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

Privacy-Preserving Adversarial Networks

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

TR2017-194 December 2017

Abstract

We propose a data-driven framework for optimizing privacy-preserving data release mechanisms toward the information-theoretically optimal tradeoff between minimizing distortion of useful data and concealing sensitive information. Our approach employs adversarially-trained neural networks to implement randomized mechanisms and to perform a variational approximation of mutual information privacy. We empirically validate our PrivacyPreserving Adversarial Networks (PPAN) framework with experiments conducted on discrete and continuous synthetic data, as well as the MNIST handwritten digits dataset. With the synthetic data, we find that our model-agnostic PPAN approach achieves tradeoff points very close to the optimal tradeoffs that are analyticallyderived 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.

arXiv

Privacy-Preserving Adversarial Networks

Ardhendu Tripathy* Ye Wang† Prakash Ishwar‡

December 18, 2017

Abstract

We propose a data-driven framework for optimizing privacy-preserving data release mechanisms toward the information-theoretically optimal tradeoff between minimizing distortion of useful data and concealing sensitive information. Our approach employs adversarially-trained neural networks to implement randomized mechanisms and to perform a variational approximation of mutual information privacy. We empirically validate our Privacy-Preserving Adversarial Networks (PPAN) framework with experiments conducted on discrete and continuous synthetic data, as well as the MNIST handwritten digits dataset. With the synthetic data, we find that 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.

1 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 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 [1, 2]. A wide variety of methods to statistically quantify and address privacy have been proposed, such as k-anonymity [3], L-diversity [4], t-closeness [5], and differential privacy [6]. 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 [7, 8, 9, 10, 11]. Unlike the methods mentioned earlier, measuring privacy via mutual information implicitly requires consideration of the underlying statistical distribution of the data. While lack of model knowledge may be a challenging issue to address in practice, entirely ignoring the data distribution 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 [12, 9, 13, 14].

We build upon the non-asymptotic, information-theoretic framework introduced by [8, 9], where the sensitive and useful data are respectively modeled as random variables X and Y. We also adopt the extension considered in [11], 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 [8, 9, 14]), as well as related asymptotic privacy frameworks where communication efficiency is also considered in a rate-distortion-privacy tradeoff [7, 10].

*A. Tripathy is with Iowa State University, Ames, IA 50011, email: ardhendu@iastate.edu, and performed 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.

In principle, when the data model distribution is known, the design of the optimal privacy-preserving mechanism can be tackled as a convex optimization problem [8, 9]. 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 toward the theoretically optimal privacy-utility tradeoffs, by learning from a set of training data rather than requiring model knowledge. We call this approach *Privacy-Preserving Adversarial Networks* (PPAN) since the mechanism is realized as a randomized neural network, which 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 [15]. While the adversary is trained to minimize the log-loss with respect to this posterior estimate, the mechanism network is trained toward the dual objectives of minimizing distortion and concealing sensitive information (by maximizing the adversarial loss).

1.1 Related Work

The general concept of adversarial training of neural networks was introduced by [16], which proposed *Generative Adversarial Networks* (GAN) for learning generative models that can synthesize new data samples. Since, their introduction, GANs have inspired an enormous number of adversarially trained neural network architectures for a wide variety of purposes [17].

The earlier works of [18, 19] 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 [18], the mechanism is realized as an autoencoder[2], and the adversary attempts to predict a binary sensitive variable from the latent representation. In the framework of [19], 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 toward 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[3] mechanisms, which generally do not achieve the optimal privacy-utility tradeoffs, although neither attempts to address information-theoretic privacy.

The recent, independent work of [20] 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 [9] 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.

1.2 Contributions and Paper Outline

Our main contributions are summarized as follows:

- Our framework, presented in Section 2, provides a data-driven approach for optimizing privacy-preserving data release mechanisms that approaches the information-theoretically optimal privacy-utility tradeoffs. The key to our approach is employing adversarial training to perform a variational approximation of mutual information privacy.
- We consider 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 framework, all of the variables involved can be discrete, continuous, and/or high-dimensional vectors. We describe specific network architectures and sampling methods appropriate for various scenarios in

[2] An autoencoder architecture itself is comprised of two networks, an encoder and a decoder. The input is first processed by the encoder to produce a latent representation (or code), which is then processed by the decoder to produce the final output.

[3] While [19] does also consider a "noisy" version of their mechanism, the randomization is limited to only independent, additive noise before or after deterministic filtering.

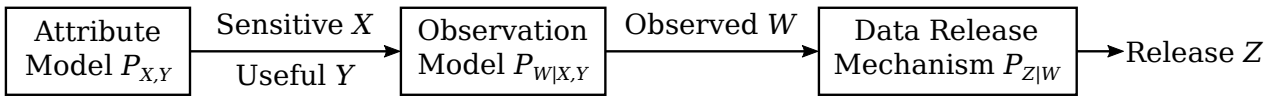


Figure 1: The observed data W is a (potentially noisy/partial) observation of the sensitive and useful data attributes (X, Y). Our goal is to optimize the data release mechanism $P_{Z|W}$ used to obtain the released data Z.

Section 2.3. When all of the variables have finite alphabets, we note 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 3 with experiments on discrete and continuous (multivariate Gaussian) synthetic data, and the MNIST handwritten digit dataset. For the synthetic data experiments, we demonstrate that PPAN closely approaches the theoretically optimal privacy-utility tradeoffs.
- For multivariate Gaussian data, with partial and full observations, we analytically derive the theoretically-optimal privacy-utility tradeoffs in Section 4, providing the theoretical baseline for our experiments with continuous synthetic data.

2 Problem Formulation and PPAN Methods

2.1 Privacy-Utility Tradeoff Optimization

We consider the privacy-utility tradeoff optimization problem described in [11], which extends the frameworks initiated by [8, 9]. 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 goal is to design 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[4] $\mathbb{E}[d(Y, Z)]$ 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,

$$\min_{P_{Z|W}} I(X; Z), \quad \text{such that} \quad \mathbb{E}[d(Y, Z)] \leq \delta \quad \text{and} \quad (X, Y) \leftrightarrow W \leftrightarrow Z, \tag{1}$$

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

As noted in [11], 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 [11] 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.

2.2 Adversarial Training for an Unknown Data Model

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 is available: $\{(w_i, x_i, y_i)\}_{i=1}^n \sim$ i.i.d. $P_{W,X,Y}$. A key to our approach is approximating $I(X; Z)$ via a variational lower bound given by [15] and also used in [21]. This bound is based on

[4] We mainly focus on expected distortion in this work, although the formulation in [11] actually allows for a more general class of distortion measures. We outline an extension of our approach for distortion measured by conditional entropy $h(Y|Z)$ in Section 5.1.

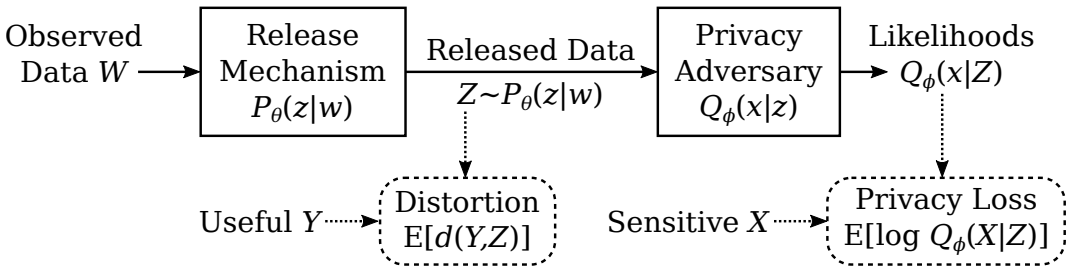


Figure 2: The release mechanism $P_\theta(z|w)$ is adversarially trained with a privacy adversary $Q_\phi(x|z)$, which estimates the posterior likelihoods of the sensitive attributes X after observing the released data Z. The mechanism is trained to minimize both the distortion and privacy loss terms, while the adversary is trained to maximize the privacy loss.

the following identity, for any conditional distribution $Q_{X|Z}$ over $\mathcal{X}$ given values in $\mathcal{Z}$,

$$\begin{aligned} I(X;Z) &= h(X) - h(X|Z) \\ &= h(X) + \int_z P_Z(z) \int_x P_{X|Z}(x|z) \log \left(P_{X|Z}(x|z) \frac{Q_{X|Z}(x|z)}{Q_{X|Z}(x|z)} \right) dx\, dz \\ &= h(X) + \mathrm{KL}(P_{X|Z} \| Q_{X|Z}) + \mathbb{E}\big[\log Q_{X|Z}(X|Z)\big], \end{aligned}$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence. Therefore, since KL divergence is nonnegative,

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

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,

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

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 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 example, 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 the adversary network that is controlled by the parameters ϕ. For each training 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

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

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

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

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

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}^i(\theta, \phi) &= \frac{1}{kn} \sum_{i=1}^{n} \sum_{j=1}^{k} [\log Q_\phi(x_i|z_{i,j}) + \lambda d(y_i, z_{i,j})] \\ &\xrightarrow{\text{a.s.}} \mathbb{E}\big[\log Q_\phi(X|Z) + \lambda d(Y, Z)\big], \end{aligned}$$

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 [16], 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.

2.3 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. In this section, we outline some of these forms and their associated sampling methods.

2.3.1 Finite Alphabets

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

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

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 [22, 23]).

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 minimal networks with no hidden layers and softmax applied to the output (to yield stochastic vectors).

2.3.2 Gaussian Approximations for Real Variables

A multivariate Gaussian release mechanism can be sampled by employing the reparameterization trick of [24], 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 $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^T$.

