InfoCensor: An Information-Theoretic Framework against Sensitive Attribute Inference and Demographic Disparity

Tianhang Zheng  
University of Toronto  
th.zheng@mail.utoronto.ca

Baochun Li  
University of Toronto  
bli@ece.toronto.edu

ABSTRACT
Deep learning sits at the forefront of many on-going advances in a variety of learning tasks. Despite its supremacy in accuracy under benign environments, Deep learning suffers from adversarial vulnerability and privacy leakage (e.g., sensitive attribute inference) in adversarial environments. Also, many deep learning systems exhibit discriminatory behaviors against certain groups of subjects (e.g., demographic disparity). In this paper, we propose a unified information-theoretic framework to defend against sensitive attribute inference and mitigate demographic disparity in deep learning for the model partitioning scenario, by minimizing two mutual information terms. We prove that as one mutual information term decreases, an upper bound on the chance for any adversary to infer the sensitive attribute from model representations will decrease. Also, the extent of demographic disparity is bounded by the other mutual information term. Since direct optimization on the mutual information is intractable, we also propose a tractable Gaussian mixture based method and a gumbel-softmax trick based method for estimating the two mutual information terms. Extensive evaluations in a variety of application domains, including computer vision and natural language processing, demonstrate our framework’s overall better performance than the existing baselines.

1 INTRODUCTION
Aided by big data and over-parameterization, modern deep learning techniques can achieve excellent performance in many learning tasks under benign environments. Therefore, at a rapid pace, deep learning is being deployed to diversified real-world applications, including some privacy or fairness-critical ones [24]. For privacy-critical applications, there is a growing concern that deep learning would be a primary source of user privacy leakage in the future, and forestalling privacy leakage from deep learning models is challenging, partially due to the models’ black-box nature. To mitigate some privacy concerns in deep learning, the community has developed several distributed machine learning schemes, such as model partitioning and federated learning [22, 27], where the clients send representations or model updates instead of the raw data to the cloud server. Although these newly emerging schemes address the direct privacy leakage from the raw data, they cannot prevent indirect privacy leakage from the representations and model updates. In particular, recent work shows that the potential adversary, with the access to the representations, can infer the associated sensitive attributes such as gender and race [32, 33]. Although the community already devoted some efforts into developing defenses to censor the sensitive attributes [9, 28, 36], [33] shows that the existing defenses, which derive from either adversarial learning or variational auto-encoding, are not very effective against sensitive attribute inference.

Beyond privacy concerns, deep learning also suffers from substantial group fairness issues. Due to data inadequacy or bias, many machine learning and deep learning models exhibit discriminatory behaviors against certain groups of subjects. In this regard, the community proposed several approaches to mitigating demographic (statistical) disparity in machine learning and deep learning. The existing approaches that are applicable to deep neural networks are mainly developed on the framework of variational auto-encoding [7, 23, 26, 28]. To our knowledge, [28] is the representative existing work that employs mutual information as the main objective for mitigating demographic disparity. However, [28] does not consider the mutual information with direct influence on demographic parity, i.e., the mutual information between model predictions and the sensitive attribute. Instead, [28] proposes to minimize a variational bound on the mutual information between the model representations and the sensitive attribute, which is a sub-optimal strategy for mitigating demographic disparity.

In this paper, we propose an information-theoretic framework, namely InfoCensor, to defend against sensitive attribute inference and mitigate demographic disparity in deep learning. As model partitioning, the system model of InfoCensor includes a feature extraction network and a prediction network. The feature extraction

CCS CONCEPTS
• Mathematics of computing → Information theory; • Computing methodologies → Artificial intelligence; • Security and privacy → Privacy protections.

KEYWORDS
Information Theory; Attribute Inference; Demographic Disparity

ACM Reference Format:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ASIA CCS ’22, May 30–June 3, 2022, Nagasaki, Japan  
© 2022 Association for Computing Machinery.  
ACM ISBN 978-1-4503-9140-5/22/05  . . $15.00  
https://doi.org/10.1145/3488932.3517402
network (also called encoder) is used for learning the representations, and the prediction network is used for performing the original task. With the support of our theoretical analysis, InfoCensor randomizes the representations with parameterized Gaussian mechanisms in both training and inference stages, and learns the parameterized Gaussian mechanisms by minimizing two mutual information terms along with the original task loss.

In terms of sensitive attribute inference, we prove that an upper bound on the chance for an arbitrary adversary to infer the sensitive attribute will drop off, as the mutual information between the representations and the attribute $I(z, s)$ decreases. Based on this theoretical result, InfoCensor mitigates the threat of sensitive attribute inference from the representations by minimizing $I(z, s)$. To mitigate demographic disparity (w.r.t. a sensitive attribute $s$), InfoCensor minimizes the mutual information between model predictions and the sensitive attribute $I(\hat{y}, s)$. We show that $I(\hat{y}, s)$ has a direct influence on demographic parity since the extent of demographic disparity is bounded by $I(\hat{y}, s)$.

All in all, our proposed objective under InfoCensor mainly incorporates $I(z, s)$, $I(\hat{y}, s)$, and the original task loss, with two hyperparameters balancing their effects. Since $I(z, s)$ is analytically intractable, we first propose a Gaussian mixture based method for estimating $I(z, s)$, provided that $p(z|x)$ is a Gaussian distribution under InfoCensor. Also, direct optimization on $I(\hat{y}, s)$ is intractable, thus we propose a gumbel-trick based method to estimate $I(\hat{y}, s)$, which enables feasible backpropagation on $I(\hat{y}, s)$. In contrast to some previous works [21, 28], our estimation methods do not need to train any additional neural network beyond the feature extraction network and the prediction network for estimating the mutual information, thus the estimations’ quality does not rely on the modeling performance or training stability of any additional networks.

We demonstrate the effectiveness of InfoCensor by extensive evaluations in different application domains. In particular, we conduct experiments on the Health Heritage dataset, the UTKface dataset, and a Twitter dataset. These datasets across different application domains represent diverse data formats—vectors, images, and text. For each dataset, InfoCensor learns the randomized representations with commonly-used neural networks, including multi-layer perceptrons (MLP), convolutional neural networks (CNN), and long short-term memory networks (LSTM). We compare InfoCensor with adversarial training [9, 36], the VAE-based information-theoretic method [28] (evaluated in [33]), and TIPRDC [21] on all the aforementioned datasets and networks. The experimental results demonstrate the superiority of InfoCensor over those baselines.

Our contributions are summarized as follows:

1. We propose an information-theoretical framework, namely InfoCensor, to defend against sensitive attribute inference and mitigate demographic disparity.

2. We establish theoretical links between sensitive attribute inference, demographic parity, and mutual information, proving that our research objective can be achieved by minimizing two mutual information terms.

3. We propose a Gaussian mixture based method and a gumbel-trick based method to estimate and optimize the two mutual information terms, without Monte Carlo sampling or training any additional neural networks.

4. We conduct an array of experiments in varied applications, including computer vision and natural language processing, which demonstrate the superior performance of InfoCensor, compared to the existing baselines.

### 2 BACKGROUND AND RELATED WORK

#### 2.1 Definitions and Notations

In this paper, we denote a data sample and its original task output by $x$ and $y$, respectively. For a classification problem, the original task output is the ground-truth label. We further denote the sensitive attribute (variable) by $s$, distributed on an alphabet $S$. Here a sensitive attribute refers to any discrete attribute that should be protected in hiring, medical, financial, real estate decisions, etc, including but not limited to identity, gender, race, etc. We refer to a value of $s$ as a sensitive class, e.g., male or female. The system model trained under InfoCensor consists of two neural networks—one for extracting representations and one for performing the original prediction task based on the representations. We denote the feature extraction network by $F_{\theta}(\cdot)$ with parameters $\theta$ and the prediction network by $f_{\phi}(\cdot)$ with parameters $\phi$. We denote the prediction by $\hat{y} = \text{argmax}_{y} f_{\phi}(x)$. We denote the probability distribution of $x$ by $p(x)$ and the mutual information between $x$ and $s$ by $I(x; s)$. For other variables, we adopt similar denotations for their probability distribution and mutual information. The KL divergence between two probability distributions $p$ and $q$ is denoted by $\text{KL}(p||q)$.

#### 2.2 Privacy in Deep Learning

With increasing client data being engaged in the development of deep learning systems, there is a growing concern that deep learning will be a primary source of privacy leakage. Many recent works have investigated and confirmed privacy concerns by showing that different levels of private information might be leaked from deep learning models under different threat models and attacks.

