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.

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:
Tianhang Zheng and Baochun Li. 2022. InfoCensor: An Information-Theoretic Framework against Sensitive Attribute Inference and Demographic Disparity. In *Proceedings of the 2022 ACM Asia Conference on Computer and Communications Security (ASIA CCS '22), May 30-June 3, 2022, Nagasaki, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3488932.3517402

ASIA CCS '22, May 30-June 3, 2022, Nagasaki, Japan

ACM ISBN 978-1-4503-9140-5/22/05…$15.00
https://doi.org/10.1145/3488932.3517402

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

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 $I(\boldsymbol{z}, \boldsymbol{s})$ decreases. Based on this theoretical result, InfoCensor mitigates the threat of sensitive attribute inference from the representations by minimizing $I(\boldsymbol{z}, \boldsymbol{s})$. To mitigate demographic disparity (w.r.t. a sensitive attribute $\boldsymbol{s}$), InfoCensor minimizes the mutual information between model predictions and the sensitive attribute $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$. We show that $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$ has a direct influence on demographic parity since the extent of demographic disparity is bounded by $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$.

All in all, our proposed objective under InfoCensor mainly incorporates $I(\boldsymbol{z}, \boldsymbol{s})$, $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$, and the original task loss, with two hyperparameters balancing their effects. Since $I(\boldsymbol{z}, \boldsymbol{s})$ is analytically intractable, we first propose a Gaussian mixture based method for estimating $I(\boldsymbol{z}, \boldsymbol{s})$, provided that $p(\boldsymbol{z}|\boldsymbol{x})$ is a Gaussian distribution under InfoCensor. Also, direct optimization on $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$ is intractable, thus we propose a gumbel-trick based method to estimate $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$, which enables feasible backpropagation on $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$. 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 perceptions (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.

The code is publicly available at https://github.com/iQua/InfoCensor.

2 BACKGROUND AND RELATED WORK

2.1 Definitions and Notations

In this paper, we denote a data sample and its original task output by $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. For a classification problem, the original task output is the ground-truth label. We further denote the sensitive attribute (variable) by $\boldsymbol{s}$, distributed on an alphabet $\mathcal{S}$. 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 $\boldsymbol{s}$ 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 $\mathbf{F}_{\boldsymbol{\theta}}(\cdot)$ with parameters $\boldsymbol{\theta}$ and the prediction network by $\mathbf{f}_{\boldsymbol{\phi}}(\cdot)$ with parameters $\boldsymbol{\phi}$. We denote the prediction by $\hat{\boldsymbol{y}} = \text{argmax}\, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{x})$. We denote the probability distribution of $\boldsymbol{x}$ by $p(\boldsymbol{x})$ and the mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$ by $I(\boldsymbol{x}; \boldsymbol{z})$. 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 $\mathsf{KL}(p|q)$.

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 $\boldsymbol{x}$ belongs to the training set based on the model prediction on $\boldsymbol{x}$ [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.

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 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 $\boldsymbol{s}$ if the prediction is independent of the sensitive attribute, *i.e.,*

$$p(\hat{\boldsymbol{y}}|\boldsymbol{s}) = p(\hat{\boldsymbol{y}}). \tag{1}$$

We note that an equivalent representation for (1) is

$$\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}})) = 0 \tag{2}$$

In this paper, demographic disparity means that the predictions are unequally distributed w.r.t $\boldsymbol{s}$, *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.,* $p(\hat{\boldsymbol{y}}|\boldsymbol{s}=0) - p(\hat{\boldsymbol{y}}|\boldsymbol{s}=1)$, especially when $\boldsymbol{s}$ 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 $p\%$-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 ($I(\boldsymbol{x};\boldsymbol{z})$ becomes infinite with continuous $\boldsymbol{x}$) 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 $I(\boldsymbol{z};\boldsymbol{s})$ and $I(\hat{\boldsymbol{y}};\boldsymbol{s})$. 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 $I^{(JSD)}(\boldsymbol{x};\boldsymbol{z},\boldsymbol{u})$ is **not** defending against attribute inference but retaining more information of $\boldsymbol{x}$ 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

client and the server (service provider) can follow the split learning protocol to train the two models. In the split learning protocol, the models can be trained by one client or multiple clients with a pipeline training scheme. In the pipeline training scheme [11], each client performs one or more iterations model training and then sends the snapshot of the feature extraction model to the next client. 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.

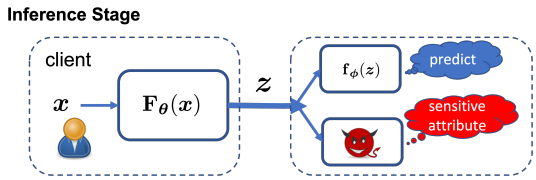

Figure 1: In the inference stage, the server may want to infer sensitive attributes of the data samples from the representations z sent by the clients [33].

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 as the threat model in [33] (ICLR20). Specifically, we assume the adversary has access to the representations and/or query access to $\mathbf{F}_\theta(\cdot)$*. 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 assess 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 for attribute inference from [33] (ICLR20).

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.

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

3.3 Sensitive Attribute Inference Attacks

Note that in the following two inference attacks, we assume the adversary has an auxiliary dataset $\mathcal{D}_{aux} \triangleq \{x_j, s_j\}$ 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 $\mathbf{g}_{\phi_{adv}}(\cdot)$ on $\{\mathbf{F}_\theta(x_m), s_m\}_{m=1}^M$, expecting that $\operatorname{argmax} \mathbf{g}_{\phi_{adv}}(\mathbf{F}_\theta(x_m)) = s_m$. Here we assume the adversary optimizes $\mathbf{g}_{\phi_{adv}}$ by minimizing the cross-entropy between $\mathbf{g}_{\phi_{adv}}(\mathbf{F}_\theta(x))$ and s using the Adam optimizer on the auxiliary dataset. After training the attack model, the adversary infers $\hat{s}$ from z by $\hat{s} = \operatorname{argmax} \mathbf{g}_{\phi_{adv}}(z)$. We detail the algorithm of the basic inference attack in Alg. 1.

Algorithm 1 Basic Sensitive Attribute Inference

Require: Auxiliary dataset $\mathcal{D}_{aux} = \{x_m, s_m\}_{m=1}^M$; target encoder $\mathbf{F}_\theta(\cdot)$; attack model $\mathbf{g}_{\phi_{adv}}(\cdot)$;
Initialize ϕ_{adv}
1. Collect $\{\mathbf{F}_\theta(x_m), s_m\}_{m=1}^M$ with the access to $\mathbf{F}_\theta(\cdot)$
2. Train $\mathbf{g}_{\phi_{adv}}(\cdot)$ on $\{\mathbf{F}_\theta(x_m), s_m\}_{m=1}^M$ to minimize the cross-entropy between $\mathbf{g}_{\phi_{adv}}(\mathbf{F}_\theta(x_m))$ and s_m

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 $\mathcal{D}_{aux}$ with an auxiliary feature extraction network (encoder) $\mathbf{F}_{\theta_{aux}}(\cdot)$ and an auxiliary prediction network $\mathbf{f}_{\phi_{aux}}(\cdot)$, by minimizing the cross-entropy between $\mathbf{f}_{\phi_{aux}}(\mathbf{F}_{\theta_{aux}}(x_m))$ and s_m. Next, the adversary learns a transform model $\mathbf{T}_\xi(\cdot)$ to transform $z_m = \mathbf{F}_\theta(x_m)$ into $\mathbf{F}_{\theta_{aux}}(x_m)$ by minimizing the mean squared error, *i.e.*, $\|\mathbf{T}_\xi(z_m) - \mathbf{F}_{\theta_{aux}}(x_m))\|_2^2$. Apparently, $\mathbf{F}_{\theta_{aux}}(x)$ leaks more information regarding s, and the adversary expects to transform z into the form of $\mathbf{F}_{\theta_{aux}}(x)$ using $\mathbf{T}_\xi(\cdot)$. The adversary then trains attack model $\mathbf{g}_{\phi_{adv}}(\cdot)$ on $\{\mathbf{T}_\xi(z_m), s_m\}$. After all the above steps, the adversary can infer $\hat{s}$ from z by $\hat{s} = \operatorname{argmax} \mathbf{g}_{\phi_{adv}}(\mathbf{T}_\xi(z))$. 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**

