TAPPFL: TASK-AGNOSTIC PRIVACY-PRESERVING REPRESENTATION LEARNING FOR FEDERATED LEARNING AGAINST ATTRIBUTE INFERENCE ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL), a new collaborative learning paradigm, has been widely studied recently due to its property to collaboratively train data from different devices without sharing the raw training data. Nevertheless, recent studies show that an adversary (e.g., an honest-but-curious server) can still be possible to infer private information about devices' training data, e.g., sensitive attributes such as income, race, and sexual orientation. To mitigate the attribute inference attacks, various existing privacy-preserving FL methods can be adopted/adapted. However, all these existing methods have key limitations: they need to know the FL task in advance, or have intolerable computational overheads or utility losses, or do not have provable privacy guarantees. We aim to address all these issues and design a task-agnostic privacy-preserving FL (short for **TAPPFL**) method against attribute inference attacks from the information-theoretic perspective. Specifically, we formally formulate TAPPFL via two mutual information goals, where one goal learns task-agnostic data representations that contain the least information about the private attribute in each device's data, and the other goal is that the learnt representations include as much information as possible about the training data to maintain utility. However, it is intractable to compute exact mutual information in general. Then, we derive tractable mutual information bounds, and each bound can be parameterized via a neural network. Next, we alternatively train these parameterized neural networks to approximate the true mutual information and learn privacy-preserving representations for device data. We also derive theoretical privacy guarantees of our TAPPFL against worst-case attribute inference attacks. Extensive results on multiple datesets and applications validate the effectiveness of our TAPPFL to protect data privacy, maintain the FL utility, and be efficient as well.

1 INTRODUCTION

The emerging collaborative data analysis using federated learning (FL) (McMahan et al., 2017a) aims to address the data insufficiency problem, and has a great potential to protect data privacy as well. In FL, the participating devices keep, analyze, and train their data locally, and only share the trained models (e.g., model gradients or parameters), instead of the raw data, with a center server (e.g., cloud). The server updates its global model by aggregating the received device models, and broadcasts the updated global model to all participating devices such that all devices *indirectly* use all data from other devices. FL has been deployed by many companies such as Google Federated Learning (2022), Microsoft Federated Learning (2022), IBM Federated Learning (2022), and Alibaba Federated Learning (2022), and applied in various privacy-sensitive applications, including on-device item ranking (McMahan et al., 2017a), content suggestions for on-device keyboards (Bonawitz et al., 2019), next word prediction (Li et al., 2020), health monitoring (Rieke et al., 2020), and medical imaging (Kaissis et al., 2020). However, recent works have shown that, though only

sharing device models, it is still possible for an adversary (e.g., an honest-but-curious server) to perform the *attribute inference attack* (Aono et al., 2017; Ganju et al., 2018; Melis et al., 2019; Dang et al., 2021; Wainakh et al., 2022) —i.e., inferring the private/sensitive information (e.g., a person's gender, race, sexual orientation, income) of device's training data. Hence, designing privacy-preserving FL mechanisms to defend against the attribute inference attack is important and necessary.

Table 1: Comparisons of the PPFL methods.

Methods	Task-Agnostic	Efficient	Provable	Accurate
MPC			✓	✓
AdvT		✓		✓
MC	✓	✓		
DP	✓	✓	✓	
TAPPFL	✓	✓	✓	✓

To mitigate the issue, various existing privacy-preserving FL methods can be adopted/adapted, including *multi-party computation (MPC)* (Danner & Jelasity, 2015; Mohassel & Zhang, 2017; Bonawitz et al., 2017; Melis et al., 2019), *adversarial training (AdvT)* (Liu et al., 2019; Li et al., 2019; Oh et al., 2017; Kim et al., 2019), *model compression (MC)* (Zhu et al., 2019), and *differential privacy (DP)* (Pathak et al., 2010; Shokri & Shmatikov, 2015; Hamm et al., 2016; McMahan et al., 2018; Geyer et al., 2017). However, all these existing methods have key limitations, thus narrowing their applicability (see Table 1). Specifically, MPC and AdvT methods have to be designed for specific FL tasks (task-dependent, which cannot be achieved in many real-world applications). For instance, a set of users collaboratively perform a FL task, and a defender aims to protect the users' data from being inferred. However, the users require a stringent confidentiality about their data and do not let the defender know their learning task, as knowing the learning task only (e.g., face recognition) may somehow disclose some sensitive information (e.g., gender) about the data. Also, MPC methods have intolerable computational overheads and AdvT methods do not have provable privacy guarantees. MC and DP methods are task-agnostic, but both of them result in high utility losses (see Figure 4). In addition, MC methods do not have provable privacy guarantees.

In this paper, we aim to design a practical privacy-preserving FL mechanism against attribute inference attacks (termed **TAPPFL**) that is *task-agnostic*, *efficient*, *accurate*, and has *privacy guarantees* as well. Our main idea is based on information theory. Specifically, we formulate TAPPFL via two mutual information (MI) goals, where one MI goal learns low-dimensional representations for device data that contain the least information about the private attribute in each device's data—thus protecting attribute privacy, and the other MI goal ensures the learnt representations include as much information as possible about the training data—thus maintaining FL utility. Our TAPPFL is task-agnostic as our formulation does not know the FL task at hand. However, the true MI values are challenging to compute, due to that they deal with high-dimensional random variables and require to compute an intractable posterior distribution. Inspired by the MI neural estimators (Belghazi et al., 2018; Chen et al., 2016; Cheng et al., 2020), we recast calculating intractable exact MI values into deriving tractable (variational) MI bounds, where each variational bound is associated with a posterior distribution that can be parameterized via a neural network. Hence, estimating the true MI values reduces to training the parameterized neural networks. We further propose an alternative learning algorithm to train these neural networks and learn task-agnostic privacy-preserving representations for device data. We also derive provable privacy guarantees of our TAPPFL against worst-case attribute inference attacks. Finally, we evaluate our TAPPFL on multiple datasets and applications (e.g., Image, Loans, and Income). Experimental results validate that the learnt devices' data representations can be used to achieve high utility and maintain attribute privacy as well. Our key contributions can be summarized as follows:

- We propose a novel privacy-preserving FL method (TAPPFL) against attribute inference attacks based on information theory. Our TAPPFL is task-agnostic, efficient, accurate, and has provable privacy guarantees.
- We formulate our TAPPFL via mutual information objectives and design tractable variational bounds to estimate intractable mutual information.
- We evaluate our TAPPFL on various datasets and applications, and experimental results demonstrate the effectiveness of our TAPPFL for privacy-preserving representation learning for FL against attribute inference attacks and show the significant advantages over the compared baselines.