The existing attacks that can cause privacy leakage include membership inference, reconstruction attack, sensitive attribute (variable) inference, model extraction, etc. Membership inference attacks aim to determine whether a data sample $x$ belongs to the training set based on the model prediction on $x$ [31]. Reconstruction attacks, which might also be referred to as model inversion attacks, aim to reconstruct the training data or representative data based on different levels of information from the models, including model parameters, model predictions, and model gradients [12, 39]. Sensitive attribute inference attacks attempt to infer sensitive attributes such as gender and race from the representations (embeddings) of deep learning models [32, 33]. Model extraction usually refers to learning a substitute model that behaves similarly to the adversary-targeted model with query access to the adversary-targeted model [14, 29, 34]. In this paper, we mainly focus on addressing sensitive attribute inference attacks in the inference stage of the model partitioning scenario [33] by InfoCensor, and we detail two commonly-used sensitive attribute inference attacks for evaluating InfoCensor in Section 3.3.

---

The code is publicly available at https://github.com/iQua/InfoCensor.
2.3 Group Fairness in Deep Learning

Many learning systems have demonstrated much or less discrimination against certain groups of subjects, leading to one group being deprived of benefits or opportunities if the systems are deployed for applications such as resource allocation or qualification. There are many ways to define and quantify the group fairness. In this paper, we mainly focus on the concept of demographic (statistical) parity [4] to investigate group fairness. Demographic parity is one of the most well-known definitions for studying group fairness. We leave the research on connecting information theory and the other definitions for future works. Under the definition of demographic parity, a predictor satisfies demographic parity (or we can say a predictor is unbiased) w.r.t. a sensitive attribute if the prediction is independent of the sensitive attribute, i.e.,

\[ p(\hat{y}|s) = p(\hat{y}). \] (1)

We note that an equivalent representation for (1) is

\[ KL(p(\hat{y}|s) || p(\hat{y})) = 0 \] (2)

In this paper, demographic disparity means that the predictions are unequally distributed w.r.t. \( s \), i.e., (severe) violation of (1). Under the definition of demographic parity, we can quantify the extent of group unfairness of a model by its deviation from demographic parity, measured by statistical parity difference (SPD), i.e., \( p(\hat{y}|s = 0) - p(\hat{y}|s = 1) \), especially when \( s \) is a binary attribute. Except for SPD, we can also use the ratio between the two conditional probability terms to as the measure of fairness, inducing the definition of \( p\%-\text{rule} \). There are some other measurements of fairness such as “decision boundary fairness” in [37]. However, this measurement is hardly applicable to deep neural networks since it needs to compute the distance between the data and the decision boundary, which is non-trivial for deep neural networks.

2.4 Related Work

Research on sensitive attribute inference is attracting increasing attention from the community due to growing concerns about privacy leakage from deep learning models [32, 33]. To defend against sensitive attribute inference, [6, 9, 13, 36] proposed to censor the sensitive attribute from the model representations based on adversarial training. The core idea of those works is involving a discriminator to infer the sensitive attribute from the representations and training the encoder to minimize the discriminator’s success. From the information-theoretic perspective, [28] proposed to train a variational auto-encoder (VAE) to censor a sensitive attribute by minimizing a variational bound on the mutual information between the representations and the sensitive attribute. [21] is another recent work for defending against attribute inference based on adversarial training and information theory. We detail the differences between our work and [21, 28] in Section 2.5.

In terms of group fairness in deep learning, [38] first proposed to learn fair representations by minimizing absolute SPD on the representations. However, the method proposed by [38] is only applicable to clustering models and binary sensitive variables. [23] modified VAE to produce fair representations with the help of maximum mean discrepancy (MMD) regularization. As mentioned above, the adversarial training methods proposed in [9, 26, 36] and the VAE-based method proposed in [28] can be used for training (group) fair deep learning models. We note that the performance of [23] is inferior to that of [28], so we only include [28] in the comparison with our framework.

2.5 Detailed Comparison with Previous Information Theory Related Methods

To our best knowledge, there are two well-known defensive methods against attribute inference related to information theory, i.e., VAE-based method [28] (NeurIPS18) and TIPRDC [21] (KDD20). InfoCensor differs from those two methods in the following aspects: First, both [28] & [21] lack theoretical bounds to guarantee the defensive performance. In contrast, we provide information-theoretic bounds on inference attack accuracy and demographic disparity. Second, we randomize the representations in both training and inference stages, while both [28] & [21] use deterministic representations in the inference stage. Without randomization in the inference stage, the encoder (feature extraction network) will degrade into a deterministic neural network, probably leading to larger mutual information \( I(\hat{y};x) \) becomes infinite with continuous \( x \) and inferior defensive performance. Third, both [28] & [21] need to train additional neural networks for estimating mutual information, thus, the estimation quality highly relies on the training stability and model performance of the additional neural networks (e.g., The mini-max game based estimation method used in [21] may suffer from training stability issues in practice). In contrast, our analytical estimation methods do not need to train any additional networks beyond the feature extractor and the classifier for estimating \( I(\hat{y};z) \). In fact, TIPRDC [21] has similar defensive performance as adversarial training. This is because, for the two terms in the objective Eq. 18 in [21], the first term Eq. 15 in [21] is the same as the max-min adversary loss. [21] also mentions that the first term is an adversarial training objective in Section 3 in [21]. The main effect of the second term \( I_{JSD}(\hat{x}, z, \hat{y}) \) is not defending against attribute inference but retaining more information of \( x \) according to Section 4.3 in [21]. Consequently, [21] achieves similar defensive performance as adversarial training.

3 PROBLEM FORMULATION

In this paper, we consider sensitive attribute inference and demographic parity. Our research objective is to defend against sensitive attribute inference attacks on model representations and mitigate demographic disparity in the inference stage of the model partitioning scenario [33]. To achieve this goal, we propose InfoCensor to randomize the model representations and minimize two mutual information terms along with the original task loss. In the following, we detail our threat model and the potential sensitive attribute inference attacks that we attempt to address with InfoCensor.

3.1 System Model

As model partitioning [5, 19, 22, 33], the proposed system consists of two models, a local feature extraction model (encoder) and a prediction model. In the inference stage, the feature extraction model is held by the client, and the prediction model is distributed at the server side as in [5, 33]. In practice, the models can be trained at a trusted cloud provider [5], which can be leased by the client for a certain amount of training time. Another choice is that the
According to [11], this pipeline training process is functionally equivalent to centralized training on a single device with the same data loading order and the same initial model weights. Eventually, in the long-term inference stage, the feature extraction model will be distributed at the client side, and the prediction model will be distributed at the server side [5, 33], as shown in Fig. 1.

### 3.2 Threat Model

Our threat model mainly considers the threat of sensitive attribute inference and group fairness in the inference stage of the model partitioning scenario. In terms of sensitive attribute inference, our threat model is similar to the threat model in [33] (ICLR20). Specifically, we assume the adversary has access to the representations and/or query access to \( F_\theta(x) \). We assume the server is honest-but-curious, which means the server will cooperate with the clients to train the prediction model but may want to infer sensitive attribute from the representations in the inference stage [33]. Actually, in the training stage, the clients can detect if the server misleads the feature extraction model to leak more information regarding \( s \) from \( z \). This is because, in such case, \( I(z; s) \) will be a large value in the training stage, according to Theorem 4.1. We note that this work mainly focuses on addressing attribute inference attacks in the inference stage [33] and leaves detecting a malicious server in the training stage for future work. Also, as the threat model of [33], we assume the adversary has access to an auxiliary dataset for training the attack model. In the experiments, we evaluate InfoCensor against the basic inference attack [32] (CCS20) and the state-of-the-art de-censoring attack [33] and leaves detecting a malicious server in the training stage [33] and leaves detecting a malicious server in the training stage [33].

In terms of group fairness, we mainly consider the concept of demographic disparity in this paper. Data inadequacy and bias are the common causes of discriminatory behaviors of the modern learning systems. Data inadequacy refers to lack of data from certain subgroups, due to the small sizes of the subgroups or inadequate data collection. Even if the data is sufficient to represent every subgroup, the data itself might still reflect historical and inherent prejudices. Our threat model assumes that the datasets might have the issue of data inadequacy (not severe) or bias, which is the nature of many real-world datasets, including the datasets used in the experiments. Our goal is to mitigate demographic disparity with modest sacrifice of the performance on the original task.

### 3.3 Sensitive Attribute Inference Attacks

Note that in the following two inference attacks, we assume the adversary has an auxiliary dataset \( D_{aux} = \{x_j, s_j\} \) and the access to model representations to train its attack model.

#### Basic Sensitive Attribute Inference

Under our threat model, the potential adversary might directly infer sensitive attributes from the model representations, and we call it “basic inference attack” in this paper. To conduct the basic inference attack, the adversary learns an attack model \( g_{\phi_{adv}}(\cdot) \) on \( \{F_\theta(x_m), s_m\}_{m=1}^M \), expecting that \( \arg\max g_{\phi_{adv}}(F_\theta(x_m)) = s_m \). Here we assume the adversary optimizes \( g_{\phi_{adv}} \) by minimizing the cross-entropy between \( g_{\phi_{adv}}(F_\theta(x)) \) and \( s \) using the Adam optimizer on the auxiliary dataset. After training the attack model, the adversary infers \( \hat{s} \) from \( z \) by \( \hat{s} = \arg\max g_{\phi_{adv}}(z) \). We detail the algorithm of the basic inference attack in Alg. 1.

