Compressive Privacy Generative Adversarial Network

Bo-Wei Tseng and Pei-Yuan Wu

***Abstract*— Machine learning as a service (MLaaS) has brought much convenience to our daily lives recently. However, the fact that the service is provided through cloud raises privacy leakage issues. In this work we propose the compressive privacy generative adversarial network (CPGAN), a data-driven adversarial learning framework for generating compressing representations that retain utility comparable to state-of-the-art, with the additional feature of defending against reconstruction attack. This is achieved by applying adversarial learning scheme to the design of compression network (privatizer), whose utility/privacy performances are evaluated by the utility classifier and the adversary reconstructor, respectively. Experimental results demonstrate that CPGAN achieves better utility/privacy trade-off in comparison with the previous work, and is applicable to real-world large datasets.**

***Index Terms*— Compressive privacy, cyber security, privacy preserving machine learning, adversarial learning, generative adversarial networks, machine learning as a service.**

I. Introduction

MACHINE learning as a service (MLaaS) has brought much convenience to our daily lives recently. However, privacy issues arise as an abundance of private information are being collected, uploaded, and stored on the cloud. For instance, In the early 2018, 87 millions of Facebook profiles were being noticed harvested by Cambridge Analytica without user consent, and such profiles had been exploited for political purposes including U.S. presidential election [1]; in March, 2018, 150 million of MyFitnessPal user's account information (e.g., usernames, email addresses) were being deduced by unauthorized third parties; in October, 2018, Cathay Pacific reported data breach involving 9.4 million passenger's personal information such as passport numbers, identity card numbers, and credit card number [2].

Privacy concerns have become an even more important issue in collaborative learning, where data from various data sources/owners are being collected and analyzed to fulfill the request of big data for better MLaaS performances. However, it is often the case where the various data owners feel reluctant to share their data amongst others, especially via the cloud. According to a survey [3], 68% out of one thousand businesses with more than one hundred employees in UK are not prepared to have their data available for open access, mostly due to concerns in corporate privacy, protection of intellectual property, as well as concerns that online data may be mismanaged. How to ensure confidential data remains protected while harnessing data for better MLaaS remains an important issue.

A. Threats Behind MLaaS

A good survey of common threats associated with the data sharing process faced in MLaaS can be found in [4]. In model inversion attack [4]–[6], the adversary aims to reconstruct the average representation (feature vector) of each category from the released model. For instance, the intruder might exploit the confidence values revealed along with the model predictions, and estimate the average representation of a category through randomly synthesizing a data entry yielding the highest confidence value of that specific category [6]. Model inversion attack often leads to reconstruction attack, since the adversary may apply their knowledge to reconstruct the private raw data from the average categorical reconstructed feature vectors. For example, Feng and Jain illustrates how an intruder is capable of reconstructing fingerprint based on the seemly compact minutiae representation [7]. Al-Rubaie and Chang [8] also illustrates the gesture (touch event) can be reconstructed from features such as velocity and direction.

In membership inference attack, the attacker aims to recognize whether or not a data entry was within the training dataset. For instance, the intruder might exploit the confidence value distribution from the target model's prediction, and then determine whether the data entry is trained on this model or not [9].

In lieu of the various threats associated with the MLaaS, and to venture MLaaS into more sensitive but usually more impactful data such as medical history, bio-metrics, personal interests and preferences, strong guarantees of privacy is necessary for collecting the private and sensitive information, and privacy protection mechanisms must be taken into account in both training and prediction in MLaaS. Several main streams of privacy protection mechanisms for ML is reviewed as follows.

B. Differential Privacy (DP)

DP [10] is based on perturbation techniques, by adding noise to either the training data [11], the iterations in an

Manuscript received May 10, 2019; revised October 28, 2019 and December 30, 2019; accepted January 4, 2020. Date of publication January 20, 2020; date of current version February 6, 2020. This work was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant MOST-107-2634-F-002-008- and Grant MOST-108-2634-F-002-005-. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Siwei Lyu. *(Corresponding author: Pei-Yuan Wu.)*

Bo-Wei Tseng is with the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan.

Pei-Yuan Wu is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, and also with the Ministry of Science and Technology Joint Research Center for AI Technology and All Vista Healthcare, Taipei 10617, Taiwan (e-mail: peiyuanwu@ntu.edu.tw).

Digital Object Identifier 10.1109/TIFS.2020.2968188

algorithm [12], [13], the objective function [14], or the output [15]. The stochastic nature in DP ensures that the removal or addition of a single entry in the training dataset does not (substantially) alter the model released by a randomized algorithm. This hinders the adversary from inferring the existence of a specific data entry based on the released model, and thus provides protection against membership inference attack.

We say a randomized algorithm $\mathscr{A}$ gives ϵ-differential privacy if for each pair of "neighboring" datasets D_1 and D_2 that differ by at most one data entry, and for all (measurable) subsets $S \subseteq Range(\mathscr{A})$, the following inequality holds:

$$\mathbb{P}[\mathscr{A}(D_1) \in S] \leq e^{\epsilon}\mathbb{P}[\mathscr{A}(D_2) \in S] \tag{1}$$

DP ensures that the existence of a single data entry does not (substantially) alter the probability distribution of the stochastic model learned from the randomized algorithm, as can be seen by rewriting (1) as[1]

$$e^{-\epsilon} \leq \frac{\mathbb{P}[\mathscr{A}(D_1) \in S]}{\mathbb{P}[\mathscr{A}(D_2) \in S]} \leq e^{\epsilon}. \tag{2}$$

It follows that the smaller value of ϵ (a.k.a. privacy budget) indicates better privacy, as the probability distribution of the stochastic model is indifferent to the existence of a data entry.

C. Local Differential Privacy (LDP)

The notion of DP is not only applied to the *global privacy* context, where institutions release databases of answer queries from a collection of people, but also to the *local privacy* context, a scenario where each individual discloses their personal information. This leads to the concept of LDP [16], where local privacy is achieved by each individual perturbing their data before releasing it to the data user.

In the setting of LDP, each individual applies a stochastic privatization mechanism Q to map their data entry $X_i \in \mathscr{X}$ stochastically to its privatized (sanitized) views $Y_i \in \mathscr{Y}$. We say a stochastic privatization mechanism Q is ϵ-locally differentially private if for all (measurable) subsets $S \subseteq \mathscr{Y}$, and for any $x, x' \in \mathscr{X}$, the following inequality holds:

$$e^{-\epsilon} \leq \frac{\mathbb{P}[Q(x) \in S]}{\mathbb{P}[Q(x') \in S]} \leq e^{\epsilon}$$

LDP preserves local privacy by ensuring that, for small values of ϵ, any data $X_i \in \mathscr{X}$ would not (substantially) alter the probability distribution of the privatized data Y_i.

LDP has been implemented in Google's RAPPOR system, which combines randomized response with bloom filters to identify popular web destinations (URLs) without revealing individual user's browsing behavior [17]. LDP has also been practiced by Microsoft in the telemetry collection over time, as well as by Apple and its developers in the collection typing and usage history [18].

Though DP provides context-free theoretical guarantees on the distinguishability between any two "neighboring" datasets from the released model, the DP guarantees often come at a price of significantly reduced utility and increase in sample complexity [19].

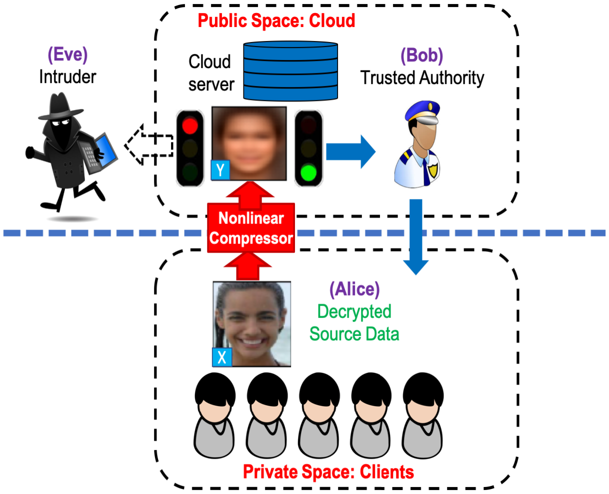


Fig. 1. The CP architecture [24]. Alice wishes to enjoy the image recognition service provided by Bob on the cloud, while concerning with the risk that her private image data might get exposed to malicious intruder (Eve) eavesdropping behind the cloud. To preserve privacy, Alice applies some privatization mechanism to her image before sending to the cloud, while Bob provides image recognition service based on the privatized image accessible on the cloud.

D. Homomorphic Encryption (HE)

HE [20] is based on cryptography approach, where arithmetic operations on the plaintext is homomorphic to certain computations on the ciphertext, thus allowing learning algorithm to compute the model without ever knowing the dataset in plaintext. To our best knowledge, Phong [21] first incorporates the additively HE scheme to the synchronous stochastic gradient descent applied to neural network. The main drawbacks of HE, however, is the grand computation cost accompanying encryption/decryption.

E. Compressive Privacy (CP)

The key idea behind CP [22] is to apply a privatization mechanism at the data owner's end, which maps the original data into some low-dimensional space by extracting the key features for the MLaaS, before sending to the cloud, that is, *data owner should have control over data privacy* [23]. In brief, CP is a dimension reduction mechanism for privacy preserving machine learning.

The CP architecture is illustrated in Figure 1, where Alice wishes to enjoy the image recognition service provided by Bob on the cloud. However, Alice is concerned with the risk that her private image data might get exposed to malicious intruders (Eve) eavesdropping behind the cloud. To preserve privacy, Alice applies some privatization mechanism to her facial image before sending to the cloud, while Bob provides image recognition service based on the privatized image accessible on the cloud. The trade-off between utility (information gained by the service) and privacy (information gained by the adversary) in the CP architecture can be analyzed under the information bottleneck (IB) paradigm, leading to a notion of "differential mutual information" (DMI) [22], [24] serving as a quantitative guideline for the design of the privatization mechanism, leading to the discriminant component analysis (DCA) privatization mechanism, as well as its

[1] Note that the roles of D_1 and D_2 can be interchanged.

extension to kernel-induced feature spaces, a.k.a. kernel-DCA (KDCA) [22]. However, to simplify analysis, in [22], [24] it is assumed that all data are Gaussian-distributed and all transformations are linear, posing a restriction on applying DMI to nonlinear CP architecture or non-Gaussian data.[2] Further studies on nonlinear CP include Chanyaswad's work on multi-kernel KDCA [25], as well as Mert's work [26] on hybrid models where fully-connected neural network is applied to the features extracted from KDCA.

Compressive privacy is also linked to the deep variational information bottleneck [27], which is a data-driven variational approximation of the information bottleneck modeled by a neural network, that aims to find a compact encoding from the input that is maximally informative for Bob to provide services.

