MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Privacy-Preserving Adversarial Networks

Tripathy, A.; Wang, Y.; Ishwar, P.

TR2019-113 October 18, 2019

Abstract

We propose a data-driven framework for optimizing privacy-preserving data release mechanisms to attain the information-theoretically optimal tradeoff between minimizing distortion of useful data and concealing specific sensitive information. Our approach employs adversarially-trained neural networks to implement randomized mechanisms and to perform a variational approximation of mutual information privacy. We validate our Privacy-Preserving Adversarial Networks (PPAN) framework via proof-of-concept experiments on discrete and continuous synthetic data, as well as the MNIST handwritten digits dataset. For synthetic data, our model-agnostic PPAN approach achieves tradeoff points very close to the optimal tradeoffs that are analytically-derived from model knowledge. In experiments with the MNIST data, we visually demonstrate a learned tradeoff between minimizing the pixel-level distortion versus concealing the written digit.

Allerton Conference on Communication, Control, and Computing

Privacy-Preserving Adversarial Networks

Ardhendu Tripathy, Ye Wang, and Prakash Ishwar

***Abstract*—We propose a data-driven framework for optimizing privacy-preserving data release mechanisms to attain the information-theoretically optimal tradeoff between minimizing distortion of useful data and concealing specific sensitive information. Our approach employs adversarially-trained neural networks to implement randomized mechanisms and to perform a variational approximation of mutual information privacy. We validate our Privacy-Preserving Adversarial Networks (PPAN) framework via proof-of-concept experiments on discrete and continuous synthetic data, as well as the MNIST handwritten digits dataset. For synthetic data, our model-agnostic PPAN approach achieves tradeoff points very close to the optimal tradeoffs that are analytically-derived from model knowledge. In experiments with the MNIST data, we visually demonstrate a learned tradeoff between minimizing the pixel-level distortion versus concealing the written digit.**

I. Introduction

Our work addresses the problem of privacy-preserving data release, where the goal is to release useful data while also limiting the exposure of associated sensitive information. Approaches that involve data modification must consider the tradeoff between concealing sensitive information and minimizing distortion to preserve data utility. However, practical optimization of this tradeoff can be challenging when we wish to quantify privacy via statistical measures (such as mutual information) and the actual statistical distributions of data are unknown. In this paper, we propose a data-driven framework involving adversarially trained neural networks to design privacy-preserving data release mechanisms that approach the information-theoretically optimal privacy-utility tradeoffs.

Privacy-preserving data release is a broad and widely explored field, where the study of principled methods have been well motivated by highly publicized leaks stemming from the inadequacy of simple anonymization techniques, such as reported in [29], [24]. A wide variety of methods to statistically quantify and address privacy have been proposed, such as k-anonymity [30], L-diversity [18], t-closeness [16], and differential privacy [5]. In our work, we focus on an information-theoretic approach where privacy is quantified by the mutual information between the data release and the sensitive information [36], [27], [3], [28], [2].

Unlike the other privacy measures mentioned earlier, mutual information depends specifically on the statistical distribution of the data. Requiring consideration of the data distribution is a practical hindrance, however measuring privacy while ignoring the data distribution altogether can weaken the scope of privacy guarantees. For example, an adversary armed with only mild knowledge about the correlation of the data[1] can undermine the practical privacy protection of differential privacy, as noted in examples given by [13], [3], [17], [35]. While model assumptions are avoided in the definition of differential privacy, independence across individuals in the dataset is implicitly required to avoid undermining privacy guarantees [13]. The example in [3, Sec. V] demonstrates that an ϵ-differentially private mechanism can leak sensitive information on the order of $O(\epsilon^2 \log n)$, in terms of mutual information, where n is size of the dataset. Moreover, differential privacy does not satisfy the so-called *linkage inequality* [35, Def. 2], which captures the notion that privacy guarantees should also limit the disclosure of other sensitive information linked to the primary data considered, as explained further in [35]. While settling the debate over which privacy measure is most appropriate is beyond the scope of this paper, we nonetheless focus on mutual information privacy, and develop a data-driven approach that addresses the practical drawback of requiring distributional knowledge for mutual information privacy.

We build upon the non-asymptotic, information-theoretic framework introduced by [27], [3], where the sensitive and useful data are respectively modeled as random variables X and Y. We also adopt the extension considered in [2], where only a (potentially partial and/or noisy) observation W of the data is available. In this framework, the design of the privacy-preserving mechanism to release Z is formulated as the optimization of the tradeoff between minimizing privacy-leakage quantified by the mutual information $I(X;Z)$ and minimizing an expected distortion $\mathbb{E}[d(Y,Z)]$. This non-asymptotic framework has strong connections to generalized rate-distortion problems (see discussion in [27], [3], [35]), as well as related asymptotic privacy frameworks where communication efficiency is also considered in a rate-distortion-privacy tradeoff [36], [28].

In principle, when the data distribution is known, the optimal design of the privacy-preserving mechanism can be tackled as a convex optimization problem [27], [3]. However, in practice, model knowledge is often missing or inaccurate for realistic data sets, and the optimization becomes intractable for high-dimensional and continuous data. Addressing these challenges, we propose a data-driven approach that optimizes the privacy-preserving mechanism to attain the theoretically optimal privacy-utility tradeoffs, by learning from a set of training data rather than requiring model knowledge. We

A. Tripathy is with University of Wisconsin-Madison, WI 53703, email: astripathy@wisc.edu, and performed a part of this work during an internship at MERL.

Y. Wang is with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, email: yewang@merl.com.

P. Ishwar is with Boston University, Boston, MA 02215, email: pi@bu.edu.

[1]Note that even when data samples are inherently independent, the prior knowledge of an adversary could become correlated when conditioned on particular side information.

call this approach *Privacy-Preserving Adversarial Networks* (PPAN) since the mechanism, realized as a randomized neural network, is trained along with an adversarial network that attempts to recover the sensitive information from the released data. The key to attaining information-theoretic privacy is that the adversarial network specifically estimates the posterior distribution (rather than only the value) of the sensitive variable given the released data to enable a variational approximation of mutual information [1]. While the adversary is trained to minimize the log-loss with respect to this posterior estimate, the mechanism network is trained to attain the dual objectives of minimizing distortion and concealing sensitive information (by maximizing the adversarial loss).

A. Related Work

The general concept of adversarial training of neural networks was introduced by [7], which proposed *Generative Adversarial Networks* (GAN) for learning generative models that can synthesize new data samples. Since their introduction, GANs have inspired a large and growing number of adversarially trained neural network architectures for a wide variety of purposes [10].

The earlier works of [6], [8], [9] have also proposed adversarial training frameworks for optimizing privacy-preserving mechanisms, where the adversarial network is realized as a classifier that attempts to recover a discrete sensitive variable. In [6], the mechanism is realized as an autoencoder, and the adversary attempts to predict a binary sensitive variable from the latent representation. In the framework of [8], [9], a deterministic mechanism is trained with the adversarial network realized as a classifier attempting to predict the sensitive variable from the output of the mechanism. Both of these frameworks additionally propose using an optional predictor network that attempts to predict a useful variable from the output of the mechanism network. Thus, while the adversarial network is trained to recover the sensitive variable, the mechanism and predictor (if present) networks are trained to realize multiple objectives: maximizing the loss of the adversary as well as minimizing the reconstruction loss of the mechanism network and/or the prediction loss of the predictor network. However, a significant limitation of both of these approaches is that they consider only deterministic[2] mechanisms, which generally do not achieve the optimal privacy-utility tradeoffs, although neither attempts to address information-theoretic privacy. The work of [23] employs an adversarial framework similar to [6] to preserve gender-privacy of face images while retaining biometric recognition utility. Within the broader context of empirical privacy measures addressed via adversarial training, [25] considers an adversarial framework for learning accurate predictive models that preserve the membership privacy of individuals that may be in the training dataset.