#### De-censoring Attack

Under our threat model, the potential adversary might also conduct an advanced de-censoring attack, proposed in [33], to infer \( s \) from \( z \). The idea of the de-censoring method is to transform the representations into a different form that might leak more information regarding \( s \). To conduct the de-censoring attack, the adversary first learns an auxiliary model on \( D_{aux} \) with an auxiliary feature extraction network (encoder) \( F_{\theta_{aux}}(\cdot) \) and an auxiliary prediction network \( \phi_{adv}(\cdot) \), by minimizing the cross-entropy between \( F_{\theta_{aux}}(F_\theta(x)) \) and \( s_{aux} \). Next, the adversary learns a transform model \( T_{\xi}(\cdot) \) to transform \( z_{aux} = F_{\theta_{aux}}(x_m) \) into \( F_{\theta_{aux}}(x_m) \) by minimizing the mean squared error, i.e., \( \| T_{\xi}(z) - F_{\theta_{aux}}(x_m) \|^2_2 \). Apparently, \( F_{\theta_{aux}}(x) \) leaks more information regarding \( s \), and the adversary expects to transform \( z \) into the form of \( F_{\theta_{aux}}(x) \) using \( T_{\xi}(\cdot) \). The adversary then trains attack model \( g_{\phi_{adv}}(\cdot) \) on \( \{T_{\xi}(x_m), s_m\}_{m=1}^M \). After all the above steps, the adversary can infer \( \hat{s} \) from \( \hat{z} \) by \( \hat{s} = \arg\max g_{\phi_{adv}}(T_{\xi}(z)) \). We refer the interested readers to [33] for more details about this de-censoring attack. Note that the de-censoring method usually outperforms the basic inference attack on deterministic representations.

### 3.4 A Potential Measurement of Group Fairness

The previous literature utilizes SPD to quantify group fairness (under the concept of demographic parity), which is a suitable measurement for binary sensitive attributes. In a more general setting, we propose \( I(\hat{y}; s) \) as a potential measurement for measuring group fairness, with a uniform prior assumption on \( p(s) \). We assume

\[
I(\hat{y}; s) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{x_m \sim D_{aux}} \left[ \mathbb{I}(\hat{s}_m \neq s_m) \right]
\]

\[
\mathbb{I}(\cdot)
\]

For other users or the server to use the model in the inference stage.

---

**Algorithm 1 Basic Sensitive Attribute Inference**

**Require:** Auxiliary dataset \( D_{aux} = \{x_m, s_m\}_{m=1}^M \); target encoder \( F_\theta(\cdot) \); attack model \( g_{\phi_{adv}}(\cdot) \);

1. Collect \( \{F_\theta(x_m), s_m\}_{m=1}^M \) with the access to \( F_\theta(\cdot) \)
2. Train \( g_{\phi_{adv}}(\cdot) \) on \( \{F_\theta(x_m), s_m\}_{m=1}^M \) to minimize the cross-entropy between \( g_{\phi_{adv}}(F_\theta(x_m)) \) and \( s_m \)
that no adversary is involved in computing the measurement, otherwise, an adversary can manipulate any existing fairness measurement. We note that the real-world priors may suffer from limitations. In the following, we will introduce and prove our theoretical results.

4 MAIN FRAMEWORK

4.1 Framework Overview

To defend against sensitive attribute inference and mitigate demographic disparity, we propose a generic objective to minimize two mutual information terms along with the original task loss. Our proposed objective can be formulated as

$$\min_{\theta, \phi} \mathcal{L}(f_\theta(z), y) + \lambda|l(z; s) + \kappa l(\hat{y}; s)$$

st. $z = f_\theta(x), \hat{y} = \text{argmax} f_\phi(z)$

where $F_\theta$ outputs a randomized representation rather than a deterministic one, i.e., $z \sim \mathcal{N}(\mu_0(x), \Sigma_0(x))$ (similar to a variational encoder [17]), as shown in Fig. 2. The main reason for randomizing the representations is that in deterministic neural networks with strictly monotone nonlinearities, certain mutual information term such as $l(x; z)$ is probably an infinite value or a constant [10]. $\mathcal{L}(f_\phi(z), y)$ refers to the loss customized for solving the original task. For a classification task, we can define the loss as the cross-entropy between the logit output $f_\phi(z)$ and the ground-truth label $y$. If the original task is an unsupervised task without task output, we might replace $\mathcal{L}(f_\phi(z), y)$ with an unsupervised loss, e.g., reconstruction loss. In this paper, we mainly focus on the classification task. $I(z; s)$ refers to the mutual information between the representations $z$ and the sensitive attribute $s$. We minimize $I(z; s)$ because as $I(z; s)$ decreases, it is more difficult for the adversary to infer the sensitive attribute from the representation (see Theorem 4.1 & 4.2 for more details). $I(\hat{y}; s)$ refers to the mutual information between the model prediction $\hat{y}$ and the sensitive variable $s$. We minimize $I(\hat{y}; s)$ since the extent of demographic disparity can be bounded by $I(\hat{y}; s)$ (see Theorem 4.4 for details). $\lambda$ and $\kappa$ are the hyperparameters that balance utility, resistance against sensitive attribute inference, and group fairness. In practice, the concrete settings of $\lambda$ and $\kappa$ may depend on the task and users’ requirements. In the following, we will introduce and prove our theoretical results.

4.2 Theoretical Connections

Sensitive Attribute Inference and $I(z; s)$. The following theorems indicate that, as the mutual information between the model representation $z$ and a sensitive attribute $s$ decreases, the sensitive attribute is more difficult to be inferred (by any adversary) from the representations.

**Theorem 4.1.** Let $s$ represents a sensitive attribute (variable) uniformly distributed on a finite alphabet $S$, $x$ and $z$ respectively refer to the corresponding data sample and the representation learned by an encoder. The chance for any adversary to (correctly) infer $s$ can be upper-bounded by

$$P[\hat{s} = s] \leq \frac{I(z; s) + \log 2}{\log |S|},$$

where $\hat{s}$ is the inferred attribute based on $z$.

**Proof of Theorem 4.1.** The proof is based on Fano’s inequality.
Lemma 1 (Fano’s Inequality). Let $s$ be a discrete random variable on a finite alphabet $S$. Let $\hat{s}$ be an estimate of $s$, then we have (Theorem 1 in [30])

$$H(s|\hat{s}) \leq H(P_e) + P_e \log(|S| - 1),$$

(6)

where $P_e$ refers to the error probability $P_e = \mathbb{P}[\hat{s} \neq s]$, and $H(\cdot)$ refers to the entropy function. $I(s; \hat{s}) = H(s) - H(s|\hat{s})$.

Based on the above lemma, we have

$$I(s; \hat{s}) \geq H(s) - H(P_e) - P_e \log(|S| - 1)$$

(7)

Since $H(P_e) \leq \log 2$, we have

$$P_e \log |S| \geq P_e \log(|S| - 1) \geq H(s) - I(s; \hat{s}) - \log 2$$

(8)

Since $z \sim N(\mu_0(x), \Sigma_0(x))$, $\hat{s}$ is inferred from $z$, and $x$ has all the information of $s$, we can have a Markov chain: $s \rightarrow x \rightarrow z \rightarrow \hat{s}$ [40]. Thus, $I(s; \hat{s}) \leq I(s; z)$. Then we have

$$\mathbb{P}[\hat{s} \neq s] = P_e \geq \frac{H(s)}{\log |S|} - \frac{I(s; \hat{s}) + \log 2}{\log |S|}$$

(9)

If $s$ is uniformly distributed over $S$, we have $H(s) = \log |S|$. Since $\mathbb{P}[\hat{s} = s] = 1 - \mathbb{P}[\hat{s} \neq s]$, we have Theorem 4.1. □

Remark 1. If $s$ is not uniformly distributed on a finite alphabet $S$, then the bound is

$$\mathbb{P}[\hat{s} = s] \leq 1 - \frac{H(s)}{\log |S|} + \frac{I(s; z) + \log 2}{\log |S|}$$

Moreover, we have a tighter bound for binary sensitive attributes as follows:

Theorem 4.2. Let $s$ represent a binary sensitive attribute with a uniform prior, $x$ and $z$ respectively refer to the corresponding data sample and the representation learned by an encoder. The chance for any adversary to (correctly) infer $s$ can be upper-bounded by

$$P[\hat{s} = s] \leq \frac{1 + \sqrt{1 - (1 - \frac{I(s; z)}{\log 2})^2}}{2},$$

(10)

where $\hat{s}$ is the inferred attribute based on $z$.