Extending this technique, a Gaussian Mixture Model (GMM) release mechanism can be realized with a neural network f_θ that produces the set of parameters $\{(\boldsymbol{\mu}_{l,i}, \mathbf{A}_{l,i}, \pi_{l,i})\}_{l=1}^m = f_\theta(w_i)$, where $\pi_{l,i}$ are the mixture weights. We then sample $z_{l,i} = \mathbf{A}_{l,i}\mathbf{u}_{l,i} + \boldsymbol{\mu}_{l,i}$ for each component distribution of the GMM, and compute the loss terms via

$$\mathcal{L}^i_{\text{GMM}}(\theta, \phi) := \sum_{l=1}^{m} \pi_{l,i}\big(\log Q_\phi(x_i|z_{l,i}) + \lambda d(y_i, z_{l,i})\big),$$

which combines the Gaussian sampling reparameterization trick with a direct expectation over the mixture component selection.

Case	Attribute Model	Observation Model	Distortion Metric
Discrete, Sec. 3.1	(X, Y) symmetric pair for $m = 10, p = 0.4$, see (7)	$W = Y$ and $W = (X, Y)$	$\Pr[Y \neq Z]$
Continuous, Sec. 3.2.2	$X = Y \sim \mathcal{N}\left(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}^2)\right)$, $\boldsymbol{\sigma}^2 = [0.47, 0.24, 0.85, 0.07, 0.66]$	$W = X = Y$	$\mathbb{E}[\|Y - Z\|^2]$
Continuous, Sec. 3.2.3	$\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right] \sim \mathcal{N}\left(\mathbf{0}, \left[\begin{smallmatrix} 1 & 0.85 \\ 0.85 & 1 \end{smallmatrix}\right]\right)$	$W = Y$ and $W = (X, Y)$	$\mathbb{E}[(Y - Z)^2]$
Continuous, Sec. 3.2.4	$\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right] \sim \mathcal{N}\left(\mathbf{0}, \left[\begin{smallmatrix} I_5 & \mathrm{diag}(\boldsymbol{\rho}) \\ \mathrm{diag}(\boldsymbol{\rho}) & I_5 \end{smallmatrix}\right]\right)$, $\boldsymbol{\rho} = [0.47, 0.24, 0.85, 0.07, 0.66]$	$W = Y$	$\mathbb{E}[\|Y - Z\|^2]$

Table 1: The models used for obtaining synthetic training and test datasets in our experiments.

2.3.3 Universal Approximators

Another approach, as seen in [25], 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 be learned, this approach could potentially approximate the universal class of distributions. 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 behavior of the network.

3 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 3.1, we also compare PPAN against the approach of [26], where an approximate discrete distribution is estimated from the training data and used in lieu of the true distribution for the optimization given by (1). For the continuous synthetic data experiments, we consider Gaussian joint distributions over the sensitive, useful, and observed data, for which we can compare the results obtained by PPAN versus the theoretically optimal tradeoffs that we derive in Section 4. We use the MNIST handwritten digits dataset for an example of applying PPAN to real data in Section 3.3, where we demonstrate optimized networks that tradeoff between concealing the digit and reducing image distortion. Table 1 summarizes the data models and distortion metrics that we use in our experiments. Our experiments were implemented using the Chainer deep learning framework [27], with optimization performed by their implementation of Adam [28].

3.1 Discrete Synthetic Data

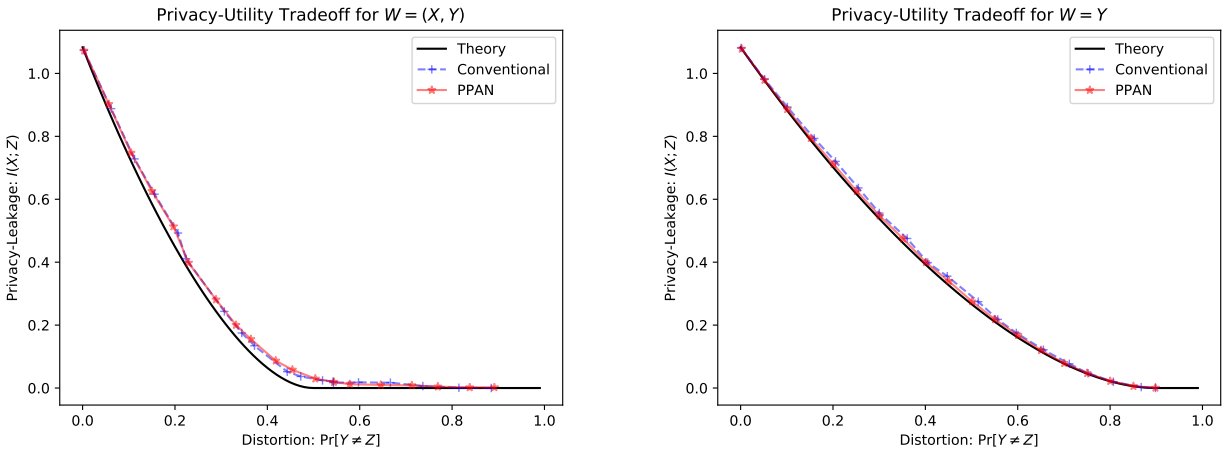
In our experiments with discrete data, we used a toy distribution for which the theoretically optimal privacy-utility tradeoffs have been analytically determined in [14]. Specifically, we consider sensitive and useful attributes that are distributed over the finite alphabets $\mathcal{X} = \mathcal{Y} = \{0, \ldots, m-1\}$, with $m \geq 2$, according to the *symmetric pair* distribution given by

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

with the parameter $p \in [0, 1]$. The mutual information of the symmetric pair distribution is given by [14] as

$$I(X;Y) = \log m - p \log(m-1) - h_2(p) =: r_m(p),$$

where $h_2(p) := -p \log p - (1-p)\log(1-p)$ is the binary entropy function, and for convenience in later discussion, we define $r_m(p)$ as a function of the distribution parameters m and p.



(a) Full data observed $W = (X, Y)$

(b) Only useful attribute observed $W = Y$

Figure 3: Comparison of PPAN performance versus the conventional model estimation approach of [26] and the theoretical optimal given in (8) and (9), for two observation scenarios: (a) full data observed, (b) only useful attributed observed. The attribute model $P_{X,Y}$ is the symmetric pair distribution in (7) with $m = 10$ and $p = 0.4$.

3.1.1 Theoretically Optimal Privacy-Utility Tradeoffs

The theoretically optimal privacy-utility tradeoffs, as defined by (1), are analytically derived in [14] for three specific data observation models, while using probability of error as the distortion metric, i.e., $\mathbb{E}[\mathbf{1}(Y \neq Z)] = \Pr[Y \neq Z]$. In one case, when the observation is the full data, i.e., $W = (X, Y)$, the optimal mutual information privacy-leakage as a function of the distortion (probability of error) limit $\delta \in [0, 1]$ is given by

$$I^*_{W=(X,Y)}(\delta) = \begin{cases} r_m(p+\delta), & \text{if } \delta \leq 1 - \frac{1}{m} - p, \\ r_m(p-\delta), & \text{if } \delta \leq p - (1 - \frac{1}{m}), \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

In another case, when the observation is only the useful attribute, i.e., $W = Y$, the optimal privacy-leakage as a function $\delta \in [0, 1]$ is given by

$$I^*_{W=Y}(\delta) = \begin{cases} r_m\left(p + \delta\left(1 - \frac{pm}{m-1}\right)\right), & \text{if } \delta < 1 - \frac{1}{m}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

We will use these two observation scenarios, full data and useful data only, in our experiments.

3.1.2 Network Architecture and Evaluation

As mentioned in Section 2.3.1, minimal network architectures can be used for the release mechanism and adversary 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

$$P_\theta(\cdot|w) = \text{softmax}(\mathbf{G}\mathbf{w}),$$

where the network parameters $\theta = \mathbf{G}$ are 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

$$Q_\phi(\cdot|z) = \text{softmax}(\mathbf{A}\mathbf{z}),$$

where $\mathbf{z}$ is the one-hot encoding of z, and the network parameters $\phi = \mathbf{A}$ are 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

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

where we use $\lambda = 500$ in these experiments.

In Figure 3, we compare the results of PPAN against the theoretical baselines of (8) and (9), as well as against a conventional approach suggested by [26], 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 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 actual performance of the optimized mechanisms with respect to the true data model, i.e., from the joint distribution combining the optimized $P_{Z|W}$ with the true $P_{X,Y,W}$.

3.2 Gaussian Synthetic Data

The experiments described previously considered the setting in which the attributes belonged to a finite discrete alphabet. In this section, we consider scalar and multivariate jointly Gaussian sensitive and useful attributes. We evaluate the performance of PPAN on synthetic data generated for this model in various scenarios. The utility metric used here is the mean squared error between the release and the useful attribute.

As we note in Section 4 the optimum release for the scenarios considered here is a random variable which is jointly Gaussian with the attributes. Thus we could potentially use a mechanism network architecture that can realize the procedure described in Section 2.3.2 to generate the release. However, since the form of the optimal release distribution will not be known in practice, we use the universal approximator technique described in Section 2.3.3. Thus we choose an architecture for the privacy mechanism which can generate real-valued release samples. 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 samples of the release random variable Z at its output. We can represent this process as the evaluation map of the function $f_\theta(w, u)$, where θ denotes the parameters of the mechanism network. Each component of the seed noise vector is an i.i.d. sample from Uniform$[-1, 1]$.