Algorithm 2 De-censoring

Require: Auxiliary dataset $\mathcal{D}_{aux} = \{\boldsymbol{x_m}, \boldsymbol{s_m}\}_{m=1}^{M}$; auxiliary encoder $\mathbf{F}_{\boldsymbol{\theta}_{aux}}(\cdot)$; auxiliary prediction network $\mathbf{f}_{\boldsymbol{\phi}_{aux}}(\cdot)$; target encoder $\mathbf{F}_{\boldsymbol{\theta}}(\cdot)$; attack model $\mathbf{g}_{\boldsymbol{\phi}_{adv}}(\cdot)$; transform model $\mathbf{T}_{\boldsymbol{\xi}}(\cdot)$; number of iterations T

Initialize $\boldsymbol{\theta}_{aux}, \boldsymbol{\phi}_{aux}, \boldsymbol{\phi}_{adv}, \boldsymbol{\xi}$

1. Train $\mathbf{F}_{\boldsymbol{\theta}_{aux}}(\cdot)$ and $\mathbf{f}_{\boldsymbol{\phi}_{aux}}(\cdot)$ on the cross-entropy between $\mathbf{f}_{\boldsymbol{\phi}_{aux}}(\mathbf{F}_{\boldsymbol{\theta}_{aux}}(\boldsymbol{x_m}))$ and $\boldsymbol{s_m}$

2. Train $\mathbf{T}_{\boldsymbol{\xi}}(\cdot)$ on the mean squared error between $\mathbf{F}_{\boldsymbol{\theta}_{aux}}(\boldsymbol{x_m})$ and $\mathbf{T}_{\boldsymbol{\xi}}(\mathbf{F}_{\boldsymbol{\theta}}(\boldsymbol{x_m}))$, *i.e.,* $\| \mathbf{T}_{\boldsymbol{\xi}}(\mathbf{F}_{\boldsymbol{\theta}}(\boldsymbol{x_m})) - \mathbf{F}_{\boldsymbol{\theta}_{aux}}(\boldsymbol{x_m}) \|_2^2$ (can be executed along with 3 simultaneously)

3. Train $\mathbf{g}_{\boldsymbol{\phi}_{adv}}(\cdot)$ on $\{\mathbf{T}_{\boldsymbol{\xi}}(\mathbf{F}_{\boldsymbol{\theta}}(\boldsymbol{x_m})), \boldsymbol{s_m}\}_{m=1}^{M}$ to minimize the cross-entropy between $\mathbf{g}_{\boldsymbol{\phi}_{adv}}(\mathbf{T}_{\boldsymbol{\xi}}(\mathbf{F}_{\boldsymbol{\theta}}(\boldsymbol{x_m})))$ and $\boldsymbol{s}_m$

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 disparity between different sensitive classes (*e.g.,* disparity between $p(\boldsymbol{s} = i)$ and $p(\boldsymbol{s} = j)$). Then without the uniform prior assumption, the measurement $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ may induce bias against the group j with small $p(\boldsymbol{s} = j)$. Specifically, for a sensitive class j, if the $p(\boldsymbol{s} = j)$ is relatively very small ($\ll 1/|\mathcal{S}|$), then even if the corresponding $\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s} = j)|p(\hat{\boldsymbol{y}}))$ is very large (*i.e.,* exhibiting severe discrimination against the group with sensitive class j), $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ is still not large. However, we expect that the measurement $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ is large no matter which sensitive class related group suffers from severe discrimination. Moreover, with the uniform prior assumption, we can bound the extent of the violation of (1) (the extent of demographic disparity) by $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ as in Theorem 4.4. Thus, we adopt a uniform prior assumption on $p(\boldsymbol{s})$ when computing the measurement $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$. Formally, our proposed measurement can be expressed as

$$I(\hat{\boldsymbol{y}}; \boldsymbol{s}) = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{s}} \mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}})), \tag{3}$$

Since $\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}})) \geq 0$, (3) is also greater than or equal to 0. And if (3) is equal to 0, then the corresponding model satisfies demographic parity, as detailed in Theorem 4.3 & 4.4. **The proposed measurement can apply to an arbitrary discrete sensitive attribute.** For a bounded continuous sensitive attribute $\boldsymbol{s} \in [a, b]^n$, we can employ $I(\hat{\boldsymbol{y}}; \boldsymbol{s}) = \int_{\boldsymbol{s}} \frac{1}{(b-a)^n} \mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}}))$ as the group fairness measurement, assuming a uniform prior on $\boldsymbol{s}$. Note that the empirical results in Section 5.5 further verifies the validity of this measurement for measuring group fairness.

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

$$\min_{\boldsymbol{\theta}, \boldsymbol{\phi}} \mathcal{L}(\mathbf{f}_{\boldsymbol{\phi}}(z), \boldsymbol{y}) + \lambda I(z; \boldsymbol{s}) + \kappa I(\hat{\boldsymbol{y}}; \boldsymbol{s}) \tag{4}$$
$$s.t.\ z = \mathbf{F}_{\boldsymbol{\theta}}(\boldsymbol{x}), \hat{\boldsymbol{y}} = \operatorname{argmax} \mathbf{f}_{\boldsymbol{\phi}}(z)$$

where $\mathbf{F}_{\boldsymbol{\theta}}(\cdot)$ outputs a randomized representation rather than a deterministic one, *i.e.,* $z \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}), \Sigma_{\boldsymbol{\theta}}(\boldsymbol{x}))$ (**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 $I(\boldsymbol{x}; z)$ is provably an infinite value or a constant [10]. $\mathcal{L}(\mathbf{f}_{\boldsymbol{\phi}}(z), \boldsymbol{y})$ 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 $\mathbf{f}_{\boldsymbol{\phi}}(z)$ and the ground-truth label $\boldsymbol{y}$. If the original task is an unsupervised task without task output, we might replace $\mathcal{L}(\mathbf{f}_{\boldsymbol{\phi}}(z), \boldsymbol{y})$ with an unsupervised loss, *e.g.,* reconstruction loss. **In this paper, we mainly focus on the classification task.** $I(z; \boldsymbol{s})$ refers to the mutual information between the representations z and the sensitive attribute $\boldsymbol{s}$. We minimize $I(z; \boldsymbol{s})$ because as $I(z; \boldsymbol{s})$ 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). $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ refers to the mutual information between the model prediction $\hat{\boldsymbol{y}}$ and the sensitive variable $\boldsymbol{s}$. We minimize $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ since the extent of demographic disparity can be bounded by $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ (see Theorem 4.4 for details). λ and κ are the hyperparameters that balance utility, resistance against sensitive attribute inference, and group fairness. In practice, the concrete settings of λ and κ may depend on the task and users' requirements. In the following, we will introduce and prove our theoretical results.

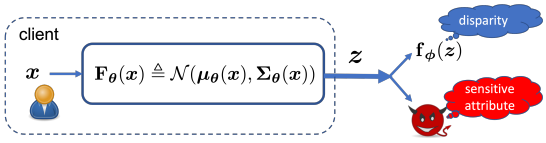

Figure 2: The representations z are sampled from the parameterized Gaussian mechanisms ($\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}), \Sigma_{\boldsymbol{\theta}}(\boldsymbol{x}))$) learned by optimizing our objective. The encoder has similar architecture as a variational encoder [17].