Proof of Theorem 4.2. The proof is also based on Fano’s inequality. Since $|S| = 2$ for a binary attribute, according to Fano’s inequality, we have $H(s|\hat{s}) \leq H(P_e)$. Let $g(t) = -t \log t$, we then have the derivative of $H^2(P_e)$ w.r.t. $P_e$ is

$$ \left( \frac{H^2(P_e)}{P_e(1 - P_e)} \right)' = \frac{g^2(P_e) - g^2(1 - P_e)}{P_e^2 (1 - P_e)^2}. $$

(11)

Since $g^2(P_e) > g^2(1 - P_e)$ when $P_e \in (0, \frac{1}{2})$, $\frac{H^2(P_e)}{P_e(1 - P_e)}$ is an increasing function when $P_e \in (0, \frac{1}{2})$. On the other hand, $g^2(P_e) < g^2(1 - P_e)$ when $P_e \in (\frac{1}{2}, 1)$, thus $\frac{H^2(P_e)}{P_e(1 - P_e)}$ is a decreasing function when $P_e \in (\frac{1}{2}, 1)$. The maximum of $\frac{H^2(P_e)}{P_e(1 - P_e)}$ is achieved at $P_e = \frac{1}{2}$, i.e., $\frac{H^2(P_e)}{P_e(1 - P_e)} \leq 4(\log 2)^2$. Thus, we have

$$H^2(P_e) \leq 4(\log 2)^2 P_e(1 - P_e).$$

(12)

Since $H(P_e) \geq H(s|\hat{s}) = H(s) - I(s; \hat{s}) \geq 0.4(\log 2)^2 P_e(1 - P_e) \geq (H(s) - I(s; \hat{s}))^2$. Due to the Markov chain $s \rightarrow x \rightarrow z \rightarrow \hat{s}$ [40], $I(s; \hat{s}) \leq I(s; z)$, and thus $H(s) - I(s; \hat{s}) \geq H(s) - I(s; z) = H(s|z) \geq 0^7$. Therefore, we have $4(\log 2)^2 P_e(1 - P_e) \geq (H(s) - I(s; z))^2$, i.e.,

$$P_e^2 - P_e + \frac{(H(s) - I(s; z))^2}{2 \log 2} \leq 0.$$ 

(13)

With a uniform prior assumption on $H(s)$ ($H(s) = \log 2$), we have

$$P_e^2 - P_e + \frac{1}{2} \frac{(I(s; z))^2}{2 \log 2} \leq 0.$$ 

(14)

According to the quadratic root formula, we have

$$P_e \geq \frac{1 - \sqrt{1 - (1 - \frac{(I(s; z))^2}{\log 2})^2}}{2}$$

(15)

Since $P[\hat{s} = s] = 1 - P_e$, we have

$$P[\hat{s} = s] \leq \frac{1 + \sqrt{1 - (1 - \frac{(I(s; z))^2}{\log 2})^2}}{2}$$

(16)

□

Based on Theorem 4.2, we have the following corollary:

Corollary 1. If $I(z; s)$ is minimized to 0, then the adversary can do better than random guess to infer the binary attribute $s$ based on $z$ under the uniform prior assumption.

Proof. If $I(z; s) = 0$, the quadratic inequality (14) in the proof of Theorem 4.2 is

$$P_e^2 - P_e + \frac{1}{4} = (P_e - \frac{1}{2})^2 \leq 0.$$ 

(17)

So we have $P_e = \frac{1}{2}$, and $P[\hat{s} = s] = \frac{1}{2}$. With the uniform prior assumption on the binary attribute $s$, this probability is equal to the probability of correct random guess. □

Remark 2. If $s$ is not a uniformly distributed binary attribute, then the bound is

$$P[\hat{s} = s] \leq \frac{1 + \sqrt{1 - (H(s) - I(s; \hat{s}))^2}}{2 \log 2}.$$ 

(18)

Theorem 4.1 & 4.2 indicate that, as $I(z; s)$ decreases, then the upper bound on the chance for any adversary to infer the sensitive attribute will also decrease. Inspired by Theorem 4.1 & 4.2, we propose to defend against sensitive attribute inference by minimizing $I(z; s)$. Since $I(z; s)$ is analytically intractable, we propose a tractable Gaussian mixture based method for estimating $I(z; s)$ (see Section 4.3). Note that in the training stage, we assume the prior $p(s)$ to be a (discrete) uniform distribution on $S$ when estimating $I(z; s)$. We do not use this assumption for estimating $I(z; s)$ in the inference stage. Also, we do not enforce this assumption on the auxiliary dataset owned by the adversary.\(^1\) This assumption might not characterize the true prior distribution of $s$ but accords with the goal to make $s$ difficult to be inferred, i.e., maximizing the uncertainty of $s$ in the model training process.\(^1\)

\(^1\)Conditional entropy of a discrete variable $s$ is non-negative.

\(^2\)The priors of the auxiliary datasets can be not uniform in the experiments.

\(^3\)A discrete uniform distribution has the maximum entropy.
Sensitive Attribute Inference and $I(x; z)$. In practice, we observe that, when $|S|$ is large, the size of the data in a minibatch is insufficient for approximating the distribution $p(z|s)$, which is used for estimating $I(z; s)$. For instance, if the number of the possible values of $s$ is larger than 20 (e.g., user identity in the Twitter dataset), and the batch size is 128, then the sample size for estimating $p(z|s)$ is smaller than 6. Such a small sample size is insufficient for estimating $p(z|s)$ with the dimension of $s$ being 64 or 128.

In such cases, we could involve $I(x; z)$ with a small coefficient as a regularizer in the objective, which can slightly improve the performance of InfoCensor. This is because (i) $I(x; z)$ is an upper bound on $I(z; s)$ due to the Markov chain $s \rightarrow x \rightarrow z \rightarrow s$ [40]; (ii) We could obtain a more accurate estimation for $I(x; z)$ than $I(z; s)$ with a minibatch of samples. Note that in the training stage, we need estimate $p(z|x)$ and $p(z)$ for approximating $I(x; z)$. Different from $p(z|s)$, $p(z|x)$ has an analytic form under InfoCensor (i.e., Gaussian distribution), and $p(z)$ can be approximated by all the samples in the minibatch. Thus, we could usually obtain a better estimation for $I(x; z)$ than $I(z; s)$ with a minibatch of training samples.

However, if $|S|$ is not large, involving $I(x; z)$ as a regularizer leads to a suboptimal solution since minimizing $I(x; z)$ censors all the information, not only the information regarding $s$, to a suboptimal solution since minimizing $I(x; z)$ is an upper bound on $I(\hat{y}; s)$. Different from $F(\cdot, \cdot)$, $F(\cdot, \cdot)$ is a deterministic function. Since $\hat{y} = f(\cdot)$, we have $I(\hat{y}; s) \leq I(z; s)$. On the other hand, we observe that minimizing $I(\hat{y}; s)$ also somehow helps defend against sensitive attribute inference attacks in practice. This is because $\hat{y}$ contains part of the information in $z$, and thus, minimizing $I(\hat{y}; s)$ also helps weaken the association between $z$ and $s$.

**Remark 3.** Some readers may think minimizing $I(\hat{y}; s)$ is not necessary, but we note that the estimation for $I(\hat{y}; s)$ is usually more accurate than the estimation for $I(z; s)$ due to the relatively low dimension of $\hat{y}$, compared to $z$. Also, $I(\hat{y}; s)$ has a more direct influence on demographic parity.

4.3 Mutual Information Estimation

In this section, we introduce how to use a Gaussian mixture based method for estimating $I(z; s)$, and how to use a gumbel-softmax trick based method for estimating $I(\hat{y}; s)$. Note that there are many methods for estimating mutual information. However, the estimations acquired by Monte Carlo or KNN-based methods [18] are either not differentiable or not very easy to differentiate and optimize. The methods proposed in [21, 28] need to train additional neural networks. Here we propose analytical estimations for $I(z; s)$ and $I(\hat{y}; s)$. Compared to the previous estimation methods, our estimation methods enable feasible backpropagation on the estimated $I(z; s)$ and $I(\hat{y}; s)$, without training any additional networks.