F. Information Theoretic (IT) Privacy

The privacy threat of a data owner releasing data to a honest-but-curious adversary can also be formulated as a modified rate-distortion problem, leading to the IT privacy. The data owner applies a privatization machanism before releasing the data, so that privacy leakage is minimized subject to constraints on distortion of useful data. The privacy leakage is predominantly measured by the mutual information an adversary gains about the sensitive data upon receiving the released data after privatization mechanism [28], [29]. The optimal privacy-utility tradeoff under various privatization mechanism constraints are further discussed in Basciftci et. al.'s work [30].

G. Generative Adversarial Privacy (GAP)

An inherit challenge in taking IT privacy approach to real world applications is the difficulty of accessing the priors, such as the joint probability distributions of the sensitive and released data. Towards this end, the data driven approach motivated by the general concept of adversarial learning framework in Generative Adversarial Networks (GAN) [31] is applied for optimizing the privatization mechanisms [32], [33], such as Generative Adversarial Privacy (GAP) [19], Privacy Preserving Adversarial Networks (PPAN) [34], Compressive Adversarial Privacy (CAP) [35], and Reconstructive Adversarial Network (RAN) [36].

Adversarial learning has been a widely applied mechanism in recent years. Motivated by Schmidhuber [37], the generative adversarial network (GAN) [31] proposed by Goodfellow had achieved remarkable success in the field of image synthesis [38]–[40]. To deal with the gradient vanishing issues in GAN, Arjovsky proposed Wasserstein GAN (WGAN) [41], [42] which measures the distance between model distribution and real distribution by earth-moving distance instead of JS-divergence, as well as adding a gradient penalty term to its objective function. Other improvements include Durugka and Gemp's work on generative multi-adversarial network [43], where the generator is guided by the ensemble of multiple discriminators to provide better image synthesis.

In GAN-inspired data driven privatization approaches, censoring representations of the private data are extracted from privatization network, while the attacker tries to recover some sensitive variable from the censoring representations through an adversarial network. An optional predictor network is often assumed that attempts to provide ML service by predicting some usable variables from the censoring representation. The privatization mechanism is optimized through a game-theoretic formulation: while the adversarial network tries to recover the sensitive variable from the censoring representation, such as by maximum a posterior (MAP) estimation, the privatization network (playing the role as the generator in GAN) aims to maximizing the loss for the adversarial network while minimizing the loss for the predictor network.

To further understand the performance of GAN-inspired privatization mechanism compared to game-theoretic optimal, various multi-dimensional Gaussian mixture data models are investigated [19], [34] with game-theoretically optimal privatization mechanisms derived. The comparison shows that the gap between theory and data-driven approaches are negligible.

H. Our Contributions

In this work the *compressive privacy generative adversarial network* (CPGAN) framework is presented, which is a data-driven local privatization scheme that creates low-dimensional representation (referred as compressing representations) to be provided for the cloud services, that removes the sensitive information from the raw data. Experimental results show that CPGAN achieves better trade-off between *privacy and utility* than previous works.

We list our main contributions below.

- We demonstrate CPGAN achieves the best utility/privacy trade-off on the benchmark dataset in comparison with the previous work. We also demonstrate that CPGAN attains comparable utility accuracy whilst resisting the reconstruction attack on the real image dataset assuming white-box attack.
- In view of the potential failure of neural network with no guaranteed global optimization solvers, we consider multiple adversary strategies for more robust evaluation of adversarial reconstruction attack. In particular, we consider additional adversary strategies with guaranteed global optimization solvers, such as linear ridge regression and kernel ridge regression.
- We incorporate the funnel layer proposed by Mert [26] into CPGAN, thus enabling the compression of raw data into lower dimension. This adds an additional factor for trading-off between privacy and utility, and the compressing representation may also find potential savings in communication cost.

Organization: The remainder of this paper is organized as follow: In Section II, we formally introduce the CPGAN framework, whilst highlighting the distinction between CPGAN and previous works. In Section III the theoretical analysis on Gaussian mixture model is elaborated.

[2] In [22] the notion of DMI is extended to kernel-based CP architectures. However, the data (after mapping to the kernel-induced feature space) in general does not follow Gaussian distribution, even if the raw data itself follows Gaussian distribution.

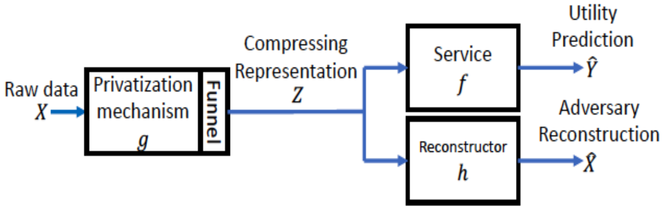


Fig. 2. The CPGAN architecture. The compressing representation generated by the privatization mechanism is sent to the cloud to enjoy the utility service, while at the same time suffering the risk of falling into the hands of the adversary, by which reconstruction attack is performed.

In Section IV we illustrate the utility/privacy tradeoff of CPGAN on both the benchmark and real datasets. Conclusions and future works are summarized in Section V.

II. Methodology

The proposed method, CPGAN (Compressive privacy generative adversarial network), is a data-driven approach, where the joint probability distributions required by IT privacy is learned from data directly. As illustrated in Figure 2, CPGAN consists of the Privatizer, Reconstructor and the Classifier, all of which are trained end-to-end through adversarial learning scheme. In Section II-A we elaborate the utility/privacy objective functions, as well as the comparison between CPGAN and other previous works. In Section II-B we elaborate multiple adversarial reconstruction strategies. The pseudo-code will be described in Section II-C.

A. Mathematical Formulation

As illustrated in Figure 2, consider the scenario where Alice wishes to enjoy some ML service (e.g., the image recognition service) provided by Bob on the cloud, where the ML service aims to extract utility variable $Y \in \mathcal{Y}$ (e.g., the category of an image) accompanying raw data entry $X \in \mathcal{X} = \mathbb{R}^m$ (e.g., the raw image). Alice applies privatization mechanism $g : \mathcal{X} \rightarrow \mathcal{Z} = \mathbb{R}^d$ to her raw data X, and releases compressing representation $Z = g(X)$ to the cloud. On one hand, Bob provides service based on the compressing representation, denoted as $f : \mathcal{Z} \rightarrow \mathcal{Y}$, and feedback prediction $\hat{Y} = f(Z)$ to Alice. On the other hand, the adversary (Eve) tries to reconstruct the raw data entry from the compressing representation through a reconstructor $h : \mathcal{Z} \rightarrow \mathcal{X}$, leading to the reconstructed data $\hat{X} = h(Z)$. In the following context, we adopt notations g_θ, h_ϕ, f_τ to further emphasize the privatizer, adversary reconstructor, and the service (also referred as utility classifier in the remaining context) are parameterized by θ, ϕ, and τ, respectively.

Besides hard decision rules, one may also consider a broader class of stochastic formulation [44], in which case the privatization mechanism g, the service f, and the reconstructor h can be instead represented as conditional probability distributions $P_g(Z|X)$, $P_f(\hat{Y}|Z)$, $P_h(\hat{X}|Z)$, respectively. In the following context we will discuss mainly under stochastic formulation, as it contains hard decision rules as special cases.

The privatization mechanism is evaluated by both the utility and privacy perspectives, as elaborated as follows:

1) *Utility Perspective:* From the utility perspective, we model the quality of service Bob provides based on the privatized data by the utility loss

$$L_{util}(g, f) = \mathbb{E}_{\substack{(X,Y)\sim P_{\mathcal{X}\times\mathcal{Y}} \\ Z|X\sim P_g(\cdot|X)}} [\ell_{util}(P_f(\cdot|Z), Y)]$$

where $P_{\mathcal{X}\times\mathcal{Y}}$ denotes the joint probability distribution of $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, and ℓ_{util} is some well-defined utility loss function that both Alice and Bob aim to minimize for better service. The utility loss can be chosen in a variety of ways:

- In regression services where $\mathcal{Y} = \mathbb{R}$, one may adopt the squared-loss function [19], [34], [35]

  $$\ell_{util}(P_f(\cdot|Z), Y) = \mathbb{E}_{\hat{Y}\sim P_f(\cdot|Z)}[(\hat{Y} - Y)^2]$$

  Note that due to the stochastic nature of the regression service, the utility loss is defined as the expectation over the squared error between ground-truth Y and the stochastic prediction $\hat{Y} \sim P_f(\cdot|Z)$. The optimal regression service is a deterministic function $f^*(Z) = \mathbb{E}[Y|Z]$, namely the conditional mean of Y given Z.
- In classification services where $\mathcal{Y}$ is categorical, one may adopt the 0-1 loss function [36], [45]

  $$\ell_{util}(P_f(\cdot|Z), Y) = -P_f(Y|Z)$$

  in which case the optimal classification service is the maximum a posterior probability (MAP) decision rule $f^*(Z) = \text{argmax}_{y\in\mathcal{Y}}\mathbb{P}[Y = y|Z]$. Note that the deterministic function (f^*) can be seen as a special case of the stochastic mapping (i.e., taking specific values with probability 1).
- In information-theoretic formulation, the log-loss function is often considered [44]

  $$\ell_{util}(P_f(\cdot|Z), Y) = -\log P_f(Y|Z).$$

  The optimal service is attained at $P_{f^*}(y|Z) = \mathbb{P}[Y = y|Z]$, in which case the utility loss is equivalent to the conditional entropy and mutual information metrics

  $$L_{util}(g, f^*) = H(Y|Z) = H(Y) - I(Z; Y)$$

  where $(X, Y) \sim P_{\mathcal{X}\times\mathcal{Y}}$, $Z|X \sim P_g(\cdot|X)$.

2) *Privacy Perspective:* From the privacy perspective, we model the quality of reconstruction by Eve from the privatized data with the adversary loss

$$L_{adv}(g, h) = \mathbb{E}_{\substack{(X,Y)\sim P_{\mathcal{X}\times\mathcal{Y}} \\ Z|X\sim P_g(\cdot|X)}} [\ell_{adv}(P_h(\cdot|Z), X)] \tag{3}$$

where ℓ_{adv} is some well-defined adversarial loss function that Eve aims to minimize to better reconstruct the raw data entries. In this work we consider the adversary loss function to be the mean squared error (MSE) [36] between the original data entry and the reconstructed data:

$$\ell_{adv}(P_h(\cdot|Z), X) = \mathbb{E}_{\hat{X}\sim P_h(\cdot|Z)}[\|\hat{X} - X\|_2^2]$$

Note that due to the stochastic nature of the adversary attack, the privacy loss is defined as the expectation over the squared error between raw data X and the stochastic reconstruction $\hat{X} \sim P_h(\cdot|Z)$.