4.2 Theoretical Connections

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

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

$$\mathbb{P}[\hat{\boldsymbol{s}} = \boldsymbol{s}] \leq \frac{I(z; \boldsymbol{s}) + \log 2}{\log |\mathcal{S}|}, \tag{5}$$

where $\hat{\boldsymbol{s}}$ 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 $\mathcal{S}$. Let $\hat{s}$ be an estimate of s, we then have (Theorem 1 in [30])*

$$H(s|\hat{s}) \leq H(P_e) + P_e \log(|\mathcal{S}| - 1), \tag{6}$$

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

$$I(s;\hat{s}) \geq H(s) - H(P_e) - P_e \log(|\mathcal{S}| - 1) \tag{7}$$

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

$$P_e \log|\mathcal{S}| \geq P_e \log(|\mathcal{S}| - 1) \geq H(s) - I(s;\hat{s}) - \log 2 \tag{8}$$

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

$$\begin{aligned} \mathbb{P}[\hat{s} \neq s] = P_e &\geq \frac{H(s)}{\log|\mathcal{S}|} - \frac{I(s;\hat{s}) + \log 2}{\log|\mathcal{S}|} \\ &\geq \frac{H(s)}{\log|\mathcal{S}|} - \frac{I(s;z) + \log 2}{\log|\mathcal{S}|} \end{aligned} \tag{9}$$

If s is uniformly distributed over $\mathcal{S}$, we have $H(s) = \log|\mathcal{S}|$. 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 $\mathcal{S}$, then the bound is $\mathbb{P}[\hat{s} = s] \leq 1 - \frac{H(s)}{\log|\mathcal{S}|} + \frac{I(z;s)+\log 2}{\log|\mathcal{S}|}$*

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

Theorem 4.2. *Let s represents 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*

$$P[\hat{s} = s] \leq \frac{1 + \sqrt{1 - (1 - \frac{I(z;s)}{\log 2})^2}}{2}, \tag{10}$$

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 $|\mathcal{S}| = 2$ 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 $\frac{H^2(P_e)}{P_e(1-P_e)}$ w.r.t. P_e is

$$(\frac{H^2(P_e)}{P_e(1-P_e)})' = \frac{g^2(P_e) - g^2(1-P_e)}{P_e^2(1-P_e)^2}. \tag{11}$$

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

$$H^2(P_e) \leq 4(\log 2)^2 P_e(1-P_e). \tag{12}$$

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 $s \to x \to z \to \hat{s}$ [40], $I(s;\hat{s}) \leq I(s;z)$, and thus $H(s) - I(s;\hat{s}) \geq H(s) - I(s;z) = H(s|z) \geq 0$[†]. Therefore, we have $4(\log 2)^2 P_e(1-P_e) \geq (H(s) - I(s;z))^2$, *i.e.,*

$$P_e^2 - P_e + (\frac{H(s) - I(s;z)}{2\log 2})^2 \leq 0. \tag{13}$$

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

$$P_e^2 - P_e + (\frac{1}{2} - \frac{I(s;z)}{2\log 2})^2 \leq 0. \tag{14}$$

According to the quadratic root formula, we have

$$P_e \geq \frac{1 - \sqrt{1 - (1 - \frac{I(s;z)}{\log 2})^2}}{2} \tag{15}$$

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

$$P[\hat{s} = s] \leq \frac{1 + \sqrt{1 - (1 - \frac{I(s;z)}{\log 2})^2}}{2} \tag{16}$$

□

Based on Theorem 4.2, we have the following corollary:

Corollary 1. *If $I(z;s)$ is minimized to 0, then the adversary can not 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

$$P_e^2 - P_e + \frac{1}{4} = (P_e - \frac{1}{2})^2 \leq 0. \tag{17}$$

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

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 $\mathcal{S}$ 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[‡]. 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[§].

[†]Conditional entropy of a discrete variable s is non-negative.
[‡]The priors of the auxiliary datasets can be not uniform in the experiments.
[§]A discrete uniform distribution has the maximum entropy.

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

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

However, if $|\mathcal{S}|$ is not large, involving $I(\boldsymbol{x};\boldsymbol{z})$ as a regularizer leads to a suboptimal solution since minimizing $I(\boldsymbol{x};\boldsymbol{z})$ censors all the information, not only the information regarding $\boldsymbol{s}$, from the representations. Therefore, in the experiments, we only add $0.001 \times I(\boldsymbol{x};\boldsymbol{z})$ to our proposed objective in (4) when training the model to protect user identity ($|\mathcal{S}| = 22$) on the Twitter dataset. For all the other cases, we do not involve $I(\boldsymbol{x};\boldsymbol{z})$ in our proposed objective.

Demographic Parity and $I(\hat{\boldsymbol{y}};\boldsymbol{s})$. We propose to mitigate demographic disparity w.r.t. a sensitive attribute $\boldsymbol{s}$ by minimizing the mutual information between the predictions $\hat{\boldsymbol{y}}$ and the sensitive attribute $\boldsymbol{s}$, *i.e,* $I(\hat{\boldsymbol{y}};\boldsymbol{s})$, based on Theorem 4.3 & 4.4. Theorem 4.3 shows a simple connection between demographic parity and $I(\hat{\boldsymbol{y}};\boldsymbol{s})$.

Theorem 4.3. *Let $\boldsymbol{s}$ represents a sensitive attribute (variable), and $\boldsymbol{x}$ and $\boldsymbol{z}$ respectively refer to the corresponding data sample and the representation. $\hat{\boldsymbol{y}}$ denotes the prediction,* i.e., $\hat{\boldsymbol{y}} = \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{z})$. *The prediction model achieves demographic parity w.r.t. $\boldsymbol{s}$ if and only if $I(\hat{\boldsymbol{y}};\boldsymbol{s})$ is minimized to* 0.

Proof. Note that $\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}}))$ must be non-negative. So if $I(\hat{\boldsymbol{y}};\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{s}}[\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}}))] = 0$, we must have for all $\boldsymbol{s}$ such that $p(\boldsymbol{s}) > 0$, $\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}})) = 0$. If $\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}})) = 0$, we must have $p(\hat{\boldsymbol{y}}|\boldsymbol{s}) = p(\hat{\boldsymbol{y}})$ (demographic parity). On the other hand, if the model achieves demographic parity, we must have $\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}})) = 0$, then $I(\hat{\boldsymbol{y}};\boldsymbol{s})$ must be minimized to 0. □

Theorem 4.4. *Given the definitions in Theorem 4.1 & 4.3 and the uniform prior assumption on $p(\boldsymbol{s})$, we have $\sum_{\hat{\boldsymbol{y}}\in\mathcal{Y}} |p(\hat{\boldsymbol{y}}|\boldsymbol{s}) - p(\hat{\boldsymbol{y}})| \le \sqrt{2|\mathcal{S}| \cdot I(\hat{\boldsymbol{y}};\boldsymbol{s})}$ ($\forall \boldsymbol{s} \in \mathcal{S}$).*

Proof. With a uniform assumption on $p(\boldsymbol{s})$, we have $p(\boldsymbol{s}) = \frac{1}{|\mathcal{S}|}$ (discrete uniform distribution). Then we have

$$I(\hat{\boldsymbol{y}};\boldsymbol{s}) = \frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{s}\in\mathcal{S}} \mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}})) \ge \frac{1}{|\mathcal{S}|}\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}})) \ (\forall \boldsymbol{s} \in \mathcal{S}) \tag{18}$$

According to Theorem B in [8], we have

$$\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}})) \ge \frac{1}{2}\Big(\sum_{\hat{\boldsymbol{y}}\in\mathcal{Y}} |p(\hat{\boldsymbol{y}}|\boldsymbol{s}) - p(\hat{\boldsymbol{y}})|\Big)^2. \tag{19}$$

Thus, we have the following bound, *i.e.,*

$$\sum_{\hat{\boldsymbol{y}}\in\mathcal{Y}} |p(\hat{\boldsymbol{y}}|\boldsymbol{s}) - p(\hat{\boldsymbol{y}})| \le \sqrt{2\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}}))} \le \sqrt{2|\mathcal{S}| \cdot I(\hat{\boldsymbol{y}};\boldsymbol{s})} \tag{20}$$