The recent, independent work of [11] proposes a similar adversarial training framework, which also realizes the necessity of and proposes randomized mechanism networks, in order to address the information-theoretically optimal privacy-utility tradeoffs. They also rediscover the earlier realization of [3] that mutual information privacy arises from an adversary (which outputs a distribution) that is optimized with respect to log-loss. However, their framework does not make the connections to a general variational approximation of mutual information applicable to arbitrary (i.e., discrete, continuous, and/or multivariate) sensitive variable alphabets, and hence their data-driven formulation and empirical evaluation is limited to only binary sensitive variables.

[2]While [8], [9] does also consider a "noisy" version of their mechanism, the randomization is limited to only independent, additive noise before or after deterministic filtering.

B. Contributions and Paper Outline

Our framework, presented in Section II, provides the first data-driven approach for optimizing privacy-preserving data release mechanisms that approaches the information-theoretically optimal privacy-utility tradeoffs. A key novelty of our approach is the use of adversarial training to perform a variational approximation of mutual information privacy. Unlike previous work, our approach can handle randomized data release mechanisms where the input to the mechanism can be a general observation of the data, e.g., a full or potentially noisy/partial view of the sensitive and useful variables.

In our proposed framework all of the variables that are involved can be discrete, continuous, and/or high-dimensional vectors. We develop specific network architectures and sampling methods appropriate for various scenarios in Section II-C. In particular, when all of the variables have finite alphabets, we demonstrate that the network architectures can be efficiently minimalized to essentially just the matrices describing the conditional distributions, and that replacing sampling with a directly computed expectation improves training performance.

We evaluate our PPAN approach in Section III with experiments on synthetic data and the MNIST handwritten digit dataset. For the synthetic data experiment, we demonstrate that PPAN closely approaches the theoretically optimal privacy-utility tradeoff. In Section III-A, we consider synthetic discrete-valued data following a *symmetric pair* distribution and compare the privacy-utility tradeoff results with an approach addressing the same problem in [20]. In Section III-B3, we consider scalar jointly Gaussian sensitive and useful attributes and benchmark the performance of PPAN against the theoretically optimal privacy-utility tradeoff. In Section III-B2, we demonstrate how the PPAN framework can be used to generate rate-distortion curves studied in information theory, purely from samples. Finally in Section IV and in the full version of the paper [34, Appendix B], we provide and derive analytical expressions for the optimal privacy-utility tradeoffs for Gaussian distributed data and mean square error distortion. Some extensions of our framework and other visualizations can be found in the rest of the appendices of [34].

II. Problem Formulation and PPAN Methods

A. Privacy-Utility Tradeoff Optimization

We consider the privacy-utility tradeoff optimization problem described in [2], which extends the frameworks initiated by [27], [3]. Figure 1 depicts the problem setting where

observed data W, sensitive attributes X, and useful attributes Y are modeled as random variables that are jointly distributed according to a data model $P_{W,X,Y}$ over the space $\mathcal{W}\times\mathcal{X}\times\mathcal{Y}$. The observed data W is a potentially noisy/partial observation of the sensitive and useful data attributes (X, Y). The goal is to design and optimize the data release mechanism, i.e., a system that processes the observed data W to produce a release $Z \in \mathcal{Z}$ that minimizes the privacy-leakage of the sensitive attributes X, while also maximizing the utility gained from revealing information about Y. This system is specified by the *release mechanism* $P_{Z|W}$, with $(W, X, Y, Z) \sim P_{W,X,Y}P_{Z|W}$, and thus $(X, Y) \leftrightarrow W \leftrightarrow Z$ forms a Markov chain. Privacy-leakage is quantified by the mutual information $I(X; Z)$ between the sensitive attributes X and the release Z. Utility is inversely quantified by the expected distortion $\mathbb{E}[d(Y, Z)]$ between the useful attributes Y and the release Z, where the distortion function $d : \mathcal{Y}\times\mathcal{Z} \to [0, \infty)$ is given by the application. The design of the release mechanism $P_{Z|W}$ is formulated as the following privacy-utility tradeoff optimization problem,

$$\min_{P_{Z|W}:(X,Y)\leftrightarrow W\leftrightarrow Z} I(X; Z), \quad \text{s.t.} \quad \mathbb{E}[d(Y, Z)] \le \delta, \qquad (1)$$

where the parameter δ indicates the distortion (or *disutility*) budget allowed for the sake of preserving privacy.

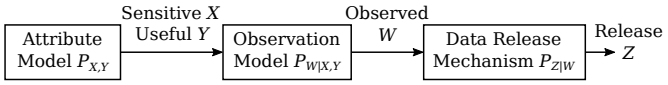


Fig. 1: Setting for privacy-utility tradeoff optimization.

As noted in [2], given a fixed data model $P_{W,X,Y}$ and distortion function d, the problem in (1) is a convex optimization problem, since the mutual information objective $I(X; Z)$ is a convex functional of $P_{Z|X}$, which is in turn a linear functional of $P_{Z|W}$, and the expected distortion $\mathbb{E}[d(Y, Z)]$ is a linear functional of $P_{Y,Z}$ and hence also of $P_{Z|W}$. While the treatment in [2] considers discrete variables over finite alphabets, the formulation of (1) need not be limited those assumptions. Thus, in this work, we seek to also address this problem with high-dimensional, continuous variables. Although outside the focus of this work, in [34, Appendix C] we discuss how mutual information privacy is impacted if there is some side information about X available to an attacker. In [34, Appendix D], we discuss how to handle mutual information as a *utility* function within the PPAN framework as opposed to expected distortion that we focus on in this work.

B. Adversarial Training for an Unknown Data Model

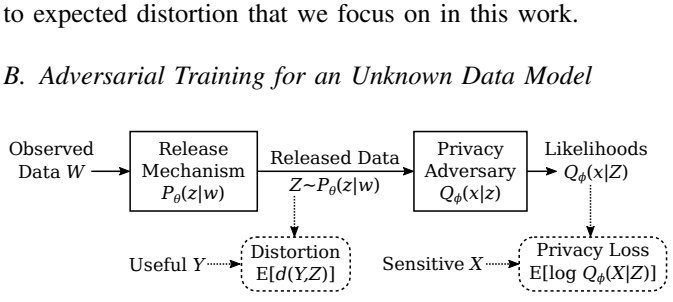


Fig. 2: Adversarial training framework.