On the one hand, data owner (Alice) wishes to find a privatization mechanism that preserves both privacy and utility. On the other hand, for a given privatization mechanism, both the service provider (Bob) and the adversary (Eve) wishes to minimize their corresponding losses. As a result, we formulate the optimization problem for the privatization mechanism as follows

$$\max_{g} \left(\min_{h} L_{adv}(g, h) - \lambda \min_{f} L_{util}(g, f) \right) \tag{4}$$

Here λ controls the trade-off between the privacy and utility. There are several distinctions between this work and previous literature:

- In GAP [34] and PPAN [19], the adversary's goal is to identify some private variable, which the data owner wishes to keep secret. However, in the case where one has no prior knowledge on which specific private variable to protect, the privatization mechanism should at least protect the raw image from reconstruction attack. As a result, in this work we consider the adversary's goal as to reconstruct data owner's private images from the compressing representations rather than specifying some private variable to identify.
- The utility loss defined in GAP, PPAN, and CAP is the distortion (often taken as the MSE) between the raw data and the best reconstruction based on the compressing representation. On the contrary, in this work the reconstruction error is viewed as the adversary loss (cf. (3)). The motivation is that without a specifically defined private variable to protect, we should at least protect the raw data from reconstruction attacks.
- This work adopts the 0-1 utility loss and ℓ_2 adversary loss, which is identical to RAN. However, there are several distinctions between CPGAN and RAN:
  - **Adversary architecture**: In RAN, the privacy is evaluated by an adversary with a single neural network [36]. In this work, to further prevent the underestimate of adversarial threat, multiple linear/nonlinear reconstruction strategies are applied by the adversary besides the neural network. The multiple reconstruction strategies will be further elaborated in Section II-B.
  - **Funnel layer**: To further enhance the privacy preserving mechanism, it is nature to place a narrow and funneling layer to retain the low dimensional representations, which preserve more utility related but less privacy related information. As a consequence, the funnel layer is adopted to further reduce and control the dimension of the compressing representation, as illustrated in Figure 2. Funnel layer serves as another factor for trading-off between privacy and utility, which will be further elaborated in Section IV-B3.
  - **Training strategy**: In RAN, in each epoch both the generator (referred as encoder in their work) and the utility classifier are first adjusted to optimize the utility loss, then the adversary adjusts the reconstructor to minimize the reconstruction MSE. In CPGAN, in each epoch the utility classifier and the adversary first optimize their own parameters independently with fixed compressing representation, after then they serve as a measure of the utility/privacy loss that leads the privatizer to adjust its parameters, which seems to more naturally resembling the GAN formulation.

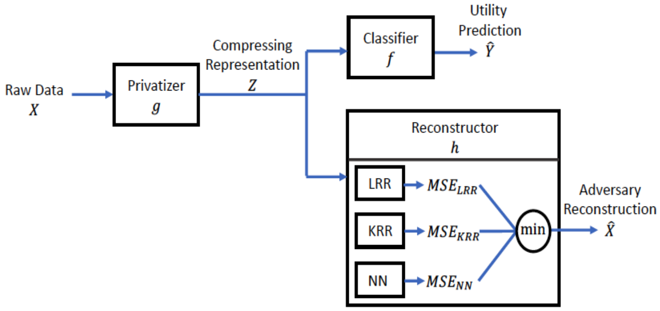


Fig. 3. The evaluation of adversarial loss via multiple reconstruction attack strategies. The adversary may train the reconstructor with multiple strategies such as LRR, KRR, or NN, and choose the strategy attaining minimum MSE as the reconstructor. At each epoch the adversarial reconstructor attained by the best strategy is chosen to update the privatizer.

B. Multiple Adversarial Reconstruction Attack Strategies

In RAN, the adversarial loss is evaluated through minimizing the reconstructor MSE, often realized by a neural network. However, training a neural network is in general a non-convex optimization problem. It is also questionable whether the neural network converges during training, or if it gets stuck at a saddle point or local optimal instead of global optimal, leading to an under-estimated adversary. To avoid updating the privatizer with the weak adversary caused by not properly trained neural network, we propose multiple adversary strategies to robustly evaluate the upper bound for the adversary loss regardless of optimization issues in neural networks.

Suppose there are N samples of raw data entry $\mathbf{x}_i \in \mathcal{X}$ with dimension m (i.e. $\mathbf{x}_i \in \mathbb{R}^m$), and their compressing representation $\mathbf{z}_i \sim P_g(\cdot|\mathbf{x}_i)$ with dimension d (i.e. $\mathbf{z}_i \in \mathbb{R}^d$), we consider various strategies for learning the adversarial reconstructor h by minimize the regularized empirical reconstruction error

$$L_{adv}^{(emp)}(g, h) = \frac{1}{N} \sum_{i=1}^{N} \ell_{adv}(P_h(\cdot|\mathbf{z}_i), \mathbf{x}_i) + \rho L_{reg}(h) \tag{5}$$

where ρ and $L_{reg}(h)$ are some regularization term/function tunable by the adversary to prevent over-fitting.

As illustrated in Figure 3, the evaluation of adversarial loss considers multiple reconstruction attack strategies. The adversary may train the re-constructor with multiple strategies such as LRR, KRR, or NN, and choose the strategy attaining minimum MSE as the reconstructor. At each epoch the adversarial reconstructor attained by the best strategy is chosen to update the privatizer. We elaborate the various reconstruction strategies with closed form solutions (LRR,KRR) in details as follows:

1) Linear Ridge Regression (LRR): In LRR, the reconstructor takes the form of a linear function $h(\mathbf{z}) = W^T \mathbf{z}$, and the regularization function is often taken as the Frobenius norm of $W \in \mathbb{R}^{d \times m}$, namely

$$L_{reg}^{LRR}(h) = \|W\|_F^2 = \sum_{i=1}^{d} \sum_{j=1}^{m} |w_{ij}|^2.$$

The regularized empirical reconstruction loss (cf. (5)) has closed-form optimal solution [46]

$$h_{LRR}(\mathbf{z}) = W_{LRR}^T \mathbf{z}$$

where

$$W_{LRR} = \left(\frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i \mathbf{z}_i^T + \rho I \right)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i \mathbf{x}_i^T \right) \tag{6}$$

In this manuscript the ridge term for LRR is set to $\rho = 0.001$.

2) Kernel Ridge Regression (KRR): The basic idea behind kernel trick is to first transform the patterns $\mathbf{z} \in \mathcal{Z}$ into some reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ through a nonlinear mapping $\phi : \mathcal{Z} \to \mathcal{H}$, on which supervised/unsupervised learning algorithms are applied [46]. The RKHS and the nonlinear mapping are both defined by a kernel function $k : \mathcal{Z} \times \mathcal{Z} \to \mathcal{H}$ that satisfies Mercer condition [47], such that the kernel function represents the inner product on the RKHS

$$k(\mathbf{z}, \mathbf{z}') = \langle \phi(\mathbf{z}), \phi(\mathbf{z}') \rangle_{\mathcal{H}}$$

Common choices include the Gaussian radial basis function (RBF) kernel $k_{RBF}(\mathbf{z}, \mathbf{z}') = \exp\left(-\frac{\|\mathbf{z}-\mathbf{z}'\|_2^2}{2\sigma^2} \right)$ and the polynomial kernel $k_{Poly\_p}(\mathbf{z}, \mathbf{z}') = \left(1 + \frac{\mathbf{z}^T \mathbf{z}'}{\sigma^2} \right)^p$. KRR is essentially LRR in the RKHS, where the reconstruction function $h(\mathbf{z}) = [h_1(\mathbf{z}) \ \cdots \ h_m(\mathbf{z})]^T$ in concern takes the form

$$h_j(\mathbf{z}) = \langle \mathbf{u}_j, \phi(\mathbf{z}) \rangle_{\mathcal{H}} = \sum_{i=1}^{N} a_{ji} k(\mathbf{z}_i, \mathbf{z})$$

where $\mathbf{u}_j$ is the weights of LRR in the RKHS, and the additional constraint posed by the second equation is justified by the representation theorem [48] or learning subspace property [46] (i.e. $\mathbf{u}_j = \phi(\mathbf{z})\mathbf{a}_j$). Under such setting, the regularized empirical reconstruction loss (cf. (5)) attains minimum by the closed-form solution

$$h_{KRR}(\mathbf{z}) = \mathbf{A}_{KRR}^T \mathbf{k}(\mathbf{z})$$

where $\mathbf{k}(\mathbf{z}) \in \mathbb{R}^N$ with $[\mathbf{k}(\mathbf{z})]_i = k(\mathbf{z}_i, \mathbf{z})$. Denote $\mathbf{K} \in \mathbb{R}^{N \times N}$ with $[\mathbf{K}]_{ij} = k(\mathbf{z}_i, \mathbf{z}_j)$, and $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_N] \in \mathbb{R}^{m \times N}$, then

$$\mathbf{A}_{KRR} = (\mathbf{K} + N\rho \mathbf{I})^{-1} \mathbf{X}^T \tag{7}$$

Though (7) is a closed form solution [46], it involves solving a system of N linear equations, which is prohibitive when N is large. Towards this end, large-scale kernel machines such as Nystroem method [49] and random Fourier features [50] may significantly reduce the training cost at the expense of approximating the RBF kernel with low rank kernels. In this manuscript we apply random Fourier features, with the parameters summarized in Table I. Note that both the two closed form adversaries (h_{LRR}, h_{KRR}) are deterministic mappings, which can be viewed as special cases of stochastic mapping function.

TABLE I

PARAMETERS OF KRR ON VARIOUS DATASETS

	Synthetic dataset	MNIST	HAR	GENKI-4K	SVHN	CIFAR-10	CelebA
Ridge	1	0.001	1	1	0.001	0.001	0.001
Mapping dimension	10000	500	5000	2048	5000	5000	2000
Gamma	0.001	0.001	0.001	0.001	0.001	0.001	0.001

3) Neural Network (NN): In this work, The NN adversary is constructed by fully connected or deconvolution (upsampling) layer, and it is optimized in the black-box manner (Gradient descent). The entire architecture of NN reconstructor are specified in Appendix. A and B.