□

Note that $\mathcal{Y}$ refers to the set of all possible predictions $\hat{\boldsymbol{y}}$. **Theorem 4.4 indicates that the extent of the violation of demographic parity (1) can be bounded by $I(\hat{\boldsymbol{y}};\boldsymbol{s})$.**

Connection between Attribute Inference and Demographic Disparity. There is also a close connection between attribute inference, demographic disparity, and mutual information. Theoretically, minimizing $I(\boldsymbol{z};\boldsymbol{s})$ also helps mitigate demographic disparity. This is because $I(\boldsymbol{z};\boldsymbol{s})$ is an upper bound on $I(\hat{\boldsymbol{y}};\boldsymbol{s})$. Different from $\mathbf{F}_{\boldsymbol{\theta}}(\cdot)$, $\mathbf{f}_{\boldsymbol{\phi}}(\cdot)$ is a deterministic function. Since $\hat{y} = \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{z})$, we have $I(\hat{\boldsymbol{y}};\boldsymbol{s}) \le I(\boldsymbol{z};\boldsymbol{s})$. On the other hand, we observe that minimizing $I(\hat{\boldsymbol{y}};\boldsymbol{s})$ also *somehow* helps defend against sensitive attribute inference attacks in practice. This is because $\hat{\boldsymbol{y}}$ contains part of the information in $\boldsymbol{z}$, and thus, minimizing $I(\hat{\boldsymbol{y}};\boldsymbol{s})$ also helps weaken the association between $\boldsymbol{z}$ and $\boldsymbol{s}$.

Remark 3. *Some readers may think minimizing $I(\hat{\boldsymbol{y}};\boldsymbol{s})$ is not necessary, but we note that the estimation for $I(\hat{\boldsymbol{y}};\boldsymbol{s})$ is usually more accurate than the estimation for $I(\boldsymbol{z};\boldsymbol{s})$ due to the relatively low dimension of $\hat{\boldsymbol{y}}$, compared to $\boldsymbol{z}$. Also, $I(\hat{\boldsymbol{y}};\boldsymbol{s})$ 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 $I(\boldsymbol{z};\boldsymbol{s})$, and how to use a gumbel-softmax trick based method for estimating $I(\hat{\boldsymbol{y}};\boldsymbol{s})$. Note that there are many methods for estimating mutual information. However, the estimations acquired by Monte Carto 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 $I(\boldsymbol{z};\boldsymbol{s})$ and $I(\boldsymbol{y};\boldsymbol{s})$. Compared to the previous estimation methods, our estimation methods enable feasible backpropagation on the estimated $I(\boldsymbol{z};\boldsymbol{s})$ and $I(\hat{\boldsymbol{y}},\boldsymbol{s})$, without training any additional networks.

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

$$I(\boldsymbol{z};\boldsymbol{s}) = E_{\boldsymbol{s}}[\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{s})|p(\boldsymbol{z}))] \tag{21}$$

Since the true $p(\boldsymbol{z}|\boldsymbol{s})$ and $p(\boldsymbol{z})$ are intractable, we propose to approximate both distributions by Gaussian mixtures. For any $\boldsymbol{s}$, we approximate $p(\boldsymbol{z}|\boldsymbol{s})$ by $p(\boldsymbol{z}|\boldsymbol{s}) \approx \frac{1}{N_s}\sum_{i=1}^{N_s} p(\boldsymbol{z}|\boldsymbol{x}_i)$, where $\{\boldsymbol{x}_i\}_{i=1}^{N_s}$ denotes the samples with one sensitive class. Likewise, we can approximate $p(\boldsymbol{z})$ by $\frac{1}{N}\sum_{n=1}^{N} p(\boldsymbol{z}|\boldsymbol{x}_n)$. Recall that, under InfoCensor, each $p(\boldsymbol{z}|\boldsymbol{x}_i)$ is a Gaussian distribution with mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$, so the approximations of $p(\boldsymbol{z}|\boldsymbol{s})$ and $p(\boldsymbol{z})$ are Gaussian mixtures.

With a uniform prior assumption (in the training stage), we can formulate an empirical approximation of (21) as

$$I(\boldsymbol{z};\boldsymbol{s}) \approx \frac{1}{|\mathcal{S}|}\sum_{s}[\mathrm{KL}(\frac{1}{N_s}\sum_{i=1}^{N_s}p(\boldsymbol{z}|\boldsymbol{x}_i)|\frac{1}{N}\sum_{n=1}^{N}p(\boldsymbol{z}|\boldsymbol{x}_n))] \tag{22}$$

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 $\alpha_{n,i} \geq 0$ and $\beta_{i,n} \geq 0$, such that $\sum_n \alpha_{n,i} = 1/N_s$ and $\sum_i \beta_{i,n} = 1/N$. We then have $p(\boldsymbol{z}|\boldsymbol{s}) \approx \sum_{n,i}\alpha_{n,i}p(\boldsymbol{z}|\boldsymbol{x}_i)$ and $p(\boldsymbol{z}) \approx \sum_{n,i}\beta_{i,n}p(\boldsymbol{z}|\boldsymbol{x}_n)$.

$$\begin{aligned}\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{s})|p(\boldsymbol{z})) &= -\int_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{s})\log\frac{p(\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{s})}d\boldsymbol{z} \\ &\approx -\int_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{s})\log\sum_{i,n}\frac{\beta_{i,n}p(\boldsymbol{z}|\boldsymbol{x}_n)}{\alpha_{n,i}p(\boldsymbol{z}|\boldsymbol{x}_i)}\frac{\alpha_{n,i}p(\boldsymbol{z}|\boldsymbol{x}_i)}{p(\boldsymbol{z}|\boldsymbol{s})}d\boldsymbol{z} \\ &\leq -\sum_{n,i}\alpha_{n,i}\int_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{x}_i)\log\frac{\beta_{i,n}p(\boldsymbol{z}|\boldsymbol{x}_n)}{\alpha_{n,i}p(\boldsymbol{z}|\boldsymbol{x}_i)}d\boldsymbol{z} \\ &= \sum_{n,i}\alpha_{n,i}\log\frac{\alpha_{n,i}}{\beta_{i,n}} + \sum_{n,i}\alpha_{n,i}\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{x}_i)|p(\boldsymbol{z}|\boldsymbol{x}_n))\end{aligned} \tag{23}$$

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

$$-\log\sum_{i,n}\frac{\beta_{i,n}p(\boldsymbol{z}|\boldsymbol{x}_n)}{\alpha_{n,i}p(\boldsymbol{z}|\boldsymbol{x}_i)}\frac{\alpha_{n,i}p(\boldsymbol{z}|\boldsymbol{x}_i)}{p(\boldsymbol{z}|\boldsymbol{s})} \leq -\sum_{i,n}\frac{\alpha_{n,i}p(\boldsymbol{z}|\boldsymbol{x}_i)}{p(\boldsymbol{z}|\boldsymbol{s})}\log\frac{\beta_{i,n}p(\boldsymbol{z}|\boldsymbol{x}_n)}{\alpha_{n,i}p(\boldsymbol{z}|\boldsymbol{x}_i)}$$

$\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{x}_i)|p(\boldsymbol{z}|\boldsymbol{x}_n))$ 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 $\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{s})|p(\boldsymbol{z}))$ is obtained by minimizing $\sum_{n,i}\alpha_{n,i}\log\frac{\alpha_{n,i}}{\beta_{i,n}} + \sum_{n,i}\alpha_{n,i}\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{x}_i)|p(\boldsymbol{z}|\boldsymbol{x}_n))$ w.r.t. $\alpha_{n,i}$ and $\beta_{i,n}$. Specifically, we optimize $\alpha_{n,i}$ and $\beta_{i,n}$ by fixing one and updating the other with its optimal solution. In each iteration, we first fix $\alpha_{n,i}$ and update $\beta_{i,n}$ by $\beta_{i,n} = \frac{\alpha_{n,i}}{N\sum_{i'}\alpha_{n,i'}}$. Then we fix $\beta_{i,n}$ and update $\alpha_{n,i}$ by $\alpha_{n,i} = \frac{\beta_{i,n}\exp(-\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{x}_i)|p(\boldsymbol{z}|\boldsymbol{x}_n)))}{N_s\sum_{n'}\beta_{i,n'}\exp(-\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{x}_i)|p(\boldsymbol{z}|\boldsymbol{x}'_{n'})))}$. We execute the above operations for several iterations and fix $\alpha_{n,i}$ and $\beta_{i,n}$ to compute the approximation. **Note that we fix $\alpha_{n,i}$ and $\beta_{i,n}$ as constants (by *detach*() in PyTorch), when we optimize the model parameters, to avoid backpropagation through the above iterations.** After fixing $\alpha_{n,i}$ and $\beta_{i,n}$, we only need to compute the gradient of the KL divergence between two Gaussian distributions, *i.e.,* $\nabla_{\boldsymbol{\theta}}\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{x}_i)|p(\boldsymbol{z}|\boldsymbol{x}_n))$, to optimize $\boldsymbol{\theta}$.

