Variational Leakage: The Role of Information Complexity in Privacy Leakage

Appendices included

Amir Ahooye Atashin*
amir.atashin@gmail.com
University of Geneva
Geneva, Switzerland

Behrooz Razeghi†*
behrooz.razeghi@unige.ch
University of Geneva
Geneva, Switzerland

Deniz Gündüz
d.gunduz@imperial.ac.uk
Imperial College London
London, United Kingdom

Slava Voloshynovskiy
svolos@unige.ch
University of Geneva
Geneva, Switzerland

ABSTRACT

We study the role of information complexity in privacy leakage about an attribute of an adversary's interest, which is not known *a priori* to the system designer. Considering the supervised representation learning setup and using neural networks to parameterize the variational bounds of information quantities, we study the impact of the following factors on the amount of information leakage: information complexity regularizer weight, latent space dimension, the cardinalities of the known utility and unknown sensitive attribute sets, the correlation between utility and sensitive attributes, and a potential bias in a sensitive attribute of adversary's interest. We conduct extensive experiments on Colored-MNIST and CelebA datasets to evaluate the effect of information complexity on the amount of intrinsic leakage.

A repository of the proposed method implementation, Colored-MNIST dataset generator and the corresponding analysis is publicly available at:

https://github.com/BehroozRazeghi/Variational-Leakage

KEYWORDS

Information complexity, privacy, intrinsic leakage, statistical inference, information bottleneck

1 INTRODUCTION

Sensitive information sharing is a challenging problem in information systems. It is often handled by obfuscating the available information before sharing it with other parties. In [17], this problem has been formalized as the **privacy funnel (PF)** in an information theoretic framework. Given two correlated random variables $\mathbf{S}$ and $\mathbf{X}$ with a joint distribution $P_{\mathbf{S},\mathbf{X}}$, where $\mathbf{X}$ represents the available information and $\mathbf{S}$ the private latent variable, the goal of the PF model is to find a representation $\mathbf{Z}$ of $\mathbf{X}$ using a stochastic mapping $P_{\mathbf{Z}|\mathbf{X}}$ such that: (i) $\mathbf{S}$–○–$\mathbf{X}$–○–$\mathbf{Z}$ form a Markov chain; and (ii) representation $\mathbf{Z}$ is maximally informative about the useful data $\mathbf{X}$ (maximizing Shannon's mutual information (MI) $\mathrm{I}(\mathbf{X};\mathbf{Z})$) while being minimally informative about the sensitive data $\mathbf{S}$ (minimizing $\mathrm{I}(\mathbf{S};\mathbf{Z})$). There have been many extensions of this model in the recent literature, e.g., [3, 9, 17, 19–23, 27].

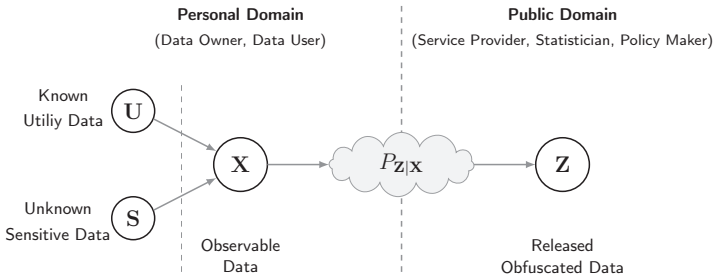


Figure 1: The general setup.

In this paper, we will consider a delicate generalization of the PF model considered in [3, 21], where the goal of the system designer is not to reveal the data that has available but another correlated utility variable. In particular, we assume that the data owner/user acquires some utility from the service provider based on the amount of information disclosed about a utility random variable $\mathbf{U}$ correlated with $\mathbf{X}$, measured by $\mathrm{I}(\mathbf{U};\mathbf{Z})$. Therefore, considering Markov chain $(\mathbf{U},\mathbf{S})$ –○–$\mathbf{X}$–○–$\mathbf{Z}$, the data owner's aim is to share a representation $\mathbf{Z}$ of *observed data* $\mathbf{X}$, through a stochastic mapping $P_{\mathbf{Z}|\mathbf{X}}$, while preserving information about *utility attribute* $\mathbf{U}$ and obfuscate information about *sensitive attribute* $\mathbf{S}$ (see Fig. 1).

The implicit assumption in the PF model presented above and the related generative adversarial privacy framework [10, 31] is to have *pre-defined interests* in the game between the 'defender' (data owner/user) and the 'adversary'; that is, the data owner knows in advance what feature/ variable of the underlying data the adversary is interested in. Accordingly, the data release mechanism can be optimized/ tuned to minimize any inference the adversary can make about this specific random variable. However, this assumption is violated in most real-world scenarios. The attribute that the defender may assume as sensitive may not be the attribute of interest for the inferential adversary. As an example, for a given utility

*Both authors contributed equally to this research.
†Work done while at Imperial College London.

task at hand, the defender may try to restrict inference on gender recognition while the adversary is interested in inferring an individual's identity or facial emotion. Inspired by [11], and in contrast to the above setups, we consider the scenario in which the adversary is curious about an attribute that is *unknown* to the system designer.

In particular, we argue that the information complexity of the representation measured by MI $\mathrm{I}(\mathbf{X};\mathbf{Z})$ can also limit the information leakage about the unknown sensitive variable. In this paper, obtaining the parameterized variational approximation of information quantities, we investigate the core idea of [11] in the supervised representation learning setup.

Notations: Throughout this paper, random vectors are denoted by capital bold letters (e.g., $\mathbf{X}$), deterministic vectors are denoted by small bold letters (e.g., $\mathbf{x}$), and alphabets (sets) are denoted by calligraphic fonts (e.g., $\mathcal{X}$). We use the shorthand $[N]$ to denote the set $\{1, 2, \ldots, N\}$. $\mathrm{H}(P_{\mathbf{X}}) \coloneqq \mathbb{E}_{P_{\mathbf{X}}}[-\log P_{\mathbf{X}}]$ denotes the Shannon's entropy, while $\mathrm{H}(P_{\mathbf{X}}\|Q_{\mathbf{X}}) \coloneqq \mathbb{E}_{P_{\mathbf{X}}}[-\log Q_{\mathbf{X}}]$ denotes the cross-entropy of the distribution $P_{\mathbf{X}}$ relative to a distribution $Q_{\mathbf{X}}$. The relative entropy is defined as $\mathrm{D}_{\mathrm{KL}}(P_{\mathbf{X}}\|Q_{\mathbf{X}}) \coloneqq \mathbb{E}_{P_{\mathbf{X}}}\left[\log \frac{P_{\mathbf{X}}}{Q_{\mathbf{X}}}\right]$. The conditional relative entropy is defined by:

$$\mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{Z}|\mathbf{X}}\|Q_{\mathbf{Z}|\mathbf{X}} \mid P_{\mathbf{X}}\right) \coloneqq \mathbb{E}_{P_{\mathbf{X}}}\left[\mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{Z}|\mathbf{X}=\mathbf{x}}\|Q_{\mathbf{Z}|\mathbf{X}=\mathbf{x}}\right)\right].$$

And the MI is defined by:

$$\mathrm{I}\left(P_{\mathbf{X}}; P_{\mathbf{Z}|\mathbf{X}}\right) \coloneqq \mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{Z}|\mathbf{X}}\|P_{\mathbf{Z}} \mid P_{\mathbf{X}}\right)$$

We abuse notation to write $\mathrm{H}(\mathbf{X}) = \mathrm{H}(P_{\mathbf{X}})$ and $\mathrm{I}(\mathbf{X};\mathbf{Z}) = \mathrm{I}\left(P_{\mathbf{X}}; P_{\mathbf{Z}|\mathbf{X}}\right)$ for random vectors $\mathbf{X} \sim P_{\mathbf{X}}$ and $\mathbf{Z} \sim P_{\mathbf{Z}}$.