Our aim is to solve the privacy-utility tradeoff optimization problem when the data model $P_{W,X,Y}$ is unknown but instead a set of training samples: $\{(w_i, x_i, y_i)\}_{i=1}^n \sim$ i.i.d. $P_{W,X,Y}$ is available.[3] A key to our approach is approximating $I(X; Z)$ via a variational lower bound given by [1] and also used in [4]. This bound is based on the following identity which holds for any distribution $Q_{X|Z}$ over $\mathcal{X}$ given values in $\mathcal{Z}$

$$-h(X|Z) = \mathrm{KL}(P_{X|Z}\|Q_{X|Z}) + \mathbb{E}\big[\log Q_{X|Z}(X|Z)\big],$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback-Leibler (KL) divergence. Therefore, since $I(X; Z) = h(X) - h(X|Z)$ and KL divergence is nonnegative,

$$h(X) + \max_{Q_{X|Z}} \mathbb{E}\big[\log Q_{X|Z}(X|Z)\big] = I(X; Z), \qquad (2)$$

where the maximum is attained when the variational posterior $Q_{X|Z} = P_{X|Z}$. Using (2) with the constant $h(X)$ term dropped, we convert the formulation of (1) to an unconstrained minimax optimization problem,

$$\min_{P_{Z|W}} \max_{Q_{X|Z}} \mathbb{E}\big[\log Q_{X|Z}(X|Z)\big] + \lambda\mathbb{E}\big[d(Y, Z)\big], \qquad (3)$$

where the expectations are with respect to $(W, X, Y, Z) \sim P_{W,X,Y}P_{Z|W}$, and the parameter $\lambda > 0$ can be adjusted to obtain various points on the optimal privacy-utility tradeoff curve. Alternatively, to target a specific distortion budget δ, the second term in (3) could be replaced with a penalty term $\lambda(\max(0, \mathbb{E}[d(Y, Z)] - \delta))^2$, where $\lambda > 0$ is made relatively large to penalize exceeding the budget. The expectations in (3) can be conveniently approximated by Monte Carlo sampling over training set batches.

The minimax formulation of (3) can be interpreted and realized in an adversarial training framework (as illustrated by Figure 2), where the variational posterior $Q_{X|Z}$ is viewed as the posterior likelihood estimates of the sensitive attributes X made by an adversary observing the release Z. The data release mechanism is trained to minimize both the distortion and privacy loss terms, while the adversary is trained to maximize the privacy loss. Specifically, the adversary attempts to maximize the negative log-loss $\mathbb{E}[\log Q_{X|Z}(X|Z)]$, which the release mechanism $P_{Z|W}$ attempts to minimize. The release mechanism and adversary are realized as neural networks, which take as inputs W and Z, respectively, and produce the parameters that specify their respective distributions $P_{Z|W}$ and $Q_{X|Z}$ within parametric families that are appropriate for the given application. For e.g., a release mechanism suitable for the release space $\mathcal{Z} = \mathbb{R}^d$ could be the multivariate Gaussian

$$P_{Z|W}(z|w) = \mathcal{N}(z; (\boldsymbol{\mu}, \boldsymbol{\Sigma}) = f_\theta(w)),$$

where the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ are determined by a neural network f_θ as a function of w and controlled by the parameters θ. For brevity of notation, we will use $P_\theta(z|w)$ to denote the distribution defined by the release mechanism network f_θ. Similarly, we will let $Q_\phi(x|z)$ denote the parametric distribution defined by the adversary network that is controlled by the parameters ϕ. For each training

[3]For the case when X is not explicitly available during training, or it is vaguely defined, please see the discussion in [34, Appendix E] .

sample tuple (w_i, x_i, y_i), we sample k independent releases $\{z_{i,j}\}_{j=1}^{k} \overset{\text{iid}}{\sim} P_\theta(z|w_i)$ to approximate the loss term with

$$\mathcal{L}^i(\theta, \phi) := \frac{1}{k} \sum_{j=1}^{k} \left[\log Q_\phi(x_i|z_{i,j}) + \lambda d(y_i, z_{i,j})\right]. \quad (4)$$

The networks are optimized with respect to these loss terms averaged over the training data (or mini-batches)

$$\min_\theta \max_\phi \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}^i(\theta, \phi), \quad (5)$$

which approximates the theoretical privacy-utility tradeoff optimization problem as given in (3), since by the law of large numbers, as $n \to \infty$,

$$\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}^i(\theta, \phi) \xrightarrow{\text{a.s.}} \mathbb{E}\big[\log Q_\phi(X|Z) + \lambda d(Y, Z)\big],$$

where the expectation is with respect to $(W, X, Y, Z) \sim P_{W,X,Y} P_\theta(z|w)$. Similarly, the second term in (4) could be replaced with a penalty term $\lambda(\max(0, d(y_i, z_{i,j}) - \delta))^2$ to target a specific distortion budget δ. Similar to GANs [7], the minimax optimization in (5) can be more practically handled by alternating gradient descent/ascent between the two networks (possibly with multiple inner maximization updates per outer minimization update) rather than optimizing the adversary network until convergence for each release mechanism network update. See [34, Appendix A] for the pseudocode description of the algorithm.

C. Sampling the Release Mechanism

To allow optimization of the networks via gradient methods, the release samples need to be generated such that the gradients of the loss terms can be readily calculated. Various forms of the release mechanism distribution $P_\theta(z|w)$ are appropriate for different applications, and each require their own specific sampling methods. Finite alphabet models are appropriate for categorical data such as star ratings and quantized census data whereas Gaussian, mixture of Gaussian or more general real-valued models are more appropriate for voice, image, video, and other physical sensor data.

1) Finite Alphabets: When the release space $\mathcal{Z}$ is a finite discrete set, we can forgo sampling altogether and calculate the loss terms via

$$\mathcal{L}^i_{\text{disc}}(\theta, \phi) := \sum_{z \in \mathcal{Z}} P_\theta(z|w_i)(\log Q_\phi(x_i|z) + \lambda d(y_i, z)), \quad (6)$$

which replaces the empirical average over k samples with the direct expectation over Z. We found that this direct expectation produced better results than estimation via sampling, such as by applying the Gumbel-softmax categorical reparameterization trick (see [19], [12]). Here we assume that the alphabet size is known. Since this is a data-driven mechanism, we will obtain good performance if the empirical distribution of the training data does not diverge much from the actual unknown dataset distribution. This is often a standard assumption in different setups, for e.g., in the Probably Approximately Correct (PAC) notion of learning. In practice, Bayesian priors for the estimation of the conditional distributions that appear in (6) could also be incorporated in order to mitigate the curse-of-dimensionality issue wherein the alphabet sizes are much larger than the size of the training set.

Further, if $\mathcal{W}$ and $\mathcal{X}$ are also finite alphabets, then $P_\theta(z|w)$ and $Q_\phi(x|z)$ can be exactly parameterized by matrices of size $|\mathcal{Z}| \times |\mathcal{W}|$ and $|\mathcal{X}| \times |\mathcal{Z}|$, respectively. Thus, in the purely finite alphabet case, with the variables represented as one-hot vectors, the mechanism and adversary are most efficiently realized as networks with no hidden layers and softmax applied to the output (to yield stochastic vectors).

2) Gaussian Approximations for Reals: A multivariate Gaussian release mechanism can be sampled by employing the reparameterization trick of [15], which first samples a vector of independent standard normal variables $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and then generates $z = \mathbf{A}\mathbf{u} + \boldsymbol{\mu}$, where the parameters $(\boldsymbol{\mu}, \mathbf{A}) = f_\theta(w)$ are produced by the release mechanism network to specify a conditional Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^T$. This approach can be extended to Gaussian Mixture Models as explained in [34, Appendix F].

3) Universal Approximators: Another approach, as seen in [22], is to directly produce the release sample as $z = f_\theta(w, u)$ using a neural network that takes random seed noise u as an additional input. The seed noise u can be sampled from a simple distribution (e.g., uniform, Gaussian, etc.) and provides the randomization of z with respect to w. Since the transformations applying the seed noise can, in principle, be learned, this approach could potentially approximate any "nice" distribution due to the universal approximation properties of neural networks. However, although it is not needed for training, it is generally intractable to produce an explicit expression for $P_\theta(z|w)$ as implied by the network.