Gaussian Mixture based Estimation. The mutual information between $z$ and $s$ can be expressed as

$$I(z; s) = E_s [KL(p(z|s)p(z))]$$

Since the true $p(z|s)$ and $p(z)$ are intractable, we propose to approximate both distributions by Gaussian mixtures. For any $s$, we approximate $p(z|s)$ by $p(z|s) \approx \frac{1}{N} \sum_{i=1}^{N} p(z|x_i)$, where $\{x_i\}_{i=1}^{N}$ denotes the samples with one sensitive class. Likewise, we can approximate $p(z)$ by $\frac{1}{N} \sum_{i=1}^{N} p(z|x_i)$. Recall that, under InfoCensor, each $p(z|x_i)$ is a Gaussian distribution with mean $\mu(x_i)$ and covariance $\Sigma(x_i)$, so the approximations of $p(z|s)$ and $p(z)$ are Gaussian mixtures.
According to (22), we only need to compute the KL divergence between two Gaussian mixtures. Unfortunately, the KL divergence between two Gaussian mixtures is also not analytically tractable. Thus, in the following, we propose to use a variational method to estimate the KL divergence. We introduce two variational variables $a_{n,i} \geq 0$ and $\beta_{n,i} \geq 0$, such that $\sum_n a_{n,i} = 1/N$ and $\sum_i \beta_{n,i} = 1/N$. We then have $p(z|s) \approx \sum_n a_{n,i} p(z|x_i)$ and $p(z) \approx \sum_i \beta_{n,i} p(z|x_i)$.

\[
KL(p(z|s)|p(z)) = -\int p(z|s) \log \frac{p(z|s)}{p(z)} dz \\
= -\int p(z|s) \log \frac{\beta_{n,i} p(z|x_i)}{a_{n,i} p(z|x_i)} dz
\]

The above inequality is derived based on Jensen’s inequality: Since $-\log x$ is a convex function, we have

\[
-\log \sum_{i,n} \frac{\beta_{n,i} p(z|x_i)}{a_{n,i} p(z|x_i)} \leq -\sum_{i,n} a_{n,i} p(z|x_i) \log \frac{\beta_{n,i} p(z|x_i)}{a_{n,i} p(z|x_i)}
\]

KL($p(z|x_i)|p(z|x_n)$) is the KL divergence between two Gaussian distributions. According to [17], the KL divergence between two Gaussian distributions has a simple analytical solution, which is easy to differentiate and optimize. We refer the interested readers to the appendix in [17] for more details. Given (23), the best approximation of KL($p(z|s)|p(z)$) is obtained by minimizing $\sum_n a_{n,i} \log \frac{a_{n,i}}{\beta_{n,i}} \sum_n a_{n,i} KL(p(z|x_i)|p(z|x_n))$ w.r.t. $a_{n,i}$ and $\beta_{n,i}$. Specifically, we optimize $a_{n,i}$ and $\beta_{n,i}$ by fixing one and updating the other with its optimal solution. In each iteration, we first fix $a_{n,i}$ and update $\beta_{n,i}$ by $\beta_{n,i} = \frac{\sum_n a_{n,i} \exp(-KL(p(z|x_i)|p(z|x_n)))}{\sum_n \sum_n a_{n,i} \exp(-KL(p(z|x_i)|p(z|x_n)))}$. Then we fix $\beta_{n,i}$ and update $a_{n,i}$ by $a_{n,i} = \frac{\beta_{n,i} \exp(-KL(p(z|x_i)|p(z|x_n)))}{\sum_n \sum_n a_{n,i} \exp(-KL(p(z|x_i)|p(z|x_n)))}$. We execute the above operations for several iterations and fix $a_{n,i}$ and $\beta_{n,i}$ to compute the approximation. Note that we fix $a_{n,i}$ and $\beta_{n,i}$ as constants (by detach() in PyTorch), when we optimize the model parameters, to avoid backpropagation through the above iterations. After fixing $a_{n,i}$ and $\beta_{n,i}$, we only need to compute the gradient of the KL divergence between two Gaussian distributions, i.e., $\nabla_{\theta} KL(p(z|x_i)|p(z|x_n))$, to optimize $\theta$.

For $I(x;z)$, we follow [28] to approximate it by

\[
I(x;z) \approx \frac{1}{N} \sum_{n=1}^{N} \frac{1}{N} \sum_{n'=1}^{N} KL(p(z|x_n)|p(z|x_{n'})).
\]

**Gumbel-Softmax Trick Based Estimation.** Finally, we propose a method using the gumbel-softmax trick [15] for estimating $I(\hat{y}; s)$.

Similar to (21), we can express $I(\hat{y}; s)$ as

\[
I(\hat{y}; s) = \mathbb{E}_{\tilde{s}}[KL(p(\hat{y}|s)|p(\hat{y}))].
\]

Note that both $p(\hat{y}|s)$ and $p(\hat{y})$ can be denoted by a vector, with each element representing the probability corresponding to a possible value of $\hat{y}$. We then estimate $p(\hat{y}|s)$ and $p(\hat{y})$ by $\frac{1}{N} \sum_n one_hot(\hat{y}_n)$ and $\frac{1}{N} \sum_i one_hot(\hat{y}_i)$, where $one_hot(\hat{y})$ denotes the one hot vector of $\hat{y}$. Specifically, if the prediction $\hat{y}$ corresponds to the $k$-th class, then the $k$-th element of $one_hot(\hat{y})$ is 1, and all the other elements are 0. We denote the $k$-th element of $one_hot(\hat{y})$ by $one_hot_k(\hat{y})$.

Since $one_hot(\cdot)$ is not differentiable, we propose to estimate it by the gumbel-softmax trick [15].

We define the predicted class probability vector as $\pi(x) \triangleq$ softmax($f_\phi(x)$). Gumbel-softmax trick estimates one-hot $p(\hat{y})$ by

\[
one_hot_k(\hat{y}) \approx \frac{\exp((\log(p_{\pi_k}(x)) + g_k)/\tau)}{\sum_k \exp((\log(p_{\pi_k}(x)) + g_k)/\tau)},
\]

where $g_k$ is randomly sampled from Gumbel(0, 1) [25], $p_{\pi_k}(x)$ is the $k$-th element of $\pi(x)$, and $\tau$ is the temperature. With (25) and a uniform prior assumption, we estimate $I(\hat{y}; s)$ by

\[
I(\hat{y}; s) \approx \frac{1}{N} \sum_{n=1}^{N} KL(p_{\pi_n}(x)|p(\hat{y})) \frac{1}{N} \sum_{n=1}^{N} one_hot_k(\hat{y}_n),
\]

where the one-hot vectors are estimated by the gumbel-softmax trick [15] (setting $\tau = 0.1$ in (25)), and then (26) can be differentiable and easy to optimize.

### 4.4 Algorithm Design

**Algorithm 3: InfoCensor**

**Require:** Training data $D = \{x_n, y_n, s_n\}_{n=1}^{N}$; feature extraction network $F_\theta$; prediction network $f_\phi$; surrogate loss $\mathcal{L}(f_\phi(z), y)$; number of iterations $T$

Initialize $\theta, \phi$

for $t = 1$ to $T$

1. Sample a minibatch from $D$ ($(x_n, y_n, s_n)_{n=1}^{N}$). The size of the minibatch is denoted by $N$. The size of samples with a specific $s$ in the minibatch is denoted by $N_s$.

2. Input the minibatch of the data samples $x_n$ into $F_\theta$ to acquire the mean and covariance matrix of the representations, i.e., $\mu_\theta(x_n)$ and $\Sigma_\theta(x_n)$.

3. Sample randomized representations $z_n \sim p(z|x_n)$, where $p(z|x_n) \triangleq N(\mu_\theta(x_n), \Sigma_\theta(x_n))$.

4. Input $z_n$ into $f_\phi$ to obtain the predictions $\hat{y}_n = f_\phi(z_n)$.

5. Update $\theta, \phi$ by the Adam optimizer to minimize (4).

end for

Output $F_\theta$ and $f_\phi$.

Our basic algorithm is detailed in Alg. 3. In each training iteration, our algorithm samples a minibatch of training data and inputs the data into the feature extraction network $F_\theta$ (encoder). The encoder outputs the mean and covariance matrix of the representations, i.e., $\mu_\theta(x)$ and $\Sigma_\theta(x)$. Under InfoCensor, $p(z|x_n) \triangleq N(\mu_\theta(x_n), \Sigma_\theta(x_n))$, so as $p(z|x)$. Given $p(z|x)$ and $p(z|x_n)$, we can estimate $I(z;s)$ by our proposed method. Then our algorithm samples randomized representations $z_n$ from $p(z|x_n)$ and input
the representations into the prediction network $f_\phi$ to obtain the predictions $\hat{y}_n = f_\phi(z_n)$, so as $\hat{y}_i$. With $\hat{y}_i$ and $\hat{y}_n$, we can estimate $\hat{L}(\hat{y}_i, \hat{y}_n)$ by our proposed method and estimate the task loss by $L(f_\phi(z), y) \approx \frac{1}{N} \sum_{n=1}^{N} L(f_\phi(z_n), y_n)$. Finally, we update $\theta, \phi$. We can execute the above training procedure at a trusted cloud provider [5] or following the split learning protocol [11]. In the split learning protocol, the client sends $z_n, y_n$ to the server (for server to update $\phi$) and receives $\nabla_{z_n} L(f_\phi(z_n), y_n)$ from the server. Then, the client estimates and computes the gradients of those two mutual information terms locally and adds $\nabla_{\theta} L(f_\theta(F_\theta(x_n)), y_n)$ to update $\theta$. Interested readers can refer to [11] for more details on split learning (we change the client’s objective into (4)).