The attacker network, with parameters denoted by ϕ, models the posterior probability $Q_\phi(X \mid Z)$ of the sensitive attribute given the release. We assume that $Q_\phi(\cdot \mid z)$ is a normal distribution with mean $\boldsymbol{\mu}_\phi(z)$ and covariance matrix $\mathrm{diag}(\boldsymbol{\sigma}^2_\phi(z))$, i.e., they are functions of the release z. For the attacker network, we use three fully connected layers to learn the mean and variance. The network takes as input the release z and outputs the pair of evaluation maps $(\boldsymbol{\mu}_\phi(z), \log \boldsymbol{\sigma}^2_\phi(z))$, where the log is applied componentwise on the variance vector. The ReLU activation function is applied at the outputs of the two hidden layers, and no activation function is used at the output layer. We use the PPAN mechanism 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., the loss terms are set to be

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

The parameter δ is swept through a linearly spaced range of values. For each value of δ, we train the adversarial networks and evaluate the performance to obtain an operating point in the privacy-leakage versus distortion plane. The data model is sampled independently to obtain a dataset realization that is used to train and evaluate the PPAN mechanism for each different value of δ. In all the scenarios described below, we used 8000 training instances sampled from the given model. 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. The PPAN networks were trained using stochastic gradient descent with minibatch size 200 for 250 epochs. In each iteration we do 5 gradient descent steps to update the parameters of the attacker network before updating the mechanism network. We evaluate the performance of PPAN mechanism on an independently generated test set of 4000 samples. We generated the corresponding releases for the test set as $z_{\text{test}} = f_{\theta^\star}(w_{\text{test}}, u)$, where u are seed noise realizations, and $\theta^\star$ denote the learned parameters for the mechanism network. The attribute model, observation scenario, testing procedure and the values of the other hyperparameters used in our experiments are described in the subsections below.

3.2.1 Estimating Mutual Information Leakage for the Test Set

The operating point of a trained PPAN mechanism is specified by the values of mutual information and distortion between the test set and its corresponding release. We can evaluate the empirical distortion using w_{test} and z_{test}. However, evaluating $I(X_{\text{test}}; Z_{\text{test}})$ requires us to know the joint distribution $P(X_{\text{test}}, Z_{\text{test}})$ in general, and here we have access to only the realizations x_{test} and z_{test}. In Section 4 we show that for the experiments considered

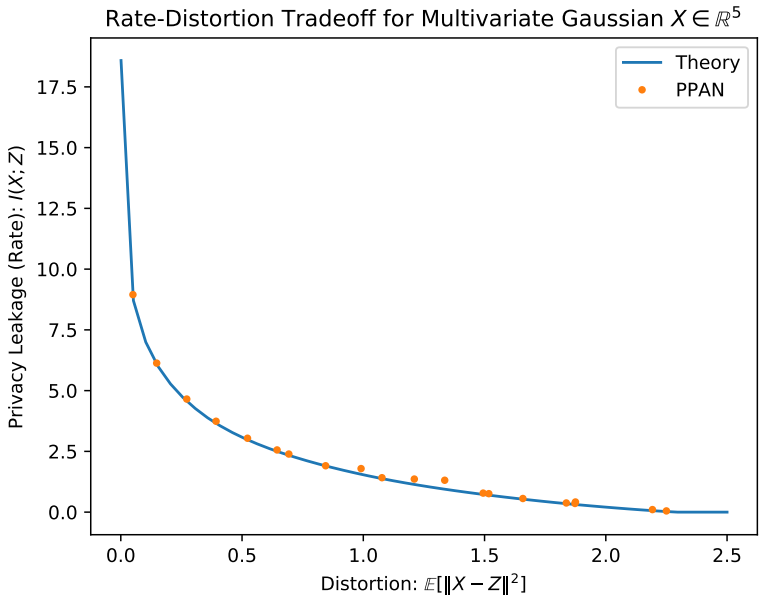


Figure 4: The figure compares the optimal rate distortion curve with the results obtained by PPAN on the test set. The adversarial networks optimize (5) for 20 linearly spaced values of the target distortion in $[0, 2.5]$. An independent dataset realization sampled from the underlying model is used for training and evaluating the PPAN mechanism at different values of the target distortion. Each dataset realization has 8000 training instances and 4000 test instances.

here, the optimal Z_{test} is jointly Gaussian with X_{test}. Motivated by this, we estimate $I(X_{\text{test}}; Z_{\text{test}})$ in the following manner. We find the empirical covariance matrix of x_{test} and z_{test}, denoted as

$$\hat{\Sigma}(x_{\text{test}}, z_{\text{test}}) = \begin{bmatrix} \hat{\Sigma}_{x_{\text{test}}} & \hat{\Sigma}_{x_{\text{test}}, z_{\text{test}}} \\ \hat{\Sigma}^T_{x_{\text{test}}, z_{\text{test}}} & \hat{\Sigma}_{z_{\text{test}}} \end{bmatrix}.$$

In all our experiments, X_{test} and Z_{test} have the same number of dimensions. Consider jointly Gaussian random variables X_g and Z_g such that $\text{Cov}(X_g, Z_g) = \hat{\Sigma}(x_{\text{test}}, z_{\text{test}})$. Then we have

$$I(X_g; Z_g) = \frac{1}{2} \log \left(\frac{\det \hat{\Sigma}_{x_{\text{test}}}}{\det \hat{\Sigma}_{x_{\text{test}}|z_{\text{test}}}} \right), \text{ where } \hat{\Sigma}_{x_{\text{test}}|z_{\text{test}}} := \hat{\Sigma}_{x_{\text{test}}} - \hat{\Sigma}_{x_{\text{test}}, z_{\text{test}}} \hat{\Sigma}^{-1}_{z_{\text{test}}} \hat{\Sigma}^T_{x_{\text{test}}, z_{\text{test}}}.$$

We use $I(X_g; Z_g)$ as an estimate $\hat{I}(X_{\text{test}}; Z_{\text{test}})$ of the mutual information leakage in the sequel for Gaussian synthetic data. We note that this underestimates the true mutual information leakage since

$$\begin{aligned} I(X_{\text{test}}; Z_{\text{test}}) &= h(X_{\text{test}}) - h(X_{\text{test}}|Z_{\text{test}}) = h(X_{\text{test}}) - h(X_{\text{test}} - \widehat{\mathbb{E}}[X_{\text{test}}|Z_{\text{test}}]|Z_{\text{test}}) \\ &\geq h(X_{\text{test}}) - h(X_{\text{test}} - \widehat{\mathbb{E}}[X_{\text{test}}|Z_{\text{test}}]) = I(X_g; Z_g) = \hat{I}(X_{\text{test}}, Z_{\text{test}}), \end{aligned}$$

where $\widehat{\mathbb{E}}[X_{\text{test}}|Z_{\text{test}}]$ is the linear MMSE estimate of X_{test} as a function of Z_{test}. We use this estimate only for its simplicity. One could certainly use other non-parametric estimates of mutual information.

3.2.2 Rate Distortion

We first apply the PPAN framework to the problem of finding the minimum required code rate in order to describe a multivariate Gaussian source $X \in \mathbb{R}^5$ within a given value of mean squared error. This is a standard problem in information theory, for example, see [29, Chap. 10]. This problem 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 mean squared error less than a distortion level while retaining as much expected uncertainty about X as possible.

We choose the attribute model $X \sim \mathcal{N}(\mathbf{0}, \text{diag}(0.47, 0.24, 0.85, 0.07, 0.66))$. For the multiplier of the distortion term in the penalty formulation, we use the value $\lambda = 500$. We run the experiment for 20 different values of the

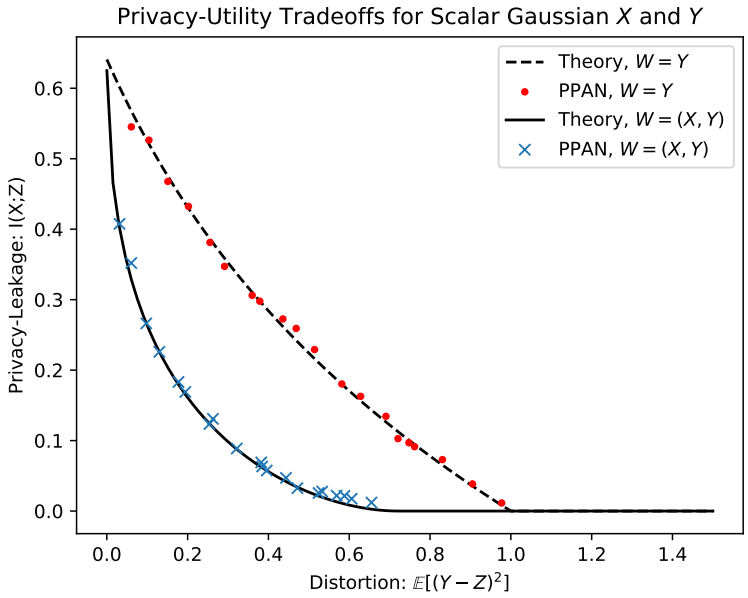


Figure 5: The two observation models of useful data only and full data for jointly Gaussian attributes are compared here. The operating points of PPAN on the test set are shown along with their respective optimal tradeoff curves. The PPAN mechanism optimizes (5) for 20 linearly spaced values of target distortion: $\delta^{\mathrm{UD}} \in [0, 1]$ and $\delta^{\mathrm{FD}} \in [0, 0.8]$. An independent dataset realization sampled from the underlying model is used for training and evaluating the PPAN mechanism at different values of the target distortion. Each dataset realization has 8000 training instances and 4000 test instances.

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 testing procedure is as follows. We evaluate the mechanism network $\boldsymbol{z}_{\mathrm{test}} = f_{\theta^\star}(\boldsymbol{w}_{\mathrm{test}}, u)$ for all $\boldsymbol{w}_{\mathrm{test}}$ in the test set. Here, $\theta^\star$ are the learned parameters and u consists of independent seed noise samples. Since $W = X$, the utility loss is quantified by the empirical average of the MSE $\|\boldsymbol{w}_{\mathrm{test}} - \boldsymbol{z}_{\mathrm{test}}\|^2$ over all test samples. The privacy loss is quantified by the estimate $\hat{I}(X_{\mathrm{test}}; Z_{\mathrm{test}})$ as described in Section 3.2.1.

The optimal privacy-utility tradeoff (or, rate-distortion) curve is given as $\sum_{j=1}^{5} \max\{0, 0.5 \log(\sigma_j^2/D_j)\}$ [29], where $\boldsymbol{\sigma}$ are the true parameters of the attribute distribution and D_j is the allowed squared error distortion in the jth component. 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.

