Subverting Privacy-Preserving GANs: Hiding Secrets in Sanitized Images

Kang Liu, Benjamin Tan, Siddharth Garg
Department of Electrical and Computer Engineering
New York University
{kang.liu, benjamin.tan, siddharth.garg}@nyu.edu

Abstract

Unprecedented data collection and sharing have exacerbated privacy concerns and led to increasing interest in privacy-preserving tools that remove sensitive attributes from images while maintaining useful information for other tasks. Currently, state-of-the-art approaches use privacy-preserving generative adversarial networks (PP-GANs) for this purpose, for instance, to enable reliable facial expression recognition without leaking users' identity. However, PP-GANs do not offer formal proofs of privacy and instead rely on experimentally measuring information leakage using classification accuracy on the sensitive attributes of deep learning (DL)-based discriminators. In this work, we question the rigor of such checks by subverting existing privacy-preserving GANs for facial expression recognition. We show that it is possible to hide the sensitive identification data in the sanitized output images of such PP-GANs for later extraction, which can even allow for reconstruction of the entire input images, while satisfying privacy checks. We demonstrate our approach via a PP-GAN-based architecture and provide qualitative and quantitative evaluations using two public datasets. Our experimental results raise fundamental questions about the need for more rigorous privacy checks of PP-GANs, and we provide insights into the social impact of these.

Introduction

The availability of large datasets and high performance computing resources has enabled new machine learning (ML) solutions for a range of application domains. However, as is often the case with transformative technologies, the ubiquity of big data and ML raises new data privacy concerns. Given the emergence of applications that use personal data, such as facial expression recognition (Chen, Konrad, and Ishwar 2018) or autonomous driving (Xiong et al. 2019) one must take care to provide data relevant to the specific application without inadvertently leaking other sensitive information. Despite recent legislative efforts to protect personal data privacy—for instance, the General Data Protection Regulation (GDPR) passed by EU—technology must also play a role in safeguarding privacy (Tene et al. 2019).

Consider a scenario where a user wants to use their private data with an untrusted application, as in Figure 1. For privacy, the user needs to remove sensitive—and application irrelevant—attributes from their data while preserving relevant details. This can be achieved using a tool typically sourced from a third-party provider (e.g. *Generated Photos*[1]). To select a tool, the end user performs their own "privacy check", evaluating that the tool satisfies their definition of privacy.

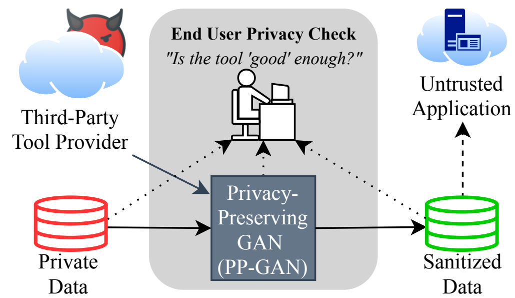

Figure 1: Typical use case for a PP-GAN sourced from a third-party, where a user wants to sanitize their data for use with an (untrusted) application.

To this end, recent research proposes the use of deep neural networks (DNNs), specifically, generative adversarial networks (GANs) for sanitizing data of sensitive attributes (Wu et al. 2019; Maximov, Elezi, and Leal-Taixé 2020; Chen, Konrad, and Ishwar 2018). These so-called "*privacy-preserving GANs*" (PP-GANs) can sanitize images of human faces such that only their facial expressions are preserved while other identifying information is replaced (Chen, Konrad, and Ishwar 2018). Other examples include: removing location-relevant information from vehicular camera data (Xiong et al. 2019), obfuscating the identity of the person who produced a handwriting sample (Feutry, Piantanida, and Duhamel 2020), and removal of barcodes from images (Raval, Machanavajjhala, and Cox 2017). Given the expertise required to train such models, one expects that users will need to acquire a privacy preservation tool from a third party or outsource GAN training, so proper privacy evaluation is paramount. In the aforementioned works, researchers note a trade-off between "utility" and "privacy" objectives—they suggest that PP-GANs offer

[1]https://generated.photos/anonymizer

a panacea that achieves both.

The privacy offered by PP-GANs is typically measured using empirical metrics of information leakage (Chen, Konrad, and Ishwar 2018; Xiong et al. 2019; Feutry, Piantanida, and Duhamel 2020). For instance, Chen, Konrad, and Ishwar (2018) use the (in)ability of deep learning (DL)-based discriminators to identify secret information from sanitized images as the metric for privacy protection. However, empirical metrics of this nature are bounded by discriminators' learning capacities and training budgets; we argue that such privacy checks lack rigor.