5 EXPERIMENTS

5.1 Experimental Settings

Datasets. We use similar datasets as in [33] for evaluating InfoCensor with varied data formats, i.e., Health Heritage [1], UTKFace [3], and Twitter [2]. Health Heritage is a dataset with the medical records of over 55,000 patients. The original task is to predict whether the Charlson Index is greater than zero, and we set the sensitive attribute as gender or age. UTKFace contains over 23,000 face images labeled with age, gender, and race. We set the original task as predicting the age and the sensitive attribute as gender or race. Twitter refers to a dataset of tweets associated with user information from [2]. For Twitter, we set the original task as predicting the age of the tweeter given a tweet, and we set the sensitive attribute as the tweeter’s identity or gender. We detail how to preprocess the datasets in Appendix A.

We randomly split each dataset into 80% training data and 20% testing data. The size of auxiliary dataset is 50% of the dataset (random split). The inference attack accuracy is measured on the remaining data. Different from [33]’s setting, we do not enforce any assumption on the membership of the auxiliary data under this setting, which means the auxiliary data owned by the adversary might contain training or testing data. The adversary’s attempt is to infer the sensitive attributes of the data it does not know, no matter whether the data is training data or unseen data. We also compute the inference attack accuracy against the models under [33]’s evaluation setting, where the auxiliary data is 50% of the training data, and the inference attack accuracy is measured on testing data. We show the results in the appendix (Table 4). The results are consistent with our claim that InfoCensor achieves the overall best performance, compared to the baselines. Also, the difference between the attack results against InfoCensor under those two settings is usually not large.

Baselines. We compare InfoCensor with four baselines, including standard training (no defense), adversarial training, the VAE-based method [28], and TIPRDC [21]. Note that adversarial training and the VAE-based method are strong baselines proposed by ICLR or NeurIPS papers, used in the advanced attribute inference attack paper [33], and verified by recent works. TIPRDC is a recent baseline based on information theory for data outsourcing against attribute inference attacks [21]. The experimental results show that InfoCensor surpasses all the baselines in defending against sensitive attribute inference and mitigating demographic disparity, with modest sacrifice of the performance on the original task.

Networks. On Health Heritage, we employ an MLP with two hidden layers as the feature extraction network (encoder). On UTKFace, we employ a convolutional neural network (CNN) with three convolutional layers followed by two fully-connected layers (similar to the architecture of LeNet [20]) as the feature extraction network. For Twitter, the feature extraction network (encoder) is built upon a long-short term memory network (LSTM). For all the encoders, there are two output layers for $\mu$ and $\Sigma$ respectively as in [17]. For all the experiments, the prediction network is a single-layer neural network (following [33]), whose input is the representation from the encoder and output is the prediction. We employ an MLP with two hidden layers as the attack model to infer the sensitive attributes from the representations. We also employ an MLP with one hidden layer as the transformer for the de-censoring method [33]. We detail the network architectures in Appendix A.

Evaluation Metrics. To evaluate the model resistance against sensitive attribute inference, we employ the inference attack accuracy by the basic inference attack or the de-censoring method as the evaluation metric. The attack accuracy is computed by $\frac{1}{N} \sum_{n=1}^{N} I(\hat{y}_n; s_n)$, where $I(\cdot;\cdot)$ is the identity function, and $s_n$ refers to the sensitive attribute predicted by the basic inference attack or the de-censoring method. To evaluate demographic disparity, we employ our proposed general measurement $I(\hat{y}; s)$ and averaged SPD (computed on the testing data) as the evaluation metrics. We detail how to compute $I(\hat{y}; s)$ and averaged SPD (Avg SPD) in Section 5.5. Note that the violation of demographic disparity can be bounded by $I(\hat{y}; s)$ as in Theorem 4.4.

Hyperparameters. By default, we set the dimension of the representations as 128. We conduct hyperparameter study on the representation dimension in Section 5.7. We train all the models with the Adam optimizer [16]. By default, we set the learning rate as 0.001. We tune the framework-specific hyperparameters for adversarial training, VAE-based method, and InfoCensor individually for each dataset. For adversarial training, we have $\lambda$ as the hyperparameter before the adversarial (discriminator) loss (same as $\gamma$ in [33]). On Health Heritage, we set $\lambda$ as 0.1 when the sensitive attribute is set as age, and we set $\lambda$ as 1.0 when the sensitive attribute is gender. We set $\lambda$ as 0.1 on both UTKFace and Twitter. Note that larger $\lambda$ (than the above settings) can lead to much worse performance on the original task in our testbed. For instance, if we set $\lambda$ as 1.0 on Twitter for adversarial training, the model accuracy on the original task will decrease by about 30% in our testbed. For TIPRDC, we set $\lambda$ as 0.9 (another $\lambda$ defined in [21]) as in [21], and we add the objective of [21] to the original task loss and set the hyperparameter before the objective as the hyperparameter in adversarial training. For the VAE-based Method, we set $\lambda$ as $1 \times 10^{-5}$ and $\beta$ as 0.001 on Health Heritage and UTKFace. On Twitter, we set $\lambda$ as $1 \times 10^{-5}$ and $\beta$ as 0.0005. For InfoCensor, we set $\kappa$ as 0.5 for all the datasets, and we set $\lambda$ as 0.5, 0.25, and 0.1 on Health Heritage, UTKFace, and Twitter, respectively. Note that we conduct hyperparameter study in Section 5.7, which shows that InfoCensor can achieve stable performance with different $\lambda$ and $\kappa$ (overall better performance.
than adversarial training and VAE-based method), as long as $\lambda$ and $\kappa$ are set within a suitable range.

In Section 5.2, we execute the basic inference attack and the de-censoring method against all the models with eight hyperparameter settings, and we report the best attack accuracy in Table 1. In the other subsections, we report the attack accuracy by the basic inference attack with one hyperparameter setting.

Note that our implementation is based on PyTorch, while [33]'s implementation is based on Keras. The difference between some of our results and [33]'s results might be due to the different data pre-processing methods and the randomness of different platforms. Our code is publicly available at https://github.com/iQua/InfoCensor.

### 5.2 Empirical Results

We first show the empirical results in Table 1. In Table 1, Original refers to the model accuracy on the original task; Basic refers to the attack accuracy of the basic inference attack; De-censoring refers to the attack accuracy of the de-censoring method. Table 1 shows that InfoCensor can defend against sensitive attribute inference and mitigate demographic disparity with modest sacrifice of the accuracy on the original task. In particular, InfoCensor surpasses the existing baselines, including adversarial training, the VAE-based method [28], and TIPRDC [21] by large margins. Also, we observe that, on the models with deterministic representations, the de-censoring method is usually superior to the basic inference attack. However, on InfoCensor, the de-censoring method might overfit on the noisy representations (obtained on the auxiliary data) in some cases, and thus does not generalize well to the other data. In such cases, the basic inference attack might obtain slightly better attack accuracy.

### 5.3 Visualization

In Fig. 3, we visualize the 2D projections of the representations using t-Distributed Stochastic Neighbor Embedding (t-SNE) [35] (on UTKFace’s test set), to provide an intuition on the difference between the representations learned by InfoCensor and those learned by the other baselines. Compared with the deterministic representations learned by the other methods, the representations learned by InfoCensor distribute in a more widespread way since we randomize the representations in the inference stage. This observation is verified by Fig. 4. Fig. 4 shows that the elements of the representations learned by InfoCensor distribute from $-20$ to $30$, while the elements of the deterministic representations learned by the other methods distribute within a much smaller range. Moreover, the representations learned by InfoCensor with different sensitive classes (male and female) distribute in a less distinguishable way since InfoCensor minimizes the mutual information between the representations and the sensitive attribute. In regard to the original task, i.e., predicting the age, the randomized representations from different classes are separable, indicating that the randomized representations can handle the original task.

### 5.4 Attribute Inference and Mutual Information

In this subsection, we study the relationship between $I(\mathbf{z} ; s)$ and sensitive attribute inference. For more accurate estimation on the mutual information here, we adopt the Monte Carlo method to estimate the KL divergence between $p(z|s)$ and $p(z)$ using 50000 random samples $z_i$ from the Gaussian mixture distribution $p(z|s)$. The Gaussian mixtures are constructed over the representations of all the remaining data. Then, we could estimate $I(\mathbf{z} ; s)$ by

$$\mathbb{E}_{p(\mathbf{s})} \left[ \frac{1}{n} \sum_{i=1}^{n} \left( \log p(z_i|\mathbf{s}) - \log p(z_i) \right) \right],$$

with $p(s)$ being estimated by $\frac{N_s}{N}$. $N_s$ and $N$ denote the number of samples from one sensitive class and the total number of the samples, respectively. Note that the above Monte Carlo estimation is more accurate, but we could not use it in the training stage since backpropagation on this Monte Carlo estimation is infeasible. Under InfoCensor, we train plenty of models with different hyperparameter settings, which yield different $I(\mathbf{z} ; s)$. We conduct the basic inference attack against those models. We then plot the $I(\mathbf{z} ; s)$ estimated by the Monte Carlo method and the corresponding inference attack accuracy in Fig. 5. The empirical results are consistent with our theoretical analysis—As $I(\mathbf{z} ; s)$ decreases, the inference attack accuracy will also decrease. The empirical results also motivate the future works to utilize $I(\mathbf{z} ; s)$ as a (supplementary) measurement for measuring the threat of sensitive attribute inference.