3.2.3 Scalar Attribute: Useful Data Only and Full Data

Here we 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 consider two different observation models here: $W = Y$, called useful data only (UD) and $W = (X, Y)$, called full data (FD). For the useful data only observation model, the input to the adversarial network is the useful attribute Y and seed noise U, while for the full data observation model, the input is the pair of attributes (X, Y) and seed noise U. In both cases, U is a scalar random variable following Uniform$[-1, 1]$. The values of the multipliers chosen are: $\lambda^{\mathrm{UD}} = 10$ and $\lambda^{\mathrm{FD}} = 50$. In each case, we run experiments for 20 different values of the target distortion with $\delta^{\mathrm{UD}} \in [0, 1]$ and $\delta^{\mathrm{FD}} \in [0, 0.8]$. The output of the mechanism in the testing phase can be denoted as

$$z_{\mathrm{test}}^{\mathrm{UD}} = f_{\theta^\star_{\mathrm{UD}}}(y_{\mathrm{test}}, u) \quad \text{and} \quad z_{\mathrm{test}}^{\mathrm{FD}} = f_{\theta^\star_{\mathrm{FD}}}(x_{\mathrm{test}}, y_{\mathrm{test}}, u),$$

where $\theta^\star_{\mathrm{UD}}, \theta^\star_{\mathrm{FD}}$ are the learned parameters in the two cases and u are independent samples of the seed noise. The utility loss is given in both cases by the empirical average of the MSE $\|\boldsymbol{w}_{\mathrm{test}} - \boldsymbol{z}_{\mathrm{test}}\|^2$ over all test samples.

The privacy loss is computed following the procedure described in Section 3.2.1. The (privacy-leakage, distortion) pairs returned by PPAN are plotted along with the optimal tradeoff curves (from Propositions 1 and 3) 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.

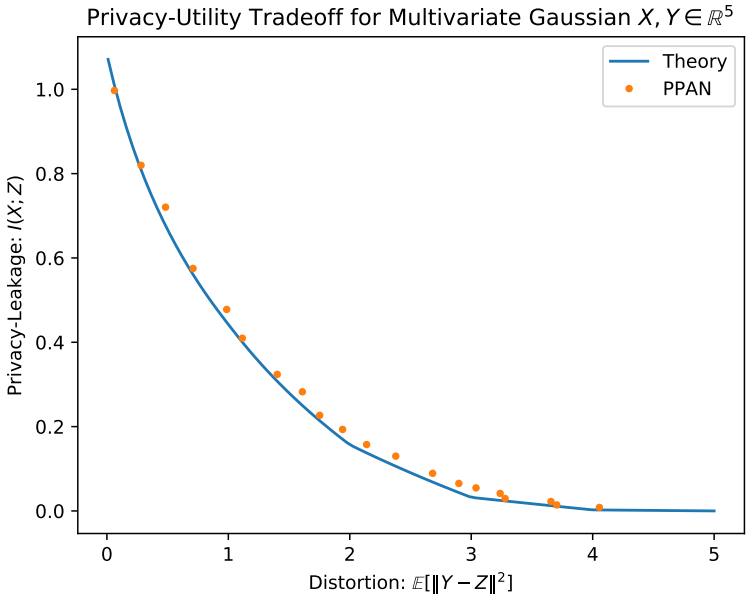


Figure 6: This figure compares the operating points achieved by the PPAN mechanism with the corresponding theoretical optimum tradeoff curve. The PPAN mechanism optimizes (5) for 20 linearly spaced values of the target distortion in $[0, 4.5]$. An independent dataset realization sampled from the underlying model is used for training and evaluating the PPAN mechanism at different values of the target distortion. Each dataset realization has 8000 training instances and 4000 test instances.

3.2.4 Vector Attribute: Useful Data Only

Here we consider multivariate jointly Gaussian sensitive and useful attributes $\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right] \sim \mathcal{N}\left(\mathbf{0}, \left[\begin{smallmatrix} I_5 & \mathrm{diag}(\boldsymbol{\rho}) \\ \mathrm{diag}(\boldsymbol{\rho}) & I_5 \end{smallmatrix}\right]\right)$ where both $X, Y \in \mathbb{R}^5$ and $\boldsymbol{\rho} = [0.47, 0.24, 0.85, 0.07, 0.66]$. We choose the multiplier $\lambda = 10$ in this case. The value of the target distortion in the penalty formulation is linearly varied in the range $[0, 4.5]$. For each value of δ, we sample an independent dataset realization which is used to train and test the adversarial networks. The seed noise is a vector random variable of length 8, each component of it being i.i.d. Uniform$[-1, 1]$. As the observation model is useful data only, we have that $\boldsymbol{z}_{\text{test}} = f_{\theta^\star}(\boldsymbol{y}_{\text{test}}, u)$. The utility loss is measured by mean squared error between $\boldsymbol{y}_{\text{test}}$ and $\boldsymbol{z}_{\text{test}}$ and the privacy-leakage $\hat{I}(\boldsymbol{X}_{\text{test}}; \boldsymbol{Z}_{\text{test}})$ is measured using the procedure in Section 3.2.1. We plot the (privacy-leakage, distortion) pairs returned by the PPAN mechanism along with the optimal tradeoff curve (from Proposition 2) in Figure 6. We see that the operating points of the PPAN mechanism are very close to the theoretically optimum tradeoff curve over a wide range of target distortion values.

3.3 MNIST Handwritten Digits

The MNIST dataset consists of 70 thousand 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 normalized vectors in $[0, 1]^{784}$.

In this experiment, we consider the image to be both the useful and observed data, i.e., $W = Y$, the digit label to be the sensitive attribute X, and the mechanism releases an image $Z \in [0, 1]^{784}$. We measure the distortion between the original and released images $Y, Z \in [0, 1]^{784}$ with

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

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 image distortion.

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

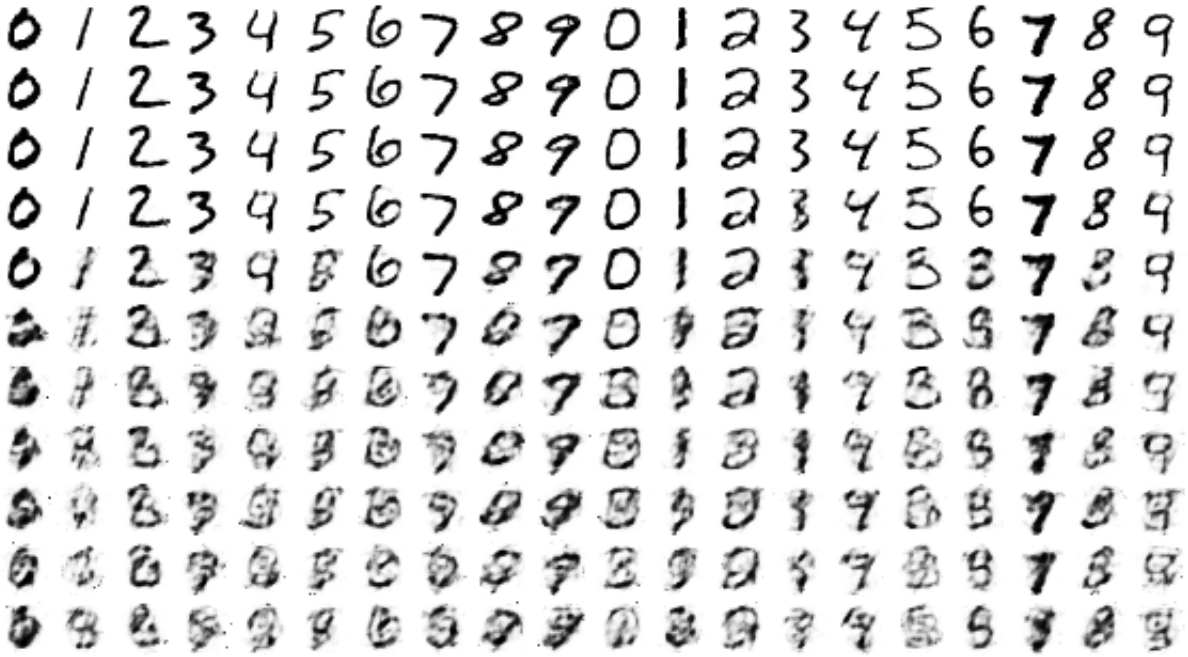


Figure 7: Example results from applying PPAN to conceal MNIST handwritten digits, without using an additional discriminator adversary (i.e., $\gamma = 0$). Top row consists of the original test set examples input to the mechanism, while the second through last rows are the corresponding outputs from mechanisms trained with $\lambda = \{35, 30, 25, 20, 17, 16, 15, 13, 10, 8\}$.

$784 + 20$ 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 2.3.3. 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 $\{0, \ldots, 9\}$.