III. Experimental Results

In this section, we present the privacy-utility tradeoffs that are achieved by our PPAN framework in experiments with synthetic and real data. For the synthetic data experiments, we show that the results obtained by PPAN (which does not require model knowledge and instead uses training data) are very close to the theoretically optimal tradeoffs obtained from optimizing (1) with full model knowledge. In the experiments with discrete synthetic data presented in Section III-A, we also compare PPAN against the approach of [20], where first an approximate discrete distribution is estimated from the training data, which is then used in place of the true distribution for the optimization in (1). This two-step procedure involves model estimation as its first step, and is in general not tractable for high-dimensional continuous distributions. For the synthetic data experiment, we consider Gaussian joint distribution over the sensitive, useful, and observed data, for which we can compare the results obtained by PPAN against the theoretically optimal tradeoffs (derived in Section IV). We use the MNIST handwritten digits dataset to illustrate the application of the PPAN framework to real data in Section III-C. We demonstrate optimized networks that can trace the tradeoff between concealing the digit and reducing image distortion. Table I summarizes the data models and distortion metrics that we use

TABLE I: The models used for obtaining synthetic training and test datasets in our experiments.

Case	Attribute Model	Observation	Distortion Metric
Discrete, Sec. III-A	(X,Y) symmetric pair for $m=10, p=0.4$, see (7)	$W=Y$ and $W=(X,Y)$	$\Pr[Y\neq Z]$
Continuous, Sec. III-B4	$\left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right]\sim\mathcal{N}\left(\mathbf{0},\left[\begin{smallmatrix}I_5 & \mathrm{diag}(\boldsymbol{\rho})\\ \mathrm{diag}(\boldsymbol{\rho}) & I_5\end{smallmatrix}\right]\right)$, $\boldsymbol{\rho}=[0.47,0.24,0.85,0.07,0.66]$	$W=Y$	$\mathbb{E}[\|Y-Z\|^2]$
Continuous, Sec. III-B3	$\left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right]\sim\mathcal{N}\left(\mathbf{0},\left[\begin{smallmatrix}1 & 0.85\\ 0.85 & 1\end{smallmatrix}\right]\right)$	$W=Y$ and $W=(X,Y)$	$\mathbb{E}[(Y-Z)^2]$
Continuous, Sec. III-B2	$X=Y\sim\mathcal{N}\left(\mathbf{0},\mathrm{diag}(\boldsymbol{\sigma}^2)\right)$, $\boldsymbol{\sigma}^2=[0.47,0.24,0.85,0.07,0.66]$	$W=X=Y$	$\mathbb{E}[\|Y-Z\|^2]$

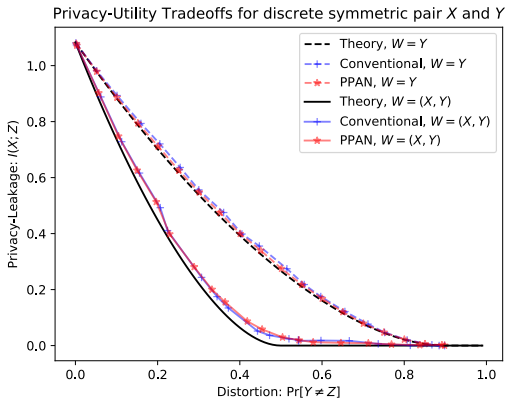


Fig. 3: Comparison of PPAN performance against the conventional model estimation approach of [20] and the theoretical optimum, for two observation scenarios: full data observed, i.e. $W=(X,Y)$, shown by solid lines, and only useful attribute observed, i.e. $W=Y$, shown by dashed lines.

in our experiments. Our experiments were implemented using the Chainer deep learning framework [33], with optimization performed by their implementation of Adam [14]. We used the Chainer-default Adam parameters in all of our experiments: $\alpha=0.001$, $\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=10^{-8}$.

A. Discrete Synthetic Data

In our experiments with discrete data, we will consider two observation models, full data (where $W=(X,Y)$) and useful data only (where $W=Y$). We use a toy distribution for the attributes for which the theoretically optimal privacy-utility tradeoffs have been analytically derived in [35], using probability of error as the distortion metric, i.e., $\mathbb{E}[\mathbf{1}(Y\neq Z)]=\Pr[Y\neq Z]$. Specifically, we consider sensitive and useful attributes that are distributed over the finite alphabets $\mathcal{X}=\mathcal{Y}=\{0,\ldots,m-1\}$, with $m\geq 2$ and parameter $p\in[0,1]$, according to the *symmetric pair* distribution given by

$$P_{X,Y}(x,y)=\begin{cases}\frac{1-p}{m}, & \text{if } x=y,\\ \frac{p}{m(m-1)}, & \text{otherwise.}\end{cases} \tag{7}$$

1) Network Architecture and Evaluation: As mentioned in Section II-C1, the network architecture for the release mechanism and adversary can be reduced to a bare minimum when all of the variables are finite-alphabet. Each network simply applies a single linear transformation (with no bias term) on the one-hot encoded input, followed by the softmax operation to yield a stochastic vector. The mechanism network takes as input w encoded as a one-hot column vector $\mathbf{w}$ and outputs $P_\theta(\cdot|w)=\mathrm{softmax}(\mathbf{Gw})$, where the network parameters $\theta=\mathbf{G}$ are the entries of a $|\mathcal{Z}|\times|\mathcal{W}|$ real matrix. Note that applying the softmax operation to each column of $\mathbf{G}$ produces the conditional distribution $P_{Z|W}$ describing the mechanism. Similarly, the attacker network is realized as $Q_\phi(\cdot|z)=\mathrm{softmax}(\mathbf{Az})$, where $\mathbf{z}$ is the one-hot encoding of z, and the network parameters $\phi=\mathbf{A}$ are entries of a $|\mathcal{X}|\times|\mathcal{Z}|$ real matrix. We optimize these networks according to (5), using the penalty term modification of the loss terms in (6) as given by

$$\mathcal{L}^i_{\text{disc}}(\theta,\phi):=\sum_{z\in\mathcal{Z}}P_\theta(z|w_i)\big(\log Q_\phi(x_i|z)+\lambda\max(0,d(y_i,z)-\delta)^2\big).$$

We use $\lambda=500$ in these experiments.

In Figure 3, we compare the results of PPAN against the theoretical baselines given by [35] (*c.f.* [34, Appendix G]), as well as against a conventional approach suggested by [20], where the joint distribution of $P_{W,X,Y}$ is estimated from the training data and then used in the convex optimization of (1). We can see that the PPAN mechanism learns a data release distribution that has close to optimal privacy leakage for a wide range of distortion values. We used 1000 training samples generated according to the symmetric pair distribution in (7) with $m=10$ and $p=0.4$. The PPAN networks were trained for 2500 epochs (for the full data observation case) with a minibatch size of 100, with each network alternatingly updated once per iteration. For the useful data only observation case, 2000 epochs were used. For evaluating both the PPAN and conventional approaches, we computed the mutual information and probability of error from the joint distribution that combines the optimized $P_{Z|W}$ with the true $P_{X,Y,W}$.

B. Gaussian Synthetic Data

In this section, we consider scalar and multivariate jointly Gaussian sensitive and useful attributes. We evaluate the performance of PPAN on synthetic data following this model in various scenarios. The distortion metric is the mean squared error between the release and the useful attribute. As we note in Section IV, the optimum release for the scenarios considered here is jointly Gaussian with the attributes. Thus we could use a mechanism network architecture that can realize the procedure described in Section II-C2 to generate the release. However, since the optimal release distribution is

not known for general attribute models, we use the universal approximator technique described in Section II-C3.

