, researchers [11] recently proposed privatizing image features by embedding them within an affine subspace containing the original feature as well as adversarial feature samples. In this paper, we propose two novel inversion attacks to show that it is possible to (approximately) recover the original image features from these embeddings, allowing us to recover privacy-critical image content. In light of such successes and the lack of theoretical privacy guarantees afforded by existing visual privacy methods, we further propose the first method to privatize image features via local differential privacy, which, unlike prior approaches, provides a guaranteed bound for privacy leakage regardless of the strength of the attacks. In addition, our method yields strong performance in visual localization as a downstream task while enjoying the privacy guarantee.

1. Introduction

The extraction and matching of image keypoints with descriptors is an essential building block for many vision problems, such as 3D reconstruction [2], image retrieval [28] and recognition [40]. Thus, modern computer vision services often require the users to share feature descriptors and/or raw images to a centralized server for downstream tasks, such as visual localization [38]. However, recent works [30, 37] show that high-quality images may be recovered from the keypoint descriptors [30] or their spatial information [37], raising serious concerns regarding the potential leakage of private information via inversion attacks.

This in turn inspires great interest in researching feature obfuscation with a view to concealing the privacy critical information in the image, mainly by perturbing either the feature descriptors or their locations. One of the recent works [11] represents a feature descriptor point as an affine subspace passing through the original point, as well as a number of other adversarial descriptors randomly sampled from a database of descriptors, as shown in Fig. 1. These adversarial descriptors serve as confounders to conceal the raw descriptor. Another line of research [38, 39, 15, 36] aims to conceal the location of 2D or 3D keypoints by lifting the point to a line passing through that point, which prevents a direct attack of the sort in [30, 37].

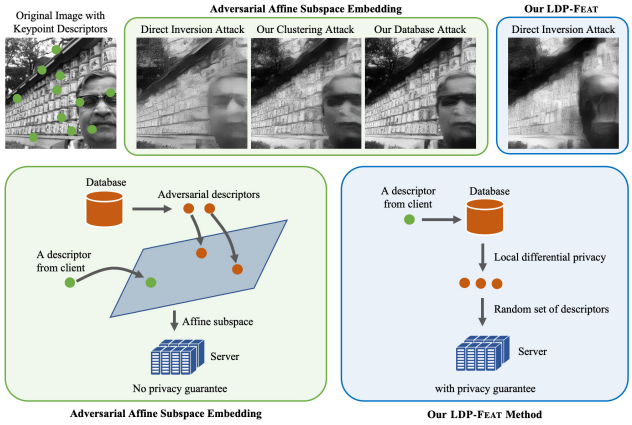

Figure 1. **Our novel privacy attacks and LDP-based privacy method.** Top row: image reconstruction attacks against adversarial affine subspace embeddings [11] and our LDP-FEAT when they have comparable performance in the downstream utility task. Bottom row: overviews of the adversarial affine subspace embeddings algorithm [11] and our LDP-FEAT algorithm

Despite their success, these works are primarily evaluated on the basis of empirical performance of a chosen attacker, without rigorous understanding of the attacker-independent, intrinsic privacy property. This causes hindrance for a method to claim privacy protection safely since there is no theoretical guarantee to assure practical applications. For instance, [7] re-investigates the privacy claim in [38] and designs a stronger attack to reveal that a significant amount of scene geometry information in fact still exists in the lifted line clouds, which can be leveraged to recover sensitive image content. In this paper, we focus on the

feature descriptor and, similar in spirit to [7], we reveal the privacy leakage in the affine subspace lifting [11]. Considering the drawbacks of the visual privacy-based method, we present the first attempt of its kind to formulate the privacy protection of image features through the lens of differential privacy [44], which permits theoretic privacy characterization, enjoys a guaranteed bound on privacy loss, and has become a gold standard notion of privacy.

More specifically, we firstly introduce two novel attacks against the adversarial affine subspace embedding [11], namely the *database attack* and the *clustering attack*. In the database attack, we assume that the database used to sample the adversarial descriptors is accessible to the attacker, while in clustering attack we relax this assumption. At its core, both attacks are established based upon the following key assumption: the low-dimensional (*e.g.* 2,4,8) affine subspace very likely only intersects with the manifold of high-dimensional (*e.g.* 128 for SIFT [23]) descriptors at those points that were intentionally selected to construct the subspace in the beginning, i.e. the raw descriptor to be concealed and the adversarial ones chosen from the database. Tha main idea of our attacks lies in identifying these intersections and further eliminating the adversarial ones. As shown in Fig. 1, our attacks recover images of higher quality than the direct inversion attack shown in [11].

Next, we propose LDP-FEAT, a novel descriptor privatization method that rests on the notion of local differential privacy (LDP) [50], as illustrated in Fig. 1. In contrast to the original centralized differential privacy which prevents private information in the database from releasing to queries, we instead aim to protect privacy in the query itself, *i.e.* the image descriptors to be sent. We propose to formulate the feature obfuscation by local differential privacy, with the so-called ω-subset mechanism [44] – we effectively replace each raw descriptor with a random set of descriptors under predefined probability distribution that endows the rigorous and quantifiable differential privacy guarantee. Further, our database and clustering attack are not applicable to LDP-FEAT, and the direct inversion attack largely fails on LDP-FEAT, as shown in Fig. 1. We demonstrate strong performance in visual localization as a downstream task while enjoying the advantageous privacy guarantee.

In summary, our contributions include:

- Two novel attacks on adversarial affine subspace embeddings [11] that enable (approximate) recovery of the original feature descriptors.
- A novel method for image feature privatization that rests on local differential privacy with favorable privacy guarantees.
- Advantageous privacy-utility trade-offs achieved via empirical results to support practical applications.

2. Related Work

Feature descriptors. Feature descriptors extracted from image key points are used for a range of computer vision tasks such as 3D scene reconstruction [2], image retrieval [28] and recognition [40]. Traditional methods for extracting such descriptors were handcrafted based on direct pixel sampling [6] or histograms of image gradients [23, 8]. More recently, a growing number of methods rely on deep learning to extract the feature descriptors [26, 18, 3].

Inverting image features. The task of reconstructing images from features has been explored to understand what is encoded by the features, as was done for SIFT by [46], HOG features by [42] and bag-of-words by [21]. Recent work has been primarily focused on inverting and interpreting CNN features [52, 51, 24]. Dosovitskiy and Brox [10] proposed encoder-decoder CNN architectures for inverting many different features and later incorporated adversarial training with a perceptual loss [9], primarily focusing on dense features. Pittaluga et al. [30] focus on inverting sparse SIFT descriptors stored along with structure-from-motion point clouds, recovering high-quality image content by training feature inversion networks. Song et al. [37] further demonstrate image recovery from just colored 3D point clouds without the descriptors. The capability enabled by these works raise significant privacy concerns, which further motivate research in privacy-preserving visual representations.

Visual privacy. McPherson et al. [25] and Vasiljevic et al. [41] showed that deep models could defeat conventional image obfuscation methods such as blurring and pixelation. To defend against CNN-based attacks, adversarial optimization has been employed to learn CNN-resistant image encodings for action recognition [47, 43], face attribute recognition [48], place recognition [29], and more [5]. Researchers also developed privacy-preserving representations for image-based localization and mapping, which is tackled from two angles: (i) 2D and 3D keypoint obfuscation [38, 39, 36, 16, 7, 15, 14] by concealing the position information of keypoints, and (ii) image feature/descriptor obfuscation [11, 27] by concealing the descriptor of keypoints. The main idea for keypoint obfuscation lies in lifting a point to a random line or plane, while descriptor obfuscation lifts the descriptor to an affine subspace [11], or directly learns attack-resistant descriptors by adversarial training [27]. Despite their empirical effectiveness, these methods do not provide theoretical guarantee and characterization on the privacy protection. In this paper, we first introduce two novel attacks against [11] that reveal its privacy leakage, and then propose a new feature privatization method that provides formal guarantees via local differential privacy.

Differential privacy. In recent years, differential privacy [12, 13] has become the gold standard for publication and analysis of sensitive data. Most differential privacy research is focused on the centralized setting [17], where raw user

data is aggregated by a trusted curator who then shares the data to the public without releasing private information. Note, the curator is assumed trusted, which, however, may not be the case in practice. Instead, the local differential privacy [44, 50] provides a means for users to privatize their data locally prior to sending it out; hence a trusted curator is not required. Our work presents the first attempt to apply local differential privacy for image feature privatization.

3. Preliminary: Affine Subspace Embedding

In order to preserve privacy in keypoint descriptors, Dusmanu et al. [11] propose to "lifting" each descriptor to an adversarial affine subspace before sharing to the curator.

Subspace lifting. Let $d \in \mathbb{R}^n$ denote a descriptor to be privatized. [11] proposes lifting d to an m-dimensional affine subspace $D \subset \mathbb{R}^n$ satisfying $d \in D$, represented by a translation vector d_0 and m basis vectors $\{d_1, ..., d_m\}$, i.e. $D = d_0 + \texttt{span}(d_1, ...d_m)$.