For some experiments, we also employ the standard GAN approach by also adding a discriminator network to further encourage the mechanism toward producing 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

$$\mathcal{L}^i(\theta, \phi, \psi) := \log Q_\phi(x_i|z_i) + \lambda d(y_i, z_i) + \gamma \log D_\psi(z_i) + \gamma \log(1 - D_\psi(y_i)), \tag{10}$$

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

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

Figures 7 and 8 show example results from applying trained privacy mechanisms to MNIST test set examples. The first, Figure 7, shows the results with the standard PPAN formulation, trained via (10) with $\gamma = 0$. The second, Figure 8, shows the results when the additional discriminator network is introduced, which is jointly trained via (10) with $\gamma = 2$. The first row of each figure 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 of the figures, the value of λ is decreased, reducing the emphasis on minimizing distortion. We see in both figures that the outputs start from accurate reconstructions and become progressively more distorted while the digit becomes more difficult to correctly recognize as λ decreases. In Figure 7, we see that mechanism

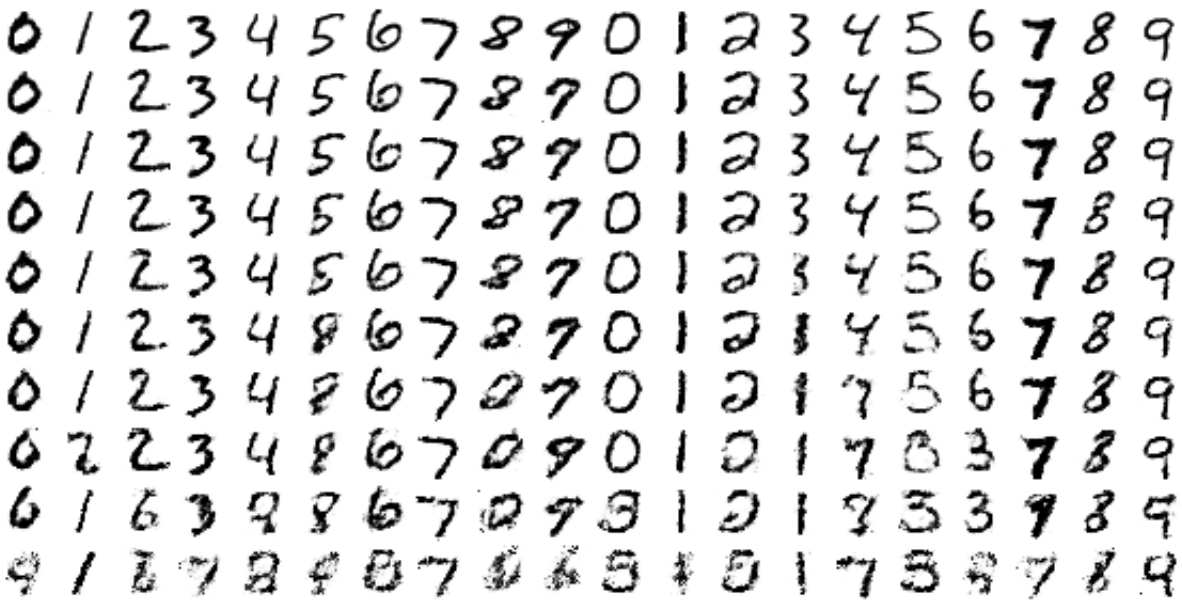
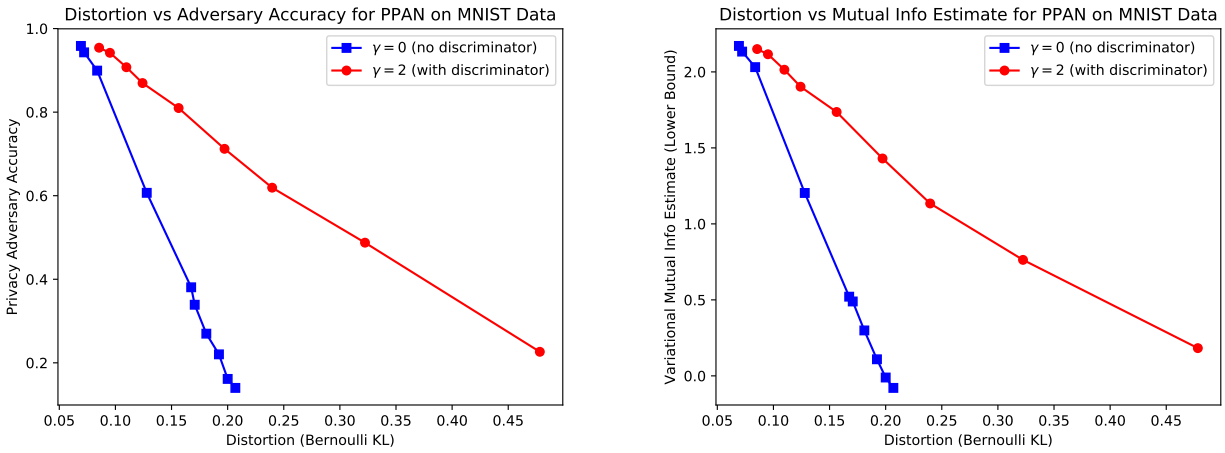


Figure 8: Example results from applying PPAN to conceal MNIST handwritten digits, while using a discriminator adversary with $\gamma = 2$. Top row consists of the original test set examples input to the mechanism, while the second through last rows are the corresponding outputs from mechanisms trained with $\lambda = \{15, 13, 11, 9, 7, 5, 4, 3, 2\}$.



(a) Distortion vs Adversary Accuracy

(b) Distortion vs Mutual Info Estimate

Figure 9: Objective evaluation of the distortion vs privacy tradeoffs for PPAN applied to the MNIST test set: (a) distortion versus the accuracy of the adversary in recognizing the original digit, (b) distortion versus the variational lower bound of mutual information calculated by the adversary.

seems to learn to minimize distortion while rendering the digit unrecognizable, which in some cases results in an output that resembles a different digit. In Figure 8, we see that the additonal discriminator network encourages outputs that more cleanly resemble actual digits, which required lower values for λ 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. We used the 10K test images to objectively evaluate the performance of the trained mechanisms for Figure 9, 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.

4 Optimum Privacy Utility Tradeoff for Gaussian Attributes

In Section 3 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*

$$\min_{P_{Z|Y}} I(X;Z), \quad \textit{such that} \quad \mathbb{E}(Y-Z)^2 \leq \delta \quad \textit{and} \quad X \leftrightarrow Y \leftrightarrow Z \tag{11}$$

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., see [8, eq. 8] and [10, example 2]. For completeness, we present the proof of this result in Appendix 6.1. 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 [8], 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 problem (11)

$$\min_{P_{Z|Y}} I(X;Z) \text{ such that } \mathbb{E}(Y-Z)^T(Y-Z) \leq \delta \text{ and } X \leftrightarrow Y \leftrightarrow Z. \tag{12}$$

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 and 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 (12) reduces to the following problem

$$\min_{P_{Z'|Y'}} I(X';Z') \text{ such that } \mathbb{E}(Y'-Z')^T(Y'-Z') \leq \delta/c \text{ and } X' \leftrightarrow Y' \leftrightarrow Z'. \tag{13}$$

Proposition 2. *If* $\left[\begin{smallmatrix} X' \\ Y' \end{smallmatrix}\right] \sim \mathcal{N}\Big(\left[\begin{smallmatrix} \mathbf{0}_m \\ \mathbf{0}_n \end{smallmatrix}\right], \left[\begin{smallmatrix} I_m & \Lambda_P \\ \Lambda_P^T & I_n \end{smallmatrix}\right]\Big)$*, then the minimizer of* (13) *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 Appendix 6.2. 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*

$$\min_{P_{Z|X,Y}} I(X; Z), \quad \textit{such that} \quad \mathbb{E}(Y - Z)^2 \leq \delta \tag{14}$$

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

$$I(X; Z) = \begin{cases} 0, & \text{if } \delta \geq \rho^2 \\ \dfrac{1}{2}\log\left(\dfrac{1}{1 - \left(\sqrt{\rho^2(1-\delta)} - \sqrt{(1-\rho^2)\delta}\right)^2}\right), & \text{if } \delta < \rho^2. \end{cases}$$

The proof of the above proposition is presented in Appendix 6.3. The theoretical tradeoff curve in Figure 5 was obtained using the above expression.

5 Conclusion

In this paper, we developed a data-driven framework for optimizing privacy-preserving data release mechanisms. 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 framework 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. While we focused on expected distortion to measure (dis)utility, our framework can be adapted to other general utility measures. For example, in the following subsection, we outline an adaptation to utility measured by the mutual information between the useful information and the released data.

5.1 Mutual Information Utility

The conditional entropy $h(Y|Z)$ is an alternative measure for distortion, which corresponds to the utility objective of maximizing the mutual information $I(Y; Z)$, since $h(Y)$ is fixed. When $h(Y|Z)$ is used as the distortion measure in a scenario where the observation $W = X$, the privacy-utility tradeoff optimization problem, as described in Section 2.1, becomes equivalent to the *Information Bottleneck* problem considered in [30]. In other scenarios where the observation $W = Y$, this problem becomes the *Privacy Funnel* problem introduced by [31]. The formulation

of (3) can be modified to address conditional entropy distortion by introducing another variational posterior $Q_{Y|Z}$ and using the following optimization, which applies a second variational approximation of mutual information,

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

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 along the optimal tradeoff curve. In a similar fashion to the approach in Section 2.2, this optimization problem can be practically addressed via the training of three neural networks, which respectively parameterize the mechanism $P_{Z|W}$ and the two variational posteriors $Q_{X|Z}$ and $Q_{Y|Z}$.

References

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

[2] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE Symp. on Security and Privacy*. IEEE, 2008, pp. 111–125.

[3] L. Sweeney, "k-anonymity: A model for protecting privacy," *Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[4] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Trans. on Knowledge Discovery from Data*, vol. 1, no. 1, p. 3, 2007.

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

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

[7] H. Yamamoto, "A source coding problem for sources with additional outputs to keep secret from the receiver or wiretappers," *IEEE Trans. on Information Theory*, vol. 29, no. 6, pp. 918–923, 1983.

[8] D. Rebollo-Monedero, J. Forné, 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, 2010.

[9] F. du Pin Calmon and N. Fawaz, "Privacy against statistical inference," in *Allerton Conf. on Comm., Ctrl., and Comp.*, 2012, pp. 1401–1408.

[10] 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*, vol. 8, no. 6, pp. 838–852, 2013.

[11] 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.

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

[13] C. Liu, S. Chakraborty, and P. Mittal, "Dependence makes you vulnberable: Differential privacy under dependent tuples," in *Network and Distributed System Security Symposium*, 2016.

[14] Y. Wang, Y. O. Basciftci, and P. Ishwar, "Privacy-utility tradeoffs under constrained data release mechanisms," *arXiv preprint arXiv:1710.09295*, 2017. [Online]. Available: https://arxiv.org/abs/1710.09295

[15] D. Barber and F. V. Agakov, "The im algorithm: A variational approach to information maximization," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schlkopf, Eds. Cambridge, MA: MIT Press, 2003, p. None. [Online]. Available: http://books.nips.cc/papers/files/nips16/NIPS2003_AA26.pdf

[16] 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*, 2014, pp. 2672–2680.