This brings us to our paper's motivating question: *are empirical privacy checks sufficient to guarantee protection against private data recovery from data sanitized by a PP-GAN?* As is common practice in the security community, we answer this question in an adversarial setting. We show that PP-GAN designs can be subverted to pass privacy checks, while still allowing secret information to be extracted from sanitized images. Our results have both foundational and practical implications. Foundationally, they establish that stronger privacy checks are needed before PP-GANs can be deployed in the real-world. From a practical stand-point, our results sound a note of caution against the use of data sanitization tools, and specifically PP-GANs, designed by third-parties. Our contributions include:

- We provide the first comprehensive security analysis of privacy-preserving GANs and demonstrate that existing privacy checks are inadequate to detect leakage of sensitive information.
- Using a novel steganographic approach, we adversarially modify a state-of-the-art PP-GAN to hide a secret (the user ID), from purportedly sanitized face images.
- Our results show that our proposed adversarial PP-GAN can successfully hide sensitive attributes in "sanitized" output images that pass privacy checks, with 100% secret recovery rate.

We first provide background on PP-GANs and associated empirical privacy checks. We then formulate an attack scenario to ask if empirical privacy checks can be subverted. Next, we outline our approach for circumventing empirical privacy checks. We present our experimental work and further discuss our findings in more detail. We frame our work with reference to prior related work before we conclude in the end.

Background

In this section, we describe the relevant background on our representative PP-GAN baseline and how their privacy guarantees are evaluated.

Representative PP-GAN Baseline

We adopt the PPRL-VGAN framework proposed by Chen, Konrad, and Ishwar (2018) as our experimental focus[2] and use similar notation. PPRL-VGAN produces a PP-GAN with a variational autoencoder-generative adversarial network (VAE-GAN) architecture. The generator network, G_0, comprises an encoder, a Gaussian sampling block, and a decoder, as shown in Figure 2. The PP-GAN is designed to replace the "user-identity" in an image while preserving facial expression information.

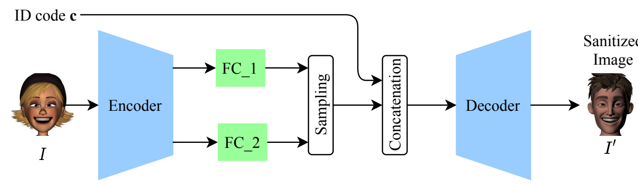

Figure 2: Baseline PP-GAN Architecture

Formally, given an input face image I with user ID $y^{id} \in \{0, 1, \cdots, N_{id} - 1\}$, expression label $y^{ep} \in \{0, 1, \cdots, N_{ep} - 1\}$, and a target ID c, the generator G_0 synthesizes a realistic face image I' that belongs to the target ID c while preserving expression y^{ep}. The user specifies the target identity using a one-hot encoded identity code $c \in \{0, 1\}^{N_{id}}$ such that $I' = G_0(I, c)$. Multiple discriminators, described next, are used to train this PP-GAN.

Discriminator Unlike conventional GAN settings (Goodfellow et al. 2014), the training of PPRL-VGAN employs three discriminators, D^0, D^1, and D^2, responsible for real/synthetic face discrimination, ID classification, and facial expression classification, respectively. Specifically, D^0 takes a real or synthetic image as input and outputs the probability of it being real. The probability of image I being classified as real is denoted by $D^0(I)$. Similarly, $D^1_{y^{id}}(I)$ and $D^2_{y^{ep}}(I)$ denote the probabilities of I being an image of user y^{id} and having expression y^{ep}, respectively. The discriminators are trained simultaneously to maximize the combined loss function $\mathcal{L}_D$ which is expressed as:

$$
\begin{aligned}
\mathcal{L}_D(D, G_0) =& \lambda_D^0 \big(E_{I \sim p_d(I)} \log D^0(I) + \\
& E_{I \sim p_d(I), c \sim p(c)} \log \big(1 - D^0\big(G_0(I, c)\big)\big)\big) + \\
& \lambda_D^1 E_{(I, y^{id}) \sim p_d(I, y^{id})} \log D^1_{y^{id}}(I) + \\
& \lambda_D^2 E_{(I, y^{ep}) \sim p_d(I, y^{ep})} \log D^2_{y^{ep}}(I)
\end{aligned} \tag{1}
$$

Here, λ_D^0, λ_D^1 and λ_D^2 are scalar constants weighting the different loss components for real/synthetic image discrimination, input image ID recognition, and input image facial expression recognition.

Generator The generator network has an encoder-decoder architecture. The encoder transforms I into two intermediate latents that are fed to a Gaussian sampling block to obtain an "identity-invariant face image representation" $f(I)$, i.e., the encoder learns a mapping function with random Gaussian sampling, $f(I) \sim q(f(I)|I)$. The decoder further maps the concatenation of $f(I)$ and c into the synthetic face image I'.