Selection of subspace. To ensure d not be easily recoverable, subspace D must intersect the manifold of real-world descriptors at multiple points. [11] proposes that half of the basis descriptors be randomly selected from a database of real-world descriptors W, and the other half be randomly generated via random sampling from a uniform distribution, *i.e.*, setting $d_0 = d$ and $d_i = a_i - d$ for $i = \{1, ..., \frac{m}{2}\}$, where $a_i \sim \mathcal{U}\{W\}$ and $d_i \sim \mathcal{U}([-1, 1])^n$ for $i = \{\frac{m}{2}+1, ..., m\}$. This way, d and $\{a_1, ..., a_{\frac{m}{2}}\}$ are contained in D. [11] refers to this half-and-half approach as *hybrid lifting*.

Re-parameterization. Evidently, the above representation of D directly exposes the descriptors, hence [11] re-parameterizes D to prevent information leakage. First, to avoid setting the translation vector as the raw descriptor d, it randomly generates a new translation vector $d_0 = p_\perp^D(e_0)$, where $p_\perp^D(e_0)$ denotes the orthogonal projection of e_0 onto D, $e_0 \sim \mathcal{U}([-1, 1])^n$. Further, to prevent an attacker from using the direction of the basis descriptors to infer the raw descriptor d, a new set of basis descriptors $d_i = p_\perp^D(e_i)$ for $i = \{1, ..., m\}$, where $e_i \sim \mathcal{U}([-1, 1])^n$, are randomly generated. Note that the above two steps only re-parameterize D without changing its intrinsic property.

Matching. With the lifted privacy-preserving representation, [11] further proposes the use of point-to-subspace and subspace-to-subspace distances for matching raw-to-lifted and lifted-to-lifted descriptors, respectively.

4. Database and Clustering Inversion Attacks

In this section, we present two attacks against adversarial lifting [11], namely the database attack and the clustering attack. In database attack, we assume the attacker has access to the database of real-world descriptors W from which the adversarial descriptors were selected, whereas in clustering attack, the attacker has no access to the database. Both attacks are based on the following key empirical assumption: a low-dimensional hybrid adversarial affine subspace D likely only intersects the high-dimensional descriptor manifold at $\frac{m}{2}+1$ points corresponding to the original descriptor d and the adversarial descriptors $\{a_1, ..., a_{\frac{m}{2}}\}$ that were sampled from the database.

4.1. Database Attack

With the above assumption, if we can identify the $\frac{m}{2}$ subspace-manifold intersections corresponding to the adversarial descriptors $\{a_1, ..., a_{\frac{m}{2}}\}$, the recovery of descriptor d from subspace D is reduced to finding the one remaining intersection. This is exactly the outline of our attack, which we illustrate in Fig. 2(a) by a toy example with n=3 and m=2, *i.e.* subspace being a plane in $\mathbb{R}^3$.

Step 1: Compute distances to the database W. We start by computing the distances $\texttt{dist}(D, w_i) = ||w_i - p_\perp^D(w_i)||_2$ between subspace D and each descriptor in the database of real-world descriptors $w_i \in W$, and then sort the descriptors in ascending order according to their respective distances.

Step 2: Recover adversarial descriptors exactly. Recall that the adversarial descriptors are selected from W, thus $\texttt{dist}(D, w_i) = 0$ holds exactly for $i = \{1, ..., \frac{m}{2}\}$, which means the first $\frac{m}{2}$ descriptors from the sorted list immediately give our estimates $\{\hat{a}_1, ..., \hat{a}_{\frac{m}{2}}\}$ for $\{a_1, ..., a_{\frac{m}{2}}\}$.

Step 3: Estimate the concealed descriptor. Unlike the adversarial descriptors, the database does not contain the original descriptor d, but we may estimate it by its close neighbors in the database. To this end, the next $|V|$ descriptors in the sorted list, $V = \{v_1, ..., v_{|V|}\}$, $|V| << |W|$, are selected, containing the descriptors nearest to D for which $\texttt{dist}(D, v_i) > 0$. More specifically, these descriptors are near to either the adversarial descriptors or the raw descriptor, and we aim to further select a subset $U \in V$ which is close to the raw descriptor d but far from the adversarial ones. U and V are both illustrated in Fig. 2(a). To select U, a score $s_i = \min_{j=1,...,\frac{m}{2}} ||\hat{a}_j - v_i||_2$ is computed for each v_i in V. The descriptors with the highest scores, $u_1, ..., u_{|U|}$, are used to estimate d via weighted average and orthogonal projection:

$$\hat{d} = p_\perp^D \left(\frac{1}{\alpha} \sum_{i=1}^{U} \frac{u_i}{\texttt{dist}(D, u_i)} \right), \tag{1}$$

where $\alpha = \sum_{i=1}^{U} \texttt{dist}(D, u_i)^{-1}$.

Remark. The intuition for why this attack works is that any descriptors from database W that are near the subspace D will likely cluster around either the original descriptor d or one of the adversarial descriptors, as these are likely to be the only points where the subspace intersects the manifold of real-world descriptors. The effectiveness of this attack is empirically validated in Sec. 6.1.

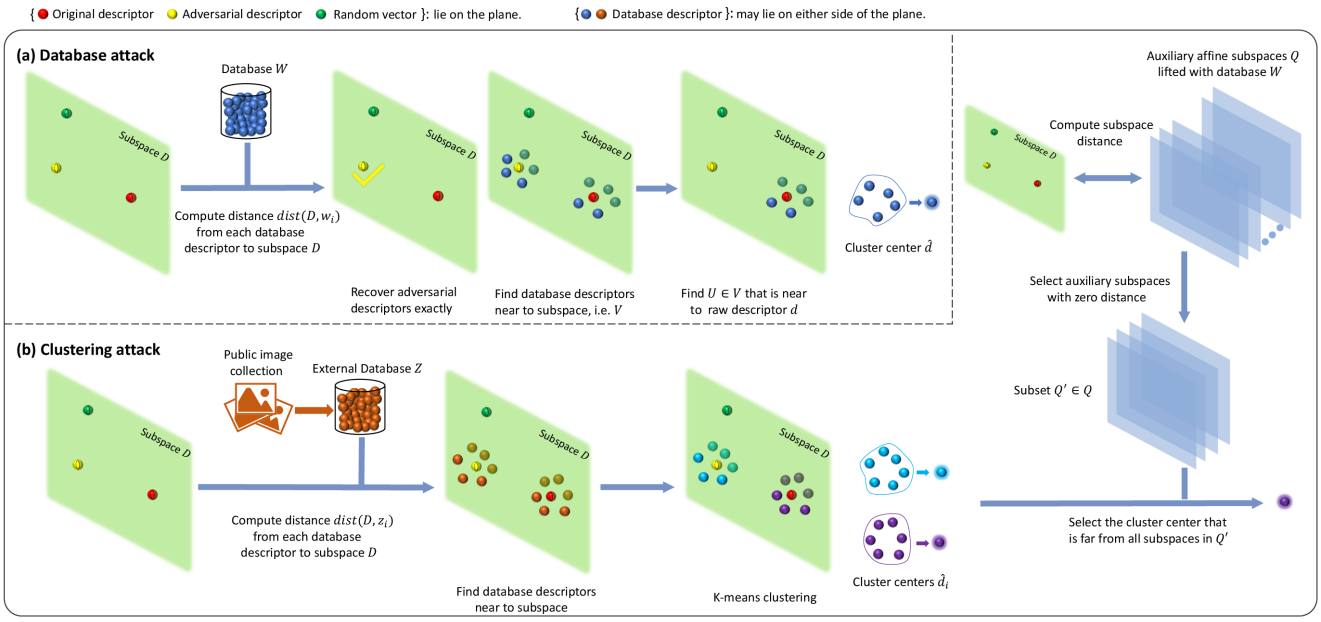

Figure 2. **Illustration of (a) database attack and (b) clustering attack**.

4.2. Clustering Attack

For this attack, we assume that the attacker does not have access to the database of real-world descriptors W from which adversarial descriptors $a_{i=1,...,\frac{m}{2}}$ are sampled, but does have access to an additional set of adversarial affine subspaces $\mathcal{Q}$ that were lifted with the same database W. They could be obtained either from other descriptors in the same image or from a set of c other images. The clustering attack is illustrated in Fig. 2(b).

Step 1: Compute distance to public database. Extract descriptors from a large set of public images and then cluster them to generate a public database of real-world descriptors Z to serve as a proxy for the private database W. Then, compute the distances $\texttt{dist}(D, z_i) = ||z_i - p^D_\perp(z_i)||_2$ between subspace D and each descriptor $z_i \in Z$.