2 PROBLEM FORMULATION

Given the observed data $\mathbf{X}$, the data owner wishes to release a representation $\mathbf{Z}$ for a utility task $\mathbf{U}$. Our aim is to investigate the potential statistical inference about a sensitive random attribute $\mathbf{S}$ from the released representation $\mathbf{Z}$. The sensitive attribute $\mathbf{S}$ is possibly also correlated with $\mathbf{U}$ and $\mathbf{X}$.

The objective is to obtain a stochastic map $P_{\mathbf{Z}|\mathbf{X}} : \mathcal{X} \to \mathcal{Z}$ such that $P_{\mathbf{U}|\mathbf{Z}} \approx P_{\mathbf{U}|\mathbf{X}}, \forall\, \mathbf{Z} \in \mathcal{Z}, \forall\, \mathbf{U} \in \mathcal{U}, \forall\, \mathbf{X} \in \mathcal{X}$. This means that the posterior distribution of the utility attribute $\mathbf{U}$ is similar when conditioned on the released representation $\mathbf{Z}$ or on the original data $\mathbf{X}$. Under logarithmic loss, one can measure the utility by Shannon's MI [17, 23, 29]. The logarithmic loss function has been widely used in learning theory [6], image processing [2], information bottleneck [8], multi-terminal source coding [7], and PF [17].

Threat Model: We make minimal assumptions about the adversary's goal, which can model a large family of potential adversaries. In particular, we have the following assumptions:

- The distribution $P_{\mathbf{S}|\mathbf{X}}$ is unknown to the data user/owner. We only restrict attribute $\mathbf{S}$ to be discrete, which captures most scenarios of interest, e.g., a facial attribute, an identity, a political preference.
- The adversary observes released representation $\mathbf{Z}$ and the Markov chain $(\mathbf{U}, \mathbf{S})$–○–$\mathbf{X}$–○–$\mathbf{Z}$ holds.
- We assume the adversary knows the mapping $P_{\mathbf{Z}|\mathbf{X}}$ designed by the data owner, i.e., the data release mechanism is public. Furthermore, the adversary may have access to a collection of the original dataset with the corresponding labels $\mathbf{S}$.

Suppose that the sensitive attribute $\mathbf{S} \in \mathcal{S}$ has a uniform distribution over a discrete set $\mathcal{S}$, where $|\mathcal{S}| = 2^L < \infty$. If $\mathrm{I}(\mathbf{S};\mathbf{Z}) \geq L - \epsilon$, then equivalently $\mathrm{H}(\mathbf{S} \mid \mathbf{Z}) \leq \epsilon$. Also note that due to the Markov chain $\mathbf{S}$–○–$\mathbf{X}$–○–$\mathbf{Z}$, we have $\mathrm{I}(\mathbf{S};\mathbf{Z}) = \mathrm{I}(\mathbf{X};\mathbf{Z}) - \mathrm{I}(\mathbf{X};\mathbf{Z} \mid \mathbf{S})$. When $\mathbf{S}$ is not known a priori, the data owner has no control over $\mathrm{I}(\mathbf{X};\mathbf{Z} \mid \mathbf{S})$. On the other hand, $\mathrm{I}(\mathbf{X};\mathbf{Z})$ can be interpreted as the information complexity of the released representation, which plays a critical role in controlling the information leakage $\mathrm{I}(\mathbf{S};\mathbf{Z})$. Note also that a statistic $\mathbf{Z} = f(\mathbf{X})$ induces a partition on the sample space $\mathcal{X}$, where $\mathbf{Z}$ is sufficient statistic for $\mathbf{U}$ if and only if the assigned samples in each partition do not depend on $\mathbf{U}$. Hence, intuitively, a larger $|\mathcal{U}|$ induces finer partitions on $\mathcal{X}$, which could potentially lead to more leakage about the unknown random function $\mathbf{S}$ of $\mathbf{X}$. This is the core concept of the notion of *variational leakage*, which we shortly address in our experiments.

Since the data owner does not know the particular sensitive variable of interest to the adversary, we argue that it instead aims to design $P_{\mathbf{Z}|\mathbf{X}}$ with the minimum (information) complexity and minimum utility loss. With the introduction of a Lagrange multiplier $\beta \in [0, 1]$, we can formulate the objective of the data owner by *maximizing* the associated Lagrangian functional:

$$\mathcal{L}\left(P_{\mathbf{Z}|\mathbf{X}}, \beta\right) = \mathrm{I}(\mathbf{U};\mathbf{Z}) - \beta\, \mathrm{I}(\mathbf{X};\mathbf{Z}). \tag{1}$$

This is the well-known **information bottleneck (IB)** principle [29], which formulates the problem of extracting, in the most succinct way, the relevant information from random variable $\mathbf{X}$ about the random variable of interest $\mathbf{U}$. Given two correlated random variables $\mathbf{U}$ and $\mathbf{X}$ with joint distribution $P_{\mathbf{U},\mathbf{X}}$, the goal is to find a representation $\mathbf{Z}$ of $\mathbf{X}$ using a stochastic mapping $P_{\mathbf{Z}|\mathbf{X}}$ such that: (i) $\mathbf{U}$–○–$\mathbf{X}$–○–$\mathbf{Z}$, and (ii) $\mathbf{Z}$ is maximally informative about $\mathbf{U}$ (maximizing $\mathrm{I}(\mathbf{U};\mathbf{Z})$) and minimally informative about $\mathbf{X}$ (minimizing $\mathrm{I}(\mathbf{X};\mathbf{Z})$).

Note that in the PF model, $\mathrm{I}(\mathbf{X};\mathbf{Z})$ measures the *useful* information, which is of the designer's interest, while in the IB model, $\mathrm{I}(\mathbf{U};\mathbf{Z})$ measures the *useful* information. Hence, $\mathrm{I}(\mathbf{X};\mathbf{Z} \mid \mathbf{S})$ in PF quantifies the *residual* information, while $\mathrm{I}(\mathbf{X};\mathbf{Z} \mid \mathbf{U})$ in IB quantifies the *redundant* information.

In the sequel, we provide the parameterized variational approximation of information quantities, and then study the impact of the information complexity $\mathrm{I}(\mathbf{X};\mathbf{Z})$ on the information leakage for an unknown sensitive variable.

2.1 Variational Approximation of Information Measures

Let $Q_{\mathbf{U}|\mathbf{Z}} : \mathcal{Z} \to \mathcal{P}(\mathcal{U})$, $Q_{\mathbf{S}|\mathbf{Z}} : \mathcal{Z} \to \mathcal{P}(\mathcal{S})$, $Q_{\mathbf{Z}} : \mathcal{Z} \to \mathcal{P}(\mathcal{Z})$ be variational approximations of the optimal utility decoder distribution $P_{\mathbf{U}|\mathbf{Z}}$, adversary decoder distribution $P_{\mathbf{S}|\mathbf{Z}}$, and latent space distribution $P_{\mathbf{Z}}$, respectively. The common approach is to use **deep neural networks (DNNs)** to model/parameterized these distributions. Let $P_{\boldsymbol{\phi}}(\mathbf{Z} \mid \mathbf{X})$ denote the family of encoding probability distributions $P_{\mathbf{Z}|\mathbf{X}}$ over $\mathcal{Z}$ for each element of space $\mathcal{X}$, parameterized by the output of a DNN $f_{\boldsymbol{\phi}}$ with parameters $\boldsymbol{\phi}$. Analogously, let $P_{\boldsymbol{\theta}}(\mathbf{U}|\mathbf{Z})$ and $P_{\boldsymbol{\xi}}(\mathbf{S} \mid \mathbf{Z})$ denote the corresponding family of decoding probability distributions $Q_{\mathbf{U}|\mathbf{Z}}$ and $Q_{\mathbf{S}|\mathbf{Z}}$, respectively,