2 RELATED WORK

Privacy-preserving FL against inference attacks. *Secure multi-party computation* (Danner & Jelasity, 2015; Mohassel & Zhang, 2017; Bonawitz et al., 2017; Melis et al., 2019), *adversarial training* (Oh et al., 2017; Wu et al., 2018; Pittaluga et al., 2019; Liu et al., 2019; Kim et al., 2019), *model compression* (Zhu et al., 2019), and *differential privacy (DP)* (Pathak et al., 2010; Shokri & Shmatikov, 2015; Hamm et al., 2016; McMahan et al., 2018; Geyer et al., 2017; Wei et al., 2020) are the four typical privacy-preserving FL methods. For example, Bonawitz et al. (2017) design a secure multi-party aggregation for FL, where devices are required to encrypt their local models before uploading them to the server. However, it incurs an intolerable computational overhead and may need to know the specific FL task in advance. Adversarial training methods are inspired by GAN (Goodfellow et al., 2014). Particularly, these methods adopt adversarial learning to learn obfuscated features from the training data so that their privacy information cannot be inferred from a learnt model. However, these methods also need to know the FL task and lack of formal privacy guarantees. Zhu et al. (2019) apply gradient compression/sparsification to defend against privacy leakage from shared local models. However, to achieve a desirable privacy protection, such approaches require high compression rates, leading to intolerable utility losses. In addition, it does not have formal privacy guarantees. Shokri & Shmatikov (2015) propose a collaborative learning method where the sparse vector is adopted to achieve DP. However, the DP methods have high utility losses, in order to protect data privacy.

Mutual information (MI) estimation. Accurately estimating MI between high dimensional random variable is challenging (Belghazi et al., 2018). To address the challenge, recent methods (Alemi et al., 2017; Belghazi et al., 2018; Oord et al., 2018; Poole et al., 2019; Hjelm et al., 2019; Cheng et al., 2020) propose to first derive (upper or lower) MI bounds by introducing auxiliary variational distributions and then train parameterized neural networks to estimate variational distributions and approximate true MI. For instance, MINE (Belghazi et al., 2018) views MI as a KL divergence between the joint and marginal distributions, converts it into the dual representation, and derives a lower MI bound. Cheng et al. (2020) design a Contrastive Log-ratio Upper Bound of MI, which connects MI with contrastive learning (Oord et al., 2018), and estimates MI as the difference of conditional probabilities between positive and negative sample pairs.

3 BACKGROUND AND PROBLEM DEFINITION

3.1 FEDERATED LEARNING

The Federated Learning (FL) paradigm enables a server to coordinate the training of multiple local devices through multiple rounds of global communications, without sharing the local data. Suppose there are M devices $\mathcal{C} = \{C_i\}_{i=1}^{M}$ and a server S participating in FL. Each device C_i is assumed to own data samples $\mathbf{x}^i$ from a distribution $\mathcal{D}^i$ over the sample space $\mathcal{X}^i$. In each round t, each device C_i first downloads the previous round's global model (e.g., Θ_{t-1}) from the server, and then updates its local model (e.g., Θ_t^i) using the local data $\{\mathbf{x}^i\}$ and global model Θ_{t-1}. The server S then randomly collects a set of (e.g., K) current local models in devices (e.g., $\mathcal{C}_K$) and updates the global model for the next round using an aggregation algorithm. For example, when using the most common FedAvg (McMahan et al., 2017b), the server updates the global model as $\Theta_t \leftarrow \sum_{i \in \mathcal{C}_K} \frac{n_i}{\sum_{i \in \mathcal{C}_k} n_i} \Theta_t^i$, where n_i is the size of the training data of device C_i.

3.2 PROBLEM DEFINITION

We assume each device C_i's data has its own *private attribute* and denote it as $\mathbf{u}^i$. Each device C_i aims to learn a feature extractor $f_{\Theta^i} : \mathcal{X}^i \rightarrow \mathcal{R}^i$, parameterized by Θ^i, that maps data samples from input space $\mathcal{X}^i$ to the latent representation space $\mathcal{R}^i$; and we denote the learnt representation for a sample $\mathbf{x}^i$ as $\mathbf{r}^i = f_{\Theta^i}(\mathbf{x}^i)$. The learnt representations can be used for downstream tasks, e.g., next-word-prediction on smart phones (Li et al., 2020). We assume the server S is honest-but-curious and it has access to the feature extractor parameters $\{\Theta^i\}$ shared by devices. The server's purpose is to infer any private attribute $\mathbf{u}^i$ through the $\{\Theta^i\}$ without tampering the FL training process. Our goal is to learn the feature extractor f_{Θ^i} per device such that it protects the private attribute $\mathbf{u}^i$, and preserves the FL utility as well. For a general purpose, we assume the FL task is unknown (i.e., task-agnostic) to the defender (i.e., who learns the feature extractor).

4 DESIGN OF TAPPFL

In this section, we will design our task-agnostic privacy-preserving FL (TAPPFL) method against attribute inference attacks. Our TAPPFL is inspired by information theory and has provable privacy guarantees.