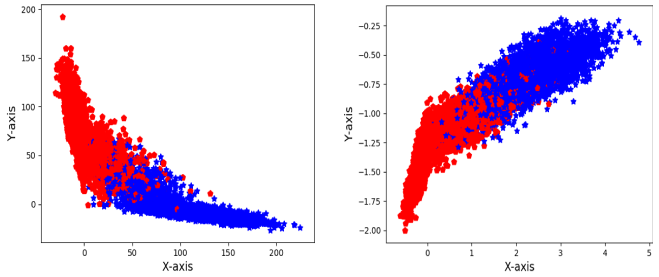
Algorithm 1 CPGAN

```
1: $g_\theta, f_\tau, h_\phi^{NN} \leftarrow$ initialize network parameters
2: for $n$ epochs do
3:     Sample minibatch of m data points $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$ from $P_{\mathcal{X}\times\mathcal{Y}}$, and sample $\mathbf{z}_i \sim P_{g_\theta}(\cdot|\mathbf{x}_i)$.
4:     Solve $W_{LRR}$ and $A_{KRR}$ according to (6) and (7).
5:     $\hat{\mathbf{x}}_{i,LRR} = W_{LRR}^T \mathbf{z}_i,\ i = 1, \ldots, m.$
6:     $L_{adv}^{LRR} = \frac{1}{m} \sum_{i=1}^m \|\hat{\mathbf{x}}_{i,LRR} - \mathbf{x}_i\|_2^2$
7:     $\hat{\mathbf{x}}_{i,KRR} = A_{KRR}^T \mathbf{k}(\mathbf{z}_i)$
8:     $L_{adv}^{KRR} = \frac{1}{m} \sum_{i=1}^m \|\hat{\mathbf{x}}_{i,KRR} - \mathbf{x}_i\|_2^2$
9:     $\hat{\mathbf{x}}_{i,NN} \sim P_{h_\phi^{NN}}(\cdot|\mathbf{z}_i)$
10:    $L_{adv}^{NN} = \frac{1}{m} \sum_{i=1}^m \|\hat{\mathbf{x}}_{i,NN} - \mathbf{x}_i\|_2^2$
11:    for $t$ steps do
12:        $h_\phi^{NN} \leftarrow h_\phi^{NN} - \alpha_h \nabla_\phi L_{adv}^{NN}$
13:    end for
14:    $L_{adv} = \min(L_{adv}^{LRR}, L_{adv}^{KRR}, L_{adv}^{NN})$        ▷ Adversary loss
15:    $L_{util} = -\frac{1}{m} \sum_{i=1}^m \log P_f(\mathbf{y}_i|\mathbf{z}_i)$        ▷ Cross entropy
16:    $f_\tau \leftarrow f_\tau - \alpha_f \nabla_\tau L_{util}$
17:    $L_{cpgan} = \lambda L_{util} - L_{adv}$
18:    $g_\theta \leftarrow g_\theta - \alpha_g \nabla_\theta L_{cpgan}$
19: end for
```

C. Algorithm

The pseudo-code for CPGAN is given in Algorithm 1. In line 4-13, three adversarial reconstruction strategies (LRR, KRR, and NN) are evaluated. Both LRR (line 5-6) and KRR (line 7-8) are evaluated through closed form, while NN is evaluated through gradient descent (line 9-13). The reconstruction MSE is then evaluated as the minimum MSE given by the various reconstruction strategies (line 14), which in combination of the utility loss (line 15) are then applied to further update the utility classifier (line 16) and the privatizer (line 17-18). Here we adopt cross entropy (log-loss) as the utility loss during the implementation, and choose $t = 15$ or 25 as the number of inner loop iterations (line 11).

It should be noted that the privatizer is bound to the machine learning service. An example of two privatization mechanisms trained over two distinct service providers with different binary



(a) Learnt from service provider 1 classifying “Male” label (b) Learnt from service provider 2 classifying “Smiling” label

Fig. 4. An example of the compressing representation (Z) given by two privatization mechanisms trained over two distinct binary classification service providers. The compressing representation is in 2D space for visualization purposes, with the color representing the label of each data point.

classification tasks is illustrated in Figure 4. In this experiment, CPGAN is first trained on two service providers classifying the label “male” and “smile” on CelebA dataset, respectively. Afterwards, the input images are compressed by the learnt privatizer whilst using PCA to further reduce the dimension into 2D space for visualization. It is clear that the compressing representations learnt over different service providers have totally different patterns, indicating the dependence of privatizer to the specific service provider.

III. Theoretical Analysis for Binar Gaussian Mixture Model

To provide a tractable theoretical benchmark for comparison to the data driven approach, in this section we extend the analysis of single-variate binary Gaussian mixture model [44] to multi-variate case.

A. Gaussian Mixture Model Settings

Consider the setting where $Y \in \{0, 1\}$, and that X is a Gaussian random variable whose mean and covariance matrix are dependent of Y. More precisely,

$$X|_{Y=0} \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0), \quad \mathbb{P}[Y=0] = p_0$$
$$X|_{Y=1} \sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1), \quad \mathbb{P}[Y=1] = p_1$$

To make the problem more tractable, we make the following simplifications:

- The two Gaussian distributions have the same co-variance matrices $\Sigma_0 = \Sigma_1 = \Sigma$.
- The privatizer is a linear mapping contaminated with noise $Z = \mathbf{A}X + E$, where $\mathbf{A} \in \mathbb{R}^{d \times m}$ and $E \sim \mathcal{N}(\mathbf{0}, \Sigma_E)$ is zero-mean noise.
- X has zero mean, namely $p_0\boldsymbol{\mu}_0 + p_1\boldsymbol{\mu}_1 = \mathbf{0}$.
- The reconstructor is a linear mapping $\hat{X} = \mathbf{W}^T Z$, where $\mathbf{W} \in \mathbb{R}^{d \times m}$. Note that adding bias in the reconstructor is reluctant as both X and Z have zero mean.

For the sake of analysis, we may write $X = \Upsilon + \Xi$, where $\Upsilon = \boldsymbol{\mu}_Y$ and $\Xi \sim \mathcal{N}(\mathbf{0}, \Sigma)$ are independent r.v.s.

B. Privacy Perspective

The adversarial loss is measured as the MSE of reconstructor. Denote $\mathbf{R}_\Upsilon = \mathbb{E}[\Upsilon\Upsilon^T]$, $\mathbf{R}_\Xi = \mathbb{E}[\Xi\Xi^T] = \Sigma$, $\mathbf{R}_E = \mathbb{E}[EE^T] = \Sigma_E$, then

$$\begin{aligned} \mathbf{R}_X &= \mathbb{E}[XX^T] = \mathbf{R}_\Upsilon + \mathbf{R}_\Xi \\ \mathbf{R}_Z &= \mathbb{E}[ZZ^T] = \mathbf{A}(\mathbf{R}_\Upsilon + \mathbf{R}_\Xi)\mathbf{A}^T + \mathbf{R}_E \\ \mathbf{R}_{ZX} &= \mathbb{E}[ZX^T] = \mathbf{A}(\mathbf{R}_\Upsilon + \mathbf{R}_\Xi) \end{aligned}$$

Denote $\boldsymbol{\mu}_{01} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. Then by assumption $\mathbb{E}[X] = p_0\boldsymbol{\mu}_0 + p_1\boldsymbol{\mu}_1 = \mathbf{0}$, one has

$$\boldsymbol{\mu}_0 = -p_1\boldsymbol{\mu}_{01}, \quad \boldsymbol{\mu}_1 = p_0\boldsymbol{\mu}_{01}$$

Hence

$$\begin{aligned} \mathbf{R}_\Upsilon &= p_0\boldsymbol{\mu}_0\boldsymbol{\mu}_0^T + p_1\boldsymbol{\mu}_1\boldsymbol{\mu}_1^T \\ &= p_0p_1^2\boldsymbol{\mu}_{01}\boldsymbol{\mu}_{01}^T + p_1p_0^2\boldsymbol{\mu}_{01}\boldsymbol{\mu}_{01}^T \\ &= p_0\ \ p_1\boldsymbol{\mu}_{01}\boldsymbol{\mu}_{01}^T. \end{aligned}$$

The best linear unbiased estimator of X given Z is $\hat{X}_{opt} = \mathbf{W}_{opt}^T Z$, where

$$\mathbf{W}_{opt} = \text{argmin}_{\mathbf{W} \in \mathbb{R}^{d \times m}} \mathbb{E}[\|X - \mathbf{W}^T Z\|_2^2] = \mathbf{R}_Z^{-1}\mathbf{R}_{ZX}$$

and the reconstruction MSE is

$$\begin{aligned} &\mathbb{E}[\|X - \hat{X}_{opt}\|_2^2] \\ &\quad = Trace\left(\mathbb{E}[(X - \mathbf{W}_{opt}^T Z)(X - \mathbf{W}_{opt}^T Z)^T]\right) \\ &\quad = Trace(\mathbf{R}_X - \mathbf{R}_{ZX}^T\mathbf{R}_Z^{-1}\mathbf{R}_{ZX}) \\ &\quad = Trace(\mathbf{R}_X - \mathbf{R}_X\mathbf{A}^T(\mathbf{A}\mathbf{R}_X\mathbf{A}^T + \mathbf{R}_E)^{-1}\mathbf{A}\mathbf{R}_X) \end{aligned}$$

C. Utility Perspective

The utiltiy loss is measured as the error in predicting Y from Z. Note that

$$\begin{aligned} Z|_{Y=0} &= \mathbf{A}(\boldsymbol{\mu}_0 + \Xi) + E \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}_0, \mathbf{A}\mathbf{R}_\Xi\mathbf{A}^T + \mathbf{R}_E) \\ Z|_{Y=1} &= \mathbf{A}(\boldsymbol{\mu}_1 + \Xi) + E \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}_1, \mathbf{A}\mathbf{R}_\Xi\mathbf{A}^T + \mathbf{R}_E) \end{aligned}$$

Denote

$$\begin{aligned} Z' &= (\mathbf{A}\mathbf{R}_\Xi\mathbf{A}^T + \mathbf{R}_E)^{-\frac{1}{2}} Z \\ \boldsymbol{\mu}'_Y &= (\mathbf{A}\mathbf{R}_\Xi\mathbf{A}^T + \mathbf{R}_E)^{-\frac{1}{2}} \mathbf{A}\boldsymbol{\mu}_Y \end{aligned}$$

then

$$Z'\big|_{Y=0} \sim \mathcal{N}(\boldsymbol{\mu}'_0, \mathbf{I}), \quad Z'\big|_{Y=1} \sim \mathcal{N}(\boldsymbol{\mu}'_1, \mathbf{I})$$

Therefore

$$\begin{aligned} \mathbb{P}[Y=1|Z'] &= \frac{p_1\mathcal{N}(Z'; \boldsymbol{\mu}'_1, \mathbf{I})}{p_0\mathcal{N}(Z'; \boldsymbol{\mu}'_0, \mathbf{I}) + p_1\mathcal{N}(Z'; \boldsymbol{\mu}'_1, \mathbf{I})} \\ &= \frac{p_1\mathcal{N}(Z''; \mu''_1, 1)}{p_0\mathcal{N}(Z''; \mu''_0, 1) + p_1\mathcal{N}(Z''; \mu''_1, 1)} \end{aligned}$$

where $\hat{\mathbf{v}} = \frac{\boldsymbol{\mu}'_1 - \boldsymbol{\mu}'_0}{\|\boldsymbol{\mu}'_1 - \boldsymbol{\mu}'_0\|_2}$, $Z'' = Z' \cdot \hat{\mathbf{v}}$, $\mu''_1 = \boldsymbol{\mu}'_1 \cdot \hat{\mathbf{v}}$, $\mu''_0 = \boldsymbol{\mu}'_0 \cdot \hat{\mathbf{v}}$. Hence the maximum a posteriori (MAP) estimation of Y given Z'' is

$$\hat{Y}_{map} = \begin{cases} 1, & \text{if } p_1\mathcal{N}(Z''; \mu''_1, 1) \geq p_0\mathcal{N}(Z''; \mu''_0, 1) \\ 0, & \text{otherwise} \end{cases}$$

which can be simplified as follows

$$\hat{Y}_{map} = \begin{cases} 1, & \text{if } Z'' \geq z''_{mid} \\ 0, & \text{otherwise} \end{cases}$$

where

$$z''_{mid} = \frac{\mu''_0 + \mu''_1}{2} + \frac{1}{\mu''_1 - \mu''_0} \log\left(\frac{p_0}{p_1}\right)$$

Denote

$$\mu''_{01} = \mu''_1 - \mu''_0 = \|\boldsymbol{\mu}'_1 - \boldsymbol{\mu}'_0\|_2 = \|(\mathbf{A}\mathbf{R}_{\xi}\mathbf{A}^T + \mathbf{R}_{\boldsymbol{\epsilon}})^{-\frac{1}{2}}\mathbf{A}\boldsymbol{\mu}_{01}\|_2,$$

then

$$\begin{aligned} &\mathbb{P}[\hat{Y}_{map} \neq Y] \\ &= \mathbb{P}[(Y, \hat{Y}_{map}) = (0,1)] + \mathbb{P}[(Y, \hat{Y}_{map}) = (1,0)] \\ &= p_0\ Q\left(z''_{mid} - \mu''_0\right) + p_1\ Q(\mu''_1 - z''_{mid}) \\ &= p_0\ Q\left(\frac{\mu''_{01}}{2} + \frac{1}{\mu''_{01}}\log\left(\frac{p_0}{p_1}\right)\right) + p_1\ Q\left(\frac{\mu''_{01}}{2} + \frac{1}{\mu''_{01}}\log\left(\frac{p_1}{p_0}\right)\right) \end{aligned}$$

where $Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$.