The mechanism implemented in these experiments consists of three fully connected layers, with the ReLU activation function applied at the outputs of the two hidden layers, and no activation function is used at the output layer. The mechanism takes as input observation W and seed noise U, and generates the release $Z = f_\theta(W, U)$, where θ denotes the parameters of the mechanism network. Components of the seed noise vector are i.i.d. Uniform$[-1, 1]$. The adversary network, with parameters denoted by ϕ, models the posterior probability $Q_\phi(X|Z)$ of the sensitive attribute given the release. We assume that $Q_\phi(\cdot|z)$ is a normal distribution with mean vector $\boldsymbol{\mu}_\phi(z)$ and covariance matrix $\mathrm{diag}(\boldsymbol{\sigma}^2_\phi(z))$, i.e., they are functions of the release z. The adversary network has three fully connected layers to learn the mean and variances. The network takes as input the release z and outputs the pair $(\boldsymbol{\mu}_\phi(z), \log \boldsymbol{\sigma}^2_\phi(z))$, where the log is applied componentwise on the variance vector. We use the adversarial networks to solve the min-max optimization problem described in (5). We choose $k = 1$ in (4), and similar to the previous section, we use the penalty modification of the distortion term, i.e.,

$$\mathcal{L}^i_{\text{gauss}}(\theta, \phi) = \log Q_\phi(x_i|z_i) + \lambda(\max(0, \|y_i - z_i\|^2 - \delta))^2. \tag{8}$$

The parameter δ is swept through a linearly spaced range of values. The values chosen for the multiplier λ and the distortion budget δ in various experiments is described in the sections below. For each value of δ, we sample the data model to obtain an independent dataset realization and use it to train and test the adversarial networks. We use 8000 training samples and evaluate the performance of PPAN on 4000 test samples. For the scalar data experiments, both networks have 5 nodes per hidden layer, while 20 nodes per hidden layer were used for the multivariate data experiments. Each hidden layer has 20 nodes. The adversarial networks were trained for 250 epochs with a minibatch size of 200. In each iteration we do 5 gradient descent steps to update the parameters of the adversary network before updating the mechanism network.

1) Estimating Mutual Information Leakage: Distortion caused by a release is estimated by the empirical mean squared error with respect to the testing samples. However, estimating mutual information to evaluate privacy leakage is less straightforward since the joint distribution $P_{X,Z}$ as realized by the optimized mechanism is not available explicitly. Since for these experiments, the optimal release Z is jointly Gaussian with X (as we show in Section IV), we estimate $I(X; Z)$ via a Gaussian approximation. Specifically, we use the expression for the mutual information of jointly Gaussian random vectors and replace all covariance matrices that appear there by their empirical counterparts, i.e., $\hat{I}(X; Z) = 0.5 \log(\det(\hat{\Sigma}_X)/\det(\hat{\Sigma}_{X|Z}))$, where $\hat{\Sigma}_{X|Z} := \hat{\Sigma}_X - \hat{\Sigma}_{X,Z}\hat{\Sigma}^+_Z\hat{\Sigma}^T_{X,Z}$ and $\hat{\Sigma}_X$ denotes the empirical self covariance matrix of X, $\hat{\Sigma}^+_Z$ denotes the pseudoinverse of the empirical self covariance matrix of Z, and $\hat{\Sigma}_{X,Z}$ denotes their empirical cross covariance matrix. This underestimates the true mutual information leakage since

$$\begin{aligned} I(X; Z) &= h(X) - h(X - \widehat{\mathbb{E}}[X|Z]|Z) \\ &\geq h(X) - h(X - \widehat{\mathbb{E}}[X|Z]) = \hat{I}(X; Z), \end{aligned}$$

where $\widehat{\mathbb{E}}[X|Z]$ is the linear MMSE estimate of X as a function of Z. We use this estimate only for its simplicity, and one could use other non-parametric estimates of mutual information [26].

2) Rate Distortion: We can apply the PPAN framework to the problem of computing the minimum required rate of a code that describes a multivariate source X to within a target value of expected distortion. This is a standard problem in information theory when the source distribution is known, for example, see Chapter 10 of [31]. However, the PPAN framework can be used to empirically approximate the rate-distortion curve from i.i.d. samples of the source without knowledge of the source distribution. The computation of the rate-distortion function can be viewed as a degenerate case of the PPAN framework with $W = X = Y$, i.e., the sensitive and useful attributes are the same and the observed dataset is the attribute. The release Z corresponds to an estimate $\hat{X}$ with expected distortion less than a target level while retaining as much expected uncertainty about X as possible.

We illustrate the PPAN approach using a Gaussian source $X \in \mathbb{R}^5$ and mean squared error distortion. For the experiment, we choose the attribute model $X \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(0.47, 0.24, 0.85, 0.07, 0.66))$ and the value $\lambda = 500$. We run the experiment for 20 different values of the target distortion, linearly spaced between 0 to 2.5. The inputs to the adversarial network are realizations of the attributes and seed noise. The seed noise is chosen to be a random vector of length 8 with each component i.i.d. Uniform$[-1, 1]$. The network architecture and values of other hyperparameters are the same as those used for multivariate Gaussian attributes in Section III-B. Using the learned parameters $\theta^\star$, the mechanism network generates a release as $Z = f_{\theta^\star}(W, U) = f_{\theta^\star}(X, U)$. The distortion is estimated by the empirical mean squared error of the release with respect to the training samples. The privacy loss is quantified by the estimate $\hat{I}(X; Z)$ as described in Section III-B1.

The optimal privacy-utility tradeoff (or, rate-distortion) curve is $R(D) = \min_{P(Z|X)\,:\,\mathbb{E}||X-Z||^2 \leq D} I(X; Z) = \sum_{j=1}^5 \max\{0, 0.5 \log((\boldsymbol{\sigma}[j])^2/D_j)\}$ [31], where $\boldsymbol{\sigma}^2$ are the true variance parameters of the attribute distribution and $\sum_{j=1}^5 D_j = D$. The values of D_j for each component is obtained using the Karush-Kuhn-Tucker (KKT) conditions for the constrained optimization problem, the solution of which is a standard waterfilling procedure.

We plot the (privacy-leakage, utility loss) pairs returned by the PPAN mechanism along with the optimal tradeoff curve in Figure 4. One can see that the operating points attained by the PPAN mechanism are very close to the theoretical optimum tradeoff for a wide range of target distortion values.

3) Scalar Gaussian Attributes: Consider jointly Gaussian sensitive and useful attributes such that $\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right] \sim \mathcal{N}\left(\left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 1 & 0.85 \\ 0.85 & 1 \end{smallmatrix}\right]\right)$. We analyze two different observation models here: $W = Y$, called useful data only (UD) and

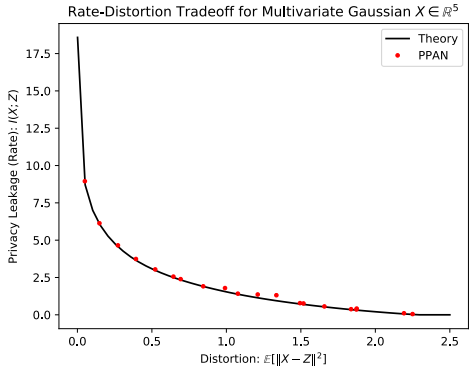


Fig. 4: Comparison of results obtained by PPAN versus the the optimal rate-distortion curve, for the rate-distortion problem where $X = Y = W$ is multivariate Gaussian.