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

$$\frac{1}{N}\sum_{n=1}^{N}\frac{1}{N}\sum_{n'=1}^{N}\mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{x}_n)|p(\boldsymbol{z}|\boldsymbol{x}_{n'})).$$

Gumbel-Softmax Trick based Estimation. Finally, we propose a method using the gumbel-softmax trick [15] for estimating $I(\hat{\boldsymbol{y}};\boldsymbol{s})$. Similar to (21), we can express $I(\hat{\boldsymbol{y}};\boldsymbol{s})$ as

$$I(\hat{\boldsymbol{y}};\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{s}}[\mathrm{KL}(p(\hat{\boldsymbol{y}}|\boldsymbol{s})|p(\hat{\boldsymbol{y}}))]. \tag{24}$$

Note that both $p(\hat{\boldsymbol{y}}|\boldsymbol{s})$ and $p(\hat{\boldsymbol{y}})$ can be denoted by a vector, with each element representing the probability corresponding to a possible value of $\boldsymbol{y}$. We then estimate $p(\hat{\boldsymbol{y}}|\boldsymbol{s})$ and $p(\hat{\boldsymbol{y}})$ by $\frac{1}{N}\sum_n \mathrm{one\_hot}(\hat{\boldsymbol{y}}_n)$ and $\frac{1}{N_s}\sum_i \mathrm{one\_hot}(\hat{\boldsymbol{y}}_i)$, where $\mathrm{one\_hot}(\hat{\boldsymbol{y}})$ denotes the one hot vector of $\hat{\boldsymbol{y}}$. Specifically, if the prediction $\hat{\boldsymbol{y}}$ corresponds to the $\tilde{k}$-th class, then the $\tilde{k}$-th element of $\mathrm{one\_hot}(\hat{\boldsymbol{y}})$ is 1, and all the other elements are 0. We denote the k-th element of $\mathrm{one\_hot}(\hat{\boldsymbol{y}})$ by $\mathrm{one\_hot}_k(\hat{\boldsymbol{y}})$. **Since $\mathbf{one\_hot}(\cdot)$ is not differentiable, we propose to estimate it by the gumbel-softmax trick [15].**

We define the predicted class probability vector as $\boldsymbol{\pi}(\boldsymbol{x}) \triangleq \mathrm{softmax}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{x}))$. Gumbel-softmax trick estimates $\mathrm{one\_hot}(\hat{\boldsymbol{y}})$ by

$$\mathrm{one\_hot}_k(\hat{\boldsymbol{y}}) \approx \frac{\exp((\log(\boldsymbol{\pi}_k(\boldsymbol{x})) + g_k)/\tau)}{\sum_{k'}\exp((\log(\boldsymbol{\pi}_{k'}(\boldsymbol{x})) + g_{k'})/\tau)}, \tag{25}$$

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

$$\frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{s}\in\mathcal{S}}\mathrm{KL}(\frac{1}{N_s}\sum_{i}\mathrm{one\_hot}(\hat{\boldsymbol{y}}_i)|\frac{1}{N}\sum_{n}\mathrm{one\_hot}(\hat{\boldsymbol{y}}_n)), \tag{26}$$

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 $\mathcal{D} = \{\boldsymbol{x}_n, \boldsymbol{y}_n, \boldsymbol{s}_n\}_{n=1}^{\mathbb{N}}$; feature extraction network $\mathbf{F}_{\boldsymbol{\theta}}$; prediction network $\mathbf{f}_{\boldsymbol{\phi}}$; surrogate loss $\mathcal{L}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{z}), \boldsymbol{y})$; number of iterations T

Initialize $\boldsymbol{\theta}, \boldsymbol{\phi}$

for t = 1 to T **do**

1. Sample a minibatch from $\mathcal{D}$ ($\{\boldsymbol{x}_n, \boldsymbol{y}_n, \boldsymbol{s}_n\}_{n=1}^{N}$). The size of the minibatch is denoted by N. The size of samples with a specific $\boldsymbol{s}$ in the minibatch is denoted by $N_{\boldsymbol{s}}$.
2. Input the minibatch of the data samples $\boldsymbol{x}_n$ into $\mathbf{F}_{\boldsymbol{\theta}}$ to acquire the mean and covariance matrix of the representations, *i.e.,* $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$ and $\Sigma_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$.
3. Sample randomized representations $\boldsymbol{z}_n \sim p(\boldsymbol{z}|\boldsymbol{x}_n)$, where $p(\boldsymbol{z}|\boldsymbol{x}_n) \triangleq \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_n), \Sigma_{\boldsymbol{\theta}}(\boldsymbol{x}_n))$
4. Input $\boldsymbol{z}_n$ into $\mathbf{f}_{\boldsymbol{\phi}}$ to obtain the predictions $\hat{\boldsymbol{y}}_n = \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{z}_n)$.
5. Update $\boldsymbol{\theta}, \boldsymbol{\phi}$ by the Adam optimizer to minimize (4).

end for

Output $\mathbf{F}_{\boldsymbol{\theta}}$ and $\mathbf{f}_{\boldsymbol{\phi}}$

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 $\mathbf{F}_{\boldsymbol{\theta}}$ (encoder). The encoder outputs the mean and covariance matrix of the representations, *i.e.,* $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $\Sigma_{\boldsymbol{\theta}}(\boldsymbol{x})$. Under InfoCensor, $p(\boldsymbol{z}|\boldsymbol{x}_n) \triangleq \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_n), \Sigma_{\boldsymbol{\theta}}(\boldsymbol{x}_n))$, so as $p(\boldsymbol{z}|\boldsymbol{x}_i)$. Given $p(\boldsymbol{z}|\boldsymbol{x}_i)$ and $p(\boldsymbol{z}|\boldsymbol{x}_n)$, we can estimate $I(\boldsymbol{z};\boldsymbol{s})$ by our proposed method. Then our algorithm samples randomized representations $\boldsymbol{z}_n$ from $p(\boldsymbol{z}|\boldsymbol{x}_n)$ and input