The generator G_0 aims to generate synthetic face image I' as real as possible to fool discriminator D^0. At the

[2] PPRL-VGAN bears similarities to other related prior works, particularly with respect to privacy checks. We discuss these in more detail in the Related Work section.

same time, G_0 learns to synthesize I's that are classified by D^1 with the target ID specified by c. In addition, the privacy-protected image I' should maintain the expression y^{ep} from I (as classified by D^2). The combined loss function $\mathcal{L}_G(D, G_0)$ to minimize is:

$$\mathcal{L}_G(D, G_0) = \lambda_G^0 E_{I\sim p_d(I),c\sim p(c)} \log D^0\big(1 - G_0(I,c)\big) + \lambda_G^1 E_{I\sim p_d(I),c\sim p(c)} \log D_c^1\big(1 - G_0(I,c)\big) + \lambda_G^2 E_{(I,y^{ep})\sim p_d(I,y^{ep}),c\sim p(c)} \log D_{y^{ep}}^2\big(1 - G_0(I,c)\big) + \lambda_G^3 KL\big(q(f(I)|I)\|p(f(I))\big) \tag{2}$$

The fourth loss term is a *KL* divergence loss used in VAE training that measures the distance between a prior distribution on the latent space $p(f(I)) \sim \mathcal{N}(0,1)$ and the conditional distribution $q(f(I)|I)$. Here, λ_G^0, λ_G^1, λ_G^2, and λ_G^3 weight the loss components between real/synthetic image discrimination, image ID classification, and expression classification respectively by D^0, D^1, and D^2, and the last *KL* divergence loss.

Empirical Privacy Checks

In the current PP-GAN literature, information leakage is measured using empirical privacy checks that quantify the ability of a separately trained DNN discriminator to pick up trace artifacts post-sanitization that correlate with the sensitive attributes. In this work, we focus on two measures based on attack scenarios (ASs) proposed by Chen, Konrad, and Ishwar (2018); we refer to these as the "weak" and "strong" privacy checks. For the subsequent discussion, we will assume that the PP-GAN is trained using a training dataset of face images and corresponding user IDs and expressions, $I_{train}, y_{train}^{id}, y_{train}^{ep}$, and a test dataset $I_{test}, y_{test}^{id}, y_{test}^{ep}$ is used to perform the privacy checks.

Weak privacy check This check corresponds to AS I of Chen et al.'s work, and examines if a discriminator trained on images from the training dataset and their corresponding IDs (i.e., $\{I_{train}, y_{train}^{id}\}$) can recover the original IDs from images $I'_{test} = G_0(I_{test}, c)$ with c picked at random from $\{0,1\}^{N_{id}}$. In other words, if the ID returned by the discriminator given I'_{test} is y_{test}^{id} then the privacy check succeeds. We refer to the output test sanitized image as $I'_{test,c}$. This check is "weaker" than the next check because its discriminator is trained on the distribution of input images and not the distribution of the PP-GAN's sanitized outputs.

Strong privacy check This check corresponds to AS II of Chen et al.'s work, where the user emulates a stronger adversary and trains a discriminator on sanitized data with the underlying ground-truth identities. To address the shortcomings of the weak check, the strong privacy check measures the classification accuracy of a discriminator trained on a dataset obtained by passing training images through G_0. That is, $\{G_0(I_{train}, c), y_{train}^{id}\}$ is used to train the discriminator. In other words, the discriminator is trained on the distribution of the PP-GAN's sanitized outputs to recover the original IDs from sanitized test images.

Subverting PP-GANs

We now ask if an adversary can train an adversarial PP-GAN G_0^{adv} that passes the weak and strong privacy checks, but enables recovery of the sensitive attribute, y^{id}, from sanitized outputs I'. This question reflects the following real-world scenario (Figure 3): an adversary, say Alice, trains G_0^{adv} and releases it publicly. A user, Bonnie, downloads G_0^{adv}, verifies that it passes both weak and strong privacy checks (using validation data), and then uses G_0^{adv} to sanitize her private test images and releases them publicly. Can Alice (or a collaborator) recover secrets from the sanitized images?

Goals Alice seeks to design a PP-GAN G_0^{adv} with the following goals:

- *Utility*: G_0^{adv}'s outputs, i.e., $I' = G_0^{adv}(I,c)$ should have the same expression as that of its input I and I' should be classified as ID c;
- *Privacy*: the sanitized images should pass both weak and strong privacy checks; and
- *Recovery*: Alice should be able to recover the ID corresponding to image I from the sanitized image I'.

Constraints In addition to meeting these goals, Alice wants to ensure that G_0^{adv} is still a plausible implementation of a PP-GAN to ensure that Bonnie does not identify it as adversarial on inspection (Bonnie is given white-box access to G_0^{adv}). As such, Alice must obey the following:

- G_0^{adv} still uses the VAE-GAN architecture, i.e., it is a neural network that comprises an encoder, a sampling block, and a decoder;
- G_0^{adv} takes in the same inputs as the baseline, which are the input image I and new ID code c;
- G_0^{adv} is allowed to be deeper than the baseline, but the extra layers must be ones that are commonly used in typical neural networks (NNs), such as convolutional layers.

While the discussion thus far has been in the context of a real-world threat, an attack that satisfies these goals and constraints has deeper implications on the PP-GANs as tools for user privacy preservation. Specifically, a successful subversion shows that existing privacy checks are insufficient to fully catch information leakage in PP-GANs. If one can design an adversarial PP-GAN that easily circumvents these

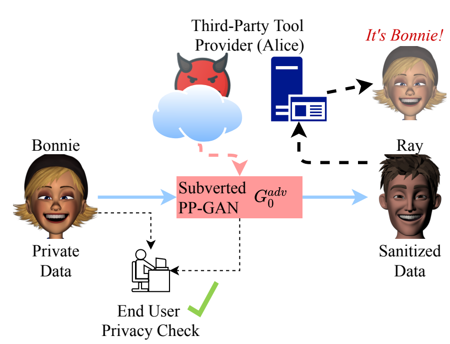

Figure 3: Overview of the Attack Scenario

privacy checks, it raises questions as to whether information could be leaked via inadvertent design errors as well.