Step 2: Select nearest neighbors. Select the $|V| \ll |Z|$ descriptors nearest to subspace D, denoted $v_{i=1,...,V}$. Note, unlike in the database attack, we can't identify $a_{i=1,...,\frac{m}{2}}$ exactly, as, in general, $\texttt{dist}(D, z_i) \neq 0$ for any i.

Step 3: Estimate candidate descriptors. Cluster V into $k = \frac{m}{2} + 1$ clusters and assign each v_i a cluster label $l_i \in \{1, ..., \frac{m}{2} + 1\}$. Similar to Eq. (1), the center of each cluster is computed separately as follows:

$$\hat{d}_i = p^D_\perp \left(\frac{1}{\alpha_i} \sum_{j=1}^{V} \mathbb{1}(l_j, i) \frac{v_i}{\texttt{dist}(D, v_i)} \right), \qquad (2)$$

where $\alpha_i = \sum_{j=1}^{V} \mathbb{1}(l_j, i) \texttt{dist}(D, j_i)^{-1}$ and $\mathbb{1}(x, y) = 1$ when $x = y$ and 0 otherwise. Note, these k descriptors represent (approximately) the intersections between subspace D and the manifold of real-world descriptors. Thus, for our attack, we assume that each descriptor $\hat{d}_i$ is near to either the original descriptor d or one of the adversarial descriptors $\{a_1, ..., a_{\frac{m}{2}}\}$. Next, we leverage the auxiliary subspace $\mathcal{Q}$ to estimate which $\hat{d}_i$ is nearest to d.

Step 4: Estimate the concealed descriptor. Recall that subspace D and the subspaces in $\mathcal{Q}$ were lifted using the same database of private descriptors W. Thus, it's likely that a subset of subspaces $\mathcal{Q}' \in \mathcal{Q}$ were lifted using one or more of the same adversarial descriptors as D, *i.e.* one of $a_i, i=\{1, ..., \frac{m}{2}\}$. We can identify this subset $\mathcal{Q}'$ by noting that each subspace in $\mathcal{Q}'_j \in \mathcal{Q}'$ intersects with D, i.e., by selecting all $\mathcal{Q}_j \in \mathcal{Q}$ for which $dist(D, \mathcal{Q}_j) = 0$. Assuming that $\mathcal{Q}$ is sufficiently large such that all a_i's were used to lift at least one of the subspaces in $\mathcal{Q}'$, this indicates that the minimal point-to-subspace distance $\min_j \texttt{dist}(a_i, Q'_j) = 0$ for $i=\{1, ..., \frac{m}{2}\}$. On the other hand, since $\mathcal{Q}'$ is selected without any knowledge or specific treatments on d, it is expected that $\min_j \texttt{dist}(d, Q'_j) \gg 0$. In this discrepancy lies the crux of our attack – while $\min_j \texttt{dist}(\hat{a}_i, Q'_j) > 0$ for our estimates of a_i, $\hat{a}_i$, we expect that $\min_j \texttt{dist}(\hat{d}, Q'_j) \gg \min_j \texttt{dist}(\hat{a}_i, Q'_j)$. Hence, we compute the score s_i for each $\hat{d}_i$ as $s_i = \min_j \texttt{dist}(\hat{d}_i, Q'_j)$ and the largest s_i yields our estimate for d. We note that it is not impossible that $\mathcal{Q}'$ may contain a database descriptor that is close to d too, but the probability of such a collision is low thanks to the high dimension of descriptors and empirically, our attack remains effective, as shown in Sec. 6.1.

Image reconstruction attack: With the recovered raw descriptors by our database/clustering attack, one may perform an image inversion attack of the sort described in [30].

5. Our LDP-FEAT

The success of our inversion attacks motivates the need for an image feature privatization method with rigorous privacy guarantee. We present the first solution towards this goal with local differential privacy.

5.1. Preliminary: Local Differential Privacy

Unlike original differential privacy [12], the local differential privacy (LDP) setting allows users to sanitize their data locally before sending to a curator, so the curator needs not be trusted. Here, we describe necessary definitions of LDP and refer readers to [49] for detailed derivations.

Definition 1 (Local Differential Privacy) *A randomized mechanism $\mathcal{M}$ satisfies ϵ-local differential privacy (ϵ-LDP), where $\epsilon \geq 0$, if and only if for any inputs x_1 and x_2,*

$$\forall y \in Range(\mathcal{M}) : \frac{\Pr[\mathcal{M}(x_1) = y]}{\Pr[\mathcal{M}(x_2) = y]} \leq e^{\epsilon}, \qquad (3)$$

where $Range(\mathcal{M})$ denotes the set of all possible outputs of $\mathcal{M}$. Note that $\mathcal{M}$ maps the input to a probability distribution rather than a single point. The ϵ controls the similarity in the output, and is termed as the *privacy budget* – a smaller ϵ indicates higher privacy protection, and vice versa. To illustrate this, we note that according to the definition of LDP, Eq. (3) holds too if we swap x_1 and x_2, *i.e.* $\Pr[\mathcal{M}(x_2){=}y] \leq e^{\epsilon} \Pr[\mathcal{M}(x_1){=}y]$. When $\epsilon{=}0$, it follows that $\Pr[\mathcal{M}(x_2){=}y]{=}\Pr[\mathcal{M}(x_1){=}y]$. This means x_1 and x_2 have an identical distribution after $\mathcal{M}$ perturbation, and are indistinguishable from each other, hence yielding strongest privacy protection. Conversely, a larger ϵ loosens the constraint in Eq. (3) and reduces privacy protection.

Selection of ϵ. While ϵ may be set as any value, it is commonly set within $[0.01, 10]$, which was shown to ensure good privacy protection in practice [19, 50].

Definition 2 (ω-Subset Mechanism) Denoting the data domain by $\mathcal{K}$, for any input $v \in \mathcal{K}$, randomly report a ω-sized subset $\mathcal{Z}$ of $\mathcal{K}$, *i.e.* $\mathcal{Z}{\subset}\mathcal{K}$ and $|\mathcal{Z}|{=}\omega$, with probability

$$\Pr(\mathcal{Z}|v) = \begin{cases} \frac{\omega e^{\epsilon}}{\omega e^{\epsilon}+|\mathcal{K}|-\omega} / \binom{|\mathcal{K}|}{\omega}, & \text{if } v \in \mathcal{Z}, \\ \frac{\omega}{\omega e^{\epsilon}+|\mathcal{K}|-\omega} / \binom{|\mathcal{K}|}{\omega}, & \text{if } v \notin \mathcal{Z}. \end{cases} \qquad (4)$$

The ω-Subset Mechanism (ω-SM) satisfies ϵ-LDP [45, 50]. It is important to note that the data domain $\mathcal{K}$ is required to be a finite space, *i.e.* consisting of countably many elements. In what follows, we formulate our image feature perturbation as ω-SM for privacy guarantee.

5.2. Image Feature Matching with LDP

Overview Similarly to affine subspace lifting [11], we apply LDP to perturb feature descriptors before sending to the curator, and the curator applies feature matching with RANSAC-based geometric verification to enable downstream tasks despite the perturbation. We note the privacy-utility trade-off here – a larger perturbation increases privacy protection but causes more significant challenges for correct matching. This trade-off is controlled in our framework by ϵ, which corresponds to a guaranteed bound of privacy loss. Next, we present in detail our LDP protocol for image features by leveraging the ω-Subset Mechanism.

Naive Approach (LDP on the full descriptor space). A straightforward approach is to apply ω-SM directly on the descriptor space for obfuscation – define $\mathcal{K}$ as the set of all possible descriptors and randomly report a subset of descriptors, which contain the raw descriptor with some probability. This is applicable to image descriptors as they have a finite domain size $|\mathcal{K}|$ as required by ω-SM: $|\mathcal{K}| = 2^{8\times128}$ for 128-dim uint8-based descriptors (*e.g.* SIFT), and $|\mathcal{K}| = 2^{32\times128}$ for 128-dim float32-based descriptors (*e.g.* HardNet [26]). However, as we shall demonstrate in Sec. 6, naively setting the output space to the full descriptor domain does not lead to a desirable privacy-utility trade-off. This is caused by the domain size being too large; we will explain this shortly after introducing the following domain with a smaller size.

Our LDP-FEAT (LDP on a dictionary of descriptors). We instead define the data domain $\mathcal{K}$ as a finite dictionary of descriptors established from real-world image collections; this dictionary serves as the database shared with all users. More specifically, the database is created by extracting descriptors from a large public database of images and then performing k-means, as in [11]. Locally, each user enforces differential privacy by the following steps.

Step 1: Replace each descriptor d that is to be sent to the curator with its nearest neighbor $d' \in \mathcal{K}$.