the representations into the prediction network $\mathbf{f}_{\boldsymbol{\phi}}$ to obtain the predictions $\hat{\boldsymbol{y}}_n = \mathbf{f}_{\boldsymbol{\phi}}(z_n)$, so as $\hat{\boldsymbol{y}}_i$. With $\hat{\boldsymbol{y}}_i$ and $\hat{\boldsymbol{y}}_n$, we can estimate $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ by our proposed method and estimate the task loss by $\mathcal{L}(\mathbf{f}_{\boldsymbol{\phi}}(z), \boldsymbol{y}) \approx \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(\mathbf{f}_{\boldsymbol{\phi}}(z_n), \boldsymbol{y}_n)$. Finally, we update $\boldsymbol{\theta}, \boldsymbol{\phi}$. 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, $\boldsymbol{y}_n$ to the server (for server to update $\boldsymbol{\phi}$) and receives $\nabla_{z_n}\mathcal{L}(\mathbf{f}_{\boldsymbol{\phi}}(z_n), \boldsymbol{y}_n)$ from the server. Then, the client estimates and computes the gradients of those two mutual information terms locally and adds $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{f}_{\boldsymbol{\phi}}(\mathbf{F}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)), \boldsymbol{y}_n)$ to update $\boldsymbol{\theta}$. 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 23000 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], 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 μ and Σ 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 $\frac{1}{N}\sum_{n=1}^{N}\mathbb{1}(\hat{\boldsymbol{s}}_n = \boldsymbol{s})$, where $\mathbb{1}(\cdot)$ is the identity function, and $\hat{\boldsymbol{s}}_n$ 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 $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ and averaged SPD (computed on the testing data) as the evaluation metrics. We detail how to compute $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ and averaged SPD (Avg SPD) in Section 5.5. Note that the violation of demographic disparity can be bounded by $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ 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 λ as the hyperparameter before the adversarial (discriminator) loss (same as γ in [33]). On Health Heritage, we set λ as 0.1 when the sensitive attribute is set as age, and we set λ as 1.0 when the sensitive attribute is gender. We set λ as 0.1 on both UTKFace and Twitter. Note that larger λ (than the above settings) can lead to much worse performance on the original task in our testbed. For instance, if we set λ 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 λ as 0.9 (another λ 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 λ as 1×10^{-5} and β as 0.001 on Health Heritage and UTKFace. On Twitter, we set λ as 1×10^{-5} and β as 0.0005. For InfoCensor, we set κ as 0.5 for all the datasets, and we set λ 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 λ and κ (overall better performance

[¶]This setting may be more suitable for some scenarios, where the adversary may also try to infer the sensitive attributes of clients' training data.

than adversarial training and VAE-based method), as long as λ and κ 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 preprocessing 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.

[‖]learning rate 0.001 or 0.0001; dropout rate 0.0 or 0.1; weight decay 0.0 or 2×10^{-4}. The number of the combinations of the above settings is 8.

[**]learning rate 0.001; dropout rate 0.0; weight decay 0.0.

5.4 Attribute Inference and Mutual Information

In this subsection, we study the relationship between $I(\boldsymbol{z}; \boldsymbol{s})$ 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 $p(\boldsymbol{z}|\boldsymbol{s})$ and $p(\boldsymbol{z})$ using 50000 random samples $\boldsymbol{z_i}$ from the Gaussian mixture distribution $p(\boldsymbol{z}|\boldsymbol{s})$. The Gaussian mixtures are constructed over the representations of all the remaining data. Then, we could estimate $I(\boldsymbol{z}; \boldsymbol{s})$ by

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

with $p(\boldsymbol{s})$ 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 $I(\boldsymbol{z}; \boldsymbol{s})$. We conduct the basic inference attack against those models. We then plot the $I(\boldsymbol{z}; \boldsymbol{s})$ 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 $I(\boldsymbol{z}; \boldsymbol{s})$ decreases, the inference attack accuracy will also decrease. The empirical results also motivate the future works to utilize $I(\boldsymbol{z}; \boldsymbol{s})$ 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 $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$ and SPD, a standard metric to measure demographic disparity (group unfairness). To estimate $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$ on the testing data, we approximate $p(\hat{\boldsymbol{y}}|\boldsymbol{s})$ and $p(\hat{\boldsymbol{y}})$ by $\frac{1}{N_s}\sum_i \text{one\_hot}(\hat{\boldsymbol{y}}_i)$ and $\frac{1}{N}\sum_n \text{one\_hot}(\hat{\boldsymbol{y}}_n)$. Note that here we do not use the gumbel-softmax trick here since we do not need to backpropagate $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$ 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 $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$. We then plot $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$ and the averaged SPD in Fig. 6. In general, as $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$ increases, the averaged SPD will also increase, indicating that $I(\hat{\boldsymbol{y}}, \boldsymbol{s})$ 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 $I(\boldsymbol{z}; \boldsymbol{s})$ or $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ and remove the other term from the objective, and then retrain the model with the remaining objective. We fine-tune λ or κ after removing $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ or $I(\boldsymbol{z}; \boldsymbol{s})$ for better overall performance. We compare the results in Table 2, where we observe that the main effect of minimizing $I(\boldsymbol{z}; \boldsymbol{s})$ is improving the model's resistance against sensitive attribute inference, and minimizing $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ mitigates demographic disparity. This observation is also verified by Fig. 7. Moreover, we observe that minimizing $I(\boldsymbol{z}; \boldsymbol{s})$ also helps

Health Heritage			**Age**					**Gender**		
	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD
Standard Training	83.11%	29.19%	29.97%	0.127	0.266	83.11%	59.06%	59.70%	1.58×10^{-4}	0.016
Adversarial Training	83.12%	28.61%	29.71%	0.125	0.258	83.00%	59.06%	59.98%	0.57×10^{-4}	0.010
TIPRDC [21]	83.25%	29.40%	29.97%	0.126	0.265	83.25%	58.88%	60.37%	0.64×10^{-4}	0.010
VAE-based Method [28]	82.98%	27.81%	28.64%	0.117	0.250	83.08%	55.25%	58.01%	1.84×10^{-4}	0.017
InfoCensor (ours)	81.95%	**20.97%**	**20.97%**	**0.092**	**0.216**	82.83%	**55.02%**	**55.05%**	$\mathbf{0.52 \times 10^{-4}}$	**0.009**
UTKFace			**Gender**					**Race**		
	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD
Standard Training	54.12%	82.20%	82.62%	0.037	0.045	54.12%	68.39%	68.40%	0.097	0.053
Adversarial Training	55.12%	78.67%	78.28%	0.036	0.047	55.08%	63.27%	63.96%	0.088	0.050
TIPRDC [21]	54.46%	76.96%	77.55%	0.034	0.047	54.12%	62.26%	63.51%	0.093	0.052
VAE-based Method [28]	55.39%	76.35%	77.28%	0.039	0.050	54.89%	56.11%	59.72%	0.091	0.051
InfoCensor (ours)	54.51%	**61.42%**	**60.95%**	**0.027**	**0.042**	54.10%	**49.24%**	**49.20%**	**0.082**	**0.049**
Twitter			**Identity**					**Gender**		
	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD
Standard Training	87.58%	45.77%	47.14%	-	-	87.58%	72.34%	74.15%	0.091	0.139
Adversarial Training	86.11%	33.28%	36.64%	-	-	86.89%	68.77%	71.97%	0.078	0.129
TIPRDC [21]	84.34%	35.30%	38.88%	-	-	87.00%	69.25%	70.76%	0.091	0.140
VAE-based Method [28]	85.42%	43.82%	48.73%	-	-	85.07%	73.30%	74.01%	0.079	0.132
InfoCensor (Ours)	85.65%	**26.62%**	**26.22%**	-	-	85.18%	**68.01%**	**68.35%**	**0.067**	**0.122**

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.

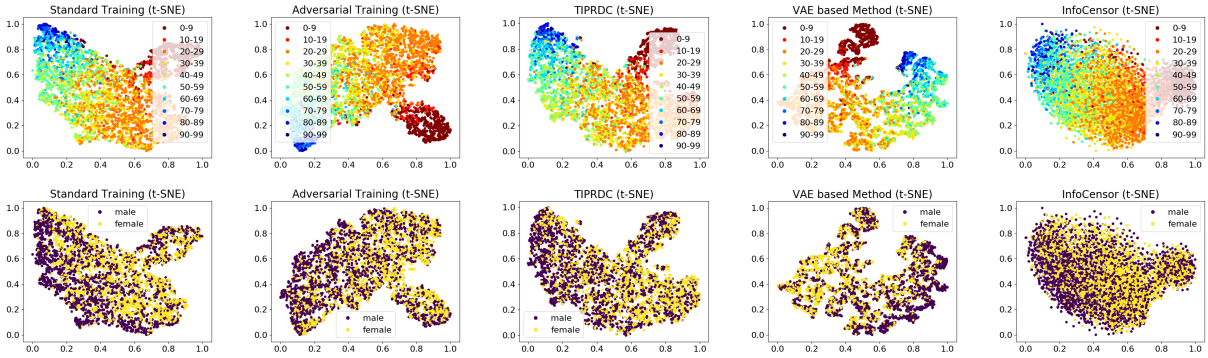

Figure 3: t-SNE (normalized) plots of the representations learned by standard training, adversarial training, VAE-based method[28], TIPRDC [21], and InfoCensor.