Proposed Approach

We now describe our proposed construction of an adversarial PP-GAN that circumvents both weak and strong privacy checks. We begin by discussing a straw man solution and then our implementation.

A "Straw Man" Solution

To embed sensitive attributes in sanitized images we turn to steganography, a family of techniques that seeks to hide secret information in ordinary files, such as images, without being detected. The secret data can then be extracted from the images received by the designated party.

One possible approach to realize our scheme is to use a conventional steganography tool (e.g., *Steghide*[3]) to embed the user y^{id} in the sanitized face images I'. In fact, we find that this straw man solution successfully hides the sensitive data. The resultant images pass both privacy checks (results are in the Experimental Work section). However, recall that we also require our adversarial modifications to be expressed as layers of a NN. Thus, as we discuss next, we attempt to implement steganographic operations directly.

Adversarial PP-GAN Design

Implementing steganography in the context of a PP-GAN poses several challenges. First, the input to an image steganography tool is a "cover image" *and* the secret; in our setting, the cover image is I' but the secret (user y^{id}) is not directly available and must be extracted from I. For this, we implement a secret extraction stage. Second, as noted above, the steganography process must be converted into NN layers which poses its own challenges. Finally, Alice must be able to extract the secret from the secret-embedded "sanitized" image I'', which also requires special steps. We illustrate the adversarial PP-GAN in Figure 4 which can subsequently be consolidated to become the final, more innocuous architecture in Figure 5. After explaining the NN building blocks for our approach, we describe the multi-step training process.

Secret extraction The first step is to extract the information pertaining to the secret y^{id} from the input image I. Since PP-GAN's encoder already has several layers to extract relevant features of I, we can extract y^{id} by adding an additional fully connected (FC) layer FC_0 in parallel with FC_1 and FC_2 (shown as ① in Figure 4). The output of FC_0 is a secret that allows recovery of y^{id}. In this work, we consider two schemes: **Scheme 1**—a direct encoding of y^{id} as a one-hot encoded vector and **Scheme 2**—a vector representation from which the original image I can be reconstructed (and from which y^{id} can be deduced).

Secret embedding Our secret embedding method is inspired by steganographic techniques that hide the bits of a secret in the cover image's frequency domain. This is achieved by manipulating the least significant bits (LSBs)

[3]http://steghide.sourceforge.net/

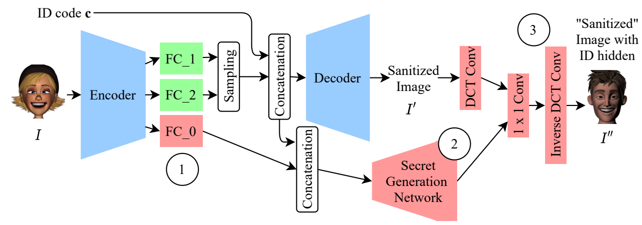

Figure 4: Initial Adversarial PP-GAN Architecture

of the discrete cosine transform (DCT) coefficients and reduces the visible impact of manipulating an image. In our approach, we embed the secret at random locations (DCT coefficients) that are *input-dependent*. To do so, we use y^{ep} and target ID c as randomness seeds for selecting the DCT coefficients for LSB replacement. Specifically, y^{ep} is used to select which DCT coefficients in which to embed the secret, while c permutes the secret before it is embedded. Note that both selection and permutation can be expressed as linear operators and are therefore implementable as NN layers (represented as ② in Figure 4). As y^{ep} is encoded in z, the PP-GAN encoder's output latent, and c is explicitly provided, we can simply use bits of their concatenated latent as randomness seeds.

To reduce the impact of LSB replacement of DCT coefficients in the final image, we embed secrets only in the LSBs of middle-frequency position (Sheisi, Mesgarian, and Rahmani 2012) and select frequency positions that tend to have larger absolute values. To determine these locations, we find the average of the absolute values of all middle-frequency DCT coefficients in a dataset of (honestly) sanitized images and choose positions with the largest mean. Once the secrets are stealthily embedded, the adversarial PP-GAN performs inverse DCT conversion (③ in Figure 4) to obtain the final "sanitized" image, I''.

Secret Recovery The secret recovery stage seeks to extract y^{id} from I''. Note that secret recovery is not part of the adversarial PP-GAN but is a separate process performed by Alice. To extract the secret ID information of the user's input image I, we first classify image I'' into expression y''_{ep} and ID y''_{id}. Assuming that $y''_{ep} \approx y_{ep}$ and $y''_{id} \approx c$, we can now recover the locations where the secret is embedded and its permutation. Finally, we perform a DCT transformation on I'' and extract the secret from the LSBs of the selected DCT coefficients. In scheme 2, where the secret is the vector representation of the original image, Alice can train and use image recovery decoder NN to reconstruct I.