4.1 FORMULATING TAPPFL VIA MUTUAL INFORMATION OBJECTIVES

For ease of description, we choose a device C_i and demonstrate how to learn the privacy-preserving feature extractor f_{Θ^i} for C_i. Our goal is to transform the data $\mathbf{x}^i \sim \mathcal{D}^i$ into a representation $\mathbf{r}^i = f_{\Theta^i}(\mathbf{x}^i)$ that satisfies the following two goals:

- **Goal 1: Privacy protection.** $\mathbf{r}^i$ contains as less information as possible about the private attribute $\mathbf{u}^i$. Ideally, when $\mathbf{r}^i$ does not include information about $\mathbf{u}^i$, it is impossible for the server to infer $\mathbf{u}^i$ from $\mathbf{r}^i$.
- **Goal 2: Utility preservation.** $\mathbf{r}^i$ should include as much information about the training data $\mathbf{x}^i$ as possible. Ideally, when $\mathbf{r}^i$ retains the most information about $\mathbf{x}^i$, the model trained on $\mathbf{r}^i$ will have the same performance as the model trained on the raw $\mathbf{x}^i$, thus preserving utility.

We propose to formalize the above two goals via mutual information (MI). In information theory, MI is a measure of shared information between two random variables, and offers a quantifiable metric for the amount of information leakage on one variable given the other. Formally, we quantify the privacy protection and utility reservation goals using two MI objectives as follows:

$$\textbf{Achieving Goal 1:} \quad \min_{\Theta^i} I(\mathbf{r}^i; \mathbf{u}^i); \quad \textbf{Achieving Goal 2:} \quad \max_{\Theta^i} I(\mathbf{x}^i; \mathbf{r}^i | \mathbf{u}^i). \tag{1}$$

where $I(\mathbf{r}^i; \mathbf{u}^i)$ is the MI between $\mathbf{r}^i$ and $\mathbf{u}^i$, and we minimize such MI to maximally reduce the correlation between $\mathbf{r}^i$ and $\mathbf{u}^i$. $I(\mathbf{x}^i; \mathbf{r}^i | \mathbf{u}^i)$ is the MI between $\mathbf{x}^i$ and $\mathbf{r}^i$ given $\mathbf{u}^i$. We maximize such MI to keep the raw information in $\mathbf{x}^i$ as much as possible in $\mathbf{r}^i$ and remove the information that $\mathbf{x}^i$ contains about $\mathbf{u}^i$.

4.2 ESTIMATING MI VIA TRACTABLE VARIATION BOUNDS

The key challenge of solving the above two MI objectives is that calculating an MI between two arbitrary random variables is likely to be infeasible (Peng et al., 2018). To address it, we are inspired by the existing MI neural estimation methods (Alemi et al., 2017; Belghazi et al., 2018; Oord et al., 2018; Poole et al., 2019; Hjelm et al., 2019; Cheng et al., 2020), which convert the intractable exact MI calculations to the tractable variational MI bounds. Specifically, we first obtain a MI upper bound for privacy protection and a MI lower bound for utility preserving via introducing two auxiliary posterior distributions, respectively. Then, we parameterize each auxiliary distribution with a neural network, and approximate the true posteriors by minimizing the upper bound and maximizing the lower bound through training the involved neural networks.

Minimizing upper bound MI for privacy protection. We propose to adapt the variational upper bound CLUB proposed in (Cheng et al., 2020). Specifically, we have

$$I(\mathbf{r}^i; \mathbf{u}^i) \leq I_{vCLUB}(\mathbf{r}^i; \mathbf{u}^i) = \mathbb{E}_{p(\mathbf{r}^i, \mathbf{u}^i)}[\log q_{\Psi^i}(\mathbf{u}^i | \mathbf{r}^i)] - \mathbb{E}_{p(\mathbf{r}^i)p(\mathbf{u}^i)}[\log q_{\Psi^i}(\mathbf{u}^i | \mathbf{r}^i)], \tag{2}$$

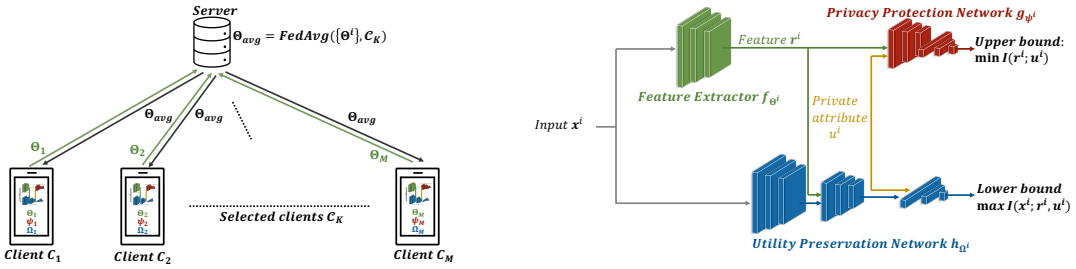
where $q_{\Psi^i}(\mathbf{u}^i | \mathbf{r}^i)$ is an auxiliary posterior distribution of $p(\mathbf{u}^i | \mathbf{r}^i)$ needing to satisfy the condition: $KL(p(\mathbf{r}^i, \mathbf{u}^i) || q_{\Psi^i}(\mathbf{r}^i, \mathbf{u}^i)) \leq KL(p(\mathbf{r}^i)p(\mathbf{u}^i) || q_{\Psi^i}(\mathbf{r}^i, \mathbf{u}^i))$. To achieve this, we need to minimize:

$$\begin{aligned} &\min_{\Psi^i} KL(p(\mathbf{r}^i, \mathbf{u}^i) || q_{\Psi^i}(\mathbf{r}^i, \mathbf{u}^i)) = \min_{\Psi^i} KL(p(\mathbf{u}^i | \mathbf{r}^i) || q_{\Psi^i}(\mathbf{u}^i | \mathbf{r}^i)) \\ &= \min_{\Psi^i} \mathbb{E}_{p(\mathbf{r}^i, \mathbf{u}^i)}[\log p(\mathbf{u}^i | \mathbf{r}^i)] - \mathbb{E}_{p(\mathbf{r}^i, \mathbf{u}^i)}[\log q_{\Psi^i}(\mathbf{u}^i | \mathbf{r}^i))] \Longleftrightarrow \max_{\Psi^i} \mathbb{E}_{p(\mathbf{r}^i, \mathbf{u}^i)}[\log q_{\Psi^i}(\mathbf{u}^i | \mathbf{r}^i)], \end{aligned} \tag{3}$$

