Variational Leakage: The role of Information Complexity in Privacy Leakage

Amir Ahooye Atashin*‡, Behrooz Razeghi*‡ §, Deniz Gündüz†, Slava Voloshynovskiy*
*University of Geneva
†Imperial College London

***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 parametrize 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, cardinality of the known utility and unknown sensitive attribute sets, the correlation between utility and sensitive attributes, and the potential bias in a sensitive attribute of adversary's interest. We conduct extensive experiments on Colored-MNIST and CelebA datasets, with a public implementation available, to evaluate the effect of information complexity on the amount of intrinsic leakage.**

***Index Terms*—Information complexity, privacy, intrinsic leakage, statistical inference, information bottleneck.**

I. 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 [1], this problem has been formalized as the *privacy funnel* in an information theoretic framework. Given two correlated random variables $\mathbf{S}$ and $\mathbf{X}$ with 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 privacy funnel model is to find a representation $\mathbf{Z}$ of $\mathbf{X}$ using a stochastic mapping $P_{\mathbf{Z}|\mathbf{X}}$ such that: (i) $\mathbf{S} -\!\circ\!- \mathbf{X} -\!\circ\!- \mathbf{Z}$ and (ii) representation $\mathbf{Z}$ is maximally informative about the useful data $\mathbf{X}$ (maximizing $\mathrm{I}(\mathbf{X};\mathbf{Z})$) 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., [1]–[9].

In this paper, we will consider a slight generalization of the privacy funnel model considered in [3], [9], where the goal of the system designer is not to reveal the data it has available, but another correlated utility variable. In particular, the data owner/user observes random data $\mathbf{X} \in \mathcal{X}$, and acquires some utility from the service provider based on the representation $\mathbf{Z} = f(\mathbf{X}) \sim P_{\mathbf{Z}|\mathbf{X}}$ she discloses. The utility acquired depends on a utility random variable $\mathbf{U}$ correlated with $\mathbf{X}$. The amount of useful information revealed to the service provider is measured by the Shannon's mutual information (MI) $\mathrm{I}(\mathbf{U};\mathbf{Z})$. At the same time, an inferential adversary observes the released representation $\mathbf{Z}$ and is interested in an attribute $\mathbf{S}$ of the original data $\mathbf{X}$. The amount of information leaked to the service provider (public domain) about the sensitive variable $\mathbf{S}$ is also measured by the mutual information, $\mathrm{I}(\mathbf{S};\mathbf{Z})$. Therefore, considering Markov chain $(\mathbf{U},\mathbf{S}) -\!\circ\!- \mathbf{X} -\!\circ\!- \mathbf{Z}$, 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).

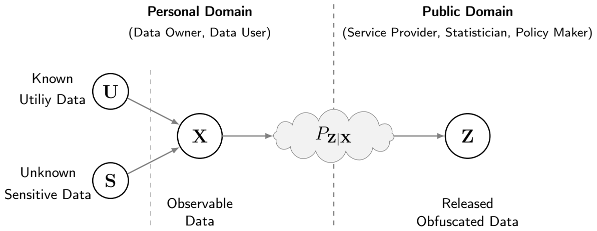


Fig. 1: The general setup.

The implicit assumption in the privacy funnel model presented above and the related generative adversarial privacy framework [10], [11] 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. In other words, the attribute that the defender may assume as sensitive, and hence, try to minimize its leakage, may not be the attribute of interest for the inferential adversary. As an example, for a given utility task at hand, the defender may try to restrict inference on gender recognition, while the adversary is interested to infer an individual's identity or facial emotion. Inspired by [12], 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 that can be revealed to the service provider can also limit the information leakage about the unknown sensitive variable. We measure the information complexity also by the mutual information $\mathrm{I}(\mathbf{X};\mathbf{Z})$. In this paper, obtaining the parametrized variational approximation of information quantities, we investigate the core idea of [12] in the supervised representation learning setup.