Step 2: Then, d' is replaced with a set of descriptors $\mathcal{Z} \subset \mathcal{K}$ of size m. We perform random sampling [45] to generate the set $\mathcal{Z}$ that satisfies the probability distribution in Eq. (4): first sample a Bernouilli scale variable u with

$$Pr(u = 1) = \frac{me^{\epsilon}}{me^{\epsilon} + |\mathcal{K}| - m}; \qquad (5)$$

then randomly sample $m - u$ descriptors $\mathcal{Y}$ from $\mathcal{K} - \{d'\}$.

Step 3: if $u = 1$, $\mathcal{Z} = \mathcal{Y} \cup \{d'\}$ else $\mathcal{Z} = \mathcal{Y}$.

This approach satisfies ϵ-LDP (See supplementary for proof). Note that the curator will receive multiple descriptors per keypoint. The curator can then discover which, if any, of the matches for a given keypoint are correct by performing RANSAC-based geometric verification. Despite the perturbation on the descriptor, good empirical performance is still observed in downstream tasks, as shown in Sec. 6.

Why domain size matters? Referring to Eq. (5), it is clear that, with a fixed value of ϵ, an extremely large value of $|\mathcal{K}|$ renders $Pr(u = 1)$ extremely small – which severely

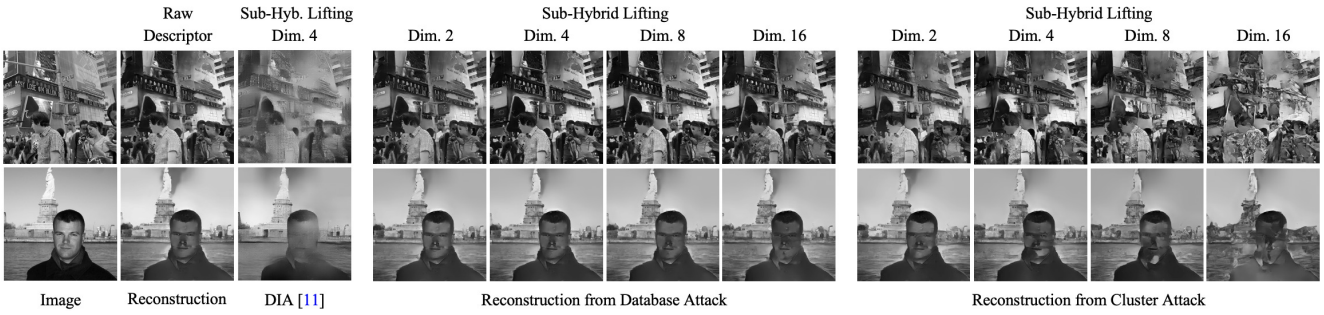

Figure 3. **Inverting Adversarial Affine Subspace Embeddings.**

limits the sending of raw descriptors to the server. This implies a very low proportion of inlier correspondences, which hinders utility. On the other hand, one may observe that increasing ϵ in tandem with $|\mathcal{K}|$ may prevent $Pr(u = 1)$ from dropping, however, a larger ϵ quickly reduces the strength of privacy protection; recall that ϵ typically is confined within $[0.01, 10]$ for practical usage [19, 50]. As such, too large a domain size may cause poor privacy-utility trade-off. Similarly, increasing m, the number of descriptors sent to the server, prevents $Pr(u = 1)$ from dropping at the cost of reducing the proportion of inliers, which inhibits utility too. These factors motivate our design choice to adopt a dictionary of descriptors-based approach.

Theorem 1 (LDP-FEAT satisfies ϵ-LDP). Our LDP-FEAT is not strictly a ω-SM per se, because of the preceding nearest neighbor mapping – we first map the raw input d to its nearest neighbor d' in the database $\mathcal{K}$, and then apply ω-SM on top of d'. We prove in supplementary that LDP-FEAT still satisfies ϵ-LDP.

Relation to affine subspace embeddings. While our method is similar to [11] in that users obfuscate descriptors by hiding them among a set of confounder descriptors randomly sampled from a database, there are two critical differences. Firstly, in our method, the set of descriptors $\mathcal{Z}$ sent by each user to the curator must be a subset of finite vocabulary $\mathcal{K}$; recall that the original descriptor d, if included, is also replaced by its nearest neighbor in $\mathcal{K}$. Hence, even if $\mathcal{K}$ is exactly known by a malicious curator, he cannot use $\mathcal{K}$ to perform a database attack of the sort described in Sec. 4.1, and the same holds for the clustering attack when $\mathcal{K}$ is not accessible. Secondly, thanks to careful design of the obfuscation protocol, our method enables rigorous accounting of privacy leakage via local differential privacy, with a guaranteed bound of privacy leakage irrespective of the strength of attacks.

6. Experimental Results

In this section, we first evaluate the efficacy of our database and clustering attack, and then evaluate our LDP-FEAT with respect to both utility and privacy.

6.1. Inversion Attacks on Adversarial Affine Subspace Embeddings

Evaluation setup. Our experimental setup is similar to [11]. For all evaluations, we employ sub-hybrid adversarial lifting of SIFT descriptors using an adversarial lifting database that was generated by clustering 10 million local features from 60,000 images from the Places365 dataset [53] into 256,000 clusters using spherical k-means [4]. As in [11], the 256,000 descriptors in the adversarial database are split into 16 sub-databases. As described in Sec. 4, in *database attack* the adversary has access to this exact adversarial lifting database, whereas in *clustering attack*, the adversary only has access to a public database of 128,00 descriptors, that was generated using the same process as above, but from a different set of 60,000 images. Additionally, as in [11], we train a U-Net [33] style CNN for image reconstruction from descriptors on the MegaDepth dataset [22] using the same architecture and loss as [30]. For all reconstructions, we report the following image reconstruction quality metrics: mean absolute error (MAE), structural similarity index measure (SSIM), and peak signal-to-noise ratio (PSNR).

Image reconstruction attack. For this evaluation, a set of descriptor-keypoint pairs is extracted from a source image, transformed into a set of subspace-keypoint pairs using sub-hybrid adversarial lifting [11], and then sent to the adversary. The goal of the adversary is to reconstruct the source image from the received subspace-keypoint pairs. To achieve this, we first employ either the *database attack* or the *cluster attack* independently on each subspace-keypoint pair to recover an estimate of the original descriptor. The descriptors so obtained are then organized into a sparse feature map $F \in \mathbb{R}^{256\times256\times128}$ using the respective keypoint locations; pixels that do not have any associated descriptor are set to zero. Feature map F is then provided as input to a pre-trained U-Net style reconstruction network to recover the original source image.

Results. We evaluate the efficacy of the attacks on the

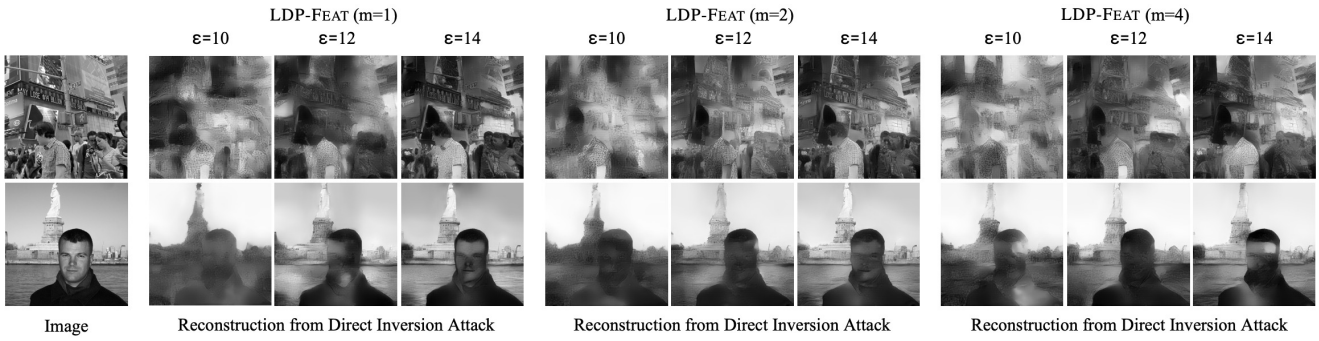

Figure 4. **Direct Inversion Attack on LDP-FEAT.** ($|\mathcal{K}| = 256k$).