NN-based Implementation

We now discuss two practical issues with respect to our adversarial PP-GAN implementation: ensuring that all functionality is implemented using NN layers, and training the final architecture.

NN Layers As previously mentioned, the secret extraction leverages existing layers of the PP-GAN (encoder) and an additional FC layer. The secret embedding stage can also

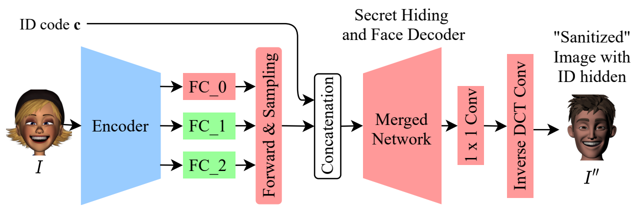

Figure 5: Proposed Adversarial PP-GAN Architecture

be implemented as a multi-layer "secret generation" NN (② in Figure 4). We use the same architecture as the benign PP-GAN's decoder with an added DCT layer at the end; it is trained to permute and embed the secret. Finally, DCT, addition of LSBs and inverse-DCT can each be implemented using single NN layers. DCT transformations are simply a linear computation which can be implemented as a convolution, so we manually design these as proposed by Liu et al. (2020). The resulting architecture is indeed a multi-layer NN although it has some parallel paths that skip across layers. Following training, we can make the adversarial PP-GAN appear less suspicious through a merging process[4] using simple transformations and increasing layer dimensions to produce the architecture in Figure 5.

Training Training the adversarial NN involves two steps. First, we need to train the adversarial PP-GAN to extract secret data corresponding to y^{id}. In scheme 1, we first train the PP-GAN with the added FC_0 layer and introduce an additional loss term to the generator loss function $\mathcal{L}_G$ that measures the L_2 distance between the output of the FC_0 and y^{id} (expressed as a one-hot vector). In doing so, the trained network produces both an honestly sanitized image I', as before, but also extracts the secret $y'_{id} \approx y^{id}$. In scheme 2, we do the same, but the additional loss term is based instead on the distance between I and the reconstructed image recovered using the output of FC_0 by an image recover decoder (see below). In both schemes, we then freeze the encoder/decoder and FC_0 weights and focus on the secret generation network to embed the secret data at locations and permutations specified by y^{ep} and c. We train the secret generation network to minimize the distance of the outputted secret matrix and the manually computed secret matrix using the output of FC_0, y^{ep} and c. The discriminators are the same as for the baseline PP-GAN.

In scheme 2, where the secret is a vector representation of I, the adversary also trains an image recovery decoder using the same architecture as the decoder in a benign PP-GAN. The recovery decoder takes as input the output of FC_0 (which will be the recovered secret).

Experimental Work[4]

Datasets

We validate our proposed approach on two facial expression datasets, FERG (Aneja et al. 2016) and MUG (Aifanti, Papachristou, and Delopoulos 2010). We split FERG into 47382 training images and 8384 test images with six subjects and seven different expressions. For MUG, we select the eight subjects with the most images available. Since MUG images are extracted from videos, the initial and final 20 frames in a clip often have neutral expressions so we ignore those frames, resulting in 8795 training images and 1609 test images with seven different expressions.

Evaluation Metrics

We evaluate the proposed adversarial PP-GAN in terms of three metrics, as described below.

Utility Measurement We measure the utility of the sanitized images via the expression classification accuracy of DL-based discriminators trained on them. Specifically, the expression classification accuracy of a discriminator trained on $\{I'_{train}, y^{ep}\}$ and tested on I'_{test} is used as the utility check for the sanitized image dataset. Ideally, we would like the utility of data sanitized by the adversarial PP-GAN to be the same as the baseline PP-GAN.

Privacy Measurement The privacy of sanitized images is measured using the weak and strong privacy checks described in the Background section. The checks measure the accuracy with which DL-based discriminators can classify the original ID from sanitized face images. Ideally, the adversarial PP-GAN should pass both privacy checks as well as the baseline.

Recoverability Finally, we measure the ability of the adversary to recover the original ID (for scheme 1), the latent representation of the input image (for scheme 2) or the original image itself (for scheme 2) from sanitized face images. The metrics used for each scenario are as follows.

ID Recovery Accuracy: is used for scheme 1 where the adversary seeks to recover the ID of input image I and is defined as the fraction of sanitized test images for which the adversary correctly recovers the secret ID.

Latent Vector Accuracy: is used for scheme 2 where the adversary seeks to recover the latent vector corresponding to input image I (from which image I can be reconstructed). Since the latent is a binary vector, we define the latent vector accuracy as the fraction of bits of the recovered latent that agree with the actual latent computed on image I.

Image Reconstruction Error: is used in scheme 2 to quantify the success of the adversary in reconstructing image I and is defined as the *MSE* distance between reconstructed images and the original input images.