[‡] The authors contributed equally.
[§] Work done while at Imperial College London.
Implementation codes and full version available at: https://github.com/BehroozRazeghi/Variational-Leakage

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

II. Problem Formulation

Given the observed data $\mathbf{X}$ the data owner wishes to release a representation $\mathbf{Z}$ for a utility task $\mathbf{U}$. In the same time, our aim is to further investigate the statistical inference about a sensitive random attribute $\mathbf{S}$ from the released representation $\mathbf{Z}$. The sensitive attribute $\mathbf{S}$ depends on $\mathbf{X}$, and it is possibly also correlated with $\mathbf{U}$. The general objective is to obtain a stochastic map $P_{\mathbf{Z}|\mathbf{X}} : \mathcal{X} \rightarrow \mathcal{Z}$ 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 the logarithmic loss, one can measure the utility by Shannon's MI [1], [8], [13]. The logarithmic loss function has been widely used in learning theory [14], image processing [15], information bottleneck [16], multi-terminal source coding [17], and privacy funnel [1].

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:

- We assume the adversary is interested in an attribute $\mathbf{S}$ of data $\mathbf{X}$, which is *not known a priori* to the data user/owner. In other words, 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 $(\mathbf{U}, \mathbf{S}) -\!\!\circ\!\!- \mathbf{X} -\!\!\circ\!\!- \mathbf{Z}$ 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, we assume 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 $\mathbf{S} -\!\!\circ\!\!- \mathbf{X} -\!\!\circ\!\!- \mathbf{Z}$, 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 $\mathcal{X}$, statistic $\mathbf{Z}$ is sufficient 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 [13], 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 of *original* IB is to find a representation $\mathbf{Z}$ of $\mathbf{X}$ using a stochastic mapping $P_{\mathbf{Z}|\mathbf{X}}$ such that: (i) $\mathbf{U} -\!\!\circ\!\!- \mathbf{X} -\!\!\circ\!\!- \mathbf{Z}$ and (ii) representation $\mathbf{Z}$ is maximally informative about $\mathbf{U}$ (maximizing $\mathrm{I}(\mathbf{U}; \mathbf{Z})$) while being minimally informative about $\mathbf{X}$ (minimizing $\mathrm{I}(\mathbf{X}; \mathbf{Z})$).

In the sequel, we provide the parametrized 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.

A. Variational Approximation of Information Measures

Let $Q_{\mathbf{U}|\mathbf{Z}} : \mathcal{Z} \rightarrow \mathcal{P}(\mathcal{U})$, $Q_{\mathbf{S}|\mathbf{Z}} : \mathcal{Z} \rightarrow \mathcal{P}(\mathcal{S})$, $Q_{\mathbf{Z}} : \mathcal{Z} \rightarrow \mathcal{P}(\mathcal{Z})$ 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 neural networks to parametrize 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 $\mathcal{X}$, parametrized by the output of a deep neural network $f_{\boldsymbol{\phi}}$ with parameters $\boldsymbol{\phi}$. Analogously, let $P_{\boldsymbol{\theta}}(\mathbf{U}|\mathbf{Z})$ and $P_{\boldsymbol{\xi}}(\mathbf{S} \,|\, \mathbf{Z})$ denote the corresponding family of decoding probability distributions $Q_{\mathbf{U}|\mathbf{Z}}$ and $Q_{\mathbf{S}|\mathbf{Z}}$, respectively, parametrized by the output of deep neural networks $g_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\xi}}$. Let $P_{\mathsf{D}}(\mathbf{X}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{x} - \mathbf{x}_n)$, $\mathbf{x}_n \in \mathcal{X}$ denote the empirical data distribution. In this case, $P_{\boldsymbol{\phi}}(\mathbf{X}, \mathbf{Z}) = P_{\mathsf{D}}(\mathbf{X}) P_{\boldsymbol{\phi}}(\mathbf{Z} \,|\, \mathbf{X})$ denotes our joint inference data distribution, and $P_{\boldsymbol{\phi}}(\mathbf{Z}) = \mathbb{E}_{P_{\mathsf{D}}(\mathbf{X})}\left[P_{\boldsymbol{\phi}}(\mathbf{Z} \,|\, \mathbf{X})\right]$ denotes the learned *aggregated* posterior distribution over latent space $\mathcal{Z}$.