parameterized by the output of DNNs $g_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\xi}}$. Let $P_{\mathrm{D}}(\mathbf{X}) = \frac{1}{N}\sum_{n=1}^{N}\delta(\mathbf{x}-\mathbf{x}_n)$, $\mathbf{x}_n \in \mathcal{X}$ denote the empirical data distribution. In this case, $P_{\boldsymbol{\phi}}(\mathbf{X},\mathbf{Z}) = P_{\mathrm{D}}(\mathbf{X})P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})$ denotes our joint inference data distribution, and $P_{\boldsymbol{\phi}}(\mathbf{Z}) = \mathbb{E}_{P_{\mathrm{D}}(\mathbf{X})}\left[P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})\right]$ denotes the learned *aggregated* posterior distribution over latent space $\mathcal{Z}$.

Information Complexity: The information complexity can be decomposed as:

$$\begin{aligned} \mathrm{I}(\mathbf{X};\mathbf{Z}) &= \mathbb{E}_{P_{\mathrm{X,Z}}}\left[\log\frac{P_{\mathrm{X,Z}}}{P_{\mathrm{X}}P_{\mathrm{Z}}}\right] = \mathbb{E}_{P_{\mathrm{X,Z}}}\left[\log\frac{P_{\mathrm{Z|X}}}{Q_{\mathrm{Z}}}\frac{Q_{\mathrm{Z}}}{P_{\mathrm{Z}}}\right] \\ &= \mathbb{E}_{P_{\mathrm{X}}}\left[\mathrm{D_{KL}}\left(P_{\mathrm{Z|X}}\|Q_{\mathrm{Z}}\right)\right] - \mathrm{D_{KL}}\left(P_{\mathrm{Z}}\|Q_{\mathrm{Z}}\right). \end{aligned} \tag{2}$$

Where Q_{Z} is the latent space's prior.

Therefore, the parameterized variational approximation of information complexity (2) can be recast as:

$$\mathrm{I}_{\boldsymbol{\phi}}(\mathbf{X};\mathbf{Z}) \coloneqq \mathrm{D_{KL}}\big(P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X}) \,\|\, Q_{\mathrm{Z}} \mid P_{\mathrm{D}}(\mathbf{X})\big) - \mathrm{D_{KL}}\big(P_{\boldsymbol{\phi}}(\mathbf{Z}) \,\|\, Q_{\mathrm{Z}}\big). \tag{3}$$

The optimal prior Q_{Z}^* minimizing the information complexity is $Q_{\mathrm{Z}}^*(\mathbf{z}) = \mathbb{E}_{P_{\mathrm{D}}(\mathbf{X})}\left[P_{\boldsymbol{\phi}}(\mathbf{Z} \mid \mathbf{X}=\mathbf{x})\right]$; however, it may potentially lead to over-fitting. A critical challenge is to guarantee that the learned aggregated posterior distribution $P_{\boldsymbol{\phi}}(\mathbf{Z})$ conforms well to thd prior Q_{Z} [4, 13, 24, 26, 30]. We can cope with this issue by employing a more *expressive* form for Q_{Z}, which would allow us to provide a good fit of an arbitrary space for $\mathcal{Z}$, at the expense of additional *computational complexity*.

Information Utility: The parameterized variational approximation of MI between the released representation $\mathbf{Z}$ and the utility attribute $\mathbf{U}$ can be recast as:

$$\begin{aligned} &\mathrm{I}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{U};\mathbf{Z}) \coloneqq \\ &\mathbb{E}_{P_{\mathrm{U,X}}}\left[\mathbb{E}_{P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})}\left[\log\frac{P_{\boldsymbol{\theta}}(\mathbf{U}|\mathbf{Z})}{P_{\mathrm{U}}}\cdot\frac{P_{\boldsymbol{\theta}}(\mathbf{U})}{P_{\boldsymbol{\theta}}(\mathbf{U})}\right]\right] = \\ &\mathbb{E}_{P_{\mathrm{U,X}}}\left[\mathbb{E}_{P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})}\left[\log P_{\boldsymbol{\theta}}(\mathbf{U}|\mathbf{Z})\right]\right] \\ &\quad - \mathbb{E}_{P_{\mathrm{U}}}\left[\log\frac{P_{\mathrm{U}}}{P_{\boldsymbol{\theta}}(\mathbf{U})}\right] + \mathbb{E}_{P_{\mathrm{U}}}\left[\log P_{\boldsymbol{\theta}}(\mathbf{U})\right] \\ &= -\mathrm{H}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{U}|\mathbf{Z}) - \mathrm{D_{KL}}\left(P_{\mathrm{U}}\|P_{\boldsymbol{\theta}}(\mathbf{U})\right) + \mathrm{H}\left(P_{\mathrm{U}}\|P_{\boldsymbol{\theta}}(\mathbf{U})\right) \\ &\geq \underbrace{-\mathrm{H}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{U}|\mathbf{Z})}_{\text{Prediction Fidelity}} - \underbrace{\mathrm{D_{KL}}\left(P_{\mathrm{U}}\,\|\,P_{\boldsymbol{\theta}}(\mathbf{U})\right)}_{\text{Distribution Discrepancy Loss}}, \end{aligned}$$

where $\mathrm{H}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{U}|\mathbf{Z}) = -\mathbb{E}_{P_{\mathrm{U,X}}}\left[\mathbb{E}_{P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})}\left[\log P_{\boldsymbol{\theta}}(\mathbf{U}|\mathbf{Z})\right]\right]$ represents the parameterized decoder uncertainty, and in the last line we use the positivity of the cross-entropy $\mathrm{H}\left(P_{\mathrm{U}}\|P_{\boldsymbol{\theta}}(\mathbf{U})\right)$.

3 LEARNING MODEL

System Designer: Given independent and identically distributed (i.i.d.) training samples $\{(\mathbf{u}_n,\mathbf{x}_n)\}_{n=1}^{N} \subseteq \mathcal{U}\times\mathcal{X}$, and using stochastic gradient descent (SGD)-type approach, DNNs $f_{\boldsymbol{\phi}}$, $g_{\boldsymbol{\theta}}$, $D_{\boldsymbol{\eta}}$, and $D_{\boldsymbol{\omega}}$ are trained together to maximize a Monte-Carlo approximation of the deep variational IB functional over parameters $\boldsymbol{\phi}$, $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, and $\boldsymbol{\omega}$ (Fig. 2). Backpropagation through random samples from the posterior distribution $P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})$is required in our framework, which is a challenge since backpropagation cannot flow via random nodes; to overcome this hurdle, we apply the reparameterization approach [14]).