### 5.5 Statistical parity difference (SPD) and Mutual Information

In this subsection, we study the relationship between $I(\hat{\mathbf{y}} , s)$ and SPD, a standard metric to measure demographic disparity (group unfairness). To estimate $I(\hat{\mathbf{y}} , s)$ on the testing data, we approximate $p(\hat{\mathbf{y}}|s)$ and $p(\hat{\mathbf{y}})$ by $\frac{1}{N_s} \sum_i \text{one}_\text{hot}(\hat{\mathbf{y}}_i)$ and $\frac{1}{N} \sum_i \text{one}_\text{hot}(\hat{\mathbf{y}}_i)$. Note that here we do not use the gumbel-softmax trick here since we do not need to backpropagate $I(\hat{\mathbf{y}} , s)$ in the inference stage. To compute the averaged SPD (Avg SPD), we take the average over all the absolute values of the SPDs between any two sensitive classes. Here we also train plenty of models with different hyperparameter settings, which yield different $I(\hat{\mathbf{y}} , s)$. We then plot $I(\hat{\mathbf{y}} , s)$ and the averaged SPD in Fig. 6. In general, as $I(\hat{\mathbf{y}} , s)$ increases, the averaged SPD will also increase, indicating that $I(\hat{\mathbf{y}} , s)$ is a valid measurement for measuring group fairness.

### 5.6 Impacts of Each MI

In this subsection, we evaluate the impacts of (minimizing) each mutual information term in our objective on sensitive attribute inference and group fairness. Specifically, we keep either $I(\mathbf{z} ; s)$ or $I(\mathbf{y} ; s)$ and remove the other term from the objective, and then retrain the model with the remaining objective. We fine-tune $\lambda$ or $\kappa$ after removing $I(\hat{\mathbf{y}}, s)$ or $I(\mathbf{z}; s)$ for better overall performance. We compare the results in Table 2, where we observe that the main effect of minimizing $I(\mathbf{z} ; s)$ is improving the model’s resistance against sensitive attribute inference, and minimizing $I(\hat{\mathbf{y}} ; s)$ mitigates demographic disparity. This observation is also verified by Fig. 7. Moreover, we observe that minimizing $I(\mathbf{z}; s)$ also helps
Table 1: The best results are marked in bold. Age, Gender, Race, and Identity refer to the sensitive attribute $s$. Basic refers to basic inference attack accuracy. We do not evaluate demographic disparity when we set identity as the sensitive attribute since the prediction results $\hat{y}$ completely depend on the identity $s$. Note that the results reported in [33] also show that adversarial training (TIPRDC achieves similar defensive adversarial training) is not useful on Health Heritage.

<table>
<thead>
<tr>
<th>Health Heritage</th>
<th>Age</th>
<th>Gender</th>
<th>Race</th>
<th>Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original</td>
<td>Basic</td>
<td>De-censoring</td>
<td>$l(\hat{y} ; s)$</td>
</tr>
<tr>
<td>Standard Training</td>
<td>83.11%</td>
<td>29.19%</td>
<td>29.97%</td>
<td>0.127</td>
</tr>
<tr>
<td>Adversarial Training</td>
<td>83.12%</td>
<td>28.61%</td>
<td>29.71%</td>
<td>0.125</td>
</tr>
<tr>
<td>VAE-based Method [28]</td>
<td>83.25%</td>
<td>29.40%</td>
<td>29.97%</td>
<td>0.126</td>
</tr>
<tr>
<td>InfoCensor (ours)</td>
<td>81.95%</td>
<td>20.97%</td>
<td>20.97%</td>
<td>0.092</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gender Race</th>
<th>Original</th>
<th>Basic</th>
<th>De-censoring</th>
<th>$l(\hat{y} ; s)$</th>
<th>Avg SPD</th>
<th>Original</th>
<th>Basic</th>
<th>De-censoring</th>
<th>$l(\hat{y} ; s)$</th>
<th>Avg SPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Training</td>
<td>54.12%</td>
<td>82.20%</td>
<td>82.62%</td>
<td>0.037</td>
<td>0.045</td>
<td>54.12%</td>
<td>68.39%</td>
<td>68.40%</td>
<td>0.097</td>
<td>0.053</td>
</tr>
<tr>
<td>Adversarial Training</td>
<td>55.12%</td>
<td>78.67%</td>
<td>78.28%</td>
<td>0.036</td>
<td>0.047</td>
<td>55.08%</td>
<td>63.27%</td>
<td>63.96%</td>
<td>0.088</td>
<td>0.050</td>
</tr>
<tr>
<td>TIPRDC [21]</td>
<td>54.46%</td>
<td>76.96%</td>
<td>77.55%</td>
<td>0.034</td>
<td>0.047</td>
<td>54.12%</td>
<td>62.26%</td>
<td>63.51%</td>
<td>0.093</td>
<td>0.052</td>
</tr>
<tr>
<td>VAE-based Method [28]</td>
<td>55.39%</td>
<td>76.35%</td>
<td>77.28%</td>
<td>0.039</td>
<td>0.050</td>
<td>54.89%</td>
<td>56.11%</td>
<td>59.72%</td>
<td>0.091</td>
<td>0.051</td>
</tr>
<tr>
<td>InfoCensor (ours)</td>
<td>54.51%</td>
<td>61.42%</td>
<td>60.95%</td>
<td>0.027</td>
<td>0.042</td>
<td>54.10%</td>
<td>49.24%</td>
<td>49.20%</td>
<td>0.082</td>
<td>0.040</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Twitter</th>
<th>Identity</th>
<th>Gender</th>
<th>Race</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original</td>
<td>Basic</td>
<td>De-censoring</td>
</tr>
<tr>
<td>Standard Training</td>
<td>87.58%</td>
<td>45.77%</td>
<td>47.14%</td>
</tr>
<tr>
<td>Adversarial Training</td>
<td>86.11%</td>
<td>33.28%</td>
<td>36.64%</td>
</tr>
<tr>
<td>TIPRDC [21]</td>
<td>84.34%</td>
<td>35.30%</td>
<td>38.88%</td>
</tr>
<tr>
<td>VAE-based Method [28]</td>
<td>85.42%</td>
<td>43.82%</td>
<td>48.73%</td>
</tr>
<tr>
<td>InfoCensor (ours)</td>
<td>85.65%</td>
<td>26.62%</td>
<td>26.22%</td>
</tr>
</tbody>
</table>

Table 2: Ablation study on the impacts of $I(z; s)$ and $I(\hat{y} ; s)$ in our proposed objective on model resistance against sensitive attribute inference and demographic disparity. Note that $I(z; s)$ and $I(\hat{y} ; s)$ are correlated, i.e., $I(z; s)$ is an upper bound on $I(\hat{y} ; s)$, so minimizing one term helps reduce the other one.
mitigate demographic disparity, and minimizing $I(\hat{y}; s)$ also helps improve model resistance against sensitive attribute inference. This is because $I(z; s)$ and $I(y; s)$ are correlated since $\hat{y} = f_\phi(z)$—Given $f_\phi$ as a deterministic function, $I(z; s)$ is an upper bound on $I(\hat{y}; s)$.

### 5.7 Hyperparameter Study

In this subsection, we study the sensitivity of InfoCensor to the hyperparameter settings, including the dimension of representations, the size of the auxiliary dataset (i.e., attack budget), and the hyperparameters $\lambda$ and $\kappa$ in our proposed objective.

### 5.8 Multiple Sensitive Attributes

In the previous experiments, we only defend against sensitive attribute inference and mitigate demographic disparity w.r.t. one sensitive attribute. Actually, InfoCensor can be adapted to the scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Original</th>
<th>Basic</th>
<th>De-censoring</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Health</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>82.48%</td>
<td>55.04%</td>
<td>55.04%</td>
</tr>
<tr>
<td>Age</td>
<td></td>
<td>21.27%</td>
<td>21.31%</td>
</tr>
<tr>
<td><strong>UTKFace</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>54.6%</td>
<td>62.72%</td>
<td>62.4%</td>
</tr>
<tr>
<td>Race</td>
<td></td>
<td>49.7%</td>
<td>49.4%</td>
</tr>
</tbody>
</table>

Table 3: The scenario with multiple sensitive attributes: The models are trained on the objective (27) ($I = 2$).

Hyperparameter $\lambda$ and $\kappa$. We train plenty of models under InfoCensor with different $\lambda$ and $\kappa$ and conduct the basic inference attack against those models. The corresponding results are shown in Fig. 7. As we increase $\lambda$ and $\kappa$, we observe a decrease in the original accuracy, the inference attack accuracy, and $I(\hat{y}; s)$ in most cases. If $\lambda$ or $\kappa$ is set as a very large value, the trained model might suffer from overfitting or completely lose the ability to solve the original task. Note that as long as $\lambda$ and $\kappa$ are set within a suitable range (as in Fig. 7), InfoCensor can improve model resistance against sensitive attribute inference and mitigate demographic disparity, with acceptable performance on the original task.