Information Complexity. The information complexity can be decomposed as:

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

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

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

The information complexity, $\mathrm{I}_{\boldsymbol{\phi}}(\mathbf{X};\mathbf{Z})$, measures the amount of MI between the parameters of the model and the dataset D, given a prior $Q_{\mathbf{Z}}$ and stochastic map $P_{\boldsymbol{\phi}}(\mathbf{Z}\,|\,\mathbf{X}) : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$. Note that the posterior distribution $P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})$ depends on the choice of the optimization algorithm; therefore, the information complexity implicitly depends on this choice. The optimal prior $Q^*_{\mathbf{Z}}$ minimizing the information complexity is $Q^*_{\mathbf{Z}}(\mathbf{z}) = \mathbb{E}_{P_{\mathrm{D}}(\mathbf{X})}\left[P_{\boldsymbol{\phi}}\left(\mathbf{Z}\mid\mathbf{X}=\mathbf{x}\right)\right]$; however, it may potentially lead to over-fitting. This is beyond the scope of this paper. A critical challenge is to guarantee that the learned aggregated posterior distribution $P_{\boldsymbol{\phi}}(\mathbf{Z})$ conforms well to proposal prior $Q_{\mathbf{Z}}$ [18]–[22]. We can cope with this issue by employing a more *expressive* form for $Q_{\mathbf{Z}}$, 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 parametrized variational approximation of MI between the released representation $\mathbf{Z}$ and the utility attribute $\mathbf{U}$ can be recast as:

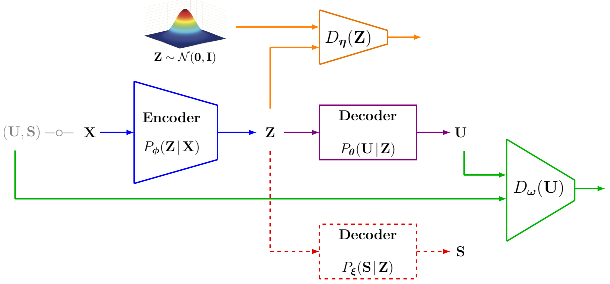
$$\begin{aligned}
&\mathrm{I}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{U};\mathbf{Z})\\
&\coloneqq \mathbb{E}_{P_{\mathbf{U},\mathbf{X}}}\left[\mathbb{E}_{P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})}\left[\log\frac{P_{\boldsymbol{\theta}}(\mathbf{U}\,|\,\mathbf{Z})}{P_{\mathbf{U}}}\cdot\frac{P_{\boldsymbol{\theta}}(\mathbf{U})}{P_{\boldsymbol{\theta}}(\mathbf{U})}\right]\right]\\
&= \mathbb{E}_{P_{\mathbf{U},\mathbf{X}}}\left[\mathbb{E}_{P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})}\left[\log P_{\boldsymbol{\theta}}(\mathbf{U}\,|\,\mathbf{Z})\right]\right]\\
&\qquad - \mathbb{E}_{P_{\mathbf{U}}}\left[\log\frac{P_{\mathbf{U}}}{P_{\boldsymbol{\theta}}(\mathbf{U})}\right] + \mathbb{E}_{P_{\mathbf{U}}}\left[\log P_{\boldsymbol{\theta}}(\mathbf{U})\right]\\
&= -\mathrm{H}_{\boldsymbol{\phi},\boldsymbol{\theta}}\left(\mathbf{U}\,|\,\mathbf{Z}\right) - \mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{U}}\|P_{\boldsymbol{\theta}}(\mathbf{U})\right) + \mathrm{H}\left(P_{\mathbf{U}}\|P_{\boldsymbol{\theta}}(\mathbf{U})\right)\\
&\geq -\mathrm{H}_{\boldsymbol{\phi},\boldsymbol{\theta}}\left(\mathbf{U}\,|\,\mathbf{Z}\right) - \mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{U}}\,\|\,P_{\boldsymbol{\theta}}(\mathbf{U})\right),
\end{aligned}$$

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

III. Learning Model

System Designer. Given a collection of 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 the stochastic gradient descent (SGD)-type algorithms, deep neural networks $f_{\boldsymbol{\phi}}$, $g_{\boldsymbol{\theta}}$, $D_{\boldsymbol{\eta}}$, and $D_{\boldsymbol{\omega}}$ are trained jointly 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). In order to have a stable gradient with respect to the encoder, the reparametrization trick [23] is used to sample from the learned posterior distribution $P_{\boldsymbol{\phi}}(\mathbf{Z}|\mathbf{X})$.