The inferred posterior distribution is typically assumed to be a multi-variate Gaussian with a diagonal co-variance, i.e., $P_{\boldsymbol{\phi}}(\mathbf{Z} \mid \mathbf{x}) = \mathcal{N}\big(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x}))\big)$. Suppose $\mathcal{Z} = \mathbb{R}^d$. We first sample a random variable $\boldsymbol{\mathcal{E}}$ i.i.d. from $\mathcal{N}(\mathbf{0},\mathbf{I}_d)$, then given data sample $\mathbf{x} \in \mathcal{X}$, we generate the sample $\mathbf{z} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x}) \odot \boldsymbol{\varepsilon}$, where $\odot$ is the element-wise (Hadamard) product. The latent space prior distribution is typically considered as a fixed d-dimensional standard isotropic multi-variate Gaussian, i.e., $Q_{\mathrm{Z}} = \mathcal{N}(\mathbf{0},\mathbf{I}_d)$. For this simple choice, the information complexity upper bound

$$\mathbb{E}_{P_{\boldsymbol{\phi}}(\mathbf{X},\mathbf{Z})}\left[\log\frac{P_{\boldsymbol{\phi}}(\mathbf{Z} \mid \mathbf{X})}{Q_{\mathrm{Z}}}\right] = \mathbb{E}_{P_{\mathrm{D}}(\mathbf{X})}\left[\mathrm{D_{KL}}\left(P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})\,\|Q_{\mathrm{Z}}\right)\right]$$

has a closed-form expression, which reads as:

$$2\,\mathrm{D_{KL}}\left(P_{\boldsymbol{\phi}}(\mathbf{Z} \mid \mathbf{X}=\mathbf{x}) \,\|\, Q_{\mathrm{Z}}\right) = \|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x})\|_2^2 + d + \sum_{i=1}^{d}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x})_i - \log\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x})_i)$$

The KL-divergences in (3) and (4) can be estimated using the density-ratio trick [18, 28], utilized in the GAN framework to directly match the data and generated model distributions. The trick is to express two distributions as conditional distributions, conditioned on a label $C \in \{0,1\}$, and reduce the task to binary classification. The key point is that we can estimate the KL-divergence by estimating the ratio of two distributions without modeling each distribution explicitly.

Consider $\mathrm{D_{KL}}\big(P_{\boldsymbol{\phi}}(\mathbf{Z}) \,\|\, Q_{\mathrm{Z}}\big) = \mathbb{E}_{P_{\boldsymbol{\phi}}(\mathbf{Z})}\left[\log\frac{P_{\boldsymbol{\phi}}(\mathbf{Z})}{Q_{\mathrm{Z}}}\right]$. We now define $\rho_{\mathrm{Z}}(\mathbf{z} \mid c)$ as $\rho_{\mathrm{Z}}(\mathbf{z} \mid c=1) = P_{\boldsymbol{\phi}}(\mathbf{Z})$, $\rho_{\mathrm{Z}}(\mathbf{z} \mid c=0) = Q_{\mathrm{Z}}$. Suppose that a perfect binary classifier (discriminator) $D_{\boldsymbol{\eta}}(\mathbf{z})$, with parameters $\boldsymbol{\eta}$, is trained to associate the label $c=1$ to samples from distribution $P_{\boldsymbol{\phi}}(\mathbf{Z})$ and the label $c=0$ to samples from Q_{Z}. Using the Bayes' rule and assuming that the marginal class probabilities are equal, i.e., $\rho(c=1) = \rho(c=0)$, the density ratio can be expressed as:

$$\frac{P_{\boldsymbol{\phi}}(\mathbf{Z}=\mathbf{z})}{Q_{\mathrm{Z}}(\mathbf{z})} = \frac{\rho_{\mathrm{Z}}(\mathbf{z} \mid c=1)}{\rho_{\mathrm{Z}}(\mathbf{z} \mid c=0)} = \frac{\rho_{\mathrm{Z}}(c=1 \mid \mathbf{z})}{\rho_{\mathrm{Z}}(c=0 \mid \mathbf{z})} \approx \frac{D_{\boldsymbol{\eta}}(\mathbf{z})}{1-D_{\boldsymbol{\eta}}(\mathbf{z})}.$$

Therefore, given a trained discriminator $D_{\boldsymbol{\eta}}(\mathbf{z})$ and M i.i.d. samples $\{\mathbf{z}_m\}_{m=1}^{M}$ from $P_{\boldsymbol{\phi}}(\mathbf{Z})$, we estimate $\mathrm{D_{KL}}\big(P_{\boldsymbol{\phi}}(\mathbf{Z}) \,\|\, Q_{\mathrm{Z}}\big)$ as:

$$\mathrm{D_{KL}}\big(P_{\boldsymbol{\phi}}(\mathbf{Z}) \,\|\, Q_{\mathrm{Z}}\big) \approx \frac{1}{M}\sum_{m=1}^{M}\log\frac{D_{\boldsymbol{\eta}}(\mathbf{z}_m)}{1-D_{\boldsymbol{\eta}}(\mathbf{z}_m)}. \tag{4}$$

Our model is trained using alternating block coordinate descend across five steps (See Algorithm 1).

Inferential Adversary: Given the publicly-known encoder $\boldsymbol{\phi}$ and K i.i.d. samples $\{(\mathbf{s}_k,\mathbf{z}_k)\}_{k=1}^{K} \subseteq \mathcal{S}\times\mathcal{Z}$, the adversary trains an inference network $\boldsymbol{\xi}$ to minimize $\mathrm{H}_{\boldsymbol{\xi}}(\mathrm{S}|\mathrm{Z})$.

4 EXPERIMENTS

In this section, we show the impact of the following factors on the amount of leakage: (i) information complexity regularizer weight $\beta \in (0,1]$, (ii) released representation dimension d_z, (iii) cardinalities of the known utility and unknown sensitive attribute sets, (iv) correlation between the utility and sensitive attributes, and (v) potential bias in a sensitive attribute of adversary's interest. We conduct experiments on the Colored-MNIST and large-scale CelebA

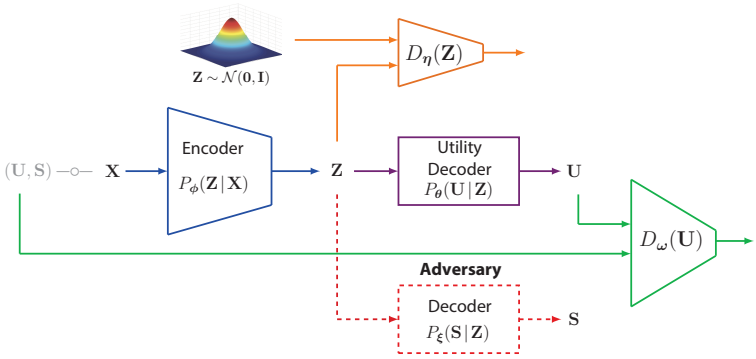


Figure 2: The training and testing architecture. During training, the data user/owner trains the parameterized networks $(\phi, \theta, \eta, \omega)$. During testing, only the encoder-decoder pair (ϕ, θ) is used. The adversary uses the publicly-known (fixed) encoder ϕ and a collection of the original dataset, and trains an inference network ξ to infer attribute S of his interest.

datasets. The Colored-MNIST[1] is *our modified* version of MNIST [15], which is a collection of 70,000 *colored* digits of size 28×28. The digits are randomly colored with red, green, or blue based on two distributions, as explained in the caption of Fig. 4. The CelebA [16] dataset contains 202,599 images of size 218×178. We used TensorFlow 2.4.1 [1] with Integrated Keras API. The method details and network architectures are provided in Appendix. A and Appendix B.

The first and second rows of Fig. 3 and Fig. 4 depict the trade-off among (i) information complexity, (ii) service provider's accuracy on utility attribute U, and (iii) adversary's accuracy on attribute S. The third row depicts the amount of information revealed about S, i.e., $\mathrm{I}(\mathbf{S};\mathbf{Z})$, for the scenarios considered in the first and second rows, which are estimated using MINE [5]. The fourth row depicts the amount of released information about the utility attribute U, i.e., $\mathrm{I}(\mathbf{U};\mathbf{Z})$, corresponding to the considered scenarios in the first and second rows, also estimated using MINE. We consider different portions of the datasets available for training adversary's network, denoted by the 'data ratio'.