In summary, the optimal utility/privacy losses given a specific privatization scheme (as represented by $\mathbf{A}$ under linear privatization setting), are given as follows:

- The optimal utility loss by service provider:

$$\begin{aligned} L_{util}^{(opt)}(\mathbf{A}) &= \mathbb{P}[\hat{Y}_{map} \neq Y] &(14)\\ &= p_0 Q\left(\frac{\mu''_{01}}{2} + \frac{1}{\mu''_{01}}\log\left(\frac{p_0}{p_1}\right)\right) \\ &\quad + p_1 Q\left(\frac{\mu''_{01}}{2} + \frac{1}{\mu''_{01}}\log\left(\frac{p_1}{p_0}\right)\right) &(15) \end{aligned}$$

  where $\mu''_{01} = \|(\mathbf{A}\Sigma\mathbf{A}^T + \Sigma_E)^{-\frac{1}{2}}\mathbf{A}\boldsymbol{\mu}_{01}\|_2$.
- The optimal privacy loss by adversary:

$$\begin{aligned} L_{adv}^{(opt)}(\mathbf{A}) &= \mathbb{E}[\|X - \hat{X}_{opt}\|_2^2] \\ &= Trace(\mathbf{R}_X - \mathbf{R}_X\mathbf{A}^T(\mathbf{A}\mathbf{R}_X\mathbf{A}^T + \Sigma_E)^{-1}\mathbf{A}\mathbf{R}_X) \\ &&(16) \end{aligned}$$

  where $\mathbf{R}_X = \Sigma + p_0\ p_1\boldsymbol{\mu}_{01}\boldsymbol{\mu}_{01}^T$.

D. Comparison Between Empirical and Theoretical Results

To compare the utility and adversary losses achieved empirically by CPGAN through gradient descent, to the theoretical results as given by (14) and (16), we conduct experiment following the settings in Sec.III-A, where we generate synthetic dataset following Gaussian mixture model with $m = 32$, $d = 8$, $p_0 = p_1 = \frac{1}{2}$, $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ are vectors with equal elements 2 and -2, respectively. $\Sigma_E = 0.001\mathbf{I}$, $\Sigma = \mathbf{U}^T\mathbf{U}$ where $\mathbf{U} = [u_{ij}]$ is a random matrix with each element u_{ij} following i.i.d. uniform distribution on [0, 1). The privatizer, reconstructor, and classifier in CPGAN are all linear matrices, trained with 20k synthetic training samples by Adam optimizer with learning rate 0.001.

Figure 5 compares the empirical and theoretical results of the utility/privacy loss over 2k synthetic validation samples. Each dot corresponds to a trade-off factor $\lambda \in \{1, \ldots, 90\}$. Note that for each λ, the privatizer (represented by $\mathbf{A}$) is determined by minimizing $L_{cpgan}(\mathbf{A}) = \lambda L_{util}^{(opt)}(\mathbf{A}) - L_{adv}^{(opt)}(\mathbf{A})$ through gradient descent.

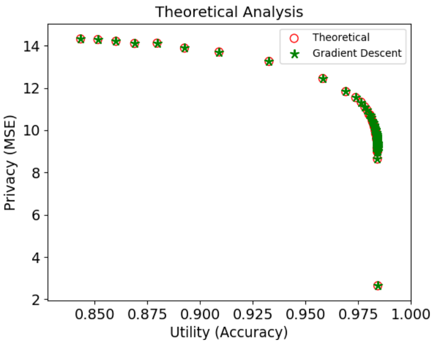


Fig. 5. Under linear case, CPGAN achieves utility/privacy trade-off nearly identical to the theoretical solution under various trade-off factors (λ).

As illustrated in Figure 5, the utility/privacy trade-off of the empirical result is very consistent to the theoretical results (namely $L_{util}^{(opt)}(\mathbf{A})$ and $L_{adv}^{(opt)}(\mathbf{A})$). It is also worth mentioning that the tuning factor λ has significant influence on the trade-off between privacy and utility.

IV. Empirical Results

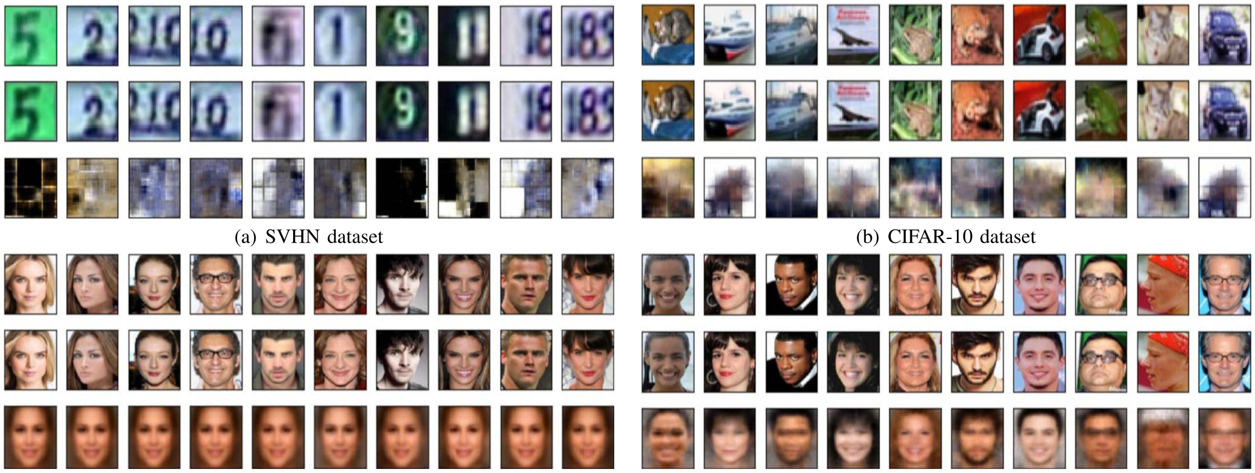
A. Privacy and Utility Trade-Off on Small Dataset

We first verify that our CPGAN can achieve the state-of-the-art privacy preserving performances on four benchmark datasets:

- **Synthetic** is described in Section III.
- **MNIST** [51] contains 60,000 handwritten digit images for the training and 10,000 handwritten digit images for the testing. All these black and white digits are size normalized, and centered in a fixed-size image with 28 × 28 pixels
- **HAR** [52] contains 10299 individual censoring signals along with six activity labels, which include walking, walking upstairs, walking downstairs, sitting, standing, laying. performing six activities.
- **GENKI-4K** [53] contains 4000 images along with expression label (i.e smile and non-smile).

And the comparison is among the following methods [36]:

- **Noise** method is a local differential privacy mechanism. It injects the Laplacian noise to the raw data X, and sends the perturbed data (X') to the deep neural network. Its privacy is evaluated by the mean square error (MSE) (i.e. $\|X - X'\|_2^2$), and the utility is evaluated by the DNN's inference accuracy.
- **DNN** method simulates the scenario that the attacker has launched the model inversion attack to intrude the feature vectors, with an aim to reconstruct the raw data and infer the sensitive information corresponding to the local users. The privacy and utility are evaluated by the reconstruction MSE and accuracy, respectively.
- **DNN(resize)** follows similar architecture design as in DNN method, with the distinction that it compresses



(a) SVHN dataset

(b) CIFAR-10 dataset

(c) CelebA dataset (Single attribute classification)

(d) CelebA dataset (Multiple attribute classification)

Fig. 6. The first row of each figure consists of randomly sampled original images. The second row consists of the reconstructed images assuming the adversary acquires the original image. The last row consists of the images reconstructed from the compressing representations under white-box attack.

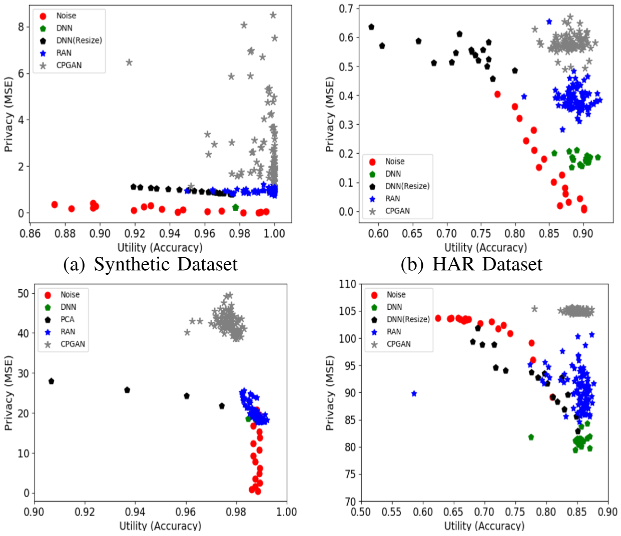
the feature vectors through principal components analysis (PCA) [54] whilst injecting the Laplician noise. It separately trains the reconstructor and classifier to evaluate the utility and privacy performance.

- **RAN** encodes the raw data into deep features (i.e the flattened feature map extracted by the convolution layers in DNN method), and applies adversarial learning to remove the privacy information in deep features. Its privacy is evaluated by the reconstruction MSE of a separately trained decoder, while its utility is evaluated by the RAN's classifier accuracy.
- **CPGAN** The utility accuracy of the compressing representations is evaluated by the classifier, and the privacy is evaluated by the reconstruction MSE of a separately trained reconstructor assuming white-box attack.

We follow Liu's implementation [36] on **Noise**, **DNN**, **DNN(resize)**, and **RAN** to serve as comparison to the proposed **CPGAN**. In both **Noise** and **DNN(resize)**, the noise follows Laplacian distribution $Lap(0, b)$ with diversity parameter $b = 0.1, 0.2, \ldots, 0.9$. The trade-off term in **RAN** is $\lambda = 0.01, 0.02, \ldots, 0.90$ [36], while in **CPGAN** we set $\lambda = 1, 2, \ldots, 90$.