Dimension of Representations. We train models under InfoCensor with varied dimensions of representations ($d$) and show the results in Fig. 8. As $d$ increases, we observe a slow increase in the original accuracy and the inference attack accuracy. The results indicate more elements in the representations may not only improve the model performance on the original task, but also may provide the adversary with more information to infer the sensitive attribute.

Inference Attack Budget. We conduct the basic inference attack with different sizes of auxiliary dataset (i.e., different attack budgets) against standard training (no defense) and InfoCensor. We show the attack results in Fig. 9. We observe that as the attack budget increases, the inference attack accuracy against the model with no defense also increases. However, the attack accuracy against InfoCensor-trained models does not have an obvious increment, as the attack budget increases. We conjecture that this is because $I(z; s)$ is minimized under InfoCensor, and thus, additional pairs of $(z_{aux}, s_{aux})$ can not bring much more (mutual) information regarding the association between $z$ and $s$ for the attack model.

---

Figure 4: Distributions of the elements of the representations (probability density can be greater than 1). Note the elements of the representations learned by InfoCensor distribute in a much larger range (from -20 to 30), compared to the other baselines.

Figure 5: The relationship between sensitive attribute inference and $I(z; s)$ on the remaining data (except the auxiliary data). The models are trained by InfoCensor with different hyperparameter settings. For Health Heritage, if we set the sensitive attribute as "gender", $I(z; s)$ is small for all the hyperparameter settings under InfoCensor (smaller than 0.005). Thus, the results on Health Heritage (gender) are not shown.

Figure 6: The relationship between averaged SPD and $I(\hat{y}; s)$ (computed on the testing data). The models are trained by InfoCensor with different hyperparameter settings. For Health Heritage, if we set the sensitive attribute as "gender", $I(\hat{y}; s)$ is small for all the hyperparameter settings under InfoCensor. Thus, the results on Health Heritage (gender) are not shown.
concerning multiple sensitive attributes \( \{s_i\}_{i=1}^I \) by modifying the objective (4) as
\[
\min_{\theta, \phi} \mathcal{L}(f_{\theta}(z), y) + \frac{1}{I} \sum_{i=1}^I \left[ \lambda I(z; s_i) + \kappa I(\hat{y}; s_i) \right] \quad (27)
\]

We train models with the above objective and show the results in Table 3 \((I = 2)\). As shown in Table 3, InfoCensor can maintain acceptable accuracy on the original task, and defend against inference attacks and mitigate demographic disparity w.r.t. multiple sensitive attributes. As shown in Table 1 & 3, even when considering two sensitive attributes concurrently, InfoCensor still outperforms the baselines that consider one sensitive attribute, in terms of the overall performance on improving model resistance against inference attacks and mitigating demographic disparity w.r.t. either sensitive attribute.

6 CONCLUSIONS

In this paper, we propose a unified information-theoretic framework to defend against sensitive attribute inference and mitigate demographic disparity in model partitioning, namely InfoCensor. Except for the original task loss, InfoCensor involves two additional mutual information terms in the main objective with theoretical justifications. Specifically, with the mutual information between the model representations and the sensitive attribute being minimized, an upper bound on the chance for any adversary to infer the sensitive attribute from the representations will be reduced. With the mutual information between the predictions and the sensitive attribute being minimized, demographic disparity w.r.t. the sensitive attribute will be mitigated. To optimize the mutual information terms, we propose a Gaussian mixture based method and a gumbel-trick based method for estimating them, enabling feasible backpropagation on those two mutual information terms. Extensive evaluations in different application domains demonstrate that, with modest sacrifice of the accuracy on the original task, InfoCensor achieves substantial performance gains against sensitive attribute inference on model representations and demographic disparity.
7 ACKNOWLEDGMENT

We thank all the anonymous reviewers for their valuable comments.

REFERENCES


A DATASETS AND NETWORKS

Note that a standard encoder has one output layer, while for a variational encoder used in InfoCensor, there are two 128-dimension output layers connected to the second-to-last layer for outputting μ and Σ, respectively.

Health Heritage. We process the data into vectors associated with labels. The information that explicitly indicates age, Charlson index, and gender is removed from the vectors (only left in the labels). Each vector consists of 71 features (elements) without binarization. We employ an MLP with two hidden layers (hidden layer dimension: 256, 128) as the feature extraction network (encoder). For the feature extraction network, the input dimension is 71, i.e., the number of elements per data vector in the processed Health Heritage dataset. The output dimension is 128 by default, which is the dimension of the model representation. The decoder is also an MLP with two hidden layers. For the decoder, the input dimension is the sum of the dimension of the representations and the number of all sensitive classes. Following [28, 33], the input of the decoder is the concatenation of the representation and the one-hot encoding of the sensitive attribute. The output of the decoder is the reconstructed data vector.

UTKFace. We resize all the images into 50 × 50 by torchvision. The range of the pixel values is [0, 1]. We employ a convolutional neural network (CNN) with three convolutional layers followed
Table 4: Here we adopt the evaluation setting of attribute inference in [33], where the size of the auxiliary dataset is 50% training data. The inference attack accuracy is only measured on the testing data. Note that the results are consistent with our claim that InfoCensor achieves the overall best performance. The difference between the inference attack accuracy against InfoCensor-trained models under those two settings is usually not large. In addition, same as Table 1, \( l(\hat{y}; s) \) is computed on the testing data.

### Health Heritage

<table>
<thead>
<tr>
<th></th>
<th>Age</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basic</td>
<td>De-censoring</td>
</tr>
<tr>
<td>Standard Training</td>
<td>29.19%</td>
<td>30.42%</td>
</tr>
<tr>
<td>Adversarial Training</td>
<td>28.65%</td>
<td>29.51%</td>
</tr>
<tr>
<td>TIPRDC [21]</td>
<td>29.69%</td>
<td>29.59%</td>
</tr>
<tr>
<td>VAE-based Method [28]</td>
<td>28.81%</td>
<td>28.60%</td>
</tr>
<tr>
<td>InfoCensor (ours)</td>
<td>21.89%</td>
<td>21.94%</td>
</tr>
</tbody>
</table>

### UTKFace

<table>
<thead>
<tr>
<th></th>
<th>Gender</th>
<th>Race</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basic</td>
<td>De-censoring</td>
</tr>
<tr>
<td>Standard Training</td>
<td>82.28%</td>
<td>82.08%</td>
</tr>
<tr>
<td>Adversarial Training</td>
<td>78.63%</td>
<td>79.15%</td>
</tr>
<tr>
<td>TIPRDC [21]</td>
<td>76.36%</td>
<td>77.74%</td>
</tr>
<tr>
<td>VAE-based Method [28]</td>
<td>75.43%</td>
<td>77.04%</td>
</tr>
<tr>
<td>InfoCensor (ours)</td>
<td>61.59%</td>
<td>61.64%</td>
</tr>
</tbody>
</table>

### Twitter

<table>
<thead>
<tr>
<th></th>
<th>Identity</th>
<th>Gender</th>
<th>De-censoring</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basic</td>
<td>De-censoring</td>
<td>Basic</td>
</tr>
<tr>
<td>Standard Training</td>
<td>42.67%</td>
<td>42.09%</td>
<td>78.95%</td>
</tr>
<tr>
<td>Adversarial Training</td>
<td>29.05%</td>
<td>29.86%</td>
<td>64.91%</td>
</tr>
<tr>
<td>TIPRDC [21]</td>
<td>32.03%</td>
<td>35.36%</td>
<td>66.65%</td>
</tr>
<tr>
<td>VAE-based Method [28]</td>
<td>36.52%</td>
<td>42.63%</td>
<td>69.48%</td>
</tr>
<tr>
<td>InfoCensor (Ours)</td>
<td>23.25%</td>
<td>23.37%</td>
<td>64.22%</td>
</tr>
</tbody>
</table>

Twitter. We remove the tweets with fewer than 20 words. We also remove the tweets from the users with fewer than 500 tweets. Then the dataset contains tweets from 22 users. We create a vocabulary for the dataset and replace each word with the corresponding index in the vocabulary. Before being input into the neural networks, the word (index) is mapped into a 32-dimensional vector (embedding). The feature extraction network (encoder) is built upon a long-short term memory network (LSTM). We first embed each word into a 32-dimensional vector, and then input the sequence of embeddings into an LSTM. The input dimension of the LSTM is 32, and the output dimension is 512. The LSTM is followed by two fully-connected layers with output dimension 256 and 128. The decoder is also built upon an LSTM followed by two fully-connected layers.

Other Networks. For all the experiments, the prediction network is a single-layer neural network, whose input is the representation from the encoder (feature extraction network) and output is the prediction. We employ an MLP with two hidden layers as the attack model to infer the sensitive attributes from the representations. We also employ an MLP with one hidden layer as the transformer for the de-censoring method [33].