The experiments on CelebA consider the scenarios in which the attributes U and S are correlated, while $|\mathcal{U}| = |\mathcal{S}| = 2$. We provide utility accuracy curves for (i) training set, (ii) validation set, and (iii) test set. As we have argued, there is a direct relationship between information complexity and intrinsic information leakage. Note that, as β increases, the information complexity is reduced, and we observe that this also results in a reduction in the information leakage. We also see that the leakage is further reduced when the dimension of the released representation Z, i.e., d_z, is reduced. This forces the data owner to obtain a more succinct representation of the utility variable, removing any extra information.

In the Colored-MNIST experiments, provided that the model eliminates all the redundant information $\mathrm{I}(\mathbf{X};\mathbf{Z}|\mathbf{U})$ and leaves only the information about U, we expect the adversary's performance

[1]Several papers have employed Colored-MNIST dataset; However, they are not unique, and researchers synthesized different versions based on their application. The innovative concept behind our version was influenced from the one used in [25].

Algorithm 1 Training Algorithm: Data Owner

Inputs: Training Dataset: $\{(\mathbf{u}_n, \mathbf{x}_n)\}_{n=1}^N$;
Hyper-Parameter: β;
$\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\omega}$ ← Initialize Network Parameters
repeat
(1) **Train the Encoder and Utility Decoder** $(\boldsymbol{\phi}, \boldsymbol{\theta})$
  Sample a mini-batch $\{\mathbf{u}_m, \mathbf{x}_m\}_{m=1}^M \sim P_{\mathrm{D}}(\mathbf{X}) P_{\mathrm{U}|\mathrm{X}}$
  Compute $\mathbf{z}_m \sim f_{\boldsymbol{\phi}}(\mathbf{x}_m), \forall m \in [M]$
  Back-propagate loss:

$$\mathcal{L}(\boldsymbol{\phi},\boldsymbol{\theta}) = -\frac{1}{M}\sum_{m=1}^{M}\big(\log P_{\boldsymbol{\theta}}(\mathbf{u}_m|\mathbf{z}_m) - \beta\, \mathrm{D}_{\mathrm{KL}}\big(P_{\boldsymbol{\phi}}(\mathbf{z}_m|\mathbf{x}_m)\|Q_{\mathrm{Z}}(\mathbf{z}_m)\big)\big)$$

(2) **Train the Latent Space Discriminator** $\boldsymbol{\eta}$
  Sample $\{\mathbf{x}_m\}_{m=1}^M \sim P_{\mathrm{D}}(\mathbf{X})$
  Sample $\{\widetilde{\mathbf{z}}_m\}_{m=1}^M \sim Q_{\mathrm{Z}}$
  Compute $\mathbf{z}_m \sim f_{\boldsymbol{\phi}}(\mathbf{x}_m), \forall m \in [M]$
  Back-propagate loss:

$$\mathcal{L}(\boldsymbol{\eta}) = -\frac{\beta}{M}\sum_{m=1}^{M}\big(\log D_{\boldsymbol{\eta}}(\mathbf{z}_m) + \log\big(1 - D_{\boldsymbol{\eta}}(\widetilde{\mathbf{z}}_m)\big)\big)$$

(3) **Train the Encoder** $\boldsymbol{\phi}$ **Adversarially**
  Sample $\{\mathbf{x}_m\}_{m=1}^M \sim P_{\mathrm{D}}(\mathbf{X})$
  Compute $\mathbf{z}_m \sim f_{\boldsymbol{\phi}}(\mathbf{x}_m), \forall m \in [M]$
  Back-propagate loss: $\mathcal{L}(\boldsymbol{\phi}) = \frac{\beta}{M}\sum_{m=1}^{M}\log D_{\boldsymbol{\eta}}(\mathbf{z}_m)$

(4) **Train the Attribute Class Discriminator** $\boldsymbol{\omega}$
  Sample $\{\mathbf{u}_m\}_{m=1}^M \sim P_{\mathrm{U}}$
  Sample $\{\widetilde{\mathbf{z}}_m\}_{m=1}^M \sim Q_{\mathrm{Z}}$
  Compute $\widetilde{\mathbf{u}}_m \sim g_{\boldsymbol{\theta}}(\widetilde{\mathbf{z}}_m), \forall m \in [M]$
  Back-propagate loss:

$$\mathcal{L}(\boldsymbol{\omega}) = -\frac{1}{M}\sum_{m=1}^{M}\big(\log D_{\boldsymbol{\omega}}(\mathbf{u}_m) + \log(1 - D_{\boldsymbol{\omega}}(\widetilde{\mathbf{u}}_m))\big)$$

(5) **Train the Utility Decoder** $\boldsymbol{\theta}$ **Adversarially**
  Sample $\{\widetilde{\mathbf{z}}_m\}_{m=1}^M \sim Q_{\mathrm{Z}}$
  Compute $\widetilde{\mathbf{u}}_m \sim g_{\boldsymbol{\theta}}(\widetilde{\mathbf{z}}_m), \forall m \in [M]$
  Back-propagate loss: $\mathcal{L}(\boldsymbol{\omega}) = \frac{1}{M}\sum_{m=1}^{M}\log\big(1 - D_{\boldsymbol{\omega}}(\widetilde{\mathbf{u}}_m)\big)$
until Convergence
return $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\omega}$

to be close to 'random guessing' since the digit color is independent of its value. We investigate the impact of the cardinality of sets $|\mathcal{U}|$ and $|\mathcal{S}|$, as well as possible biases in the distribution of S. The results show that it is possible to reach the same level of accuracy on the utility attribute U, while reducing the intrinsic leakage by increasing the regularizer weight β, or equivalently, by reducing the information complexity $\mathrm{I}_{\boldsymbol{\phi}}(\mathbf{X};\mathbf{Z})$. An interesting possible scenario is to consider correlated attributes U and S with different cardinality sets $\mathcal{U}$ and $\mathcal{S}$. For instance, utility task U is personal