Finally, our **Goal 1** for privacy protection is reformulated as solving the below min-max objective function:

$$\min_{\Theta^i} \min_{\Psi^i} I_{vCLUB}(\mathbf{r}^i; \mathbf{u}^i) \Longleftrightarrow \min_{\Theta^i} \max_{\Psi^i} \mathbb{E}_{p(\mathbf{r}^i, \mathbf{u}^i)}[\log q_{\Psi^i}(\mathbf{u}^i | \mathbf{r}^i)], \quad \mathbf{r}^i = f_{\Theta^i}(\mathbf{x}^i). \tag{4}$$

We note that Equation (4) can be interpreted as an *adversarial game* between: (1) an adversary q_{Ψ^i} (i.e., attribute inference classifier) who aims to infer the private attribute $\mathbf{u}^i$ from the representation $\mathbf{r}^i$; and (2) a defender (i.e., the feature extractor f_{Θ^i}) who aims to protect the private attribute $\mathbf{u}^i$ from being inferred.



(a) TAPPFL scheme (b) TAPPFL device training

Figure 1: (a) Scheme of our task-agnostic privacy-preservation representation learning framework for FL (TAPPFL); and (b) TAPPFL training on a single device.

Maximizing lower bound MI for utility preservation. We adopt the MI estimator proposed in (Nowozin et al., 2016) to estimate the lower bound of $I(\mathbf{x}^i; \mathbf{r}^i | \mathbf{u}^i)$. Specifically, we have

$$I(\mathbf{x}^i; \mathbf{r}^i | \mathbf{u}^i) \geq \mathbb{E}_{p(\mathbf{x}^i, \mathbf{r}^i, \mathbf{u}^i)} \left[\log q_{\Omega^i}[(\mathbf{x}^i | \mathbf{r}^i, \mathbf{u}^i)] \right] - I(\mathbf{x}^i; \mathbf{u}^i) + H(\mathbf{x}^i),$$

where q_{Ω^i} is an arbitrary auxiliary posterior distribution that aims to maintain the information in $\mathbf{x}^i$, and $H(\mathbf{x}^i)$ is the entropy of $\mathbf{x}^i$. Note that $I(\mathbf{x}^i; \mathbf{u}^i)$ and $H(\mathbf{x}^i)$ are constants as $\mathbf{x}^i$ and $\mathbf{u}^i$ are fixed. Hence, our **Goal 2** for utility preservation can be rewritten as the following max-max objective function:

$$\max_{\Theta^i} I(\mathbf{x}^i; \mathbf{r}^i | \mathbf{u}^i) \Longleftrightarrow \max_{\Theta^i} \max_{\Omega^i} \mathbb{E}_{p(\mathbf{x}^i, \mathbf{r}^i, \mathbf{u}^i)} \left[\log q_{\Omega^i}[(\mathbf{x}^i | \mathbf{r}^i, \mathbf{u}^i)] \right], \quad \mathbf{r}^i = f_{\Theta^i}(\mathbf{x}^i). \tag{5}$$

We note that Equation (5) can be interpreted as a *cooperative game* between the feature extractor f_{Θ^i} and q_{Ω^i} who aim to preserve the utility collaboratively.

Objective function of TAPPFL. By combining Equations (4) and (5) and considering all devices, our final objective function of learning the task-agnostic privacy-preserving representations in FL is as follows:

$$\sum_{C_i \in \mathcal{C}} \min_{\Theta^i} \Big(\lambda_i \max_{\Psi^i} \mathbb{E}_{p(\mathbf{u}^i, \mathbf{r}^i)} \left[\log q_{\Psi^i}(\mathbf{u}^i | f_{\Theta^i}(\mathbf{x}^i)) \right] - (1 - \lambda_i) \min_{\Omega^i} \mathbb{E}_{p(\mathbf{x}^i, \mathbf{r}^i, \mathbf{u}^i)} \left[\log q_{\Omega^i}[(\mathbf{x}^i | f_{\Theta^i}(\mathbf{x}^i), \mathbf{u}^i)] \right] \Big), \tag{6}$$

where $\lambda_i \in [0, 1]$ achieves a tradeoff between privacy and utility for the device C_i. I.e., a larger λ_i indicates a stronger attribute privacy protection, while a smaller λ_i indicates a better utility preservation for C_i.

4.3 Implementation via training parameterized neural networks

In practice, Equation (6) can be solved via training three parameterized neural networks, i.e., the feature extractor f_{Θ^i}, the privacy-protection network g_{Ψ^i} associated with the auxiliary distribution q_{Ψ^i}, and the utility-preservation network h_{Ω^i} associated with the auxiliary distribution q_{Ω^i}, using sampled data from each device C_i. Specifically, in each device C_i, we first collect a set of samples $\{\mathbf{x}_j^i\}$ and the associated private attributes $\{\mathbf{u}_j^i\}$ from $\mathcal{D}^i$. Note that, as our TAPPFL is task-agnostic, we do not know the sample labels for the FL task. With it, we can then approximate the expectation terms in Equation (6). Specifically, we approximate the expectation associated with the auxiliary distribution q_{Ψ^i} as $\mathbb{E}_{p(\mathbf{u}^i, f_{\Theta^i}(\mathbf{x}^i))} \log q_{\Psi^i}(\mathbf{u}^i | f_{\Theta^i}(\mathbf{x}^i)) \approx -\sum_j CE(\mathbf{u}_j^i, g_{\Psi^i}(f_{\Theta^i}(\mathbf{x}_j^i)))$, where $CE(\cdot)$ means the cross-entropy error function. Moreover, we approximate the expectation associated with the auxiliary distribution q_{Ω^i} via the *Jensen-Shannon* (JSD) MI estimator (Hjelm et al., 2019; Nowozin et al., 2016). That is, $\mathbb{E}_{p(\mathbf{x}^i, f_{\Theta^i}(\mathbf{x}^i), \mathbf{u}^i)} \log q_{\Omega^i}(\mathbf{x}^i | f_{\Theta^i}(\mathbf{x}_i), \mathbf{u}^i) \approx I^{(JSD)}_{\Theta^i, \Omega^i}(\mathbf{x}^i; f_{\Theta^i}(\mathbf{x}^i), \mathbf{u}^i) = \mathbb{E}_{(\mathbf{x}^i, \mathbf{u}^i)}[-\text{sp}(-h_{\Omega^i}(\mathbf{x}^i, f_{\Theta^i}(\mathbf{x}^i), \mathbf{u}^i)] - \mathbb{E}_{(\mathbf{x}^i, \mathbf{u}^i, \mathbf{x}'^i)}[\text{sp}(h_{\Omega^i}(\mathbf{x}'^i, f_{\Theta}(\mathbf{x}^i), \mathbf{u}^i)]$ with $\mathbf{x}'^i$ an independent and random sample from the same distribution as $\mathbf{x}^i$, and the expectation can be replaced by the samples $\{\mathbf{x}_j^i, \mathbf{x}'^i_j, \mathbf{u}_j^i\}$. $\text{sp}(z) = \log(1 + \exp(z))$ is the softplus function.