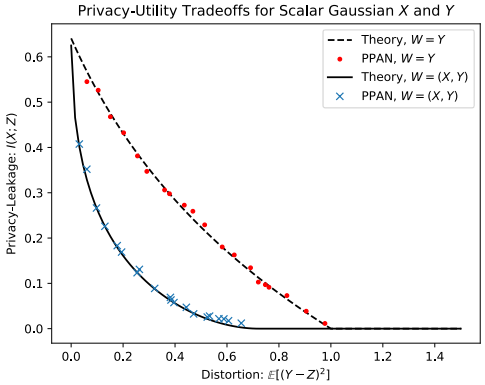


Fig. 5: Comparison of the results achieved by PPAN versus the theoretical optimum tradeoff curve, with jointly Gaussian scalar (X, Y), for the useful data only (i.e., $W = Y$) and the full data (i.e., $W = (X, Y)$) observation models.

$W = (X, Y)$, called full data (FD). The distortion metric is the mean squared error between the release and the useful attribute. The seed noise is a scalar random variable following Uniform$[-1, 1]$. The values of the multipliers chosen are: $\lambda^{\text{UD}} = 10$ and $\lambda^{\text{FD}} = 50$. In each case, we run experiments for 20 different values of the target distortion with $\delta^{\text{UD}} \in [0, 1]$ and $\delta^{\text{FD}} \in [0, 0.8]$. The privacy-leakage and distortion values returned by the PPAN mechanism on the test set are plotted along with the optimal tradeoff curves (from Propositions 1 and 3 in Section IV) in Figure 5. In both the observation models, we observe that the PPAN mechanism generates releases that have nearly optimal privacy-leakage over a range of distortion values.

4) Multivariate Gaussian Attributes: Consider multivariate jointly Gaussian sensitive and useful attributes $\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right] \sim \mathcal{N}\left(\mathbf{0}, \left[\begin{smallmatrix} I_5 & \text{diag}(\boldsymbol{\rho}) \\ \text{diag}(\boldsymbol{\rho}) & I_5 \end{smallmatrix}\right]\right)$ where both $X, Y \in \mathbb{R}^5$ and $\boldsymbol{\rho} = [0.47, 0.24, 0.85, 0.07, 0.66]$. The observation model is UD, i.e., $W = Y$. We choose the multiplier $\lambda = 10$ and 20 linearly spaced values for δ in the range $[0, 4.5]$. The seed noise is a vector with 8 components. We plot the privacy-leakage and distortion values returned by the PPAN mechanism on the test set along with the optimal tradeoff curve (from Proposition 2 in Section IV) in Figure 6. The privacy-

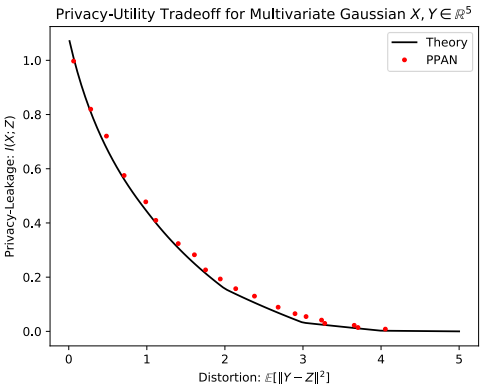


Fig. 6: Comparison of the results achieved by PPAN versus the theoretical optimum tradeoff curve for the useful data only observation model (i.e., $W = Y$) for multivariate Gaussian (X, Y).

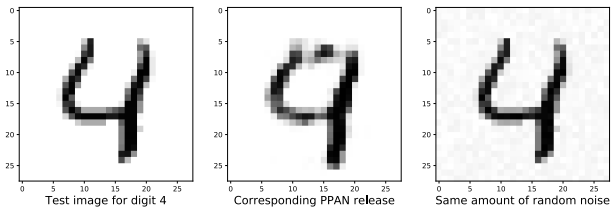


Fig. 7: A digit '4' from the MNIST test set and the release generated by a PPAN mechanism trained at $\lambda = 25$. Adding random noise to each pixel for the same amount of total noise results in the third image.

leakage values were estimated following the procedure in Section III-B1. The performance of the PPAN mechanism is very close to the theoretically optimum tradeoff curve over a wide range of target distortion values. We visualize the true data attributes and the released attributes obtained by a trained PPAN mechanism using scatter plots in [34, Appendix H].

C. MNIST Handwritten Digits

The MNIST dataset consists of 70K labeled images of handwritten digits split into training and test sets of 60K and 10K images, respectively. Each image consists of 28×28 grayscale pixels, which we handle as vectors in $[0, 1]^{784}$. In the first set of experiments, we consider the image to be both the useful and the observed data, i.e., $W = Y$, the digit label to be the sensitive attribute X, and the mechanism release as an image $Z \in [0, 1]^{784}$. We measure the distortion between the original and released images Y, Z as

$$d(Y, Z) := \frac{-1}{784} \sum_{i=1}^{784} Y[i] \log(Z[i]) + (1-Y[i]) \log(1-Z[i]),$$

which, for a fixed Y, corresponds to minimizing the average KL-divergence between corresponding pixels that are each treated as a Bernoulli distribution. Thus, the privacy objective is to conceal the digit, while the utility objective is to minimize (average pixel-level) image distortion.

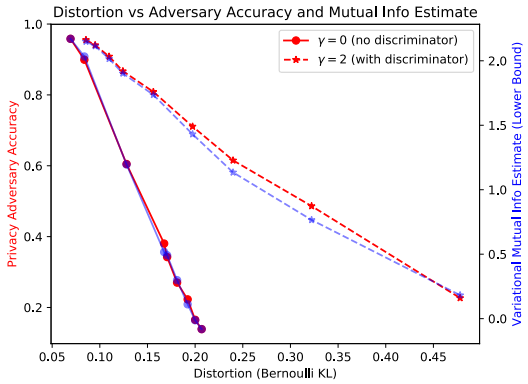


Fig. 8: Evaluation of the distortion vs privacy tradeoffs for PPAN applied to the MNIST test set, with privacy measured by adversary accuracy (in red) and estimated mutual information (in blue).

The mechanism and adversary networks both use two hidden layers with 1000 nodes each and fully-connected links between all layers. The hidden layers use $\tanh$ as the activation function. The mechanism input layer uses $784+20$ nodes for the image concatenated with 20 random Uniform$[-1,1]$ seed noise values. The mechanism output layer uses 784 nodes with the sigmoid activation function to directly produce an image in $[0,1]^{784}$. Note that the mechanism network is an example of the universal approximator architecture mentioned in Section II-C3. The attacker input layer uses 784 nodes to receive the image produced by the mechanism. The attacker output layer uses 10 nodes normalized with a softmax activation function to produce a distribution over the digit labels $\{0,\ldots,9\}$. We focus on a particular digit and the corresponding release generated by PPAN in Figure 7. PPAN learns to add noise at strategic pixels so as to best confound the digit. The third panel shows that adding random noise to each pixel, while keeping the total amount of noise added the same, is not effective at concealing the digit.