The inferred posterior distribution is typically a multivariate Gaussian with diagonal co-variance, i.e., $P_{\boldsymbol{\phi}}(\mathbf{Z}\mid\mathbf{x}) = \mathcal{N}\big(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \mathsf{diag}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x}))\big)$. 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 multivariate Gaussian, i.e., $Q_{\mathbf{Z}} = \mathcal{N}(\mathbf{0},\mathbf{I}_d)$. For this simple explicit choice, the information complexity upper bound $\mathbb{E}_{P_{\mathrm{D}}(\mathbf{X})}\left[\mathrm{D}_{\mathrm{KL}}\left(P_{\boldsymbol{\phi}}(\mathbf{Z}\,|\,\mathbf{X})\,\|Q_{\mathbf{Z}}\right)\right]$ has a closed-form expression for a given sample $\mathbf{x}$, which reads as $2\,\mathrm{D}_{\mathrm{KL}}\left(P_{\boldsymbol{\phi}}(\mathbf{Z}\,|\,\mathbf{X}=\mathbf{x})\,\|\,Q_{\mathbf{Z}}\right) = \|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x})\|_2^2 + d + \sum_{i=1}^{d}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x})_i - \log\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x})_i)$.



$\mathbf{Z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
$D_{\eta}(\mathbf{Z})$
$(\mathbf{U},\mathbf{S})$
$\mathbf{X}$
Encoder
$P_{\phi}(\mathbf{Z}|\mathbf{X})$
$\mathbf{Z}$
Decoder
$P_{\theta}(\mathbf{U}|\mathbf{Z})$
$\mathbf{U}$
$D_{\omega}(\mathbf{U})$
Decoder
$P_{\xi}(\mathbf{S}|\mathbf{Z})$
$\mathbf{S}$

Fig. 2: The training and testing architecture. During training, the data user/owner trains the parametrized networks $(\boldsymbol{\phi},\boldsymbol{\theta},\boldsymbol{\eta},\boldsymbol{\omega})$. During testing, the data user/owner only uses the encoder-decoder pair $(\boldsymbol{\phi},\boldsymbol{\theta})$. The adversary uses the publicly-known (fixed) encoder $\boldsymbol{\phi}$ as well as a collection of the original dataset, and trains an inference network $\boldsymbol{\xi}$ to infer attribute $\mathbf{S}$ of his interest.

The KL-divergences in (3) and (4) can be estimated using the density-ratio trick [24], [25], utilized in the GAN framework to directly match the data distribution and generated model distribution. 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 $\mathrm{D}_{\mathrm{KL}}(P_{\boldsymbol{\phi}}(\mathbf{Z})\,\|\,Q_{\mathbf{Z}}) = \mathbb{E}_{P_{\boldsymbol{\phi}}(\mathbf{Z})}[\log\frac{P_{\boldsymbol{\phi}}(\mathbf{Z})}{Q_{\mathbf{Z}}}]$. We now define $\rho_{\mathbf{Z}}(\mathbf{z}\,|\,c)$ as $\rho_{\mathbf{Z}}(\mathbf{z}\,|\,c\!=\!1) = P_{\boldsymbol{\phi}}(\mathbf{Z})$, $\rho_{\mathbf{Z}}(\mathbf{z}\,|\,c\!=\!0) = Q_{\mathbf{Z}}$. 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 $Q_{\mathbf{Z}}$. 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:

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

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

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

Our model is trained using alternating block coordinate descend across five steps. The training algorithm is given in 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 its uncertainty $\mathrm{H}_{\boldsymbol{\xi}}(\mathbf{S}\,|\,\mathbf{Z})$.