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

[18] H. Edwards and A. J. Storkey, "Censoring representations with an adversary," *CoRR*, vol. abs/1511.05897, 2015. [Online]. Available: http://arxiv.org/abs/1511.05897

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

[20] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *arXiv preprint arXiv:1710.09549*, 2017.

[21] 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 *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2172–2180. [Online]. Available: http://papers.nips.cc/paper/6399-infogan-interpretable-representation-learning-by-information-maximizing-generative-adversarial-nets.pdf

[22] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[23] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

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

[26] A. Makhdoumi and N. Fawaz, "Privacy-utility tradeoff under statistical uncertainty," in *Allerton Conf. on Comm., Ctrl., and Comp.*, 2013, pp. 1627–1634.

[27] 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. [Online]. Available: http://learningsys.org/papers/LearningSys_2015_paper_33.pdf

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[30] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Allerton Conf. on Comm., Ctrl., and Comp.*, 1999, pp. 368–377.

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

6 Appendix

6.1 Proof of Proposition 1

Proof. We can expand the mutual information term as follows,

$$\begin{aligned}
I(X;Z) &= h(X) - h(X|Z), \\
&= h(X) - h(X - \rho\sigma_X Z/\sigma_Y|Z), \\
&\geq h(X) - h(X - \rho\sigma_X Z/\sigma_Y), && (15)\\
&\geq 0.5\log 2\pi e\sigma_X^2 - h\left(\mathcal{N}(0, \mathbb{E}[(X - \rho\sigma_X Z/\sigma_Y)^2]\right), && (16)\\
&= \frac{1}{2}\log\left(\frac{\sigma_X^2}{\mathbb{E}[(X - \rho\sigma_X Z/\sigma_Y)^2]}\right). && (17)
\end{aligned}$$

Inequality (15) is true because conditioning can only reduce entropy and inequality (16) is true since the zero-mean normal distribution has the maximum entropy for a given value of the second moment. Let $T := X - \rho\sigma_X Y/\sigma_Y$, then T is jointly Gaussian and we have that

$$\mathrm{Cov}\left(\begin{bmatrix} T \\ Y \end{bmatrix}\right) = \begin{bmatrix} 1 & -\rho\sigma_X/\sigma_Y \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\rho\sigma_X/\sigma_Y & 1 \end{bmatrix} = \begin{bmatrix} \sigma_X^2(1-\rho^2) & 0 \\ 0 & \sigma_Y^2 \end{bmatrix}.$$

Hence, T is independent of Y. Since $X \leftrightarrow Y \leftrightarrow Z$ also forms a Markov chain, we have that T is conditionally independent of Z given Y. Due to the distortion constraint, we can upper bound $\mathbb{E}[(X - \rho\sigma_X Z/\sigma_Y)^2]$ in the following manner.

$$\begin{aligned}
\mathbb{E}\Big[\left(X - \frac{\rho\sigma_X}{\sigma_Y}Z\right)^2\Big] &= \sigma_X^2 + \frac{\rho^2\sigma_X^2}{\sigma_Y^2}\mathbb{E}[Z^2] - 2\frac{\rho\sigma_X}{\sigma_Y}\mathbb{E}\left[\left(T + \frac{\rho\sigma_X}{\sigma_Y}Y\right)Z\right], \\
&\le \sigma_X^2 + \frac{\rho^2\sigma_X^2}{\sigma_Y^2}(\delta - \sigma_Y^2 + 2\mathbb{E}[YZ]) - 2\frac{\rho\sigma_X}{\sigma_Y}\left(\frac{\rho\sigma_X}{\sigma_Y}\mathbb{E}[YZ] + \mathbb{E}[TZ]\right), \qquad (18) \\
&= \sigma_X^2(1-\rho^2) + \rho^2\delta\sigma_X^2/\sigma_Y^2. \qquad (19)
\end{aligned}$$

Inequality (18) is true because $\mathbb{E}[(Y-Z)^2] \le \delta$, and equation (19) is true because

$$\mathbb{E}[TZ] = \mathbb{E}_Y[\mathbb{E}_{T,Z|Y}[TZ]] \overset{\text{(i)}}{=} \mathbb{E}_Y\left[\mathbb{E}_{T|Y}[T]\mathbb{E}_{Z|Y}[Z]\right] \overset{\text{(ii)}}{=} \mathbb{E}_Y\left[\mathbb{E}_T[T]\mathbb{E}_{Z|Y}[Z]\right] \overset{\text{(iii)}}{=} 0,$$

where (i) is true because $T \perp\!\!\!\perp Z \mid Y$, (ii) is true because $T \perp\!\!\!\perp Y$ and (iii) is true because T has zero mean. Thus by equations (17) and (19), we get that

$$\min_{X \leftrightarrow Y \leftrightarrow Z,\ \mathbb{E}[(Y-Z)^2] \le \delta} I(X;Z) \ge \max\left\{0, \frac{1}{2}\log\left(\frac{1}{1-\rho^2+\rho^2\delta/\sigma_Y^2}\right)\right\}. \qquad (20)$$

For the choice of Z as stated in the proposition, we can check that X and Z are jointly Gaussian with $I(X;Z) = 0.5\log_2(\sigma_X^2/\mathrm{Var}(X|Z))$ and $\mathrm{Var}(X|Z) = \sigma_X^2(1-\rho^2+\rho^2\delta/\sigma_Y^2)$. Thus Z attains the lower bound for the privacy-leakage in (20) when $\delta < \sigma_Y^2$. Otherwise, the lower bound on mutual information is 0 and can be attained by $Z = 0$. □

6.2 Proof of Proposition 2

Proof. (a) If $m \le n$, then $(X_1', Y_1'), \ldots, (X_m', Y_m'), Y_{m+1}', \ldots, Y_n'$ are independent because they are jointly Gaussian and for all $i \ne j$, $\mathrm{Cov}(X_i', X_j') = \mathrm{Cov}(X_i', Y_j') = \mathrm{Cov}(Y_i', Y_j') = 0$. Similarly, if $m \ge n$, then $(X_1', Y_1'), \ldots, (X_n', Y_n')$, $X_{n+1}', \ldots, X_m'$ are independent.

In the following, we use the following well-known properties of mutual information and conditional mutual information. For any three random variables A, B, C, (b) $I(A;B) = I(B;A) \ge 0$, (c) $I(A;B) = 0 \Leftrightarrow A \perp\!\!\!\perp B$, (d) $I(A;C|B) \ge 0$, (e) $I(A;C|B) = 0 \Leftrightarrow (A \perp\!\!\!\perp C)|B$, (f) $I(A;B,C) = I(A;B) + I(A;C|B)$ so that $I(A;B,C) \ge I(A;B)$.

If $m \le n$, then $I(X';Z') = I(X_1', \ldots, X_m'; Z') \overset{\text{(f)}}{=} \sum_{i=1}^m I(X_i; Z'|X_1, \ldots, X_{i-1}) \overset{\text{(f,a,c)}}{=} \sum_{i=1}^m I(X_i; Z', X_1, \ldots, X_{i-1}) \overset{\text{(f)}}{\ge} \sum_{i=1}^m I(X_i; Z') \overset{\text{(f)}}{\ge} \sum_{i=1}^m I(X_i; Z_i)$. Similarly, if $m \ge n$, $I(X';Z') \ge \sum_{i=1}^m I(X_i;Z') \ge \sum_{i=1}^n I(X_i;Z_i)$. Thus in general,

$$I(X';Z') \ge \sum_{i=1}^{\min\{m,n\}} I(X_i;Z_i).$$

The distortion constraint in (13) implies that $\sum_{i=1}^{\min\{m,n\}} \mathbb{E}[(Y_i' - Z_i')^2] \le \delta/c$. Thus the optimal function value in (13) is lower bounded by the optimum value of the following problem.

$$\begin{aligned}
&\min_{P_{Z'|Y'}} \sum_{i=1}^{\min\{m,n\}} I(X_i';Z_i') \quad \text{s.t.} \quad \sum_{i=1}^{\min\{m,n\}} \mathbb{E}[(Y_i'-Z_i')^2] \le \delta/c \quad \text{and} \quad X' \leftrightarrow Y' \leftrightarrow Z'. \\
\equiv &\min_{\sum \delta_i' \le \delta/c, P_{Z'|Y'}} \sum_{i=1}^{\min\{m,n\}} I(X_i';Z_i') \quad \text{s.t.} \quad \forall i, \mathbb{E}[(Y_i'-Z_i')^2] \le \delta_i' \quad \text{and} \quad X' \leftrightarrow Y' \leftrightarrow Z'. \qquad (21)
\end{aligned}$$

Let $Y_{\sim i} := \{Y_1, \ldots, Y_n\} \backslash Y_i$. Since $X' - Y' - Z'$ forms a Markov chain, if $m \leq n$, we have $0 \overset{\text{(e)}}{=} I(X'; Z'|Y') = I(X_1, \ldots, X_m; Z'|Y_1, \ldots, Y_n) \overset{\text{(f)}}{=} \sum_{i=1}^{m} I(X_i; Z'|Y_1, \ldots, Y_n, X_1, \ldots, X_{i-1}) \overset{\text{(f,a,c)}}{=} \sum_{i=1}^{m} I(X_i; Z', X_1, \ldots, X_{i-1}, Y_{\sim i}|Y_i) \overset{\text{(f)}}{\geq} \sum_{i=1}^{m} I(X_i; Z_i|Y_i) \overset{\text{(d)}}{\geq} 0$. Thus,

$$0 \geq \sum_{i=1}^{m} I(X_i; Z_i|Y_i) \geq 0.$$

A similar expression can be derived for the case $m \geq n$. In general, for all $i = 1, \ldots, \min\{m, n\}$, $X_i \leftrightarrow Y_i \leftrightarrow Z_i$ forms a Markov chain. Thus for output perturbation, the Markov constraint on the vectors passes through as a Markov constraint on the individual components of the variables. We can therefore rewrite problem (21) as follows,

$$\min_{\sum \delta_i' \leq \delta/c} \sum_{i=1}^{\min\{m,n\}} \min_{P_{Z_i'|Y_i'}} I(X_i'; Z_i'), \quad \text{s.t.} \quad \forall i, \mathbb{E}[(Y_i' - Z_i')^2] \leq \delta_i' \quad \text{and} \quad X_i' \leftrightarrow Y_i' \leftrightarrow Z_i'.$$