Figure 7 illustrates the utility/privacy trade-off on the four small datasets, with the utility measured by classifier accuracy, and the privacy measured by the reconstruction MSE from the adversary. Here the reconstruction MSE is evaluated as the minimum MSE achieved under multiple reconstruction schemes (NN, LRR, KRR). This is to simulate a strong adversary who not only relies on NN reconstructor, but also applies LRR and KRR which are convex optimization problems with guaranteed achievable global optimal solutions. This is further justified in Figure 8, where NN does not necessarily achieve the minimal MSE compared to LRR and KRR.

In Synthetic, HAR, and GENKI-4K datasets, the best utility accuracies CPGAN achieves are comparable or better than all other methods, while achieving a much higher adversary reconstruction MSE (and hence better privacy). In MNIST dataset, the best utility accuracy achieved by CPGAN is slightly inferior than RAN by 0.5%, but with significantly higher adversary reconstruction error. This provides a circumstantial evidence that CPGAN provides better utility/privacy trade-off compared to previous works.



(a) Synthetic Dataset

(b) HAR Dataset

(c) MNIST Dataset

(d) Genki-4K Dataset

Fig. 7. CPGAN achieves the best privacy/utility trade-off compared to five privacy mechanisms on the benchmark datasets. That is, we can attain the best adversary reconstruction MSE while maintaining the utility accuracy comparable to other methods.

We also follow the experimental setting from RAN [36] to give a qualitative analysis for utility/privacy tradeoff comparison. As illustrated in Table II, the reconstructed image by CPGAN is the most unrecognizable among the various methods in comparison, while still achieving satisfactory utility accuracy result.

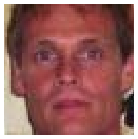
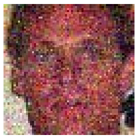
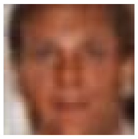
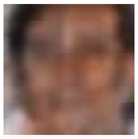
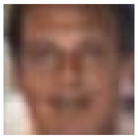
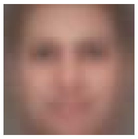
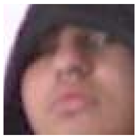
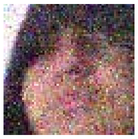
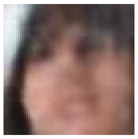
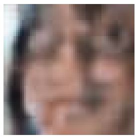
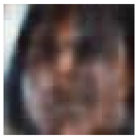
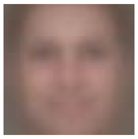
B. CPGAN on Real Datasets

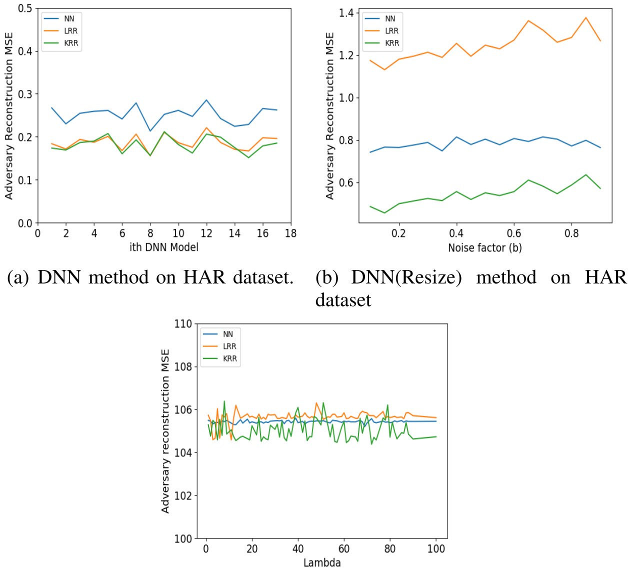
We apply CPGAN to real world datasets as described below:

- **SVHN [55]** The Street View House Numbers (SVHN) dataset contains 32 × 32 RGB images of numerical digits obtained from house numbers in Google Street

TABLE II

RECONSTRUCTED IMAGES FROM FIVE PRIVACY PRESERVING MECHANISMS ON GENKI-4K DATASET

	Original	Noise	DNN	DNN (Resize)	RAN	CPGAN
Average Utility Accuracy		69.83%	85.19%	77.70%	84.89%	84.93%
Image1						
Image2						



(a) DNN method on HAR dataset.

(b) DNN(Resize) method on HAR dataset

(c) CPGAN method on Genki-4k dataset

Fig. 8. Adversary reconstruction MSE comparison among three adversaries (LRR, KRR, NN), these results demonstrates that the adversary constructed by NN is not always the most intrusive one.

View images, which has been applied to various object recognition applications. There are 10 classes (digits 0-9), with 604388 digits for training and 26032 digits for testing. Examples are given in the top row of Figure 6(a).

- **CIFAR-10 [56]** The CIFAR-10 dataset contains 32 × 32 RGB images of animals and vehicles, which has been applied to tasks including objection recognition, image synthesis, few shot learning e.t.c. There are ten classes, with 50000 images for training[3] and 10000 images for testing. Examples are given in the top row of Figure 6(b).
- **CelebA [58]** The Large-scale CelebFaces Attributes (CelebA) dataset contains celebrity facial images, which has been applied to various computer vision tasks such as face attribute classification, face detection, landmark (or facial part) localization, and multi-task learning e.t.c. There are 202599 facial images among a total of 10177 celebrities, where each image has 40 binary attributes annotation.[4] Examples are given in the top row of Figure 6(c) and Figure 6(d).

1) Experiment on SVHN and CIFAR-10 Datasets: As the main focus of this work lies on the improvement of the privacy perspective on top of the existing utility services, we incorporate our CPGAN architecture directly to the state-of-the-art image classification models. More specifically, for CIFAR-10 and SVHN, we build the utility classifier with shake shake regularization [57] and wide residual networks [60] architectures, respectively. Note that the compressing representation has the same dimension as the original raw data, allowing the state-of-the-art classifiers to be cascaded directly to the privatizer without additional modifications. The whole implementation is trained and tested using Tensorflow [61], with the hyperparameters and the network structures specified in Appendix A.

Without the privatizer (namely, the privatizer is taken as the identity mapping), the adversary directly gains access to the raw image. Though it may seem trivial for adversary to reconstruct the raw image under such scenario (that is, by taking the reconstructor as identity mapping), we still have to verify if the identity mapping is achievable by the reconstructor. This is confirmed by the perfect reconstruction as illustrated in the second row of Figure 6(a).

On the other hand, it becomes a difficult task for adversary to reconstruct the raw image in the presence of privatizer. As illustrated in the third row of Figure 6(a) and Figure 6(b), the reconstructed images are still extremely blurred and unrecognizable, even though the reconstructor is trained by supervised learning with raw images as ground-truth.[5] which is in fact most likely an over-estimate of the adversary since only the compressing representations are released to the cloud. This shows CPGAN is indeed capable of resisting the reconstruction attacks. However, privacy comes at a price. Comparing the utility accuracy performance of CPGAN to the state-of-the-art methods (without privacy protection mechanism),

[3] We follow the data augmentation scheme in [57] to increase the training dataset by randomly cropping and flipping the raw images.

[4] We apply the *FaceNet* [59] open source code to preprocess all facial images into uniform size by cropping and alignment. All images are resized to 112 × 112 / 175 × 175 for single / multiple attributes classification, respectively.

[5] In our experiments, the training phase of the reconstructor network is terminated only after the loss function converges.

TABLE III
ACCURACY COMPARISON ON CIFAR-10 AND SVHN DATASETS

	SVHN	CIFAR-10
ResNet-20 [62]	97.70%	92.28%
Zagoruyko [60]	98.46%	96.2%
Xavier [57]	98.6%	96.45%
CPGAN	97.68%	93.87%

TABLE IV
PARAMETERS AND COMPUTATION COST ON SVHN AND CIFAR-10 DATASET

	SVHN		
	Parameters	Addition	Multiplication
Privatizer	1647	1631241	1631232
Classifier	10961834	1548669958	1548637696

	CIFAR-10		
	Parameters	Addition	Multiplication
Privatizer	1647	1631241	1631232
Classifier	2923162	877552920	879002880

CPGAN preserves privacy at the cost of slightly inferior classification accuracy. As illustrated in Table III, the utility accuracy drops by 0.92% for SVHN dataset (from 98.6% to 97.68%), and drops by 2.58% (from 96.45% to 93.87%) for CIFAR-10 dataset.

The privatizer should admit a light-weighted design, as it is executed at the local user side where computation power may be limited. As illustrated in Table IV, the privatizer requires much less parameters/additions/multiplications compared to the utility classifier. Compared with the utility classifier, the privatizer has a 6655x/949x/949x-fold savings in parameters/additions/multiplications for SVHN dataset (with shake shake regularization as utility classifier), and a 1774x/537x/538x-fold savings in parameters/additions/multiplications for CIFAR-10 dataset (with wide residual networks as utility classifier).

2) Experiment on CelebA Datasets: We apply our CPGAN to the single and multiple attribute classification on the CelebA dataset, referred as single/multi-attribute CelebA respectively in this manuscript. For single-attribute CelebA, the privatizer is built of convolution layers followed by the funnel layer, while for multi-attribute CelebA, the privatizer consists of the convolution part (from *conv1* to *concat* layer) of ATNET_G [63]. The hyerparameters and detailed network structure are given in Appendix B. All implementations on Tensorflow.

With the low dimensional compressing representations generated from the funnel layer, the utility classifier is still capable of attaining comparable utility classification accuracy as the state-of-the-art. As illustrated in Table V, Table VI and Appendix C, the utility accuracy only drops by 0.05% (from 89.97% to 89.92%) on single-attribute CelebA, and (to our surprise) increases about 0.12% (from 90.18% to 90.30%) on multi-attribute CelebA, both compared to the state-of-the-art methods (without privacy preserving mechanism). Note that here we do not compare with Han's work [64], as in their work the model is pre-trained with the CASIA-WebFace database [65]. Though pre-training is commonly adopted for extracting better features, it suffers the risk of leaking privacy information about the pretraining dataset when privacy is concerned.

TABLE V
AVERAGE ACCURACY OF SINGLE ATTRIBUTE CPGAN

	LNets+ANets [58]	Zhong [66]	CPGAN
Accuracy	87.30%	89.97%	89.92%

TABLE VI
AVERAGE ACCURACY OF MULTIPLE ATTRIBUTE CPGAN

	Han [64]	ATNET_GT [63]	CPGAN
Accuracy	92.52%	90.18%	90.30%

The privacy preserving perspective of the compressing representations is further elaborated as follows: First, following the same setting deployed on CIFAR-10 and SVHN, we verify that the reconstructor is capable of achieving the identity mapping on the CelebA dataset, where the adversary achieves perfect reconstruction when the privatizer is disabled, as illustrated in the second row of Fig 6(c) and 6(d). On the other hand, with the privatizer enabled, it becomes difficult for the adversary to reconstruct the private facial images. As illustrated in the third row of Figure 6(c), for single-attribute CelebA, the reconstructed image is roughly the mean face (at least to the authors' perspective) with person identity unrecognizable. For multi-attribute CelebA, as illustrated in the third row of 6(d), the reconstructed image is rather blurred, though facial attributes (e.g., race, mustache, smiles, etc) are still somehow recognizable. This may due to the fact that such information must be retained in the compressing representation for the utility classifier to perform its facial attribute classification task well.