IV. 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

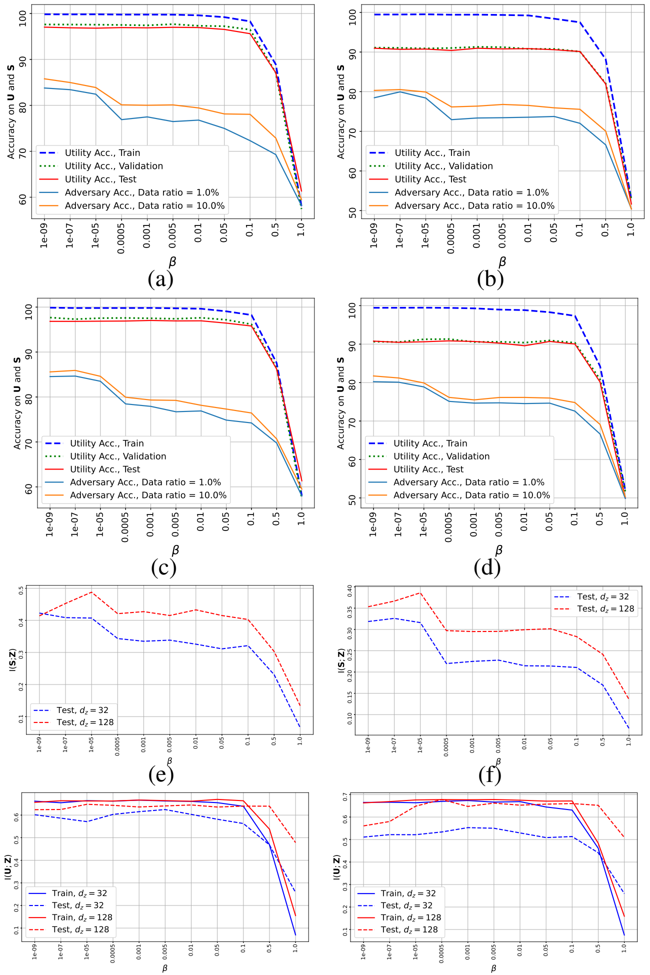
Algorithm 1 Training Algorithm: Data Owner

```
**Inputs:** Training Dataset: $\{(\mathbf{u}_n, \mathbf{x}_n)\}_{n=1}^{N}$;
            Hyper-Parameter: $\beta$;
$\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\omega} \leftarrow$ Initialize Network Parameters
**repeat**
    (1) **Train the Encoder and Utility Decoder** $(\boldsymbol{\phi}, \boldsymbol{\theta})$
    Sample a mini-batch $\{\mathbf{u}_m, \mathbf{x}_m\}_{m=1}^{M} \sim P_{\mathrm{D}}(\mathbf{X}) P_{\mathbf{U}|\mathbf{X}}$
    Compute $\mathbf{z}_m \sim f_{\boldsymbol{\phi}}(\mathbf{x}_m), \forall m \in [M]$
    Back-propagate loss:
        $\mathcal{L}(\boldsymbol{\phi},\boldsymbol{\theta}) = -\frac{1}{M}\sum_{m=1}^{M}\big(\log P_{\boldsymbol{\theta}}(\mathbf{u}_m \,|\, \mathbf{z}_m)$
            $-\beta\, \mathrm{D}_{\mathrm{KL}}(P_{\boldsymbol{\phi}}(\mathbf{z}_m \,|\, \mathbf{x}_m) \| Q_{\mathbf{Z}}(\mathbf{z}_m))\big)$
    (2) **Train the Latent Space Discriminator** $\boldsymbol{\eta}$
    Sample $\{\mathbf{x}_m\}_{m=1}^{M} \sim P_{\mathrm{D}}(\mathbf{X})$
    Sample $\{\widetilde{\mathbf{z}}_m\}_{m=1}^{M} \sim Q_{\mathbf{Z}}$
    Compute $\mathbf{z}_m \sim f_{\boldsymbol{\phi}}(\mathbf{x}_m), \forall m \in [M]$
    Back-propagate loss:
        $\mathcal{L}(\boldsymbol{\eta}) = -\frac{\beta}{M}\sum_{m=1}^{M}(\log D_{\boldsymbol{\eta}}(\mathbf{z}_m) + \log(1 - D_{\boldsymbol{\eta}}(\widetilde{\mathbf{z}}_m)))$
    (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 $\{\mathbf{u}_m\}_{m=1}^{M} \sim P_{\mathbf{U}}$
    Sample $\{\widetilde{\mathbf{z}}_m\}_{m=1}^{M} \sim Q_{\mathbf{Z}}$
    Compute $\widetilde{\mathbf{u}}_m \sim g_{\boldsymbol{\theta}}(\widetilde{\mathbf{z}}_m), \forall m \in [M]$
    Back-propagate loss:
        $\mathcal{L}(\boldsymbol{\omega}) = -\frac{1}{M}\sum_{m=1}^{M}(\log D_{\boldsymbol{\omega}}(\mathbf{u}_m) + \log(1 - D_{\boldsymbol{\omega}}(\widetilde{\mathbf{u}}_m)))$
    (5) **Train the Utility Decoder** $\boldsymbol{\theta}$ **Adversarially**
    Sample $\{\widetilde{\mathbf{z}}_m\}_{m=1}^{M} \sim Q_{\mathbf{Z}}$
    Compute $\widetilde{\mathbf{u}}_m \sim g_{\boldsymbol{\theta}}(\widetilde{\mathbf{z}}_m), \forall m \in [M]$
    Back-propagate loss: $\mathcal{L}(\boldsymbol{\omega}) = \frac{1}{M}\sum_{m=1}^{M}\log(1 - D_{\boldsymbol{\omega}}(\widetilde{\mathbf{u}}_m))$
**until** Convergence
**return** $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\omega}$
```