Figure 1 illustrates our task-agnostic privacy-preserving learning framework for FL. Our TAPPFL needs to simultaneously train three neural networks, i.e., the feature extractor f_{Θ^i}, the privacy-protection network g_{Ψ^i}, and the utility-preservation network h_{Ω^i}, in each device C_i. In particular, the server first initializes a global model Θ_0 and broadcasts Θ_0 to all devices; and the devices initializes $\{\Psi_0^i\}$ and $\{\Omega_0^i\}$ locally. Then the training procedure involves two iterative steps. For example, in the t-th round: In Step I, each device updates Θ_t^i using the received Θ_{t-1} from the server, and locally updates Ψ_t^i and Ω_t^i using Ψ_{t-1}^i and Ω_{t-1}^i based on its training data; and the devices send the updated $\{\Theta_t^i\}$ to the server. In Step II, the server selects a set of $\{\Theta_t^i\}$ and updates the global model Θ_t by aggregating these models via, e.g., Fedvg (McMahan et al., 2017b), and broadcasts Θ_t to all devices. We repeat the two steps alternately until convergence or reaching the maximum number of iterations. Algorithm 1 in Appendix details the TAPPFL training process.

5 Theoretical Results

5.1 Inherent Tradeoff between Utility Preservation and Attribute Privacy Leakage

We consider the attribute has a binary value and the primary FL task is binary classification. We will leave it as a future work to generalize our results to multi-value attributes and multi-class classification.

Let $\mathcal{A}$ be the set of all binary attribute inference classifiers, i.e., $\mathcal{A} = \{A : \mathbf{r}^i \in \mathcal{R}^i \rightarrow \{0,1\}, \forall C_i\}$. Let $\mathcal{D}^i$ be a joint distribution over the input $\mathbf{x}^i$, sensitive attribute $\mathbf{u}^i$, and label $\mathbf{y}^i$ for device C_i. W.l.o.g, we assume the representation space is bounded, i.e., $\max_{C_i \in \mathcal{C}} \max_{\mathbf{r}^i \in \mathcal{R}^i} \|\mathbf{r}^i\| \leq R$. Moreover, we denote the binary task classifier as $c : \mathbf{r}^i \rightarrow \{0,1\}$, which predicts data labels based on the learnt representation. We further define the *advantage* of the worst-case adversary with respect to the joint distribution $\mathcal{D}^i$ as below:

$$\text{Adv}_{\mathcal{D}^i}(\mathcal{A}) = \max_{A \in \mathcal{A}} |\text{Pr}_{\mathcal{D}^i}(A(\mathbf{r}^i) = a | \mathbf{u}^i = a) - \text{Pr}_{\mathcal{D}^i}(A(\mathbf{r}^i) = a | \mathbf{u}^i = 1 - a)|, \forall a = \{0,1\}. \tag{7}$$

If $\text{Adv}_{\mathcal{D}^i}(\mathcal{A}) = 1$, this means an adversary can *completely* infer the privacy attribute through the learnt representations. On the other hand, if $\text{Adv}_{\mathcal{D}^i}(\mathcal{A}) = 0$, an adversary can obtain the inference performance that is *random guessing*. Our goal is thus to learn the representations such that $\text{Adv}_{\mathcal{D}^i}(\mathcal{A})$ is small per device.

Theorem 1. *Let $\mathbf{r}^i$ be the representation with a bounded norm R (i.e., $\max_{\mathbf{r}^i \in \mathcal{R}^i} \|\mathbf{r}^i\| \leq R$) learnt by the feature extractor f_{Θ_i} for device C_i's data $\mathbf{x}^i$, and $\mathcal{A}$ be the set of all binary attribute inference classifiers. Assume the primary task classifier c is C_L-Lipschitz, i.e., $\|c\|_L \leq C_L$. Then, the device C_i's utility loss (i.e., classification error) err_i can be bounded as:*

$$err_i = CE_{\mathbf{u}^i=0}(\mathbf{y}^i, c(\mathbf{r}^i)) + CE_{\mathbf{u}^i=1}(\mathbf{y}^i, c(\mathbf{r}^i)) \geq \Delta_{\mathbf{y}^i|\mathbf{u}^i} - 2R \cdot C_L \cdot Adv_{\mathcal{D}^i}(\mathcal{A}), \tag{8}$$

where $CE_{\mathbf{u}^i=a}(\mathbf{y}^i, c(\mathbf{r}^i))$ is the conditional cross-entropy error of predicting $\mathbf{y}^i$ using $\mathbf{r}^i$ given the attribute $\mathbf{u}^i = a \in \{0,1\}$; $\Delta_{y^i|\mathbf{u}^i} = |Pr_{\mathcal{D}^i}(y^i = 1 | \mathbf{u}^i = 0) - Pr_{\mathcal{D}^i}(y^i = 1 | \mathbf{u}^i = 1)|$ is a device-dependent constant.