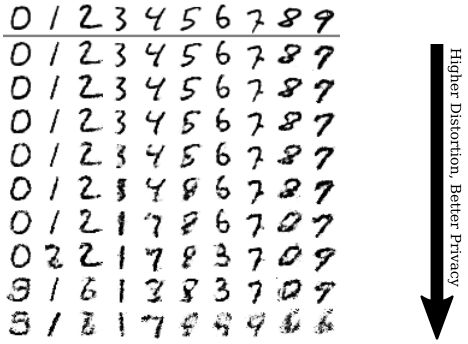
In a second set of experiments, we employ the standard GAN approach of adding a discriminator network to further encourage the mechanism to produce output images that resemble realistic digits. The discriminator network architecture uses a single hidden layer with 500 nodes, and has an output layer with one node that uses the sigmoid activation function. The discriminator network, denoted by D_ψ with parameters ψ, attempts to distinguish the outputs of the mechanism network from the original training images. Its contribution to the overall loss is controlled by a parameter $\gamma \geq 0$ (with zero indicating its absence). Incorporating this additional network, the training loss terms are given by

$$\mathcal{L}^i_{\text{mnist}}(\theta,\phi,\psi) := \log Q_\phi(x_i|z_i) + \lambda d(y_i,z_i) + \gamma \log D_\psi(z_i) + \gamma \log(1 - D_\psi(y_i)), \quad (9)$$

where z_i is generated from the input image $w_i = y_i$ by the mechanism network controlled by the parameters θ. The overall adversarial optimization objective with both the privacy adversary and the discriminator is given by

$$\min_\theta \max_{\phi,\psi} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}^i_{\text{mnist}}(\theta,\phi,\psi).$$

(a) Without additional discriminator (i.e., $\gamma = 0$).



(b) With additional discriminator ($\gamma = 2$).

Fig. 9: Examples from applying PPAN to conceal MNIST handwritten digits. Top row consists of the original test set examples input to the mechanism, while other rows show corresponding mechanism outputs at different tradeoff points.

We used the 10K test images to objectively evaluate the performance of the trained mechanisms for Figure 8, which depicts image distortion versus privacy measured by the accuracy of the adversary in recognizing the original digit and the variational lower bound for mutual information obtained by using the posterior distribution of the sensitive attribute learnt by the adversary in (2).

Figure 9 shows example results from applying trained privacy mechanisms to MNIST test set examples. The first row depicts the original test set examples input to the mechanism, while the remaining rows each depict the corresponding outputs from a mechanism trained with different values for λ. From the second to last rows, the value of λ is decreased (from 35 to 8), reducing the emphasis on minimizing distortion. We see that the outputs start from accurate reconstructions and become progressively more distorted while the digit becomes more difficult to correctly recognize as λ decreases. Figure 9a shows the results with the standard PPAN formulation, trained via (9) with $\gamma = 0$, where we see that the mechanism seems to learn to minimize distortion while rendering the digit unrecognizable, which in some cases results in an output that resembles a different digit. Figure 9b shows the results for the second set of experiments when the additional discriminator

network is introduced, which is jointly trained via (9) with $\gamma = 2$. There we see that the additional discriminator network encourages outputs that more cleanly resemble actual digits, which required lower values for λ (ranging from 15 to 2) to generate distorted images and also led to a more abrupt shift toward rendering a different digit. For both sets of experiments, the networks were each alternatingly updated once per batch (of 100 images) over 50 epochs of the 60K MNIST training set images.

IV. Optimum Privacy Utility Tradeoff for Gaussian Attributes

In Section III we compare the (privacy, distortion) pairs achieved by the model-agnostic PPAN mechanism with the optimal model-aware privacy-utility tradeoff curve. For jointly Gaussian attributes and mean squared error distortion, we can obtain, in some cases, analytical expressions for the optimal tradeoff curve as described below. Some of the steps in the proofs use bounding techniques from rate-distortion theory, which is to be expected given the tractability of the Gaussian model and the choice of mutual information and mean squared error as the privacy and utility metrics respectively.

Proposition 1. *(Useful Data only: Scalar Gaussian with mean squared error) In problem* (1)*, let X, Y be jointly Gaussian scalars with zero means $\mu_X = \mu_Y = 0$, variances σ_X^2, σ_Y^2 respectively, and correlation coefficient $\rho \in [-1, 1]$. Let mean squared error be the distortion measure. If the observation $W = Y$ (Useful Data only observation model), then the optimal release Z corresponding to*

$$\min_{P_{Z|Y}} I(X;Z), \text{ s.t. } \mathbb{E}(Y-Z)^2 \leq \delta \text{ and } X \leftrightarrow Y \leftrightarrow Z \quad (10)$$

is given by

$$Z = \begin{cases} 0, & \text{if } \delta \geq \sigma_Y^2 \\ (1-\delta/\sigma_Y^2)Y + U, & \text{if } \delta < \sigma_Y^2 \end{cases}$$

where $U \perp\!\!\!\perp (X, Y)$ and $U \sim \mathcal{N}(0, \delta(1-\delta/\sigma_Y^2))$. The mutual information leakage caused by releasing Z is

$$I(X;Z) = \max\left\{0, \frac{1}{2}\log\left(\frac{1}{1-\rho^2+\rho^2\delta/\sigma_Y^2}\right)\right\}.$$

The result of Proposition 1 is known in the existing literature, e.g., [27] (see Eq. 8) and [28] (see Example 2). For completeness, we present the proof of this result in [34, Appendix B-A]. The theoretical tradeoff curve in Figure 5 was obtained using the expressions in Proposition 1.

The case of Useful Data only observation model for jointly Gaussian *vector* attributes and mean squared error is also considered in [27], where they provide a numerical procedure to evaluate the tradeoff curve. Here, we focus on a special case where we can compute the solution analytically.

Consider the generalization to vector variables of (10)

$$\min_{P_{Z|Y}} I(X;Z) \text{ such that } \mathbb{E}(Y-Z)^T(Y-Z) \leq \delta \quad \text{and } X \leftrightarrow Y \leftrightarrow Z. \quad (11)$$

Let X, Y be jointly Gaussian vectors of dimensions m and n respectively. We assume that X, Y have zero means $\mu_X = \mu_Y = 0$ and non-singular covariance matrices $\Sigma_X, \Sigma_Y \succ 0$. Let Σ_{XY} denote the cross-covariance matrix and $P := \Sigma_X^{-\frac{1}{2}}\Sigma_{XY}\Sigma_Y^{-\frac{1}{2}}$ the normalized cross-covariance matrix with singular value decomposition $P = U_P\Lambda_P V_P^\top$. We assume that all singular values of P, denoted by $\rho_i', i = 1, \ldots, \min\{m, n\}$, are strictly positive. If

$$X' := U_P^\top \Sigma_X^{-\frac{1}{2}} X, \quad Y' := V_P^\top \Sigma_Y^{-\frac{1}{2}} Y, \quad \text{and } Z' := V_P^\top \Sigma_Y^{-\frac{1}{2}} Z$$

denote reparameterized variables, then X', Y' are zero-mean, jointly Gaussian, with identity covariance matrices I_m, I_n respectively and $m \times n$ diagonal cross-covariance matrix Λ_P. Since the transformation from (X, Z) to (X', Z') is invertible, $I(X'; Z') = I(X; Z)$. The mean squared error between Y, Z:

$$\mathbb{E}\left[(Y-Z)^\top(Y-Z)\right] = \mathbb{E}\left[(Y'-Z')^\top(V_P^\top\Sigma_Y V_P)(Y'-Z')\right].$$

For the special case when $V_P^\top \Sigma_Y V_P = cI_n$ for some $c > 0$, the vector problem (11) reduces to the following problem:

$$\min_{P_{Z'|Y'}} I(X';Z') \text{ such that } \mathbb{E}(Y'-Z')^T(Y'-Z') \leq \delta/c \quad \text{and } X' \leftrightarrow Y' \leftrightarrow Z'. \quad (12)$$