d_{z}, (iii) cardinality 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 datasets. The Colored-MNIST is *our modified* version MNIST [26], 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 caption of Fig. 4. The CelebA [27] dataset contains 202,599 colored images of size 218 × 178. We used TensorFlow 2.4 [28] with Integrated Keras API to implement and train the above-explained algorithm. The training details and network architectures are provided in Appendices A and B in the longer version of this paper.

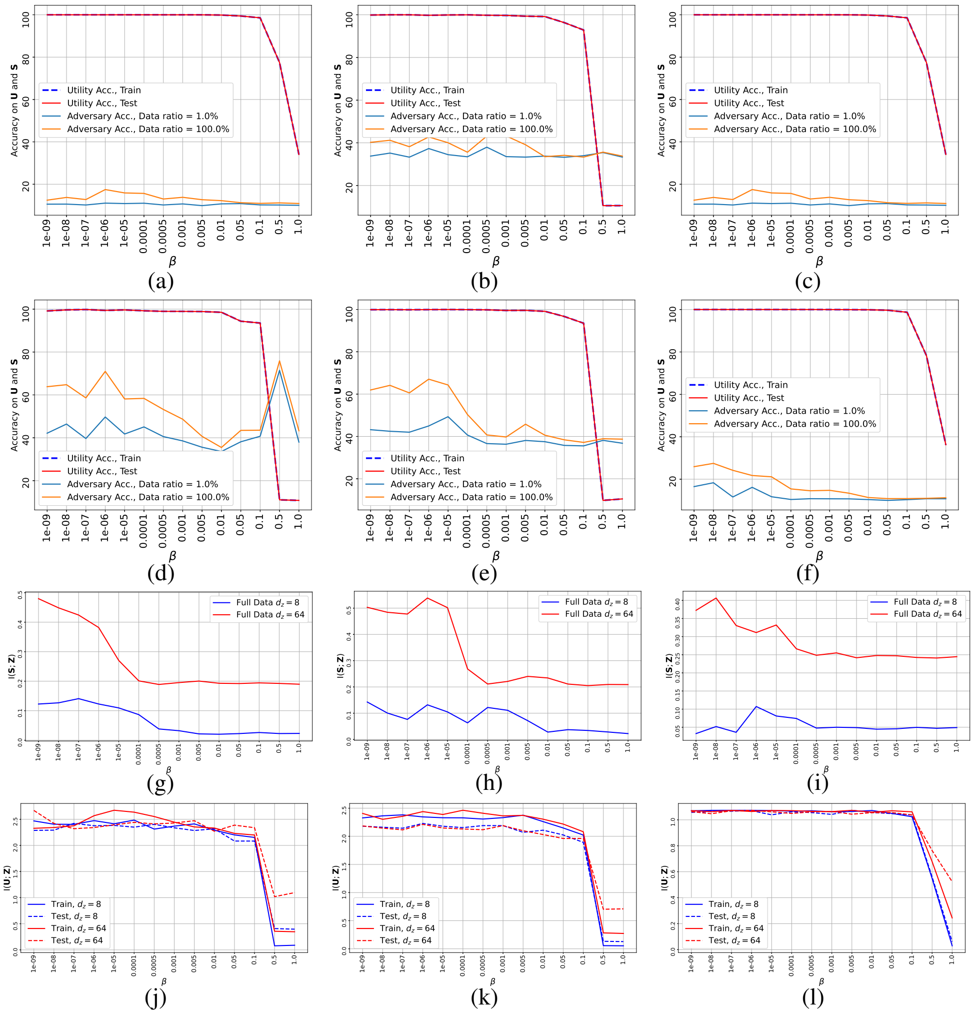
The first and second rows of Fig. 3 and Fig. 4 depict the trade-off among (i) information complexity, (ii) service provider's accuracy on utility attribute $\mathbf{U}$, and (iii) adversary's accuracy on attribute $\mathbf{S}$. The third row depicts the amount of information revealed about a sensitive random attribute $\mathbf{S}$, i.e., $\mathrm{I}(\mathbf{S};\mathbf{Z})$, corresponding to the considered scenarios in the first and second rows, which is estimated using MINE [29]. The fourth row depicts the amount of released useful information about the utility attribute $\mathbf{U}$, 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'.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3: The results on CelebA dataset, considering isotropic Gaussian prior. (First Row): $d_{\mathrm{z}} = 64$; (Second Row): $d_{\mathrm{z}} = 128$; (Third Row): Estimated information leakage $\mathrm{I}(\mathbf{S};\mathbf{Z})$ using MINE; (Fourth Row): Estimated useful information $\mathrm{I}(\mathbf{U};\mathbf{Z})$ 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 ($|\mathcal{S}| = 2$).

The experiments on CelebA consider the scenarios in which the attributes $\mathbf{U}$ and $\mathbf{S}$ are correlated, while $|\mathcal{U}| = |\mathcal{S}| = 2$. We provide the utility accuracy curves for (i) training set, (ii) validation set, and (iii) test set. As we have argued, there is a direct relationship between the information complexity and the intrinsic information leakage. Note that, as β increases, the