Experimental Results

We begin by discussing our results on the FERG dataset for which the adversary's goal is ID recovery (scheme 1). The adversarially trained PP-GAN for FERG has the same utility accuracy as the baseline (100%). Results for the privacy and recoverability metrics are shown in Table 1 for the baseline PP-GAN (Baseline), the adversarial PP-GAN (Adv.), and for the purposes of comparison, a straw man solution in which we use the *Steghide* binary to embed the ID into the baseline PP-GAN's sanitized output.

[4]More details are in the Appendix in the arXiv version https://arxiv.org/pdf/2009.09283.pdf.

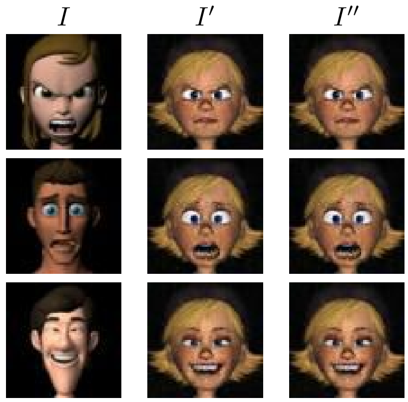

Figure 6: Selected Adversarial PP-GAN input images I, baseline sanitized images I', and outputs I'' with hidden secret (FERG dataset)

Metric	Baseline	*Steghide*	Adv.
Weak Privacy Check	0.17	0.17	0.18
Strong Privacy Check	0.30	0.30	0.30
ID Recovery Acc.	-	1.0	1.0

Table 1: Privacy and recovery metrics on the FERG dataset for the baseline PP-GAN, using *Steghide*, and with the proposed adversarial PP-GAN (Adv.).

We observe that the adversarial PP-GAN has the same accuracy for the strong privacy check and only marginally higher accuracy for the weak privacy check compared to the baseline (recall that lower accuracies imply greater privacy). We conclude that the adversarial PP-GAN would therefore pass the privacy checks. At the same time, the adversarial PP-GAN is able to recover the correct ID from sanitized images in all cases. The results from *Steghide* are identical except that it has the same accuracy for the weak privacy check as the baseline. This is because *Steghide* algorithm is fairly sophisticated, but cannot directly be used for our purposes since it is not implemented as an NN.

Figure 6 shows examples of images sanitized by the baseline and adversarial PP-GANs (centre and right columns, respectively) along with the input images (left column). Note that the sanitized images produced by the baseline and adversarial networks are visually indistinguishable.

Next we present our results on the MUG dataset for which the adversary's goal is ID recovery (scheme 1) and input image recovery (scheme 2). As with FERG dataset, the adversarially trained PP-GANs have the same utility accuracy as the baseline (100%).

The privacy and recoverability metrics for ID recovery are shown in Table 2; as with the FERG dataset, we observe that the weak and strong checks on the adversarial PP-GAN have only marginally higher accuracy compared to the baseline, and that adversary is able to recover the correct ID from 97% of sanitized images. *Steghide* has the same privacy check accuracy as the baseline and 100% recovery rate.

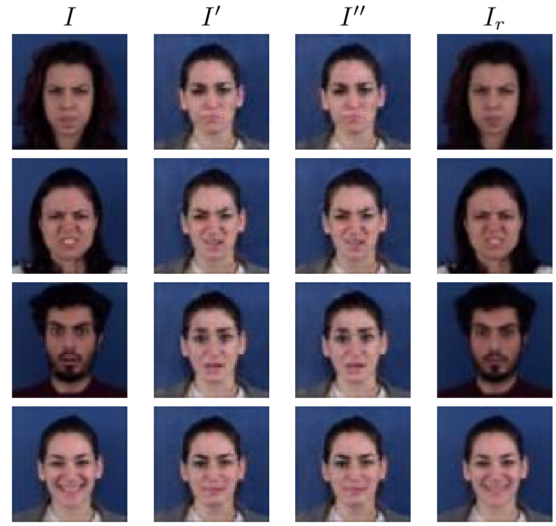

Figure 7: Selected Adversarial PP-GAN input images I, baseline sanitized images I', outputs I'' with hidden secret, and reconstructed input images I_r (MUG dataset, 18-bit secrets).

Metric	Baseline	*Steghide*	Adv.
Weak Privacy Check	0.14	0.14	0.18
Strong Privacy Check	0.29	0.29	0.30
ID Recovery Acc.	-	1.0	0.97

Table 2: Privacy and recovery metrics on the MUG dataset for the baseline PP-GAN, using *Steghide*, and with the proposed adversarial PP-GAN (Adv.).

Figure 7 shows examples of sanitized images for the baseline and adversarial PP-GANs as well as the images recovered by the adversary. The sanitized images from the baseline and adversarial PP-GANs are visually indistinguishable while the recovered images closely resemble the originals.

Table 3 tabulates the strong privacy check metric, latent vector reconstruction error and image reconstruction error for different latent vector sizes. As larger latent vectors are steganographically embedded in sanitized images, we observe lower reconstruction error at the expense of an increase in the accuracy of the strong privacy check. In all cases, the latent vector is reconstructed with $> 97\%$ accuracy. Overall, an 18-bit latent suffices to pass the privacy checks with low reconstruction errors.