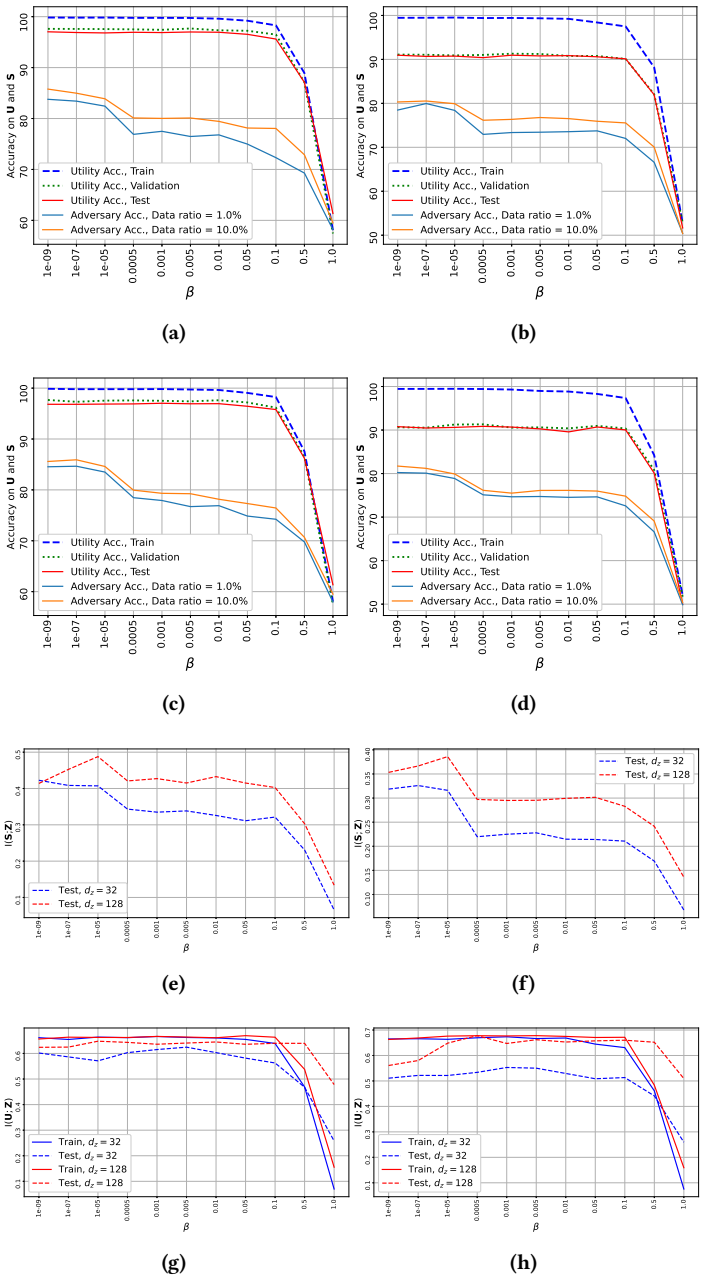


Figure 3: The results on CelebA dataset, considering isotropic Gaussian prior. (First Row): $d_z = 32$; (Second Row): $d_z = 128$; (Third Row): Estimated information leakage $I(S;Z)$ using MINE; (Fourth Row): Estimated useful information $I(U;Z)$ using MINE. (First Column): utility task is gender recognition ($|\mathcal{U}| = 2$), adversary's interest is heavy makeup ($|\mathcal{S}| = 2$); (Second Column): utility task is emotion (smiling) recognition ($|\mathcal{U}| = 2$), adversary's interest is mouth slightly open ($|\mathcal{S}| = 2$).

identification, while the adversary's interest S is gender recognition.

5 CONCLUSION

We studied the *variational leakage* to address the amount of potential privacy leakage in a supervised representation learning setup. In contrast to the PF and generative adversarial privacy models, we consider the setup in which the adversary's interest is not known a priori to the data owner. We study the role of information complexity in information leakage about an attribute of an adversary interest. This was addressed by approximating the information quantities using DNNs and experimentally evaluating the model on large-scale image databases. The proposed notion of *variational leakage* relates the amount of leakage to the minimal sufficient statistics.

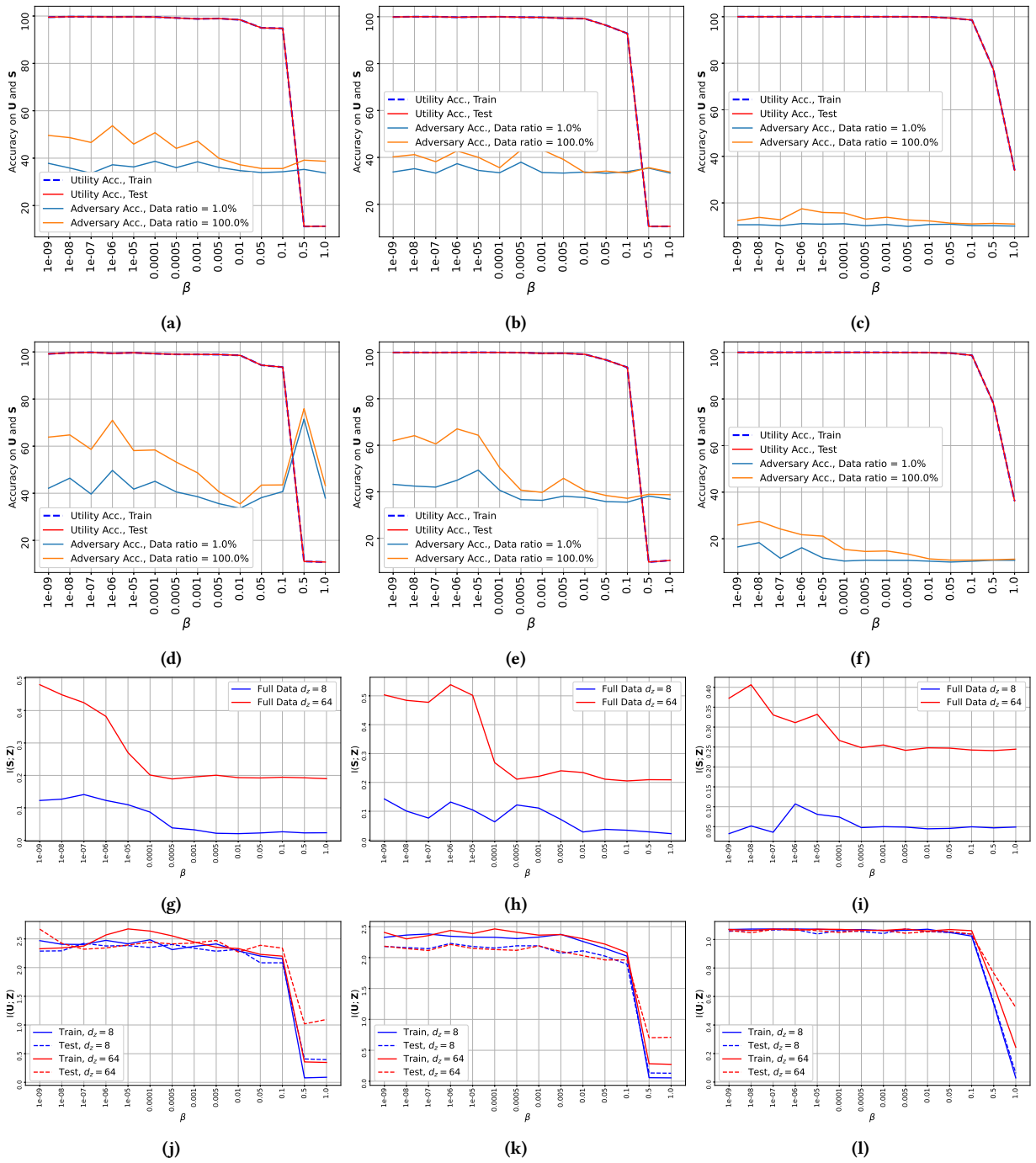


Figure 4: The results on Colored-MNIST dataset, considering isotropic Gaussian prior. (First Row): $d_z = 8$; (Second Row): $d_z = 64$; (Third Row): estimated information leakage $I(S;Z)$ using MINE; (Fourth Row): estimated useful information $I(U;Z)$ using MINE. (First Column): utility task is digit recognition ($|\mathcal{U}| = 10$), while the adversary's goal is the digit color ($|\mathcal{S}| = 3$), setting $P_S(\text{Red}) = P_S(\text{Green}) = P_S(\text{Blue}) = \frac{1}{3}$; (Second Column): utility task is digit recognition ($|\mathcal{U}| = 10$), while the adversary's goal is the digit color, setting $P_S(\text{Red}) = \frac{1}{2}$, $P_S(\text{Green}) = \frac{1}{6}$, $P_S(\text{Blue}) = \frac{1}{3}$; (Third Column): utility task is digit color recognition ($|\mathcal{U}| = 3$), while the adversary's interest is the digit number ($|\mathcal{S}| = 10$).

REFERENCES

[1] Martín Abadi et al. 2016. Tensorflow: A system for large-scale machine learning. In *{USENIX} Symp. on Operating Sys. Design and Impl. ({OSDI})*. 265–283.

[2] Thomas Andre, Marc Antonini, Michel Barlaud, and Robert M Gray. 2006. Entropy-based distortion measure for image coding. In *2006 International Conference on Image Processing*. IEEE, 1157–1160.