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

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

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 $\mathbf{Z}$, i.e., d_{z}, is reduced. This forces the data owner to obtain a more succinct representation of the utility random variable, removing any extra information.

In the Colored-MNIST experiments, provided that the model eliminates all the redundant information $\mathrm{I}(\mathbf{X};\mathbf{Z} \mid \mathbf{U})$ and leaves only the information about $\mathbf{U}$, we expect the adversary's performance 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 $\mathbf{S}$. The results show that it is possible to reach the same level of accuracy on the utility attribute $\mathbf{U}$, while reducing the intrinsic leakage by increasing the regularizer weight β, or equivalently, by reducing the information complexity $\mathrm{I}_{\phi}(\mathbf{X};\mathbf{Z})$. A possible interesting scenario is to consider correlated attributes $\mathbf{U}$ and $\mathbf{S}$ with different cardinality sets $\mathcal{U}$ and $\mathcal{S}$. For instance, utility task $\mathbf{U}$ is personal identification, while the adversary's interest $\mathbf{S}$ is gender recognition.

V. Conclusion

We studied the *variational leakage* to address the amount of potential privacy leakage in a supervised representation learning setup. In contrast to the privacy funnel and the 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's interest. This was addressed by approximating the information quantities using neural networks 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.

References

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

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

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

[4] Sreejith Sreekumar and Deniz Gündüz, "Optimal privacy-utility trade-off under a rate constraint," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 2159–2163.