Proposition 2. *If $\left[\begin{smallmatrix} X' \\ Y' \end{smallmatrix}\right] \sim \mathcal{N}\left(\left[\begin{smallmatrix} \mathbf{0}_m \\ \mathbf{0}_n \end{smallmatrix}\right], \left[\begin{smallmatrix} I_m & \Lambda_P \\ \Lambda_P^T & I_n \end{smallmatrix}\right]\right)$, then the minimizer of* (12) *is given by*

$$Z_i' = (1-\delta_i')Y_i' + U_i, i = 1, \ldots, \min\{m, n\},$$

where $(U_1, \ldots, U_{\min\{m,n\}}) \perp\!\!\!\perp (X', Y')$ and for all i, $U_i \sim \mathcal{N}(0, \delta_i'(1-\delta_i'))$, $\delta_i' := \min\{1, t - ({\rho_i'}^{-2} - 1)\}$, where $\rho_i' > 0$ denotes the i-th main diagonal entry of Λ_P, and the value of parameter t can be found by the equation $\sum_i \delta_i' = \delta/c$. The mutual information between the release and the sensitive attribute is $I(X', Z') = \sum_{i=1}^{\min\{m,n\}} \max\{0, -0.5\log(1 - {\rho_i'}^2 + {\rho_i'}^2\delta_i')\}$.

The proof of the above proposition is given in [34, Appendix B-B]. We evaluate the above parametric expression for various values of δ in order to obtain the theoretical tradeoff curves in Figure 6.

For the case of full data observation, we have the following result.

Proposition 3. *(Full Data: Scalar Gaussian with mean squared error) In problem* (1)*, let X, Y be jointly Gaussian scalars with zero means, unit variances, and correlation coefficient $\rho \in [0, 1]$. Let mean squared error be the distortion measure. If the observation $W = (X, Y)$ (full data observation model), then the optimal release Z corresponding to*

$$\min_{P_{Z|X,Y}} I(X;Z), \quad \text{such that} \quad \mathbb{E}(Y-Z)^2 \leq \delta \quad (13)$$

is given by

$$Z = (1-\delta)Y - (X-\rho Y)\sqrt{\frac{\delta(1-\delta)}{1-\rho^2}}.$$

The mutual information leakage caused by this release is $I(X;Z) = 0$ if $\delta \geq \rho^2$, and if $\delta < \rho^2$:

$$I(X;Z) = \frac{1}{2}\log\left(\frac{1}{1-\left(\sqrt{\rho^2(1-\delta)} - \sqrt{(1-\rho^2)\delta}\right)^2}\right).$$

The proof of the above proposition is presented in [34, Appendix B]. The theoretical tradeoff curve in Figure 5 was obtained using the above expression.

V. Conclusion

In this work we introduced and developed a practical, data-driven method for optimizing privacy-preserving data release mechanisms within the well-established information-theoretic framework. The key to this approach is the application of adversarially-trained neural networks, where the mechanism is realized as a randomized network, and a second network acts as a privacy adversary that attempts to recover sensitive information. By estimating the posterior distribution of the sensitive variable given the released data, the adversarial network enables a variational approximation of mutual information. This allows our method to approach the information-theoretically optimal privacy-utility tradeoffs, which we demonstrate in experiments with discrete and continuous synthetic data. We also conducted experiments with the MNIST handwritten digits dataset, where we trained a mechanism that trades off between minimizing the pixel-level image distortion and concealing the digit.

References

[1] D. Barber and F. Agakov. The IM algorithm: A variational approach to information maximization. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 201–208, Cambridge, MA, USA, 2003. MIT Press.

[2] Y. O. Basciftci, Y. Wang, and P. Ishwar. On privacy-utility tradeoffs for constrained data release mechanisms. In *Information Theory and Applications Workshop*, Feb. 2016.

[3] F. P. Calmon and N. Fawaz. Privacy against statistical inference. In *Allerton Conf. on Comm., Ctrl., and Comp.*, pages 1401–1408, 2012.

[4] X. Chen, X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2172–2180. Curran Associates, Inc., 2016.

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

[6] H. Edwards and A. J. Storkey. Censoring representations with an adversary. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

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

[8] J. Hamm. Enhancing utility and privacy with noisy minimax filters. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6389–6393, March 2017.

[9] J. Hamm. Minimax filter: Learning to preserve privacy from inference attacks. *The Journal of Machine Learning Research*, 18(1):4704–4734, 2017.

[10] A. Hindupur. The GAN zoo. https://deephunt.in/the-gan-zoo-79597dc8c347, 2017.

[11] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal. Context-aware generative adversarial privacy. *Entropy*, 19(12), 2017.

[12] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[13] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204. ACM, 2011.

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[15] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[16] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE Intl. Conf. on Data Eng.*, pages 106–115. IEEE, 2007.

[17] C. Liu, S. Chakraborty, and P. Mittal. Dependence makes you vulnberable: Differential privacy under dependent tuples. In *Network and Distributed System Security Symposium*, pages 21–24, 2016.

[18] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), Mar. 2007.

[19] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[20] A. Makhdoumi and N. Fawaz. Privacy-utility tradeoff under statistical uncertainty. In *Allerton Conf. on Comm., Ctrl., and Comp.*, pages 1627–1634, 2013.

[21] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard. From the information bottleneck to the privacy funnel. In *IEEE Information Theory Workshop*, pages 501–505, 2014.

[22] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[23] V. Mirjalili, S. Raschka, A. Namboodiri, and A. Ross. Semi-adversarial networks: Convolutional autoencoders for imparting privacy to face images. In *2018 International Conference on Biometrics (ICB)*, pages 82–89, 2018.

[24] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symp. on Security and Privacy*, pages 111–125. IEEE, 2008.

[25] M. Nasr, R. Shokri, and A. Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 634–646, 2018.

[26] B. Poczos and J. Schneider. Nonparametric estimation of conditional information and divergences. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 914–923, 21–23 Apr 2012.

[27] D. Rebollo-Monedero, J. Forné, and J. Domingo-Ferrer. From t-closeness-like privacy to postrandomization via information theory. *IEEE Trans. Knowl. Data Eng.*, 22(11):1623–1636, 2010.

[28] L. Sankar, S. R. Rajagopalan, and H. V. Poor. Utility-privacy tradeoffs in databases: An information-theoretic approach. *IEEE Trans. on Information Forensics and Security*, 8(6):838–852, 2013.

[29] L. Sweeney. Simple demographics often identify people uniquely. *Carnegie Mellon University, Data Privacy Working Paper*, 2000.

[30] L. Sweeney. k-anonymity: A model for protecting privacy. *Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[31] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2 edition, 2012.

[32] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Allerton Conf. on Comm., Ctrl., and Comp.*, pages 368–377, 1999.

[33] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[34] A. Tripathy, Y. Wang, and P. Ishwar. Privacy-preserving adversarial networks. *arXiv preprint arXiv:1712.07008*, 2017.

[35] Y. Wang, Y. O. Basciftci, and P. Ishwar. Privacy-utility tradeoffs under constrained data release mechanisms. *arXiv preprint arXiv:1710.09295*, 2017.

[36] H. Yamamoto. A source coding problem for sources with additional outputs to keep secret from the receiver or wiretappers. *IEEE Trans. on Information Theory*, 29(6):918–923, 1983.