Attack	Dim.	MAE ↓	SSIM ↑	PSNR ↑
Nearest [11]	2	.1690	.3611	14.00
	4	.1913	.3273	12.84
	8	.1985	.2481	12.29
	16	.1873	.2457	12.76
DIA [11]	2	.1194	.5005	16.35
	4	.1468	.4190	14.79
	8	.1635	.3676	14.01
	16	.1761	.3496	13.43
Clustering (Ours)	2	.1087	.5919	17.30
	4	.1142	.5652	16.51
	8	.1254	.5160	15.83
	16	.1424	.4122	14.73
Database (Ours)	2	.0947	.6566	18.26
	4	.0950	.6543	18.27
	8	.1000	.6385	17.89
	16	.1063	.5882	17.47
Raw	n/a	.0913	.6878	18.59

Table 1. **Inverting Adversarial Affine Subspace Embeddings.**

10 holiday images from Flickr selected by [11]. For comparison, we also report the results of two other approaches from [11]: *Nearest* and *Direct inversion attack (DIA)*. For *Nearest*, each adversarial subspace is replaced with its nearest neighbor in the public database described above. For *DIA*, the reconstruction network is trained to recover images directly from the affine subspace parameters. As shown by the example qualitative results in Fig. 3, our database attack is able to recover high-quality image content. The clustering attack lags slightly behind, but still reveals a significant amount of private information. We report quantitative reconstruction quality in Tab. 1, for various lifting dimensions. As a reference for upper bound, we also present results from *Raw*, where the original raw descriptor is input to the reconstruction network. As can be seen, both of our attacks are capable of recovering high-quality images from the estimated original descriptor, even for an adversarial dimension of 16.

	DB Size	Privacy	# Desc.	Day		
	$\lvert\mathcal{K}\rvert$	ϵ	m	0.25m, 2°	0.5m, 5°	5.0m, 10°
Accuracy Upper Bound	2^{1024}	∞	1	84.1	91.7	96.4
	1024k	∞	1	79.7	89.9	94.9
	512k	∞	1	78.0	87.4	93.3
	256k	∞	1	76.8	86.3	91.6
	128k	∞	1	73.2	82.8	88.1
Impact of Database Size	2^{1024}	10	2	0.00	0.00	0.00
	1024k	10	2	33.3	37.1	42.1
	512k	10	2	37.4	43.2	48.1
	256k	10	2	42.1	49.3	54.4
	128k	10	2	39.4	45.4	50.2
Privacy Guarantee	512k	16	4	76.1	85.0	90.4
	512k	14	4	73.9	84.5	90.2
	512k	12	4	69.4	77.9	84.6
	512k	10	4	42.1	49.6	55.0
	256k	16	2	75.4	85.3	90.2
	256k	14	2	75.1	84.7	89.4
	256k	12	2	69.5	78.4	84.1
	256k	10	2	42.1	49.3	54.4
Impact of Subset Size	256k	10	1	34.6	39.7	44.7
	256k	10	2	42.1	49.3	54.4
	256k	10	4	42.1	49.0	55.1
	256k	10	8	39.1	46.4	51.8
	256k	10	16	32.8	38.0	44.3

Table 2. **Aachen Day-Night Localization Challenge.**

6.2. Applying LDP-FEAT to Visual Localization

As in [11], we evaluate LDP-FEAT on the task of visual localization on a pre-built map from the Aachen Day-Night long-term visual localization challenge [1]. Following [11], we privatize descriptors extracted from the query images, but not the reference images, to simulate an application aiming to protect the user privacy in an image-based localization service, such as Google Visual Positioning System [31] or Microsoft Azure Spatial Anchors [20].

Preliminaries. We generate the pre-built map by extracting SIFT descriptors from the reference images and triangulating the database model from the camera poses and intrinsics provided by [1]. Next, we privatize the raw descriptors of all query images using LDP-FEAT. For matching, we retrieve the top 20 reference images for each query image

Ablation	Variable	Aachen Day		
		0.25m, 2°	0.5m, 5°	5.0m, 10°
Matching Algorithm	voc-match	42.1	49.3	54.4
	mutual-nn	18.3	22.0	26.1
Public Database	Aachen Ref.	42.1	49.3	54.4
	Places365	19.3	21.8	24.8
RANSAC Iterations	10M	42.1	49.3	54.4
	1M	40.2	45.6	51.3
	100k	38.2	44.7	49.9
	10k	33.0	38.7	44.5

Table 3. **Ablation Study.** ($|K|$=$256k$, ϵ=10, m=2).

using ApGem [32] and match the privatized descriptors to the reference descriptors. Finally, we obtain poses for the query images using the COLMAP [34] image registrator with fixed intrinsics. The poses are then submitted to the long-term visual localization benchmark. Below, we analyze the behavior of LDP-FEAT with the evaluation metric being the percentages of localized query images for the day queries for different thresholds in the translation and rotation error, as shown in Tab. 2.

Localization accuracy upper bound. For the first five rows in Tab. 2, we set m=1 and ϵ=∞, meaning that LDP-FEAT simply returns the nearest neighbor d'. Thus, these accuracies represent the upper bound performance for each database size. As expected, the localization accuracy decreases as the database size decreases, due to the quantization step in LDP-FEAT that replaces a raw descriptor with its nearest neighbor in $\mathcal{K}$. Note, $|\mathcal{K}|$=2^{1024} corresponds to our naive LDP method described in Sec. 5.2 that sets $\mathcal{K}$ as the set of all possible descriptors.

Impact of database size. Recall that LDP-FEAT privatizes a raw descriptor d by returning a random subset $\mathcal{Z}$⊂$\mathcal{K}$ of size m. Recall further that for a fixed ϵ, the probability that d'∈$\mathcal{Z}$, where d' denotes the nearest neighbor of d in $\mathcal{K}$, is inversely related to the size of the database $\mathcal{K}$. This creates an inherent privacy-utility tradeoff, as a larger database increases the resolution of the quantization step, but decreases the probability that d'∈$\mathcal{Z}$. To illustrate this, consider the following examples: (1) In the extreme case of $|\mathcal{K}|$=m, *i.e.* always outputting the entire database $\mathcal{K}$ for any input, the privacy is fully preserved since none of the input is distinguishable in their output, but there is not any utility for feature matching; (2) In the case of $|\mathcal{K}|$ being extremely large, $\Pr(d'$∈$\mathcal{Z})$ = $\Pr(u$=1) is nearly zero, meaning the nearest neighbor of the raw descriptor may never be included in the output subset $\mathcal{Z}$, which, evidently, leads to poor utility. In Tab. 2, we empirically investigate the impact of the database size on this tradeoff by fixing ϵ=10 and m=2, and varying $|\mathcal{K}|$. Interestingly, we find that $|\mathcal{K}|$=$256k$ is the best operating point, which demonstrates the need to find the right quantization level.

DB Size	Privacy	# Desc.	MAE	SSIM	PSNR
$\lvert\mathcal{K}\rvert$	ϵ	m	(↓)	(↑)	(↑)
256k	16	1	.1069	.6248	17.35
256k	14	1	.1137	.5923	16.90
256k	12	1	.1317	.4668	15.77
256k	10	1	.1701	.3651	13.72
256k	16	2	.1173	.5582	16.50
256k	14	2	.1208	.5417	16.31
256k	12	2	.1366	.4655	15.44
256k	10	2	.1660	.3676	13.84
256k	16	4	.1261	.4976	15.94
256k	14	4	.1322	.4868	15.47
256k	12	4	.1438	.4511	14.93
256k	10	4	.1758	.3666	13.48
512k	16	4	.1226	.4994	16.02
512k	14	4	.1254	.4866	15.95
512k	12	4	.1511	.4188	14.55
512k	10	4	.1685	.3647	13.93

Table 4. **Direct Inversion Attack on LDP-FEAT.**

Privacy guarantee. The value of ϵ does not indicate the strength of privacy protection, but rather a *bound* on the privacy leakage. As discussed above, the actual strength of the privacy protection of LDP-FEAT depends not just on ϵ, but also the database $\mathcal{K}$ and the subset size m. The unique advantage of LDP-FEAT compared to existing visual privacy methods (*e.g.* [11]) is that LDP-FEAT provides a privacy guarantee. We analyze in Tab. 2 the impact of varying ϵ on localization accuracy for two different operating points: (i) ($|\mathcal{K}| = 512k, m = 4$) and (ii) ($|\mathcal{K}| = 256k, m = 2$). First, observe the decreasing accuracy with decreasing ϵ. Interestingly, we further find that the two operating points achieve almost identical accuracy for the same ϵ value. A likely explanation for this is that the probability that $p(d' \in Z)$ is equal for both operating points, so the improved resolution of the operating point with larger $|\mathcal{K}|$ is negated by having to assign each keypoint $m = 4$ instead of $m = 2$ descriptors, in order to preserve the same ϵ value. Note, for a fixed value of ϵ, we can search the parameters of $\mathcal{K}$ and m for best utility while being assured that the privacy leakage is always bounded.

Impact of Subset Size. In the last five rows of Tab. 2, we investigate the impact of varying the subset size of LDP-FEAT. Empirically, we find that for a fixed database size $|K| = 256k$ and $\epsilon = 10$, the subset size $m = 2$ and $m = 4$ achieve the best localization accuracy. Again, this search is guarded by the privacy guarantee of LDP-FEAT.

Ablation Study. We perform additional ablation experiments in Tab. 3, where we fix $|K|$=$256k$, ϵ=10 and $m = 2$. In the first two rows, we examine the impact of the matching criteria on localization accuracy and find that vocabulary-based matching [28] improves localization performance

dramatically compared to mutual nearest neighbor. Next, we vary the source of the public database and find that extracting the descriptors from the Aachen reference images results in much better performance (more discussions below.). Finally, we report localization accuracy when varying the number of RANSAC iterations. As expected, more RANSAC iterations leads to better performance.