[5] Hsiang Hsu, Shahab Asoodeh, and Flavio P. Calmon, "Obfuscation via information density estimation," in *Proceeding of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[6] Borzoo Rassouli, Fernando E Rosas, and Deniz Gündüz, "Data disclosure under perfect sample privacy," *IEEE Transactions on Information Forensics and Security*, 2019.

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

[8] Behrooz Razeghi, Flavio P. Calmon, Deniz Gündüz, and Slava Voloshynovskiy, "On perfect obfuscation: Local information geometry analysis," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2020, pp. 1–6.

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

[10] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, pp. 656, 2017.

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

[12] Ibrahim Issa, Aaron B Wagner, and Sudeep Kamath, "An operational approach to information leakage," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1625–1657, 2019.

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

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

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

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

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

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

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

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

[21] Jakub Tomczak and Max Welling, "VAE with a VampPrior," in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1214–1223.

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

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

[24] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5847–5861, 2010.

[25] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori, "Density-ratio matching under the Bregman divergence: A unified framework of density-ratio estimation," *Annals of the Institute of Statistical Mathematics*, vol. 64, no. 5, pp. 1009–1044, 2012.

[26] Yann LeCun and Corinna Cortes, "MNIST handwritten digit database," 2010.

[27] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *International Conference on Computer Vision (ICCV)*, December 2015.

[28] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

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

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

Appendix A
Training Details

All the experiment in the paper has been done with the following structure:

1) *Pre-Training Phase:*

We utilize the warm-up phase before running training Algorithm 1 for the Variational Leakage framework within all experiments. In the warm-up phase, we pre-trained encoder (f_ϕ) and utility-decoder (g_θ) together for the few epochs via backpropagation with the Adam optimizer [30]. We found out the warm-up stage was helpful in method convergence speed. 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

2) *Main Block-wised Training Phase:*

In contrast to the most neural network models that only have one forward step and a backward step in each epoch to update all network weights, our learning algorithm consists of several blocks in which forward and backward steps have been done on different paths. The model's parameters update based on the corresponding path.

Since it was not possible for us to use default model training function of the Keras API, we implement Algorithm 1 from scratch in the Tensorflow. It is important to remember that we initialize all parameters to zero expect for the (ϕ, θ) values which acquired in the previous stage. Furthermore we set the learning rate of 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

Appendix B
Network Architectures

1) *Mutual Information Estimation:*

For all experiments in this paper, we report estimation of MI between the released representation and sensitive attribute, i.e., $\mathrm{I}(\mathbf{S};\mathbf{Z})$, as well as the MI between the released representation and utility attribute, i.e., $\mathrm{I}(\mathbf{U};\mathbf{Z})$. To approximate mutual information, we employed the MINE model [29]. The architecture of the model is depicted in I. Note that MINE's network for estimating $\mathrm{I}(\mathbf{S};\mathbf{Z})$ has the same architecture as shown in Table I.

2) *Colored-MNIST:*

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

3) *CelebA:*

In this experiment, we considered three scenarios for data utility and privacy leakage evaluation. These setups are shown in Table III. Note that all of the utility and sensitive attributes are binary data. The architecture of networks are given in Table IV.

MINE $\mathrm{I}(\mathbf{U};\mathbf{Z})$
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 I: The architecture of MINE network for mutual information estimation.

Encoder f_ϕ
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_θ
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_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($\lvert\mathcal{U}\rvert \times 8$), BN, LeakyReLU
FC($\lvert\mathcal{U}\rvert \times 8$), BN, LeakyReLU
FC(1), Sigmoid

TABLE II: The architecture of networks for the Colored-MNIST experiments.

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

TABLE III: The considered scenarios for the CelebA experiments.

Encoder $f_{\boldsymbol{\phi}}$
Input $\mathbf{x} \in \mathbb{R}^{64\times 64\times 3}$ 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 $g_{\boldsymbol{\theta}}$
Input $\mathbf{z} \in \mathbb{R}^{d_z}$ Code
FC(d_z), 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($\lvert\mathcal{U}\rvert \times 4$), BN, LeakyReLU
FC($\lvert\mathcal{U}\rvert$), BN, LeakyReLU
FC(1), Sigmoid

TABLE IV: The architecture of networks for the CelebA experiments.