Discussion

Privacy-preserving GANs have been viewed as somewhat of a panacea to the increasing concerns around surveillance technology. Our results indicate that this view might be too optimistic. Given the concerns that our work raises about the rigor of empirical privacy checks, there is a need for better evaluations of privacy. In this section, we discuss further insights following on from our experimental work.

Latent Vector Bit-length	18	24	30	36	42	48	54	60
Strong Check Acc.	0.300	0.318	0.322	0.300	0.328	0.384	0.371	0.365
Latent Vector Recons. Acc.	0.982	0.982	0.981	0.978	0.979	0.978	0.980	0.979
Image Recons. Error	6.00e-4	4.60e-4	4.22e-4	4.55e-4	3.82e-4	3.77e-4	3.36e-4	3.60e-4

Table 3: MUG dataset, Results after Training Adversarial PP-GAN architecture

Secret Hiding

More secret data bits can be hidden by the adversarial PP-GAN trained for the MUG dataset compared to that trained for the FERG dataset. This is because real human face data provides more texture variations in pixel-space, which is also reflected in the output sanitized images, and this likely introduces more "distractions" for the privacy check discriminators. The fact that DL-based discriminators are sensitive to such distractions gives us further skepticism into their use as privacy checks.

Embedding more bits in scheme 2 should help to reduce the image reconstruction error but this is counterbalanced by increasing the difficulty of learning to perform secret embedding. If the secret embedding stage is imperfect, there is higher latent vector recovery error and this results in better subversion of the privacy check, as in the case where we hide a 60-bit latent (Table 3).

DNN-based Steganography

Given our goal of having the adversarial PP-GAN implemented entirely as a NN, one possible solution is to attach a DNN-based steganography tool (Zhang et al. 2019; Hayes and Danezis 2017; Zhu et al. 2018) to the benign PP-GAN. The hiding network takes as inputs a cover image and a secret and outputs an image with the secret message hidden. The goal of these DNN-based steganography approaches is to produce secret-embedded images that are indistinguishable from cover images with respect to the probability that they contain a secret (as measured by empirically a DL-based discriminator). An accompanying reveal network extracts the secret message from the secret-embedded image. However, this approach is insufficient for our adversarial setting as the resultant network will not pass the strong privacy check. As long as there is a DL-based reveal network for secret extraction, we surmise that it is possible to train a DL-based discriminator to classify sanitized images into classes of sensitive attributes.

Threats to Validity

The privacy check measures privacy leakage via classification accuracy of a DL-based discriminator on sensitive attributes. Such networks' accuracy is affected by various factors, including the size of the training dataset, network architectures, optimization techniques, weights initialization, and training epochs, *etc*. Thus, the privacy check measurement reported in this paper is only representative of our experimental settings. We would expect different accuracy obtained for a larger dataset, different network architectures, or even more training epochs, but this again points to the unreliability of empirical privacy checks.

Related Work

Conventional privacy-preserving techniques anonymize the sensitive attributes of structured, low-dimensional and static datasets, such as *k-anonymity* (Sweeney 2002) and *l-diversity* (Machanavajjhala et al. 2007). *Differential privacy* (Dwork 2008) was proposed as a more formal privacy guarantee and can be applied to continuous and high-dimensional attributes. However, these approaches provide guarantees only when the relationship between sensitive attributes and data samples can be precisely characterized.

For applications with high-dimensional data, non-sensitive and sensitive attributes intertwine in distributions without a relation model that can be precisely extracted. Hence, empirical and task-dependent privacy checks are used to provide a holistic measure of privacy. Recent work leverages adversarial networks to sanitize input images and adopts similar DL-based discriminators for privacy examination (Edwards and Storkey 2016; Raval, Machanavajjhala, and Cox 2017; Pittaluga, Koppal, and Chakrabarti 2019; Chen, Konrad, and Ishwar 2018; Wu et al. 2018; Tseng and Wu 2020; Feutry, Piantanida, and Duhamel 2020; Maximov, Elezi, and Leal-Taixé 2020; Xiong et al. 2019). These works employ adversarial training to jointly optimize both privacy and utility objectives. Edwards and Storkey (2016) and Raval, Machanavajjhala, and Cox (2017) perform simple sanitizing tasks such as removing the QR code from a CIFAR-10 image, or removing the text in a face image, where the sensitive attributes in these cases are artificial and implicit to learn. Pittaluga, Koppal, and Chakrabarti (2019) learn the privacy preserving encodings via a similar approach but without requiring the sanitized output to be realistic looking. Similarly, Wu *et al.* aim to generate degraded versions of the input image to sanitize sensitive attributes. The idea of adversarial learning was introduced by Schmidhuber (1992), and motivated GANs as proposed by Goodfellow et al. (2014).