Remark. Theorem 1 says that, for a device-dependent constant $\Delta_{\mathbf{y}^i|\mathbf{u}^i}$, any primary task classifier using representations learnt by the feature extractor f_{Θ_i} has to incur a classification error on at least a private attribute. Specifically, the smaller/larger the advantage $\text{Adv}_{\mathcal{D}_i}(\mathcal{A})$ is, the larger/smaller the lower bound error. Note that the lower bound is independent of the adversary, meaning it covers the *worst-case* adversary. Hence, Equation (8) reflects an inherent trade-off between utility preservation and attribute privacy leakage.

5.2 Provable Privacy Guarantees against Attribute Inference Attacks

The attribute inference accuracy incurred by the worst-case adversary is bounded in the following theorem:

Theorem 2. *Let Θ_*^i (resp. $\mathbf{r}_*^i$) be the learnt optimal feature extractor parameters (resp. optimal representations) by Equation (6) for device C_i's data. Define $H_*^i = H(\mathbf{u}^i | \mathbf{r}_*^i)$. Then, for any attribute inference adversary $\mathcal{A} = \{A : \mathbf{r}^i \rightarrow \mathbf{u}^i\}$, $Pr(A(\mathbf{r}_*^i) = \mathbf{u}^i) \leq 1 - \frac{H_*^i}{2\log_2(\frac{6}{H_*^i})}$.*

Remark. Theorem 2 shows that when the conditional entropy $H_*^i = H(\mathbf{u}^i | \mathbf{r}_*^i)$ is larger, the attribute inference accuracy induced by any adversary is smaller, i.e., the less attribute privacy is leaked. From another perspective, as $H(\mathbf{u}^i | \mathbf{r}_*^i) = H(\mathbf{u}^i) - I(\mathbf{u}^i; \mathbf{r}_*^i)$, achieving the largest $H(\mathbf{u}^i | \mathbf{r}_*^i)$ indicates minimizing the mutual information $I(\mathbf{u}^i; \mathbf{r}_*^i)$ —This is exactly our **Goal 1** aims to achieve.

Table 2: Testing accuracy vs. attribute inference accuracy on the considered four datasets.

CIFAR10			
λ	Testing Acc	Attr. Infer. Acc	Gap
Private attribute: Animal or not (binary)			
0	0.82	0.74	0.26
0.25	0.80	0.64	0.14
0.5	0.76	0.60	0.10
0.75	0.76	0.56	0.06
1	0.48	0.52	0.02

Loans			
λ	Testing Acc	Attr. Infer. Acc	Gap
Private attribute: Race (binary)			
0	0.9993	0.833	0.333
0.25	0.9993	0.733	0.233
0.5	0.9993	0.733	0.233
0.75	0.8000	0.633	0.133
1	0.7333	0.567	0.067

Adult income			
λ	Testing Acc	Attr. Infer. Acc	Gap
Private attribute: Gender (binary)			
0	0.875	0.700	0.20
0.25	0.750	0.550	0.05
0.5	0.750	0.550	0.05
0.75	0.825	0.550	0.05
1	0.700	0.525	0.025

Adult income			
λ	Testing Acc	Attr. Infer. Acc	Gap
Private attribute: Marital status (7 values)			
0	0.825	0.375	0.232
0.25	0.800	0.275	0.112
0.5	0.800	0.250	0.107
0.75	0.725	0.243	0.043
1	0.700	0.175	0.032

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUP

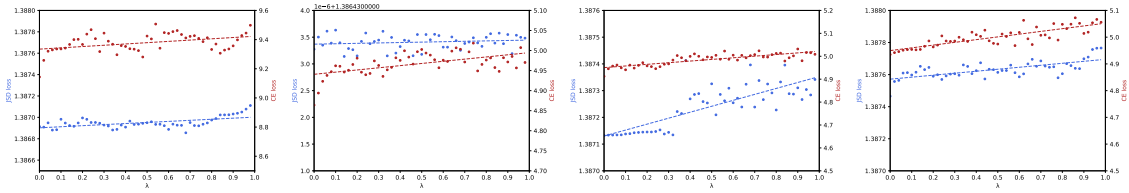
Datasets and applications. We evaluate our TAPPFL using three datasets from different applications. CIFAR-10 (Krizhevsky, 2009) is a widespread image dataset. The primary task is to predict the label of the image, while the private attribute is a binary attribute indicating if an image belongs to an animal or not. For the Loans dataset (Hardt et al., 2016), the primary task is to accurately predict the affordability of the person asking for the loan while protecting their race. Finally, for the Adult Income dataset (Dua & Graff, 2017), predicting whether the income of a person is above \$50,000 or not is the primary task. The private attributes are the gender and the marital status. More detailed descriptions of these datasets and the training/testing sets can be found on Appendix B.

Parameter settings. We use a total of 100 devices participating in FL training. By default, the server randomly selects 10% devices and uses FedAvg (McMahan et al., 2017b) to aggregate devices' feature extractor parameters in each round. In each device, we train the three parameterized neural networks via the Stochastic Gradient Descent (SGD) algorithm, where we set the local batch size to be 10 and use 10 local epochs, and the learning rate in SGD is 0.01. A detailed architecture of each neural network can be found in Table 3 in Appendix B. Before the overall learning, we first pretrain the feature extractor network only to obtain a good initialization, i.e., high utilty. The number of global rounds is set to be 20. In TAPPFL, for simplicity, we set $\lambda_i = \lambda$ for all devices and all devices share the same private attribute. The TAPPFL algorithm is implemented in PyTorch. We use the Chameleon Cloud platform offered by the NSF (Keahey et al., 2020) (CentOS7-CUDA 11 with Nvidia Rtx 6000). Our code is available at `https://github.com/anonymousesubmission`.