Health Heritage			**Age**					**Gender**		
	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD
InfoCensor	81.95%	20.97%	20.97%	0.092	0.216	82.83%	55.02%	55.05%	0.52×10^{-4}	0.009
Only $I(\boldsymbol{z};\boldsymbol{s})$	82.38%	21.40%	21.44%	0.102	0.231	82.73%	55.03%	55.05%	0.65×10^{-4}	0.010
Only $I(\hat{\boldsymbol{y}};\boldsymbol{s})$	82.24%	22.80%	22.87%	0.095	0.221	82.93%	55.02%	55.04%	0.41×10^{-4}	0.008
UTKFace			**Gender**					**Race**		
	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD	Original	Basic	De-censoring	$I(\hat{\boldsymbol{y}};\boldsymbol{s})$	Avg SPD
InfoCensor	54.51%	61.42%	60.95%	0.027	0.042	54.10%	49.24%	49.20%	0.082	0.049
Only $I(\boldsymbol{z};\boldsymbol{s})$	54.78%	63.08%	62.90%	0.035	0.048	54.44%	50.11%	50.04%	0.090	0.051
Only $I(\hat{\boldsymbol{y}};\boldsymbol{s})$	55.26%	79.78%	79.42%	0.030	0.041	54.89%	62.79%	61.82%	0.079	0.048

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.

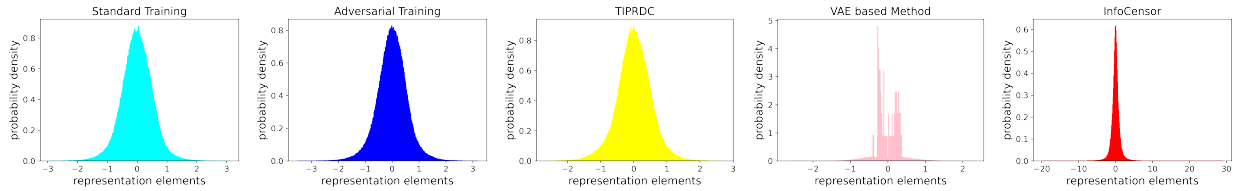

Figure 4: Distributions of the elements of the representations (probability density can be greater than 1). Note the elements of the representations learned by InfoCensor distribute in a much larger range (from -20 to 30), compared to the other baselines.

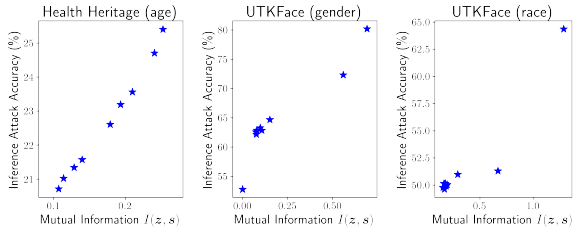

Figure 5: The relationship between sensitive attribute inference and $I(z;s)$ on the remaining data (except the auxiliary data). The models are trained by InfoCensor with different hyperparameter settings. For Health Heritage, if we set the sensitive attribute as "gender", $I(z;s)$ is small for all the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

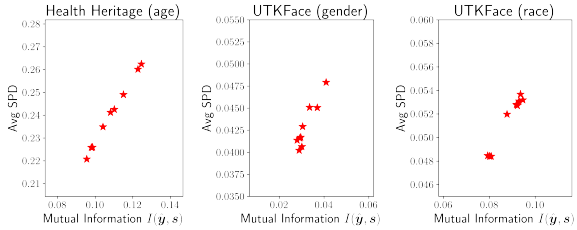

Figure 6: The relationship between averaged SPD and $I(\hat{y};s)$ (computed on the testing data). The models are trained by InfoCensor with different hyperparameter settings. For Health Heritage, if we set the sensitive attribute as "gender", $I(\hat{y};s)$ is small for all the hyperparameter settings under InfoCensor. Thus, the results on Health Heritage (gender) are not shown.

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 $I(\hat{y};s)$ are correlated since $\hat{y} = \mathbf{f}_{\phi}(z)$—Given $\mathbf{f}_{\phi}$ 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 λ and κ in our proposed objective.

Health	Original	Basic	De-censoring	$I(\hat{y};s)$
Gender	82.48%	55.04%	55.04%	0.72×10^{-4}
Age		21.27%	21.31%	0.099
UTKFace	Original	Basic	De-censoring	$I(\hat{y};s)$
Gender	54.46%	62.72%	62.41%	0.029
Race		49.70%	49.45%	0.086

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

Hyperparameter λ and κ. We train plenty of models under InfoCensor with different λ and κ and conduct the basic inference attack against those models. The corresponding results are shown in Fig. 7. As we increase λ and κ, we observe a decrease in the original accuracy, the inference attack accuracy, and $I(\hat{y};s)$ in most cases. If λ or κ 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 λ and κ 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.

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

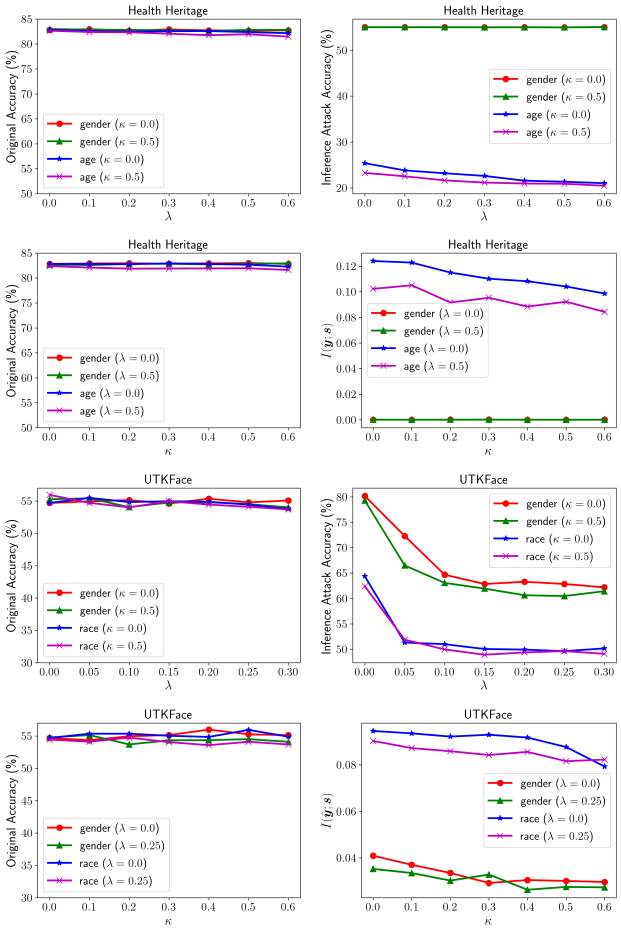

Figure 7: The performance of InfoCensor with different λ and κ.

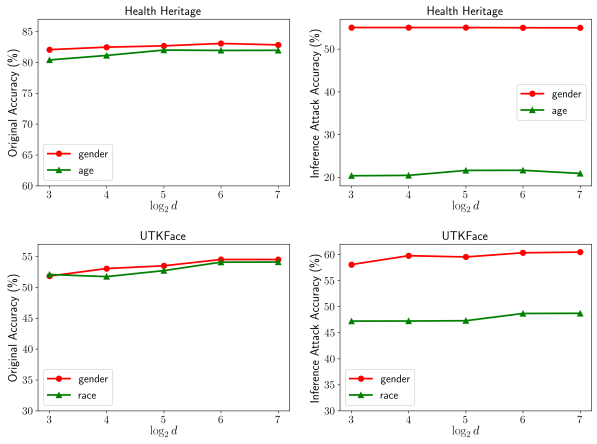

Figure 8: InfoCensor with different dimensions (d) of representations.