For each i, the solution to the inner constrained minimization problem is given by Proposition 1. Plugging in the solution we arrive at the following constrained convex minimization problem

$$\min_{\sum \delta_i' \leq \delta/c} \sum_{i=1}^{\min\{m,n\}} \max\left\{0, \frac{1}{2}\log\left(\frac{1}{1 - {\rho_i'}^2 + {\rho_i'}^2 \delta_i'}\right)\right\}$$

where $\rho_i' = \mathbb{E}[X_i' Y_i']$ and we have used the expression for the optimal privacy-leakage in the scalar case, i.e., Eq. (20) with $\sigma_Y^2 = 1$. The Lagrangian of the above convex program has the following form

$$\mathcal{L}(\boldsymbol{\delta'}, \boldsymbol{\eta}, \zeta) := \sum_{i=1}^{\min\{m,n\}} \frac{1}{2}\log\left(\frac{1}{1 - {\rho_i'}^2 + {\rho_i'}^2 \delta_i'}\right) + \zeta\left(\sum_{i=1}^{\min\{m,n\}} \delta_i' - \frac{\delta}{c}\right) + \sum_{i=1}^{\min\{m,n\}} \eta_i(\delta_i' - 1),$$

where $\boldsymbol{\delta'} = (\delta_1', \ldots, \delta_{\min\{m,n\}}')$, $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{\min\{m,n\}})$, and ζ and all the η_i's are non-negative Lagrange multipliers. Here, the non-negativity condition associated with $\max\{0, \cdot\}$ has been subsumed by requirement that $\delta_i' \leq 1$ for all i. The Karush-Kuhn-Tucker (KKT) conditions for optimality are as follows,

$$\sum_{i=1}^{\min\{m,n\}} \delta_i' = \frac{\delta}{c}, \forall i, 0 \leq \delta_i' \leq 1, \eta_i \geq 0, \eta_i(\delta_i' - 1) = 0, \frac{\partial \mathcal{L}}{\partial \delta_i'} = 0 \Rightarrow \eta_i = \frac{1}{2(\delta_i' - 1 + {\rho_i'}^{-2})} - \zeta.$$

This implies that if for any i,

$$(2\zeta)^{-1} > {\rho_i'}^{-2} \Leftrightarrow \eta_i > 0, \text{ then } \delta_i' = 1, \text{ otherwise } \delta_i' = (2\zeta)^{-1} - ({\rho_i'}^{-2} - 1).$$

The value of $(2\zeta)^{-1}$ can be found by the equation

$$\sum_{i=1}^{\min\{m,n\}} \max\left\{0, \min\{1, (2\zeta)^{-1} - ({\rho_i'}^{-2} - 1)\}\right\} = \frac{\delta}{c},$$

which is a modified water-filling solution. Based on the value of δ_i', we can construct a Z_i' that attains the lower bound on the mutual information by setting $\sigma_Y^2 = 1$ in the results of Proposition (1). □

6.3 Proof of Proposition 3

Proof. In this proposition, X and Y are assumed to be jointly Gaussian with zero means, unit variances, and correlation coefficient $\rho \in [0, 1]$. Consider the Linear Minimum Mean Squared Error (LMMSE) estimate of X given Z denoted as $\widehat{\mathbb{E}}[X|Z] = \mathbb{E}[XZ]Z/\mathbb{E}[Z^2]$. Then, similar to the proof of Proposition 1, we can expand the mutual information in the following manner.

$$\begin{aligned} I(X; Z) &= h(X) - h(X|Z) = h(X) - h(X - \widehat{\mathbb{E}}[X|Z]|Z) \\ &\geq h(X) - h(X - \widehat{\mathbb{E}}[X|Z]) \geq h(X) - h\Big(\mathcal{N}\Big(0, \mathbb{E}\Big[\Big(X - \widehat{\mathbb{E}}[X|Z]\Big)^2\Big]\Big)\Big) \\ &= -\frac{1}{2}\log\left(1 - \frac{(\mathbb{E}[XZ])^2}{\mathbb{E}[Z^2]}\right), \end{aligned}$$

where in writing the last equality we have used the fact that $\mathbb{E}\left[\widehat{\mathbb{E}}[X|Z](X-\widehat{\mathbb{E}}[X|Z])\right]=0$ by the orthogonality principle of least squares estimation. Thus, we have that

$$\min_{\mathbb{E}[(Y-Z)^2]\leq\delta} I(X;Z) \geq -\frac{1}{2}\log\left[1-\min_{\mathbb{E}[(Y-Z)^2]\leq\delta}\frac{(\mathbb{E}[XZ])^2}{\mathbb{E}[Z^2]}\right] \tag{22}$$

Below, we focus on the minimization problem on the right side of Eq. (22). It will turn out that for the minimizing $Z^\star$, we will have equality in Eq. (22). In what follows, it is helpful to think of the random variables X, Y, Z as vectors in the vector space $\mathcal{L}_2$ of all random variables with finite second moments over the underlying probability space. We will emphasize the vector nature by denoting the random variables X, Y, Z by their corresponding bold lowercase letters $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ respectively. The expectation operator on the product of two random variables in $\mathcal{L}_2$ is an inner product, and hence we can write the optimization problem of interest as follows,

$$\boldsymbol{z}^\star := \arg\min_{\boldsymbol{z}:\|\boldsymbol{z}-\boldsymbol{y}\|^2\leq\delta}\frac{|\langle\boldsymbol{x},\boldsymbol{z}\rangle|^2}{\|\boldsymbol{z}\|^2} = \arg\min_{\boldsymbol{z}:\|\boldsymbol{z}-\boldsymbol{y}\|^2\leq\delta}|\langle\boldsymbol{x},\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}\rangle|^2, \tag{23}$$

where, $\|\boldsymbol{x}\|=\|\boldsymbol{y}\|=1$, and $\langle\boldsymbol{x},\boldsymbol{y}\rangle=\rho$. Let $\hat{\imath}:=\boldsymbol{x}$, $\hat{\jmath}:=\frac{1}{\sqrt{1-\rho^2}}(\boldsymbol{y}-\rho\boldsymbol{x})=\frac{\boldsymbol{y}-\text{Proj}_{\text{Span}(\boldsymbol{x})}(\boldsymbol{y})}{||\boldsymbol{y}-\text{Proj}_{\text{Span}(\boldsymbol{x})}(\boldsymbol{y})||}$, and $\hat{k}:=\frac{\boldsymbol{z}-\text{Proj}_{\text{Span}(\boldsymbol{x},\boldsymbol{y})}(\boldsymbol{z})}{||\boldsymbol{z}-\text{Proj}_{\text{Span}(\boldsymbol{x},\boldsymbol{y})}(\boldsymbol{z})||}$. Then $\hat{\imath},\hat{\jmath},\hat{k}$ are unit vectors along three orthogonal coordinate axes and $\boldsymbol{y}=\rho\hat{\imath}+\sqrt{1-\rho^2}\hat{\jmath}$. Let $\boldsymbol{t}:=\boldsymbol{z}-\boldsymbol{y}=t_1\hat{\imath}+t_2\hat{\jmath}+t_3\hat{k}$ so that $\boldsymbol{z}=(t_1+\rho)\hat{\imath}+(t_2+\sqrt{1-\rho^2})\hat{\jmath}+t_3\hat{k}$. Then the problem in Eq. (23) is equivalent to the following one

$$(t_1^\star,t_2^\star,t_3^\star) := \arg\min_{t_1,t_2,t_3:t_1^2+t_2^2+t_3^2\leq\delta}\left[\frac{(t_1+\rho)^2}{t_1^2+t_2^2+t_3^2+2t_1\rho+2t_2\sqrt{1-\rho^2}+1}\right]. \tag{24}$$

Case $\rho^2\leq\delta$: If $\rho^2\leq\delta$, then $t_1^\star=-\rho, t_2^\star=t_3^\star=0$ is a minimizer of the problem in (24) and $\boldsymbol{z}^\star=\sqrt{1-\rho^2}\hat{\jmath}=\boldsymbol{y}-\rho\boldsymbol{x}$. This solution is displayed along with $\boldsymbol{x}$ and $\boldsymbol{y}$ in Figure 10 and has an immediate geometric interpretation. One can see that $\langle\boldsymbol{x},\boldsymbol{z}\rangle=0$ which implies that $X\perp\!\!\!\perp Z$ because then Z and X are uncorrelated and Z, being a linear combination of X and Y, is jointly Gaussian with them. Also then, $\|\boldsymbol{z}-\boldsymbol{y}\|^2\leq\delta$, or equivalently, $\mathbb{E}[(Y-Z)^2]\leq\delta$. Thus, the lower bound of 0 for $I(X;Z)$ is attained in (14) by this solution.