Evaluation metrics. We evaluate TAPPFL on both utility preservation and privacy protection. We use the testing accuracy (i.e., device's feature extractor + utility network on the primary task's test set) to measure utility preservation; and attribute inference accuracy (i.e., device's feature extractor + privacy network on the privacy task's test set) to measure the privacy leakage. The larger testing accuracy, the better utility preservation; and the attribute inference accuracy closer to random guessing, the less attribute privacy leakage.

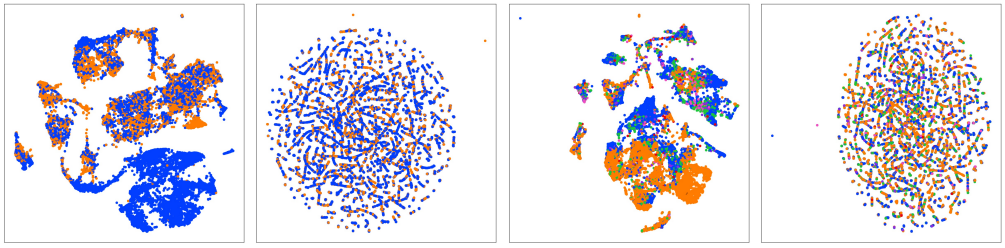
6.2 EXPERIMENTAL RESULTS

Utility-privacy tradeoff. Accordingly to Equation (6), when $\lambda = 0$ the first term of the objective function is disregarded, meaning that the protection of the private attribute is not considered. On the contrary, the second term is disappeared when $\lambda = 1$, or in other words, we only consider protecting the private attribute and utility is not preserved. Our goal is to achieve a better trade-off by tuning λ within $[0, 1]$, which allows

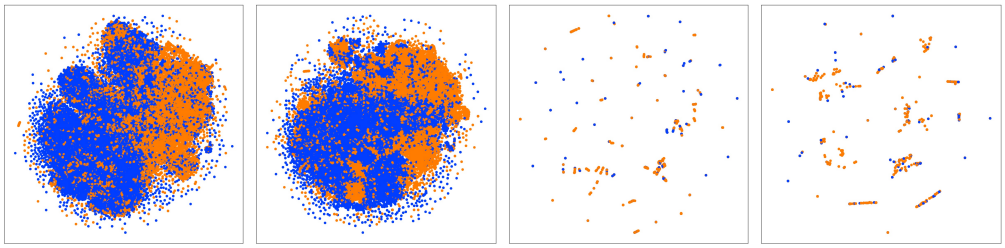


(a) CIFAR10: Animal or not (b) Loans: Race (c) Adult: Gender (d) Adult: Marital status

Figure 2: Mutual information vs. λ. Note that the CE loss and JSD loss are inversely proportional to the MIs in the two Goals. Each point corresponds to a CE loss or JSD loss at a selected λ.



(a) Adult (Gender): Raw input (b) Adult (Gender): Learnt rep. (c) Adult (Marital status): Raw input (d) Adult (Marital status): Learnt rep.



(e) CIFAR10: Raw input (f) CIFAR10: Learnt rep. (g) Loans: Raw input (h) Loans: Learnt rep.

Figure 3: 2D t-SNE embeddings of learnt representations by TAPPFL and of the raw input. Each color corresponds to a private attribute value.

preserving the FL utility and protecting the attribute privacy at the same time. Table 2 shows the testing accuracy and average attribute inference accuracy of all devices in the considered datasets, where we set five different λ values, i.e., 0, 0.25, 0.5, 0.75, and 1.0. We also show the gap between the attribute inference accuracy and the random guessing. The smaller the gap, the better the privacy protection. Ideally, when there is no gap, the learnt representation by our TAPPFL does not allow the adversary (i.e., the server) to infer *any* information related to the private attribute. Specifically, we have the following observations: 1) The testing accuracy is the largest when $\lambda = 0$, hence the utility is maintained the most. However, the attribute inference accuracy is also the highest, indicating leaking the most attribute privacy. 2) The attribute inference accuracy is the closest to random guessing when $\lambda = 1$, meaning the attribute privacy is protected the most. However, the testing accuracy is also the smallest, indicating the utility is not well maintained. 3) When $0 < \lambda < 1$, our TAPPFL achieves both reasonable testing accuracy and attribute inference accuracy—This indicates TAPPFL has a better utility-privacy tradeoff. Note that our TAPPFL does not know the labels of the primary task and learns the task-agnostic representations for device data during the entire training.

Mutual information scores vs. tradeoff parameter λ. Furthermore, we analyze our TAPPFL via plotting the two MI scores (i.e., the CE loss associated with **Goal 1 (privacy protection)**) and JSD loss associated

with **Goal 2 (utility preservation)** vs. λ. Note that the CE loss and JSD loss are inversely proportional to the MI in the two goals. Figure 2 shows the results on CIFAR10. Each point corresponds to either a CE loss or JSD loss at a selected λ. The tendency of these scores in function of λ is presented by a trend line, which is computed using a least squares polynomial fit of first degree. We observe that: 1) When the trade-off parameter λ is low, the privacy protection is not carefully considered, which is translated into a high MI between the learnt representation and the private attribute, thus the CE loss is relatively small. On the other hand, the utility preservation is maximized, resulting also into a high MI of the input given the learnt representation and the private attribute, as the JSD loss is relatively small. 2) Contrarily, for high values of λ, the privacy is largely protected in exchange for a large utility loss. Specifically, as λ increases, the CE between the private attribute and the learnt representation increases, which is translated into a decrease of their MI, thus better protecting attribute privacy. Though not easily appreciated in the curves, the JSD loss tends to increase, thus reducing the utility.

Visualization of the learnt representations. In this experiment, we leverage the t-SNE embedding algorithm (Van der Maaten & Hinton, 2008) to visualize the learnt representations by our trained feature extractor for the device data, and those without our feature extractor. λ is chosen in Table 2 that achieves the best utility-privacy tradeoff. Figure 3 shows the 2D t-SNE visualization results, where each color corresponds to a private attribute value. We can observe that the 2D t-SNE embeddings of the raw input data form some clusters for the private attributes, meaning the private attributes can be easily inferred, e.g., the t-SNE embedded representations via training a multi-class classifier. On the contrary, the 2D t-SNE embeddings of the learnt representations by our TAPPFL for different attribute values are completely mixed, which thus makes it difficult for a malicious server to infer the private attributes from the learnt representations.