[3] Yuksel Ozan Basciftci, Ye Wang, and Prakash Ishwar. 2016. On privacy-utility tradeoffs for constrained data release mechanisms. In *2016 Information Theory and Applications Workshop (ITA)*. IEEE, 1–6.

[4] Matthias Bauer and Andriy Mnih. 2019. Resampled priors for variational autoencoders. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 66–75.

[5] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *International Conference on Machine Learning*. 531–540.

[6] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, learning, and games*. Cambridge university press.

[7] Thomas A Courtade and Richard D Wesel. 2011. Multiterminal source coding with an entropy-based distortion measure. In *2011 IEEE International Symposium on Information Theory Proceedings*. IEEE, 2040–2044.

[8] Peter Harremoës and Naftali Tishby. 2007. The information bottleneck revisited or how to choose a good distortion measure. In *2007 IEEE International Symposium on Information Theory*. IEEE, 566–570.

[9] Hsiang Hsu, Shahab Asoodeh, and Flavio P. Calmon. 2019. Obfuscation via Information Density Estimation. In *Proceeding of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[10] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. 2017. Context-aware generative adversarial privacy. *Entropy* 19, 12 (2017), 656.

[11] Ibrahim Issa, Aaron B Wagner, and Sudeep Kamath. 2019. An operational approach to information leakage. *IEEE Transactions on Information Theory* 66, 3 (2019), 1625–1657.

[12] D. P Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*. 4743–4751.

[14] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*.

[15] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. (2010). http://yann.lecun.com/exdb/mnist/

[16] Z. Liu, P. Luo, X. Wang, and X. Tang. 2015. Deep Learning Face Attributes in the Wild. In *International Conference on Computer Vision (ICCV)*.

[17] Ali Makhdoumi, Salman Salamatian, Nadia Fawaz, and Muriel Médard. 2014. From the information bottleneck to the privacy funnel. In *2014 IEEE Information Theory Workshop (ITW 2014)*. IEEE, 501–505.

[18] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. 2010. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory* 56, 11 (2010), 5847–5861.

[19] Flavio P. Calmon, Ali Makhdoumi, and Muriel Médard. 2015. Fundamental limits of perfect privacy. In *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 1796–1800.

[20] Borzoo Rassouli and Deniz Gündüz. 2019. Optimal utility-privacy trade-off with total variation distance as a privacy measure. *IEEE Transactions on Information Forensics and Security* 15 (2019), 594–603.

[21] Borzoo Rassouli and Deniz Gündüz. 2021. On perfect privacy. In *to appear in IEEE Journal on Selected Areas in Information Theory (JSAIT)*. IEEE.

[22] Borzoo Rassouli, Fernando E Rosas, and Deniz Gündüz. 2019. Data Disclosure under Perfect Sample Privacy. *IEEE Trans. on Inform. Forensics and Security* (2019).

[23] Behrooz Razeghi, Flavio P. Calmon, Deniz Gündüz, and Slava Voloshynovskiy. 2020. On Perfect Obfuscation: Local Information Geometry Analysis. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–6.

[24] Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*. 1530–1538.

[25] Borja Rodríguez-Gálvez, Ragnar Thobaben, and Mikael Skoglund. 2020. A Variational Approach to Privacy and Fairness. *arXiv preprint arXiv:2006.06332* (2020).

[26] Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. 2018. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847* (2018).

[27] Sreejith Sreekumar and Deniz Gündüz. 2019. Optimal Privacy-Utility Trade-off under a Rate Constraint. In *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2159–2163.

[28] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 2012. Density-ratio matching under the Bregman divergence: A unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics* 64, 5 (2012), 1009–1044.

[29] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. In *IEEE Allerton*.

[30] Jakub Tomczak and Max Welling. 2018. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics*. 1214–1223.

[31] Ardhendu Tripathy, Ye Wang, and Prakash Ishwar. 2019. Privacy-preserving adversarial networks. In *57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 495–505.

Appendices

A TRAINING DETAILS

All the experiments in the paper have been carried out with the following structure:

A.0.1 Pre-Training Phase.
We utilize this phase to warm-up our model before running the main training Algorithm 1 for the Variational Leakage framework within all experiments. In the warm-up phase, we pre-trained encoder ($f_{\boldsymbol{\phi}}$) and utility-decoder ($g_{\boldsymbol{\theta}}$) together for the few epochs via **backpropagation (BP)** with the Adam optimizer [12]. We found out the warm-up stage was helpful for faster convergence. Therefore, we initialize the encoder and the utility-decoder weights with the obtained values rather than random or zero initialization. For each experiment, the hyper-parameters of the learning algorithm in this phase were:

Experiment Dataset	Learning Rate	Max Iteration	Batch Size
Colored-MNIST (both version)	0.005	50	1024
CelebA	0.0005	100	512

A.0.2 Main Block-wise Training Phase.
In contrast to the most DNNs training algorithms, each iteration only has one forward step through the network's weights and then update weights via BP approach. Our training strategy is block-wise and consists of multiple blocks in the main algorithm loop. At each block, forward and backward steps have been done through the specific path in our model, and then corresponding parameters update based on the block's output loss path.

Since it was not possible for us to use the Keras API's default model training function, we implement Algorithm 1 from scratch in the Tensorflow. It is important to remember that we initialize all parameters to zero except for the $(\boldsymbol{\phi}, \boldsymbol{\theta})$ values which acquired in the previous stage. Furthermore, we set the learning rate of the block (1) in the Algorithm 1, five times larger than other blocks. The hyper-parameters of the Algorithm 1 for each experiment shown in the following table:

Experiment Dataset	Learning Rate [blocks (2)-(5)]	Max Iteration	Batch Size
Colored-MNIST (both version)	0.0001	500	2048
CelebA	0.00001	500	1024

B NETWORK ARCHITECTURES

B.0.1 MI Estimation.
For all experiments in this paper, we report estimation of MI between the released representation and sensitive attribute, i.e., $I(\mathbf{S};\mathbf{Z})$, as well as the MI between the released representation and utility attribute, i.e., $I(\mathbf{U};\mathbf{Z})$. To estimate MI, we employed the MINE model [5]. The architecture of the model is depicted in Table. 1.

B.0.2 Colored-MNIST.
In the Colored-MNIST experiment, we had two versions for data utility and privacy leakage evaluation. In the first version, we set the utility data to the class' label of the input image and consider the color of the input image as sensitive data, and for the second one, we did vice versa. It is worth mentioning that both balanced and unbalanced Colored-MNIST datasets are applied with the same architecture given in Table 2.

MINE I $(\mathbf{U};\mathbf{Z})$
Input $\mathbf{z} \in \mathbb{R}^{d_z}$ Code; $\mathbf{u} \in \mathbb{R}^{\lvert\mathcal{U}\rvert}$
x = Concatenate([z, u])
FC(100), ELU
FC(100), ELU
FC(100), ELU
FC(1)

Table 1: The architecture of the MINE network.