Impact of database content. Following the above discussion, we further note that the database content in $\mathcal{K}$ also has an impact on the privacy-utility trade-off through the quantization step. Specifically, if the database descriptors are all significantly distinct from the input descriptor, it leads to large quantization errors as a result, indicating none of the database descriptors in the output subset $\mathcal{Z}$ can well represent the input descriptor. This certainly yields strong privacy but leads to poor utility. In Tab. 3, we have compared the performance of using Aachen reference images to build the database versus using the external Places365 dataset. The former achieves significantly superior performance due to its higher resemblance to the query images. Another example scenario where the database content plays a role lies in the Aachen night-time localization challenge. As we show in the supplementary, given night-time query images and the database $\mathcal{K}$ built solely from day-time images, the localization accuracy is degraded despite the strong privacy.

6.3. Direct Inversion Attack on LDP-FEAT.

We start by noting that our database and clustering attack are not applicable to LDP-FEAT since all descriptors sent are samples from the database. Instead, we evaluate the performance from a direct inversion attack. To achieve this, we stack all m descriptors sent by the user and create the sparse feature map $F \in \mathbb{R}^{256\times256\times(128\times m)}$, similarly to what is described in Sec. 6.1. F is then fed as input to a U-Net style neural network trained to reconstruct the source image from such a feature map. As shown qualitatively in Fig. 4 with different combinations of ϵ and m, the attack generally fails to recover high-quality image content that reveals privacy. We further report quantitative results in Tab. 4. One observes that the image reconstruction quality is, in general, lower than that for the affine embeddings shown in Tab. 1, indicating the privacy protection of LDP-FEAT.

7. Conclusion

In this paper, we propose two novel attacks to reveal the privacy leakage underneath the adversarial affine subspace embeddings [11]. Following this, we further propose LDP-FEAT, a new and more rigorous privacy-preserving protocol that formulates image feature matching under the umbrella of local differential privacy. This makes our method the first of its kind that enjoys the theoretical privacy guarantees offered by differential privacy, all the while achieving strong utility in downstream tasks, such as visual localization. We envision that our work will inspire more research efforts on specialized LDP protocols for image matching and other vision tasks.

References

[1] Long-term visual localization benchmark. https://www.visuallocalization.net. 7

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *Communications of the ACM*, 2011. 1, 2

[3] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, 2016. 2

[4] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006. 6

[5] Zhipeng Cai, Zuobin Xiong, Honghui Xu, Peng Wang, Wei Li, and Yi Pan. Generative adversarial networks: A survey toward private and secure applications. *ACM Computing Surveys (CSUR)*, 54(6):1–38, 2021. 2

[6] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief: Computing a local binary descriptor very fast. *PAMI*, 2011. 2

[7] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How privacy-preserving are line clouds? recovering scene details from 3d lines. In *CVPR*, 2021. 1, 2

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[9] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016. 2

[10] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *CVPR*, 2016. 2

[11] Mihai Dusmanu, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy-preserving image features via adversarial affine subspace embeddings. In *CVPR*, 2021. 1, 2, 3, 5, 6, 7, 8, 9, S14, S15

[12] Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, Lecture Notes in Computer Science, 2006. 2, 5

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006. 2

[14] Marcel Geppert, Viktor Larsson, Johannes L Schönberger, and Marc Pollefeys. Privacy preserving partial localization. In *CVPR*, 2022. 2

[15] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schönberger, and Marc Pollefeys. Privacy preserving structure-from-motion. In *ECCV*, 2020. 1, 2

[16] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schonberger, and Marc Pollefeys. Privacy preserving localization and mapping from uncalibrated cameras. In *CVPR*, 2021. 2

[17] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. Differential privacy techniques for cyber physical systems: a survey. *IEEE Communications Surveys & Tutorials*, 22(1):746–789, 2019. 2

[18] Kun He, Yan Lu, and Stan Sclaroff. Local descriptors optimized for average precision. In *CVPR*, 2018. 2

[19] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, 2014. 5, 6

[20] Neena Kamath. Announcing azure spatial anchors for collaborative, cross-platform mixed reality apps. https://azure.microsoft.com/en-us/blog/announcing-azure-spatial-anchors-for-collaborative-cross-platform-mixed-reality-apps, 2019. 7

[21] Hiroharu Kato and Tatsuya Harada. Image reconstruction from bag-of-visual-words. In *CVPR*, 2014. 2

[22] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 6

[23] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2

[24] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015. 2

[25] Richard McPherson, Reza Shokri, and Vitaly Shmatikov. Defeating image obfuscation with deep learning. *arXiv preprint arXiv:1609.00408*, 2016. 2

[26] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *NIPS*, 2017. 2, 5

[27] Tony Ng, Hyo Jin Kim, Vincent T Lee, Daniel DeTone, Tsun-Yi Yang, Tianwei Shen, Eddy Ilg, Vassileios Balntas, Krystian Mikolajczyk, and Chris Sweeney. Ninjadesc: content-concealing visual descriptors via adversarial learning. In *CVPR*, 2022. 2

[28] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 1, 2, 8

[29] Francesco Pittaluga, Sanjeev Koppal, and Ayan Chakrabarti. Learning privacy preserving encodings through adversarial training. In *WACV*, 2019. 2

[30] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *CVPR*, 2019. 1, 2, 4, 6

[31] Tilman Reinhardt. Google visual positioning service. https://ai.googleblog.com/2019/02/usingglobal-localization-to-improve.html, 2019. 7

[32] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019. 8

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 6

[34] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 8, S13

[35] Johannes Lutz Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative Evaluation of Hand-Crafted and Learned Local Features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. S13

[36] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. Privacy preserving visual slam. In *ECCV*, 2020. 1, 2

[37] Zhenbo Song, Wayne Chen, Dylan Campbell, and Hongdong Li. Deep novel view synthesis from colored 3d point clouds. In *ECCV*, 2020. 1, 2

[38] Pablo Speciale, Johannes L Schonberger, Sing Bing Kang, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image-based localization. In *CVPR*, 2019. 1, 2

[39] Pablo Speciale, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image queries for camera localization. In *ICCV*, 2019. 1, 2

[40] Matthew Turk and Alex Pentlandus. Eigenfaces for recognition. In *Journal of Cognitive Neuroscience*, 1991. 1, 2

[41] Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*, 2016. 2

[42] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. In *CVPR*, 2013. 2

[43] Haotao Wang, Zhenyu Wu, Zhangyang Wang, Zhaowen Wang, and Hailin Jin. Privacy-preserving deep visual recognition: An adversarial learning framework and a new dataset. *arXiv preprint arXiv:1906.05675*, 4, 2019. 2

[44] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745, 2017. 2, 3

[45] Teng Wang, Xuefeng Zhang, Jingyu Feng, and Xinyu Yang. A comprehensive survey on local differential privacy toward data statistics and analysis. *Sensors*, 20(24):7030, 2020. 5

[46] Philippe Weinzaepfel, Hervé Jégou, and Patrick Pérez. Reconstructing an image from its local descriptors. In *CVPR*, 2011. 2

[47] Zhenyu Wu, Zhangyang Wang, Zhaowen Wang, and Hailin Jin. Towards privacy-preserving visual recognition via adversarial training: A pilot study. In *ECCV*, 2018. 2

[48] Taihong Xiao, Yi-Hsuan Tsai, Kihyuk Sohn, Manmohan Chandraker, and Ming-Hsuan Yang. Adversarial learning of privacy-preserving and task-oriented representations. In *AAAI*, 2020. 2

[49] Xingxing Xiong, Shubo Liu, Dan Li, Zhaohui Cai, and Xiaoguang Niu. A comprehensive survey on local differential privacy. *Security and Communication Networks*, 2020, 2020. 5

[50] Mengmeng Yang, Lingjuan Lyu, Jun Zhao, Tianqing Zhu, and Kwok-Yan Lam. Local differential privacy and its applications: A comprehensive survey. *arXiv preprint arXiv:2008.03686*, 2020. 2, 3, 5, 6

[51] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *ICML Workshop on Deep Learning*, 2015. 2

[52] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2

[53] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *PAMI*, 2017. 6

Supplementary Material
LDP-FEAT: Image Features with Local Differential Privacy

Francesco Pittaluga
francescopittaluga@nec-labs.com
Bingbing Zhuang
bzhuang@nec-labs.com
NEC Labs America

The supplementary material contains (1) a proof that LDP-FEAT satisfies ϵ-LDP, (2) additional experimental results on Aachen night-time localization and Structure-from-Motion (SfM), and (3) an analysis of the paper's assumptions.