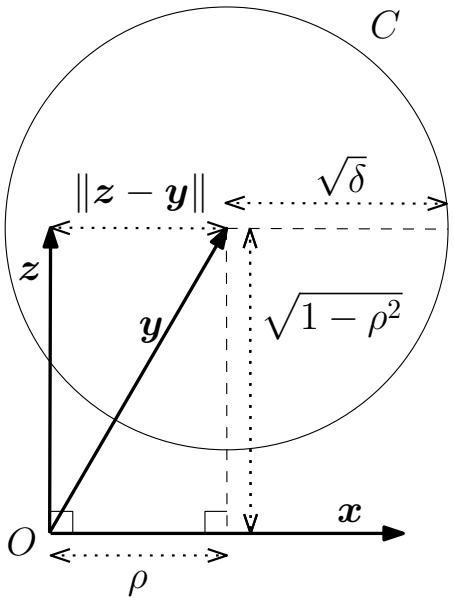


Figure 10: Solution to problem (14) for the case when $0\leq\rho\leq+\sqrt{\delta}$. In the figure, $\boldsymbol{x},\boldsymbol{y}$ represent unit length vectors with inner product equal to $\rho\in[0,1]$. The vector $\boldsymbol{z}$ is perpendicular to $\boldsymbol{x}$ and lies within a distortion sphere of radius $\sqrt{\delta}$ around $\boldsymbol{y}$. The circle C is the projection of the distortion sphere on the $\boldsymbol{x}$-$\boldsymbol{y}$ plane and point O is the origin. The dotted lines with hollow arrowheads denote the lengths of various quantities.

Case $\rho^2 > \delta$: If (t_1, t_2, t_3) is feasible in (24), i.e., $t_1^2 + t_2^2 + t_3^3 \leq \delta$ then so is $(t_1', t_2', t_3') := (t_1, +\sqrt{t_2^2 + t_3^2}, 0)$. If $t_3 \neq 0$ then (t_1', t_2', t_3') strictly dominates (t_1, t_2, t_3) because the denominator of the objective function in (24) is strictly larger for (t_1', t_2', t_3') than for (t_1, t_2, t_3). Thus, we must have

$$t_3^\star = 0, \tag{25}$$

otherwise we can strictly improve (i.e., strictly decrease) the objective function value contradicting the optimality of $(t_1^\star, t_2^\star, t_3^\star)$. Geometrically, this means that $\boldsymbol{z}^\star$ must lie in the two dimensional subspace spanned by $\boldsymbol{x}$ and $\boldsymbol{y}$. Consequently, the minimization problem in (24) reduces to

$$(t_1^\star, t_2^\star) = \arg \min_{t_1, t_2 : t_1^2 + t_2^2 \leq \delta} \left[\frac{(t_1 + \rho)^2}{t_1^2 + t_2^2 + 2t_1\rho + 2t_2\sqrt{1-\rho^2} + 1} \right]. \tag{26}$$

If (t_1, t_2) is feasible in (26), i.e., $t_1^2 + t_2^2 \leq \delta$ then so is $(t_1', t_2') := (t_1, +\sqrt{t_2^2 + (\delta - t_1^2 - t_2^2)})$. If $t_1^2 + t_2^2 < \delta$, then (t_1', t_2') strictly dominates (t_1, t_2) because the denominator of the objective function in (26) is strictly larger for (t_1', t_2') than for (t_1, t_2). Thus, we must have

$$(t_1^\star)^2 + (t_2^\star)^2 = \delta, \tag{27}$$

otherwise we can strictly improve (i.e., strictly decrease) the objective function value contradicting the optimality of $(t_1^\star, t_2^\star)$. Geometrically, this means that $\boldsymbol{z}^\star$ must lie on the circle of radius $+\sqrt{\delta}$ centered at $\boldsymbol{y}$.

Finally, we observe that if $\boldsymbol{z}$ is feasible in (23), i.e., $\|\boldsymbol{y} - \boldsymbol{z}\|^2 \leq \delta$, then so is $\boldsymbol{z}' := \text{Proj}_{\text{Span}(\boldsymbol{z})}(\boldsymbol{y}) = \gamma \boldsymbol{z}$, where $\gamma = \frac{\langle \boldsymbol{y}, \boldsymbol{z} \rangle}{||\boldsymbol{z}||^2}$. This is because the orthogonal projection of a vector onto a subspace is the vector in the subspace closest to it so that $||\boldsymbol{y} - \boldsymbol{z}'||^2 \leq ||\boldsymbol{y} - \boldsymbol{z}||^2$. Also observe that the value of the objective function in (23) is the same for both $\boldsymbol{z}$ and $\gamma \boldsymbol{z}$ and that $(\boldsymbol{y} - \boldsymbol{z}') \perp \boldsymbol{z}'$. Thus, we may assume that there is an optimal solution $\boldsymbol{z}^\star$ such that $(\boldsymbol{y} - \boldsymbol{z}^\star) \perp \boldsymbol{z}^\star$ for if not, we can rescale $\boldsymbol{z}^\star$ suitably to ensure this property without affecting the objective function or violating the distortion constraint. Since $(\boldsymbol{z}^\star - \boldsymbol{y}) = t_1^\star \hat{\imath} + t_2^\star \hat{\jmath}$ and $\boldsymbol{y} = \rho \hat{\imath} + \sqrt{1-\rho^2}\hat{\jmath}$, the orthogonality condition $(\boldsymbol{y} - \boldsymbol{z}^\star) \perp \boldsymbol{z}^\star$ can be restated as

$$t_1^\star(t_1^\star + \rho) + t_2^\star(t_2^\star + \sqrt{1-\rho^2}) = 0$$

which simplifies to

$$(t_1^\star)^2 + (t_2^\star)^2 + t_1^\star \rho + t_2^\star \sqrt{1-\rho^2} = 0 \tag{28}$$

Combining (28) with (27) we get

$$\delta + t_1^\star \rho + \sqrt{(\delta - (t_1^\star)^2)(1-\rho^2)} = 0.$$

This reduces to the following quadratic equation for $t_1^\star$ with two real roots

$$(t_1^\star)^2 + 2\delta\rho t_1^\star + \delta^2 - \delta(1-\rho^2) = 0 \Rightarrow t_1^\star = -\delta\rho \pm \sqrt{\delta(1-\delta)(1-\rho^2)}.$$

We note that $\delta < 1$ since we are considering the case $\delta < \rho^2 \leq 1$. Of the two real roots, $t_1^\star = -\delta\rho - \sqrt{\delta(1-\delta)(1-\rho^2)}$ has a lower objective value in (26). Using this value for $t_1^\star$ and setting $t_2^\star = \sqrt{\delta - (t_1^\star)^2}, t_3^\star = 0$, we can conclude that for the case when $\delta < \rho^2$, the random variable

$$Z^\star := (1-\delta)Y - (X - \rho Y)\sqrt{\frac{\delta(1-\delta)}{1-\rho^2}} \tag{29}$$

attains the lower bound on the mutual information, which equals

$$I(X;Z) = \frac{1}{2}\log\left(\frac{1}{1 - \left(\sqrt{\rho^2(1-\delta)} - \sqrt{(1-\rho^2)\delta}\right)^2} \right). \tag{30}$$

We can interpret the solution geometrically as shown in Figure 11. Unlike the previous case ($0 \leq \rho \leq +\sqrt{\delta}$), here the feasible distortion sphere does not allow $\boldsymbol{z}$ to be perpendicular to $\boldsymbol{x}$. However, one can see that the solution must lie on a tangent from the origin to the distortion sphere. The optimum $\boldsymbol{z}$ in this case (Eq. (29)) is the addition of two vectors, one along $\boldsymbol{y}$ and the other perpendicular to $\boldsymbol{y}$ (the unit vector along which is $-(\boldsymbol{x} - \rho\boldsymbol{y})/\sqrt{1-\rho^2}$). The coefficients for the linear combination can be inferred from the geometry of the figure. □

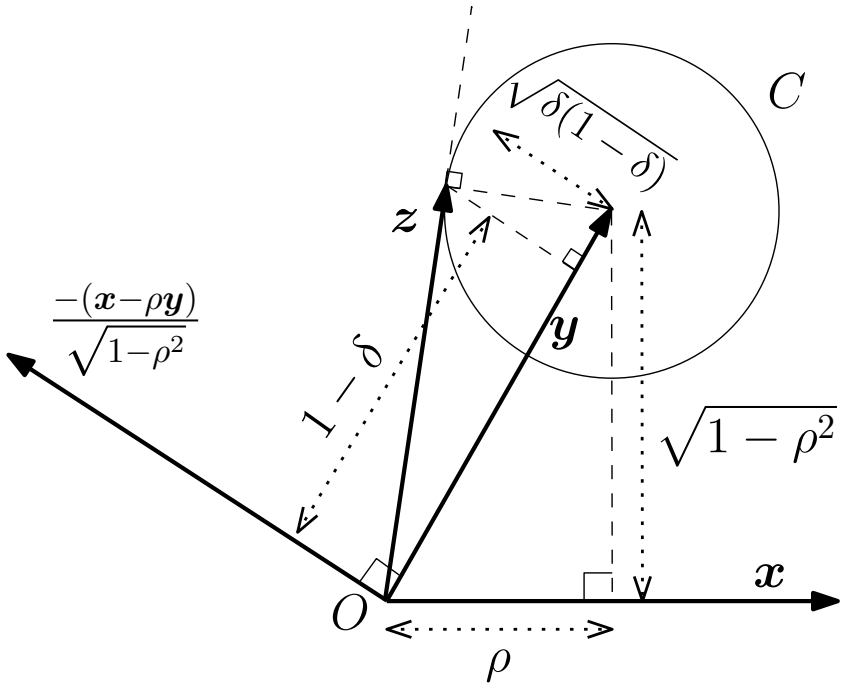


Figure 11: Geometric interpretation of the solution to the problem in (26) for the case when $\rho > +\sqrt{\delta}$. Here, $\boldsymbol{x}, \boldsymbol{y}$ are unit length vectors with inner product equal to ρ. The unit vector perpendicular $\boldsymbol{y}$ is given by $-(\boldsymbol{x}-\rho\boldsymbol{y})/\sqrt{1-\rho^2}$. The circle C is the projection of the feasible distortion sphere onto the $\boldsymbol{x}$-$\boldsymbol{y}$ plane. The problem in (26) can be stated as finding $\boldsymbol{z}$ within the feasible distortion sphere which minimizes the cosine of the angle between $\boldsymbol{z}$ and $\boldsymbol{x}$. The optimum $\boldsymbol{z}$ lies on the tangent from the origin O to the circle C on the far side of $\boldsymbol{x}$. The dotted lines with hollow arrowheads denote the lengths of various quantities.