Encoder $f_{\boldsymbol{\phi}}$
Input $\mathbf{x} \in \mathbb{R}^{28\times28\times3}$ Color Image
Conv(64,5,2), BN, LeakyReLU
Conv(128,5,2), BN, LeakyReLU
Flatten
FC($d_z \times 4$), BN, Tanh
μ: FC(d_z), σ: FC(d_z)
z=SamplingWithReparameterizationTrick[μ,σ]
Utility Decoder $g_{\boldsymbol{\theta}}$
Input $\mathbf{z} \in \mathbb{R}^{d_z}$ Code
FC($d_z \times 4$), BN, LeakyReLU
FC($\lvert\mathcal{U}\rvert$), SOFTMAX
Latent Space Discriminator $D_{\boldsymbol{\eta}}$
Input $\mathbf{z} \in \mathbb{R}^{d_z}$ Code
FC(512), BN, LeakyReLU
FC(256), BN, LeakyReLU
FC(1), Sigmoid
Utility Attribute Class Discriminator $D_{\boldsymbol{\omega}}$
Input $\mathbf{u} \in \mathbb{R}^{\lvert\mathcal{U}\rvert}$
FC($\lvert\mathcal{U}\rvert \times 8$), BN, LeakyReLU
FC($\lvert\mathcal{U}\rvert \times 8$), BN, LeakyReLU
FC(1), Sigmoid

Table 2: The architecture of DNNs used in for the Colored-MNIST experiments.

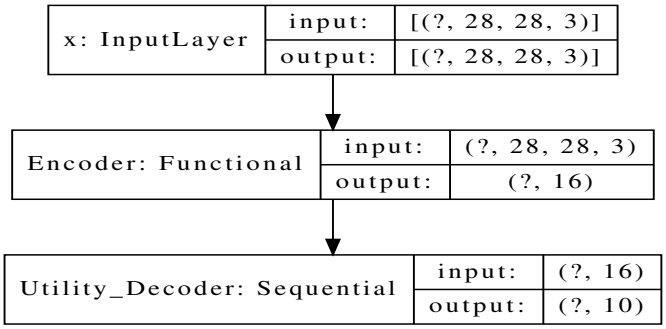
B.0.3 CelebA.
In this experiment, we considered three scenarios for data utility and privacy leakage evaluation, as shown in Table 3. Note that all of the utility and sensitive attributes are binary. The architecture of the networks are presented in Table 4.

Scenario Number	Utility Attribute	Sensitive Attribute
1	Gender	Heavy Makeup
2	Mouth Slightly Open	Smiling
3	Gender	Blond Hair

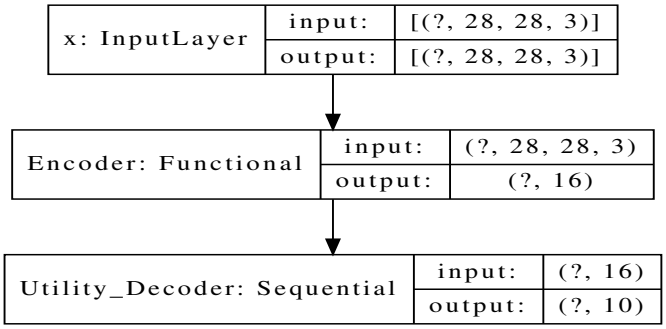
Table 3: Scenarios considered for CelebA experiments.

Encoder f_{ϕ}
Input $\mathbf{x} \in \mathbb{R}^{64\times64\times3}$ Color Image
Conv(16,3,2), BN, LeakyReLU
Conv(32,3,2), BN, LeakyReLU
Conv(64,3,2), BN, LeakyReLU
Conv(128,3,2), BN, LeakyReLU
Conv(256,3,2), BN, LeakyReLU
Flatten
FC($d_z \times 4$), BN, Tanh
μ: FC(d_z), σ: FC(d_z)
z=SamplingWithReparameterizationTrick[μ,σ]
Utility Decoder g_{θ}
Input $\mathbf{z} \in \mathbb{R}^{d_z}$ Code
FC(d_z), BN, LeakyReLU
FC($\lvert\mathcal{U}\rvert$), SOFTMAX
Latent Space Discriminator D_{η}
Input $\mathbf{z} \in \mathbb{R}^{d_z}$ Code
FC(512), BN, LeakyReLU
FC(256), BN, LeakyReLU
FC(1), Sigmoid
Utility Attribute Class Discriminator D_{ω}
Input $\mathbf{u} \in \mathbb{R}^{\lvert\mathcal{U}\rvert}$
FC($\lvert\mathcal{U}\rvert \times 4$), BN, LeakyReLU
FC($\lvert\mathcal{U}\rvert$), BN, LeakyReLU
FC(1), Sigmoid

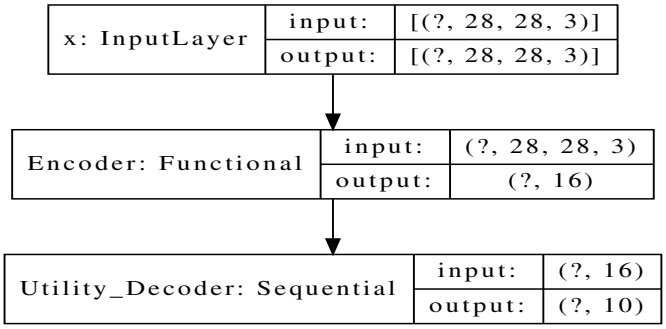
Table 4: The architecture of networks for the CelebA experiments.



(a) Autoencoder module



(b) The encoder and the latent space discriminator module



(c) utility attribute decoder and the utility discriminator module.

Figure 5: The three sub-modules that make up the main network.

C IMPLEMENTATION OVERVIEW

Fig. 6 demonstrates all sub networks of the proposed framework that attached together and their parameters learned in the Algorithm 1.

However, we did not define and save our model in this form because of technical reasons to efficiently implement the training algorithm. As shown in Algorithm 1, the main loop consists of five blocks where only some networks are used in the forward phase, and mostly one of them would update their parameters via BP in each block. Therefore, we shattered the model into three sub-modules in the training stage for simplicity and performance. Fig. 5 shows the corresponding sub-modules of Fig. 6, which are used in our implementation. During training, all of the sub-module (a) parameters, call "autoencoder part" would update with BP after each forward step. For the (b) and (c) sub-modules, only the parameters of one network are updated when the corresponding error function values backpropagate, and we freeze the other networks parameters in the sub-module. For example, block (3) of Algorithm 1 is related to the (b) sub-module, but at the BP step, the latent space discriminator is frozen to prevent its parameters from updating. This procedure is vice versa for module (b) at block (2).

It should be mentioned that during our experiments, we found out that before running our main algorithm, it is beneficial to pretrain the autoencoder sub-module since we need to sample from the latent space, which uses in other parts of the main model during training. We justify this by mentioning that sampling meaningful data rather than random ones from latent variables from

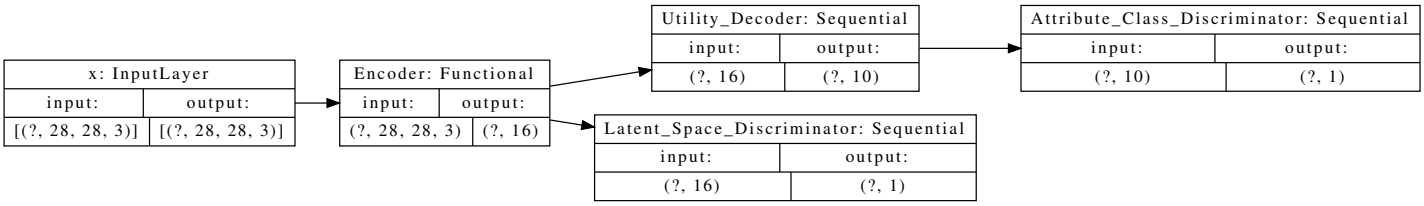


Figure 6: Complete model structure in the training phase. The above model defines for Colored-MNIST dataset with the number of classes as utility attributes and digit colors as sensitive attributes. Also, the encoder output is set to 16 neurons. (The adversary model is not part of the data owner training algorithm)

the beginning of learning helps the model to converge better and faster in comparison with starting Algorithm 1 with a randomly initiated autoencoder.