The phenomenon that the facial attributes somehow remains recognizable in the reconstructed image draws our attention to explore the utility/privacy trade-off, as to be evaluated in next subsection.

3) Dimension of the Funnel Layer: To illustrate the tradeoff between privacy and utility, we tune the output unit of the funnel layer, which controls the dimension of the compressing representation, referred as compressive dimension in the following context. Table VII illustrates the reconstructed images of ten random samples as well as the utility accuracy with various compressive dimension. Here with each compressive dimension, the CPGAN is retrained following the same setting as deployed in the mutli-attribute CelebA (see Appendix B).

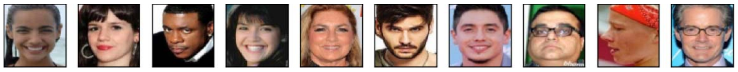
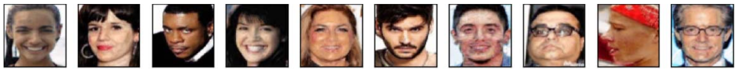
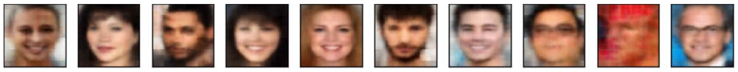
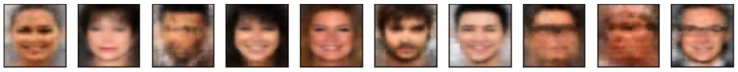
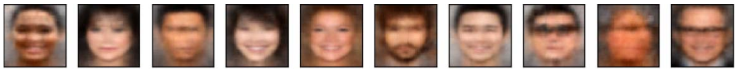
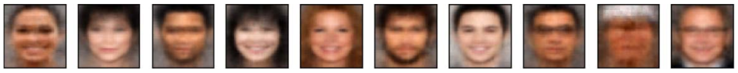
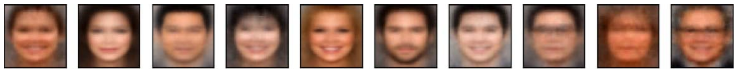
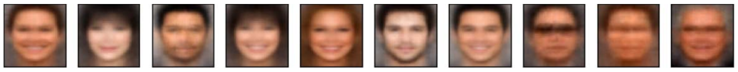
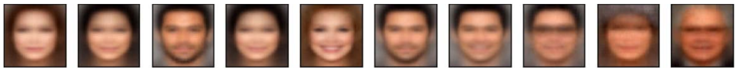
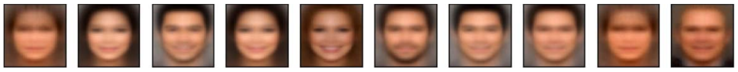
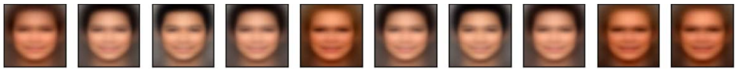
One observes that with the decreasing of the compressive dimension (from 175*175*3 to 1*2), the utility accuracy degrades (from 90.81% to 80.5%) while the adversary reconstructed image becomes more blurry and unrecognizable. This indicates that the compressive dimension indeed serves as a tuning factor for trade-off between privacy and utility, besides the concept of trade-off factor λ that also appears in previously work [36].

C. Summarizing Remarks

It may seem counter-intuitive for CPGAN, say in the example of SVHN dataset, to recognize digits based on compressing

TABLE VII

UTILITY/PRIVACY TRADE-OFF AMONG VARIOUS COMPRESSIVE DIMENSIONS

Compressive Dimension	Accuracy	Reconstructed Images
Raw images		
G=Identity[a]	90.81%	
1728*2[b]	90.21%	
128*2	90.21%	
64*2	90.19%	
32*2	89.92%	
16*2	87.63%	
8*2	87.21%	
4*2	87.06%	
2*2	85.92%	
1*2	80.5%	
Majority Classifier[c]	80.52%	

[a] The notation "G=identity" indicates the model without privacy preserving mechanism.

[b] The factor 2 is due to the fact that in multiple attribute classification problem, there are two compressing representations sent to the cloud.

[c] Majority classifier always outputs the class that is in the majority in the training set.

representations that yield non-sense reconstruction results. We hypothesize that since there are various images correspond to the same digit, the mutual information between raw image and its corresponding digit label is far less compared to the information in raw image, and most information is discarded in the compressing representation. On the other hand, in the case of CelebA dataset with multiple attribute classification (cf. Figure 6(d)), if the appearance of an image is mostly determined by the attribute information that the compressing representation aims to convey to the service provider, then it is more likely for a well-trained adversary to reconstruct recognizable raw image.

We summarize our findings as follows:

- The neural network reconstructor is not always the most intrusive adversaries in terms of minimizing reconstruction error. This motivates the design of multiple adversary reconstruction strategy.
- CPGAN achieves better utility and privacy trade-off on benchmark datasets, compared with privacy preserving mechanisms reported in literature (Noise, DNN, DNN (Resize), RAN).
- CPGAN not only attains utility accuracy comparable to the state-of-the-art model, but also considers privacy in defending the reconstruction attack, assuming threat model where the malicious attacker has white box access to the compressing representations, as well as access to the raw sensitive data (in the training phase) for building its reconstructor.
- By leveraging the funnel layer, CPGAN has two tuning factors (λ and compressive dimension d, respectively) for manipulating the trade-off between privacy and utility.

V. CONCLUSION AND FUTURE WORK

In this work we propose CPGAN as an improved data-driven adversarial learning framework for generating compressing representations, which removes the sensitive information from the raw data before sending to the cloud services. The compressing representation is extracted

TABLE VIII

IMPLEMENTATION DETAIL OF PROPOSED CPGAN ON SVHN AND CIFAR-10 DATASET

	SVHN				CIFAR-10			
	Layers	Units	Optimizer	Learning rate	Layers	Units	Optimizer	Learning rate
Privatizer	13-layer Residual Network [62], [68]		Adam	0.001	13-layer Residual Network		Adam	0.001
Reconstructor	Conv-T,[1] stride=1 Conv-T, stride=1 Conv-T, stride=1 Conv-T, stride=1	128 64 32 3	Adam	0.001	Conv-T, stride=1 Conv-T, stride=1 Conv-T, stride=1 Conv-T, stride=1	128 64 32 3	Adam	0.001
Classifier	16-8 Wide Residual Networks [60]		Adam	0.01 [2]	26-2x32d Shake-shake Regularization [57]		Momentum [69]	0.01 [2]
Epochs	160				1800			

[1] The notation ”Conv-t” means the deconvolution layers (upsampling).
[2] We apply cosine learning rate decay [57].

TABLE IX

IMPLEMENTATION DETAIL OF PROPOSED CPGAN ON CelebA DATASET

	Single attribute CelebA					Multiple attribute CelebA				
	Layers	Units	Optimizer	Learning rate	Parameter	Layers	Units	Optimizer	Learning rate	Parameter
Privatizer	Conv, stride=2 Conv, stride=2 Conv, stride=2 Conv, stride=2 Fully Connceted	64 128 256 512 compressive-d[1]	Adam [70]	0.001	414466	From ”conv_1” to ”concat” in ATNET_GT [63] Fully Connected with compressive-d units		Adam	0.001	673600
Reconstructor	Fully Connected Reshape Conv-T,[2] stride=2 Conv-T, stride=2 Conv-T, stride=2 Conv-T, stride=2	192 128 64 32 3	Adam	0.001		Fully Connected Batch Norm [71] Reshape Conv-T, stride=2 Conv-T, stride=2 Conv-T, stride=2 Conv-T, stride=2 Conv-T, stride=2	5*5*128 128 128 64 32 3	Adam	0.001	
Classifier	Fully Connected Batch Norm Fully Connected Fully Connected	256 256 1	Adam	0.001	68098	Fully Connected Batch Norm Fully Connected Fully Connected (40 branches)	64 64 1	Adam	0.001	30160
Epochs	30					30				

[1] The notation ”compressive-d” means that the dimension of the compressing representations.
[2] The notation ”Conv-t” means the deconvolution layers (upsampling).

TABLE X

ACCURACY (%) OF EACH ATTRIBUTE ON CelebA DATASET

	5 o Clock Shadow	Arched Eyebrows	Attractive	Bags Under Eyes	Bald	Bangs	Big Lips	Big Nose	Black Hair	Blond Hair	Blurry	Brown Hair	Bushy Eyebrows	Chubby	Double Chin	Eyeglasses	Goatee	Gray Hair	Heavy Makeup	High Cheekbones
LNets+ANets [58]	91	79	81	79	98	95	68	78	88	95	84	80	90	91	92	99	95	97	90	87
Zhong [66]	93	83	81	82	98	96	70	83	86	95	96	84	92	95	96	100	97	98	90	86
Han [64]	95	86	83	85	99	99	96	85	91	96	96	88	92	96	97	99	99	98	92	88
ATNET_GT [63]	92	81	81	84	99	96	71	83	89	95	96	87	92	94	96	99	97	98	90	86
Single CPGAN	92	82	80	83	98	95	71	83	89	95	95	85	90	95	96	99	96	98	90	85
Multi CPGAN	93	82	82	84	98	95	71	83	88	96	96	88	92	95	96	99	97	98	91	86

	Male	Mouth S. Open	Mustache	Narrow Eyes	No Beard	Oval Face	Pale Skin	Pointy Nose	Receding Hairline	Rosy Cheeks	Sideburns	Smiling	Straight Hair	Wavy Hair	Wearing Earrings	Wearing Hat	Wearing Lipstick	Wearing Necklace	Wearing Necktie	Young
LNets+ANets [58]	98	92	95	81	95	66	91	72	89	90	96	92	73	80	82	99	93	71	93	87
Zhong [66]	98	93	97	87	95	71	97	76	92	94	97	92	80	77	87	99	92	86	94	88
Han [64]	98	94	97	90	96	78	97	78	94	96	98	94	85	87	91	99	93	89	97	90
ATNET_GT [63]	97	93	97	86	94	76	97	75	93	95	97	92	80	82	89	99	93	86	96	88
Single CPGAN	100	93	97	89	91	72	96	75	94	95	96	92	79	78	88	99	92	83	94	87
Multi CPGAN	96	93	96	87	96	74	97	76	93	95	97	91	82	81	88	99	93	86	96	87

by privatizer, whose utility/privacy performances are evaluated by the utility classifier and the adversary reconstructor, respectively. CPGAN adopts multiple adversary reconstruction strategies besides neural networks, so as to prevent the privatizer from updating according to falsely evaluated adversary loss in the case when the neural network reconstructor is not properly optimized. It is demonstrated that CPGAN achieves better utility/privacy trade-off in comparison with the previous work, and is also applicable to real-world large datasets.