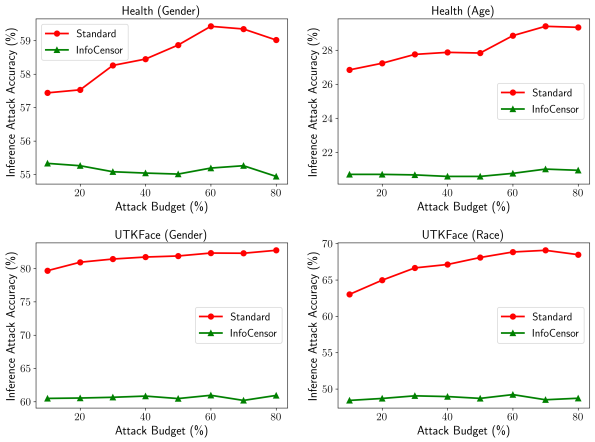

Figure 9: The attack results with different attack budgets on Standard Training and InfoCensor.

concerning multiple sensitive attributes $\{\boldsymbol{s}_i\}_{i=1}^{I}$ by modifying the objective (4) as

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{z}), \boldsymbol{y}) + \frac{1}{I}\sum_{i=1}^{I}[\lambda I(\boldsymbol{z}; \boldsymbol{s}_i) + \kappa I(\hat{\boldsymbol{y}}; \boldsymbol{s}_i)] \tag{27}$$

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 ACKNOWLEDGEMENT

We thank all the anonymous reviewers for their valuable comments.

REFERENCES

[1] [n.d.]. Health Heritage. https://www.kaggle.com/c/hhp.
[2] [n.d.]. Twitter (PAN). https://pan.webis.de/clef21/pan21-web/author-profiling.html.
[3] [n.d.]. UTKFace. https://susanqq.github.io/UTKFace/.
[4] Toon Calders and Sicco Verwer. 2010. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21, 2 (2010), 277–292.
[5] Jianfeng Chi, Emmanuel Owusu, Xuwang Yin, Tong Yu, William Chan, Patrick Tague, and Yuan Tian. 2018. Privacy partitioning: Protecting user data during the deep learning inference phase. *arXiv preprint arXiv:1812.02863* (2018).
[6] Maximin Coavoux, Shashi Narayan, and Shay B Cohen. 2018. Privacy-preserving Neural Representations of Text. In *2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1–10.
[7] Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. 2019. Flexibly fair representation learning by disentanglement. In *International Conference on Machine Learning*. PMLR, 1436–1445.
[8] Sever S Dragomir, ML Scholz, and J Sunde. 2000. Some upper bounds for relative entropy and applications. *Computers & Mathematics with Applications* 39, 9-10 (2000), 91–100.
[9] Harrison Edwards and Amos Storkey. 2015. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897* (2015).
[10] Ziv Goldfeld, Ewout Van Den Berg, Kristjan Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. 2019. Estimating Information Flow in Deep Neural Networks. In *International Conference on Machine Learning*. PMLR, 2299–2308.
[11] Otkrist Gupta and Ramesh Raskar. 2018. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications* 116 (2018), 1–8.
[12] Zecheng He, Tianwei Zhang, and Ruby B Lee. 2019. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 148–162.
[13] Yusuke Iwasawa, Kotaro Nakayama, Ikuko Yairi, and Yutaka Matsuo. 2017. Privacy Issues Regarding the Application of DNNs to Activity-Recognition using Wearables and Its Countermeasures by Use of Adversarial Training.. In *IJCAI*. 1930–1936.
[14] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. High accuracy and high fidelity extraction of neural networks. In *29th USENIX Security Symposium (USENIX Security 20)*. 1345–1362.
[15] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).
[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[17] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[18] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical review E* 69, 6 (2004), 066138.
[19] Nicholas D Lane and Petko Georgiev. 2015. Can deep learning revolutionize mobile sensing?. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. 117–122.
[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[21] Ang Li, Yixiao Duan, Huanrui Yang, Yiran Chen, and Jianlei Yang. 2020. TIPRDC: task-independent privacy-respecting data crowdsourcing framework for deep learning with anonymized intermediate representations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 824–832.
[22] Meng Li, Liangzhen Lai, Naveen Suda, Vikas Chandra, and David Z Pan. 2017. Privynet: A flexible framework for privacy-preserving deep neural network training. *arXiv preprint arXiv:1709.06161* (2017).
[23] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S Zemel. 2016. The Variational Fair Autoencoder. In *ICLR*.
[24] Cuicui Luo, Desheng Wu, and Dexiang Wu. 2017. A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence* 65 (2017), 465–470.
[25] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712* (2016).
[26] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 2018. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*. PMLR, 3384–3393.
[27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.
[28] Daniel Moyer, Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. 2018. Invariant representations without adversarial training. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 9102–9111.
[29] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4954–4963.
[30] Jonathan Scarlett and Volkan Cevher. 2019. An Introductory Guide to Fano's Inequality with Applications in Statistical Estimation. *arXiv preprint arXiv:1901.00555* (2019).
[31] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.
[32] Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 377–390.
[33] Congzheng Song and Vitaly Shmatikov. 2020. Overlearning Reveals Sensitive Attributes. In *8th International Conference on Learning Representations, ICLR 2020*.
[34] Jean-Baptiste Truong, Pratyush Maini, Robert Walls, and Nicolas Papernot. 2020. Data-Free Model Extraction. *arXiv preprint arXiv:2011.14779* (2020).
[35] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
[36] Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 585–596.
[37] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*. PMLR, 962–970.
[38] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International conference on machine learning*. PMLR, 325–333.
[39] Ligeng Zhu and Song Han. 2020. Deep leakage from gradients. In *Federated Learning*. Springer, 17–31.
[40] Sicheng Zhu, Xiao Zhang, and David Evans. 2020. Learning adversarially robust representations via worst-case mutual information maximization. In *International Conference on Machine Learning*. PMLR, 11609–11618.

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

Health Heritage	**Age**		**Gender**	
	Basic	De-censoring	Basic	De-censoring
Standard Training	29.19%	30.42%	59.58%	60.28%
Adversarial Training	28.65%	29.51%	58.70%	59.78%
TIPRDC [21]	29.69%	29.59%	59.03%	59.81%
VAE-based Method [28]	28.81%	28.60%	55.85%	56.67%
InfoCensor (ours)	**21.89%**	**21.94%**	**55.57%**	**55.52%**
UTKFace	**Gender**		**Race**	
	Basic	De-censoring	Basic	De-censoring
Standard Training	82.28%	82.08%	67.50%	67.64%
Adversarial Training	78.63%	79.15%	62.93%	64.05%
TIPRDC [21]	76.36%	77.74%	62.37%	63.21%
VAE-based Method [28]	75.43%	77.04%	54.78%	58.80%
InfoCensor (ours)	**61.59%**	**61.64%**	**49.60%**	**49.56%**
Twitter	**Identity**		**Gender**	
	Basic	De-censoring	Basic	De-censoring
Standard Training	42.67%	42.09%	70.95%	71.91%
Adversarial Training	29.05%	29.86%	64.91%	68.59%
TIPRDC [21]	32.03%	35.36%	66.65%	67.89%
VAE-based Method [28]	36.52%	42.63%	69.48%	70.29%
InfoCensor (Ours)	**23.25%**	**23.37%**	**64.22%**	**64.06%**

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, $I(\hat{\boldsymbol{y}}; \boldsymbol{s})$ is computed on the testing data.

by two fully-connected layers (similar to the architecture of LeNet [20]) as the feature extraction network. The kernel size is 3, and the stride is set as 1, for those convolutional layers. The decoder for the VAE-based method consists of a fully-connected layer and three deconvolutional layers.

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].