S1. Local Differential Privacy of LDP-FEAT

Here, we prove that LDP-FEAT satisfies ϵ-LDP. For clarity, we first prove that ω-SM satisfy ω-LDP and the proof for LDP-FEAT follows very similarly.

S1.1. Theorem 1 (ω-Subset satisfies ϵ-LDP)

For any inputs v_1 and v_2, and their output $\mathcal{Z}_1$ and $\mathcal{Z}_2$ returned by ω-SM, there are four possible scenarios $\{v_1{\in}\mathcal{Z}_1, v_2{\in}\mathcal{Z}_2\}$, $\{v_1{\notin}\mathcal{Z}_1, v_2{\in}\mathcal{Z}_2\}$, $\{v_1{\in}\mathcal{Z}_1, v_2{\notin}\mathcal{Z}_2\}$, $\{v_1{\notin}\mathcal{Z}_1, v_2{\notin}\mathcal{Z}_2\}$, each with different probability distributions. Below, we show that the probability inequality required by ϵ-LDP, *i.e.* Eq. (3) of the main paper, is satisfied for all the four scenarios.
1) $\{v_1{\in}\mathcal{Z}_1, v_2{\in}\mathcal{Z}_2\}$. In this case,

$$\begin{aligned}\Pr(\mathcal{Z}_1|v_1) &= \frac{\omega e^{\epsilon}}{\omega e^{\epsilon} + |\mathcal{K}| - \omega} \Big/ \binom{|\mathcal{K}|}{\omega}, \\ \Pr(\mathcal{Z}_2|v_2) &= \frac{\omega e^{\epsilon}}{\omega e^{\epsilon} + |\mathcal{K}| - \omega} \Big/ \binom{|\mathcal{K}|}{\omega},\end{aligned} \tag{S1}$$

meaning that $\Pr(\mathcal{Z}_1|v_1){=}\Pr(\mathcal{Z}_2|v_2)$, hence $\Pr(\mathcal{Z}_1|v_1) \leq e^{\epsilon}\Pr(\mathcal{Z}_2|v_2)$ holds.
2) $\{v_1{\notin}\mathcal{Z}_1, v_2{\in}\mathcal{Z}_2\}$. In this case,

$$\begin{aligned}\Pr(\mathcal{Z}_1|v_1) &= \frac{\omega}{\omega e^{\epsilon} + |\mathcal{K}| - \omega} \Big/ \binom{|\mathcal{K}|}{\omega}, \\ \Pr(\mathcal{Z}_2|v_2) &= \frac{\omega e^{\epsilon}}{\omega e^{\epsilon} + |\mathcal{K}| - \omega} \Big/ \binom{|\mathcal{K}|}{\omega},\end{aligned} \tag{S2}$$

meaning that $\Pr(\mathcal{Z}_1|v_1) = e^{-\epsilon}\Pr(\mathcal{Z}_2|v_2)$, hence $\Pr(\mathcal{Z}_1|v_1) \leq e^{\epsilon}\Pr(\mathcal{Z}_2|v_2)$ holds since $\epsilon > 0$.
3) $\{v_1{\in}\mathcal{Z}_1, v_2{\notin}\mathcal{Z}_2\}$. In this case,

$$\begin{aligned}\Pr(\mathcal{Z}_1|v_1) &= \frac{\omega e^{\epsilon}}{\omega e^{\epsilon} + |\mathcal{K}| - \omega} \Big/ \binom{|\mathcal{K}|}{\omega}, \\ \Pr(\mathcal{Z}_2|v_2) &= \frac{\omega}{\omega e^{\epsilon} + |\mathcal{K}| - \omega} \Big/ \binom{|\mathcal{K}|}{\omega},\end{aligned} \tag{S3}$$

meaning that $\Pr(\mathcal{Z}_1|v_1) = e^{\epsilon}\Pr(\mathcal{Z}_2|v_2)$, hence $\Pr(\mathcal{Z}_1|v_1) \leq e^{\epsilon}\Pr(\mathcal{Z}_2|v_2)$ holds.
4) $\{v_1{\notin}\mathcal{Z}_1, v_2{\notin}\mathcal{Z}_2\}$. In this cases,

$$\begin{aligned}\Pr(\mathcal{Z}_1|v_1) &= \frac{\omega}{\omega e^{\epsilon} + |\mathcal{K}| - \omega} \Big/ \binom{|\mathcal{K}|}{\omega}, \\ \Pr(\mathcal{Z}_2|v_2) &= \frac{\omega}{\omega e^{\epsilon} + |\mathcal{K}| - \omega} \Big/ \binom{|\mathcal{K}|}{\omega},\end{aligned} \tag{S4}$$

meaning that $\Pr(\mathcal{Z}_1|v_1){=}\Pr(\mathcal{Z}_2|v_2)$, hence $\Pr(\mathcal{Z}_1|v_1) \leq e^{\epsilon}\Pr(\mathcal{Z}_2|v_2)$ holds.
This concludes our proof.

S1.2. Theorem 2 (LDP-FEAT satisfies ϵ-LDP)

For any input descriptor d, the output set $\mathcal{Z}$ are obtained by: first map d to an element (let us denote it as $\bar{d}$) in the database $\mathcal{K}$, and then $\bar{d}$ is mapped to the random set $\mathcal{Z}$. Hence,

$$\begin{aligned}\Pr(\mathcal{Z}|d) &= \sum_{\bar{d}\in\mathcal{K}} \Pr(\mathcal{Z}, \bar{d}|d) \\ &= \sum_{\bar{d}\in\mathcal{K}} \Pr(\mathcal{Z}|\bar{d})\Pr(\bar{d}|d).\end{aligned} \tag{S5}$$

Since the mapping from d to $\bar{d}$ is deterministic – it is mapped to the nearest neighbor d' in the database, we have

$$\Pr(\bar{d}|d) = \begin{cases}1, & \text{if } \bar{d} = d, \\ 0, & \text{if } \bar{d} \neq d.\end{cases} \tag{S6}$$

Plugging Eq. (S6) into Eq. (S5) yields

$$\Pr(\mathcal{Z}|d) = \Pr(\mathcal{Z}|d') \tag{S7}$$

	DB Size	Privacy	# Desc.	Day			Night		
	$\|\mathcal{K}\|$	ϵ	m	0.25m, 2°	0.5m, 5°	5.0m, 10°	0.25m, 2°	0.5m, 5°	5.0m, 10°
Accuracy Upper Bound	128k	∞	1	73.2	82.8	88.1	24.6	28.8	33.5
	256k	∞	1	76.8	86.3	91.6	28.8	34.6	42.4
	512k	∞	1	78.0	87.4	93.3	33.5	40.8	51.3
	1024k	∞	1	79.7	89.9	94.9	36.1	42.4	51.8
	2^{1024}	∞	1	**84.1**	**91.7**	**96.4**	**50.3**	**61.8**	**73.8**
Impact of Database Size	128k	10	2	39.4	45.4	50.2	1.60	2.10	3.70
	256k	10	2	**42.1**	**49.3**	**54.4**	**3.70**	**5.20**	**6.30**
	512k	10	2	37.4	43.2	48.1	2.10	3.70	4.70
	1024k	10	2	33.3	37.1	42.1	2.10	3.70	3.70
	2^{1024}	10	2	0.00	0.00	0.00	0.00	0.00	0.00
Privacy Guarantee	256k	10	2	42.1	49.3	54.4	3.70	5.20	6.30
	256k	12	2	69.5	78.4	84.1	16.8	19.4	23.6
	256k	14	2	75.1	84.7	89.4	**23.0**	**27.7**	**33.0**
	256k	16	2	**75.4**	**85.3**	**90.2**	**23.0**	27.2	31.9
	512k	10	4	42.1	49.6	55.0	5.20	7.90	9.90
	512k	12	4	69.4	77.9	84.6	19.4	22.5	26.2
	512k	14	4	73.9	84.5	90.2	23.6	29.8	34.6
	512k	16	4	**76.1**	**85.0**	**90.4**	**24.6**	**29.8**	**34.0**
Impact of Subset Size	256k	10	1	34.6	39.7	44.7	2.60	3.70	4.70
	256k	10	2	**42.1**	**49.3**	54.4	**3.70**	**5.20**	**6.30**
	256k	10	4	**42.1**	49.0	**55.1**	**3.70**	4.70	5.20
	256k	10	8	39.1	46.4	51.8	4.70	5.80	6.30
	256k	10	16	32.8	38.0	44.3	2.60	2.60	3.70

Table S1. **Aachen Day-Night Localization Challenge.**

For any input descriptor d_1 and d_2, their nearest neighbor d'_1 and d'_2, and their output $\mathcal{Z}_1$ and $\mathcal{Z}_2$, there are four possible scenarios $\{d'_1 \in \mathcal{Z}_1, d'_2 \in \mathcal{Z}_2\}$, $\{d'_1 \notin \mathcal{Z}_1, d'_2 \in \mathcal{Z}_2\}$, $\{d'_1 \in \mathcal{Z}_1, d'_2 \notin \mathcal{Z}_2\}$, $\{d'_1 \notin \mathcal{Z}_1, d'_2 \notin \mathcal{Z}_2\}$, each with different probability distributions. Since $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are sampled using the ω-SM, we have shown above that $Pr(\mathcal{Z}_1|d'_1) \leq e^{\epsilon} Pr(\mathcal{Z}_2|d'_2)$ holds for all the four scenarios, and given Eq. (S7), we have $Pr(\mathcal{Z}_1|d_1) \leq e^{\epsilon} Pr(\mathcal{Z}_2|d_2)$. This means that LDP-FEAT satisfies ϵ-LDP.