We list several extensions of CPGAN for future improvements as follows:

- **Light-weighted privatizer design** As the privatizer resides in the data owner's devices, a light-weighted privatizer design is essential for CPGAN to be adopted to mobile device applications. Though it is demonstrated such light-weighted design is possible for small-sized images, we have not yet come up with a light-weighted CPGAN design that also achieves satisfactory utility/privacy trade-off for large-sized images.
- **Better quantitative privacy measure** In this work the privacy is measured in terms of MSE of the adversary reconstructed image, which may not be the best metric. A metric that better reflects human's visual perception may lead to better evaluation of adversary loss.
- **Privacy issue in the training phase** Currently CPGAN is trained with raw sensitive data, which poses privacy concern that a corrupted data aggregater may leak the raw sensitive data during the training of CPGAN. Training CPGAN with homomorphic encryption applied to the raw sensitive data may be a work-around if one is willing to afford the grave encryption/decryption computational cost. More efficient remedies are up to exploration.
- **Optimization of GAN-type objective function** It is unclear if the min-max formulation (cf.(4)) is well-posed provided both L_{adv} and L_{util} are estimated empirically from samples. The analytical framework behind WGAN [41] and fGAN [67] may lead to a better understanding of such GAN-type formulation.

Appendix

A. Implementation Detail of Proposed CPGAN on SVHN and CIFAR-10 Dataset

See Table 8.

B. Implementation Detail of Proposed CPGAN on CelebA Dataset

See Table 9.

C. Accuracy (%) of Each Attribute on CelebA Dataset

See Table 10.

Acknowledgment

This work was supported in part by Ministry of Science and Technology (MOST), Taiwan, under grants MOST-107-2634-F-002-008- and MOST-108-2634-F-002-005-.

References

[1] M. Hron. *Top 10 Biggest Data Breaches in 2018*. Accessed: Dec. 2018. [Online]. Available: https://blog.avast.com/biggest-data-breaches

[2] D. Kwok. *Cathay Pacific Faces Probe Over Massive Data Breach. Technology News*. Accessed: Nov. 2018. [Online]. Available: https://www.reuters.com/article/us-cathaypacific-cyber/cathay-pacific-faces-probe-over-massive-data-breach-idUSKCN1NB0JY

[3] C. Arthur. *Businesses Unwilling to Share Data, But Keen On Government Doing It. The Guardian*. Accessed: Jun. 2010. [Online]. Available: https://www.theguardian.com/technology/2010/jun/29/business-data-sharing-unwilling

[4] S. Chang and C. Li, "Privacy in neural network learning: threats and countermeasures," *IEEE Netw.*, vol. 32, no. 4, pp. 61–67, Jul. 2018.

[5] M. Veale, R. Binns, and L. Edwards, "Algorithms that remember: Model inversion attacks and data protection law," *Philos. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 376, no. 2133, Nov. 2018, Art. no. 20180083, doi: 10.1098/rsta.2018.0083.

[6] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2015, pp. 1322–1333, doi: 10.1145/2810103.2813677.

[7] J. Feng and A. K. Jain, "Fingerprint reconstruction: From minutiae to phase," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 209–223, Feb. 2011.

[8] M. Al-Rubaie and J. M. Chang, "Reconstruction attacks against mobile-based continuous authentication systems in the cloud," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2648–2663, Dec. 2016.

[9] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.

[10] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Berlin, Germany: Springer, 2008, pp. 1–19.

[11] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze gauss: Optimal bounds for privacy-preserving principal component analysis," in *Proc. 46th Annu. ACM Symp. Theory Comput. (STOC)*, 2014, pp. 11–20.

[12] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2016, pp. 308–318.

[13] M. Hardt and E. Price, "The noisy power method: A meta algorithm with applications," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2861–2869.

[14] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Mar. 2011.

[15] K. Chaudhuri, A. D. Sarwate, and K. Sinha, "A near-optimal algorithm for differentially-private principal components," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2905–2943, Jan. 2013.

[16] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE 54th Annu. Symp. Found. Comput. Sci.*, Oct. 2013, pp. 429–438.

[17] U. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proc. 21st ACM Conf. Comput. Commun. Secur.*, 2014, pp. 1054–1067. [Online]. Available: https://arxiv.org/pdf/1407.6981

[18] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, "Privacy at scale: Local differential privacy in practice," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2018, pp. 1655–1658.

[19] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Generative adversarial privacy," 2018, *arXiv:1807.05306v3*. [Online]. Available: https://arxiv.org/pdf/1807.05306v3

[20] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2009. [Online]. Available: https://crypto.stanford.edu/craig

[21] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.

[22] S. Kung, "Compressive privacy: From informationestimation theory to machine learning [lecture notes]," *IEEE Signal Process. Mag.*, vol. 34, no. 1, pp. 94–112, Jan. 2017.

[23] S.-Y. Kung, T. Chanyaswad, J. M. Chang, and P. Wu, "Collaborative PCA/DCA Learning Methods for Compressive Privacy," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 3, pp. 1–18, Jul. 2017.

[24] S. Kung, "A compressive privacy approach to generalized information bottleneck and privacy funnel problems," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1846–1872, Mar. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0016003217303162

[25] T. Chanyaswad, J. M. Chang, and S. Y. Kung, "A compressive multi-kernel method for privacy-preserving machine learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 4079–4086.

[26] M. Al, T. Chanyaswad, and S.-Y. Kung, "Multi-kernel, deep neural network and hybrid models for privacy preserving machine learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2891–2895.

[27] A. Alemi, I. Fischer, J. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–19. [Online]. Available: https://arxiv.org/pdf/1612.00410

[28] D. Rebollo-Monedero, J. Forne, and J. Domingo-Ferrer, "From t-closeness-like privacy to postrandomization via information theory," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 11, pp. 1623–1636, Nov. 2010.

[29] F. Du Pin Calmon and N. Fawaz, "Privacy against statistical inference," in *Proc. 50th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2012, pp. 1401–1408.

[30] Y. O. Basciftci, Y. Wang, and P. Ishwar, "On privacy-utility tradeoffs for constrained data release mechanisms," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Jan. 2016, pp. 1–6.

[31] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2672–2680.

[32] H. Edwards and A. Storkey, "Censoring representations with an adversary," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–14.

[33] J. Hamm, "Enhancing utility and privacy with noisy minimax filters," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 6389–6393.

[34] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-preserving adversarial networks," 2017, *arXiv:1712.07008v3*. [Online]. Available: http://arxiv.org/pdf/1712.07008v3

[35] X. Chen, P. Kairouz, and R. Rajagopal, "Understanding compressive adversarial privacy," 2018, *arXiv:1809.08911*. [Online]. Available: http://arxiv.org/pdf/1809.08911

[36] S. Liu, A. Shrivastava, J. Du, and L. Zhong, "Better accuracy with quantified privacy: Representations learned via reconstructive adversarial network," 2019, *arXiv:1901.08730*. [Online]. Available: http://arxiv.org/pdf/1901.08730

[37] J. Schmidhuber, "Learning factorial codes by predictability minimization," *Neural Comput.*, vol. 4, no. 6, pp. 863–879, 1992, doi: 10.1162/NECO.1992.4.6.863.

[38] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 2642–2651.

[39] Y. U. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017.

[40] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2172–2180.

[41] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 214–223.

[42] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017. [Online]. Available: https://dblp.org/rec/bibtex/conf/iclr/ArjovskyB17

[43] I. Durugkar, I. Gem, and S. Mahadevan, "Generative multi-adversarial networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–14.

[44] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, p. 656, Dec. 2017.

[45] T. T. Nguyen and S. Sanner, "Algorithms for direct 0–1 loss optimization in binary classification," in *Proc. Int. Conf. Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 1085–1093.

[46] S. Y. Kung, *Kernel Methods and Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[47] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Philos. Trans. Roy. Soc. London*, vol. 209, pp. 415–446, Jan. 1909.

[48] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. 14th Annu. Conf. Comput. Learn. Theory 5th Eur. Conf. Comput. Learn. Theory (COLT/EuroCOLT)*. London, U.K.: Springer-Verlag, 2001, pp. 416–426.

[49] C. K. I. Williams and M. Seeger, "Using the Nystr ”om method to speed up kernel machines," in *Proc. Int. Conf. Neural Inf. Process. System (NIPS)*, 2001, pp. 682–688.

[50] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2007, pp. 1177–1184.

[51] Y. LeCun and C. Cortes. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[52] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[53] *The MPLab GENKI-4K Database*. Accessed: Jan. 28, 2020. [Online]. Available: http://mplab.ucsd.edu/

[54] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, Jul. 2010, doi: 10.1002/wics.101.

[55] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011. [Online]. Available: https://research.google/pubs/pub37648/

[56] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

[57] X. Gastaldi, "Shake-shake regularization," 2017, *arXiv:1705.07485*. [Online]. Available: http://arxiv.org/pdf/1705.07485

[58] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

[59] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[60] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2016, pp. 1–15.

[61] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: http://tensorflow.org/

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Visio (ECCV)*, 2016, pp. 630–645.

[63] D. Gao, P. Yuan, N. Sun, X. Wu, and Y. Cai, "Face attribute prediction with convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2017, pp. 1294–1299.

[64] H. Han, A. K. Jain, F. Wang, S. Shan, and X. Chen, "Heterogeneous face attribute estimation: A deep multi-task learning approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2597–2609, Nov. 2018.

[65] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," 2014, *arXiv:1411.7923*. [Online]. Available: http://arxiv.org/pdf/1411.7923

[66] Y. Zhong, J. Sullivan, and H. Li, "Face attribute prediction using off-the-shelf CNN features," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2016, pp. 1–7.

[67] S. Nowozin, B. Cseke, and R. Tomioka, "F-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 271–279.

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[69] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999, doi: 10.1016/s0893-6080(98)00116-6.

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–15.

[71] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1–11.

Bo-Wei Tseng was born in Hsinchu, Taiwan, in 1994. He received the M.S. degree from the Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, in 2019. In February 2020, he will join MediaTek Research Lab, Taipei, as a Deep Learning Researcher. His research interest lies in artificial intelligence and privacy preserving machine learning.

Pei-Yuan Wu was born in Taipei, Taiwan, in 1987. He received the B.S.E. degree in electrical engineering from National Taiwan University in 2009 and the M.A. and Ph.D. degrees in electrical engineering from Princeton University in 2012 and 2015, respectively. He was with Taiwan Semiconductor Manufacturing Company from 2015 to 2017. He has been an Assistant Professor with National Taiwan University, since 2017. His research interests include artificial intelligence, signal processing, estimation and prediction, and cyber-physical system modeling. He was a recipient of the Gordon Y.S. Wu Fellowship in 2010 and Outstanding Teaching Assistant Award at Princeton University in 2012.