Conclusion

Privacy leakage of sanitized images produced by privacy-preserving GANs (PP-GANs) is usually measured empirically using DL-based privacy check discriminators. To illustrate the potential shortcomings of such checks, we produced an adversarial PP-GAN that appeared to remove sensitive attributes while maintaining the utility of the sanitized data for a given application. While our adversarial PP-GAN passed all privacy checks, it actually hid secret data pertaining to the sensitive attributes, even allowing for reconstruction of the original private image. Our experimental results highlighted the insufficiency of existing DL-based privacy checks, and potential risks of using untrusted third-party PP-GAN tools.

Ethical Impact

Our work critiques empirical privacy check metrics often used in privacy-preserving generative adversarial networks (PP-GANs). Given increasing privacy concerns with data sharing, we believe that our work provides timely insights on the implications of such approaches to privacy and will hopefully encourage more work on the rigor of privacy checks. Our work describes a technical approach that could allow an adversary to prepare and release an adversarial PP-GAN, although given that PP-GANs are not yet in widespread general use, we expect the negative impact to be minimal.

References

Aifanti, N.; Papachristou, C.; and Delopoulos, A. 2010. The MUG facial expression database. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, 1–4. IEEE.

Aneja, D.; Colburn, A.; Faigin, G.; Shapiro, L.; and Mones, B. 2016. Modeling Stylized Character Expressions via Deep Learning. In *Asian Conference on Computer Vision*, 136–153. Springer.

Chen, J.; Konrad, J.; and Ishwar, P. 2018. VGAN-Based Image Representation Learning for Privacy-Preserving Facial Expression Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1651–1660. doi:10.1109/CVPRW.2018.00207.

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

Edwards, H.; and Storkey, A. J. 2016. Censoring Representations with an Adversary. In *4th International Conference on Learning Representations, ICLR 2016*.

Feutry, C.; Piantanida, P.; and Duhamel, P. 2020. Learning Semi-Supervised Anonymized Representations by Mutual Information. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3467–3471. doi:10.1109/ICASSP40776.2020.9053379. ISSN: 2379-190X.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Hayes, J.; and Danezis, G. 2017. Generating steganographic images via adversarial training. In *Advances in Neural Information Processing Systems*, 1954–1963.

Liu, K.; Yang, H.; Ma, Y.; Tan, B.; Yu, B.; Young, E. F.; Karri, R.; and Garg, S. 2020. Adversarial Perturbation Attacks on ML-based CAD: A Case Study on CNN-based Lithographic Hotspot Detection. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 25(5): 1–31.

Machanavajjhala, A.; Kifer, D.; Gehrke, J.; and Venkitasubramaniam, M. 2007. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1(1): 3–es.

Maximov, M.; Elezi, I.; and Leal-Taixé, L. 2020. CIAGAN: Conditional Identity Anonymization Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5447–5456.

Pittaluga, F.; Koppal, S.; and Chakrabarti, A. 2019. Learning privacy preserving encodings through adversarial training. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 791–799. IEEE.

Raval, N.; Machanavajjhala, A.; and Cox, L. P. 2017. Protecting visual secrets using adversarial nets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1329–1332. IEEE.

Schmidhuber, J. 1992. Learning factorial codes by predictability minimization. *Neural computation* 4(6): 863–879.

Sheisi, H.; Mesgarian, J.; and Rahmani, M. 2012. Steganography: Dct coefficient replacement method and compare with JSteg algorithm. *International Journal of Computer and Electrical Engineering* 4(4): 458–462.

Sweeney, L. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(05): 557–570.

Tene, O.; Evans, K.; Gencarelli, B.; Maldoff, G.; and Zanfir-Fortuna, G. 2019. GDPR at Year One: Enter the Designers and Engineers. *IEEE Security & Privacy* 17(6): 7–9. ISSN 1540-7993, 1558-4046. doi:10.1109/MSEC.2019.2938196. URL https://ieeexplore.ieee.org/document/8886909/.

Tseng, B.-W.; and Wu, P.-Y. 2020. Compressive Privacy Generative Adversarial Network. *IEEE Transactions on Information Forensics and Security* 15: 2499–2513.

Wu, Y.; Yang, F.; Xu, Y.; and Ling, H. 2019. Privacy-Protective-GAN for Privacy Preserving Face De-Identification. *Journal of Computer Science and Technology* 34(1): 47–60. ISSN 1000-9000, 1860-4749. doi:10.1007/s11390-019-1898-8. URL http://link.springer.com/10.1007/s11390-019-1898-8.

Wu, Z.; Wang, Z.; Wang, Z.; and Jin, H. 2018. Towards privacy-preserving visual recognition via adversarial training: A pilot study. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 606–624.

Xiong, Z.; Li, W.; Han, Q.; and Cai, Z. 2019. Privacy-Preserving Auto-Driving: A GAN-Based Approach to Protect Vehicular Camera Data. In *2019 IEEE International Conference on Data Mining (ICDM)*, 668–677. doi:10.1109/ICDM.2019.00077. ISSN: 2374-8486.

Zhang, K. A.; Cuesta-Infante, A.; Xu, L.; and Veeramachaneni, K. 2019. SteganoGAN: High capacity image steganography with GANs. *arXiv preprint arXiv:1901.03892* .

Zhu, J.; Kaplan, R.; Johnson, J.; and Fei-Fei, L. 2018. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, 657–672.