S2. Additional Results

S2.1. Aachen Night Localization

Similarly to the Tab.2 of the main paper, we report in Tab. S1 the localization accuracy for night-time queries in the Aachen Day-Night localization challenge. Overall, we observe a degradation of performance compared to the day-time queries. This is mainly because our database $\mathcal{K}$ was built from the Aachen reference images which contain day-time images only. As aforementioned, this causes a large quantization error Δd in LDP-FEAT, which certainly enhances privacy protection but compromises the utility. We leave the pursuit of a better privacy-utility trade-off for night-time localization as a future work.

S2.2. Structure-from-Motion

We further demonstrate the utility of LDP-FEAT on Structure-from-Motion, as shown in Fig. S1. We adopt COLMAP [34] for SfM by customizing its feature extraction and matching using LDP-FEAT. As an indicator for SfM performance, we report the number of registered images, the number of reconstructed sparse 3D points, the average keypoint track length, and the average reprojection error.

We report results on the "South Building" and "Fountain" scene from the 3D reconstruction benchmark [35]. We first report the results for ($|\mathcal{K}| = 2^{1024}$, $\epsilon = \infty$, $m = 1$). This corresponds to the oracle setting where only the raw descriptor is sent without any privacy protection, and which serves a performance upper bound. We then use a dictionary with $512k$ descriptors, *i.e.* ($|\mathcal{K}| = 512k$, $\epsilon = \infty$, $m = 1$) where the quantization step, *i.e.* mapping the raw descriptor

Scene	Dict. Size $\|\mathcal{K}\|$	Privacy ϵ	# Desc. m	Reg. Images	Sparse Points	Track Length	Reproj. Error
South Building	2^{1024}	∞	1	128	110,714	5.66	1.29
	512k	∞	1	128	62,194	4.85	1.12
	512k	10	4	88	8,668	3.56	0.85
	512k	10	8	75	10,554	3.66	1.05
	512k	10	16	123	25,451	3.62	0.89
	512k	10	32	123	26,760	3.52	0.94
	512k	10	64	124	21,596	3.31	1.28
Fountain	2^{1024}	∞	1	11	15,332	4.42	2.82
	512k	∞	1	11	8,612	3.80	2.39
	512k	10	4	11	983	2.86	1.26
	512k	10	8	11	1,827	2.98	1.35
	512k	10	16	11	2,598	3.03	1.50
	512k	10	32	11	3,078	3.02	1.52
	512k	10	64	11	3,242	2.89	1.41

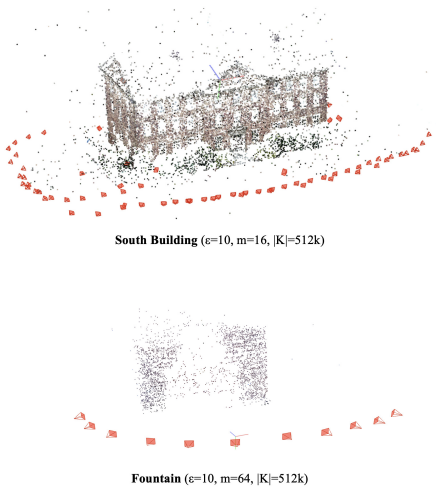

Figure S1. **Local Feature Evaluation Benchmark.** Structure-from-Motion results using LDP-FEAT with different configurations.

Dim	Success Rate (%)			
m	N=50	N=20	N=10	N=5
4	93.89	95.60	96.61	97.46
16	87.27	90.54	92.73	94.38

Table S2. **Intersecting Adversarial Subspaces.**

Dim=2	Dim=4	Dim=8	Dim=16
97.42%	94.30%	85.46%	73.03%

Table S3. **Clustering Attack Collisions.**

d to its nearest neighbor d', introduces a degree of privacy protection and thus degrades the performance accordingly. Next, we fix $|\mathcal{K}| = 512k$ and $\epsilon = 10$, while increasing m from 4 to 64. The performance varies, and we observe that m=32 yields the best performance. Overall, one observes that good SfM results are obtained from LDP-FEAT under different settings; in particular, most of the cameras are successfully registered, despite the reconstructed points being sparser. We demonstrate the qualitative reconstruction results in Fig. S1.

S3. Analysis of Assumptions

S3.1. Intersecting Adversarial Subspaces

As discussed in the paper, our proposed Database and Clustering attacks are based on the following key empirical assumption: a low-dimensional hybrid adversarial affine subspace D likely only intersects the high-dimensional descriptor manifold at $\frac{m}{2}+1$ points corresponding to the original descriptor d and the adversarial descriptors $\{a_1, ..., a_{\frac{m}{2}}\}$ that were sampled from the database. Here, we generate subspaces for 100K descriptors and report how often our assumption holds, i.e., for each subspace, we select the top N database descriptors closest to the subspace, and match them to the $\frac{m}{2}+1$ descriptors forming the subspace. Using the standard ratio test (>0.8) we report the percentage of successful matches in Tab. S2. The high success rates empirically validate our assumption regarding the rareness of intersections beyond the $\frac{m}{2}+1$ forming descriptors. We also note that our assumption is implied by the success of feature matching in [11].

S3.2. Clustering Attack Collisions

For our clustering attack, we assume that the attacker does not have access to the database of real-world descriptors W from which adversarial descriptors $a_{i=1,...,\frac{m}{2}}$ for subspace D are sampled, but does have access to an additional set of adversarial affine subspaces $\mathcal{Q}$ that were lifted with the same database W. We can identify the subset of subspaces $\mathcal{Q}' \in \mathcal{Q}$ that were lifted using one or more of the same adversarial descriptors as D, by noting that each subspace in $\mathcal{Q}'_j \in \mathcal{Q}'$ intersects with D. Assuming that $\mathcal{Q}$ is sufficiently large such that all a_i's were used to lift at least one of the subspaces in $\mathcal{Q}'$, this indicates that the minimal point-to-subspace distance $\min_j \texttt{dist}(a_i, Q'_j) = 0$ for i=$\{1, ..., \frac{m}{2}\}$. On the other hand, since $\mathcal{Q}'$ is selected without any knowledge or specific treatments on d, it is expected that $\min_j \texttt{dist}(d, Q'_j) \gg 0$. In this discrepancy lies the crux of our attack – while $\min_j \texttt{dist}(\hat{a}_i, Q'_j) > 0$ for our estimates of a_i, $\hat{a}_i$, we expect that $\min_j \texttt{dist}(\hat{d}, Q'_j) \gg$

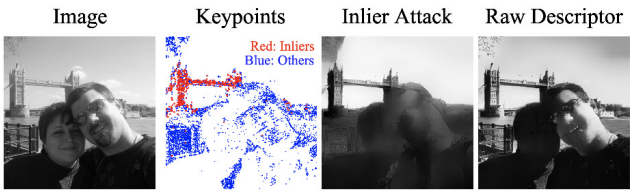

Figure S2. **Inlier Attack.**

$\min_j \texttt{dist}(\hat{a}_i, Q'_j)$. Hence, we compute the score s_i for each $\hat{d}_i$ as $s_i = \min_j \texttt{dist}(\hat{d}_i, Q'_j)$ and the largest s_i yields our estimate for d. We note that it is not impossible that $\mathcal{Q}'$ may contain a database descriptor that is close to d too, but the probability of such a collision is low thanks to the high dimension of descriptors. In Tab. S3, we validate this assumption by lifting all the descriptors of our 10 test images to adversarial subspaces and reporting the percentage of them that have no collisions in our attack.

S3.3. Sensitivity of Inlier Content

Since inlier correspondences emerge from RANSAC in the geometric tasks, one natural attack one may think of is leveraging these inlier features to reveal the image content; we term this as inlier attack. We note that this attack is generally applicable to all privacy protocols that are capable of geometric utility tasks where RANSAC returns inliers, *e.g.* ours and [11]. However, RANSAC inliers typically consist of only static background scenes without dynamic foreground (e.g. faces). We clarify that the privacy protection mainly targets at the foreground in both our and [11]'s problem setup, and thus the inlier attack was not a concern. Nonetheless, we perform inlier attack here and show example result in S2. As expected, the attack works only for the background bridge, but not for the foreground faces.