Comparison with the state-of-the-art defenses. In our last experiment, we compare our TAPPFL with the two task-agnostic privacy protection methods, i.e., differential privacy (DP) (Wei et al., 2020) and model compression (MC) (Zhu et al., 2019) (See Table 1). MC prunes the devices' feature extractor parameters whose magnitude are smaller than a threshold, and the devices only share parameters larger than the threshold to the server. DP protects privacy with theoretical guarantees. Specifically, DP randomly injects noise into the feature extractor's parameters and uploads the noisy parameters to the server. The server then performs the aggregation using the noisy parameters. Here, we consider applying the Gaussian noise and Laplacian noise to develop two DP baselines, i.e., DP-Gaussian and DP-Laplace (note that the DP protection in (Wei et al., 2020) is very weak due to very high ϵ such as 50 and 100). Thus, we tune the hyperparameter, i.e., noise variance in DP and pruning rate in MC, such that DP and MC have the same attribute inference accuracy as TAPPFL, and then compare their utility/testing accuracy. Figure 4 shows the comparison results on CIFAR10, where we set five attribute inference accuracies as 0.55, 0.60, 0.65, 0.70, and 0.75, respectively. We can see that our TAPPFL achieves the best privacy-utility tradeoff and is significantly better than the compared defenses.

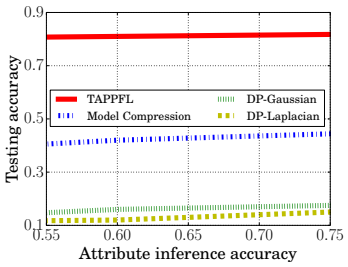


Figure 4: Compared defense results on CIFAR10.

7 CONCLUSION

We study privacy-preserving federated learning (FL) against the attribute inference attack, i.e., an honest-but-curious server infers sensitive information in the device data from shared device models. To this end, we design a task-agnostic and provable privacy-preserving representation learning framework for FL (TAPPFL) from the information-theoretic perspective. TAPPFL is formulated via two mutual information goals: one goal learns low-dimensional representations for device data that contain the least information about the data's private attribute, and the other one includes as much information as possible about the training data, in order to maintain FL utility. TAPPFL can also bound the privacy leakage of the private attributes. Extensive results on various datasets from different applications show that, by tuning the utility-privacy tradeoff parameter, our TAPPFL can well protect the attributes (i.e., attribute inference accuracy is close to random guessing), and obtains a high utility. TAPPFL is also shown to significantly outperform the state-of-the-art defenses.

REFERENCES

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2017.

Alibaba Federated Learning. `https://federatedscope.io/`, 2022.

Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning: Revisited and enhanced. In *International Conference on Applications and Techniques in Information Security*, 2017.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *ICML*, 2018.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *ACM SIGSAC Conference on Computer and Communications Security*, 2017.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 2019.

Chris Calabro. *The exponential complexity of satisfiability problems*. University of California, San Diego, 2009.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *ICML*, 2020.

Trung Dang, Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, Peter Chin, and Françoise Beaufays. Revealing and protecting labels in distributed training. In *NeurIPS*, 2021.

Gábor Danner and Márk Jelasity. Fully distributed privacy preserving mini-batch gradient descent learning. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, 2015.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml/datasets/adult`.

Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *CCS*, 2018.

Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv*, 2017.

Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

Google Federated Learning. `https://federated.withgoogle.com/`, 2022.

Jihun Hamm, Yingjun Cao, and Mikhail Belkin. Learning privately from multiparty data. In *International Conference on Machine Learning*, 2016.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 33233331, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

IBM Federated Learning. `https://www.ibm.com/docs/en/cloud-paks/cp-data/4.0?topic=models-federated-learning`, April 2022.

Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2020.

Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. Lessons learned from the chameleon testbed. In *USENIX ATC*. 2020.

Tae-hoon Kim, Dongmin Kang, Kari Pulli, and Jonghyun Choi. Training with the invisibles: Obfuscating images to share safely for learning visual recognition models. *arXiv preprint arXiv:1901.00098*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Ang Li, Jiayi Guo, Huanrui Yang, and Yiran Chen. Deepobfuscator: Adversarial training framework for privacy-preserving image classification. *arXiv preprint arXiv:1909.04126*, 2019.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 2020.

Peiyuan Liao, Han Zhao, Keyulu Xu, Tommi Jaakkola, Geoffrey J Gordon, Stefanie Jegelka, and Ruslan Salakhutdinov. Information obfuscation of graph neural networks. In *ICML*, 2021.

Sicong Liu, Junzhao Du, Anshumali Shrivastava, and Lin Zhong. Privacy adversarial network: Representation learning for mobile data privacy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):1–18, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282, 2017a.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017b.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE SP*, 2019.

Microsoft Federated Learning. `https://www.microsoft.com/en-us/research/blog/flute-a-scalable-federated-learning-simulation-platform/`, May 2022.

Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

Seong Joon Oh, Mario Fritz, and Bernt Schiele. Adversarial image perturbation for privacy protection a game theory perspective. In *ICCV*, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018.

Manas Pathak, Shantanu Rane, and Bhiksha Raj. Multiparty differential privacy via aggregation of locally trained classifiers. In *Advances in Neural Information Processing Systems*, 2010.

Xue Bin Peng, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine. Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. *arXiv preprint arXiv:1810.00821*, 2018.

Francesco Pittaluga, Sanjeev Koppal, and Ayan Chakrabarti. Learning privacy preserving encodings through adversarial training. In *WACV*, 2019.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. In *ICML*, 2019.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, Sebastian Ourselin, Micah Sheller, Ronald Summer, Andrew Trask, Daguang Xu, Maximilian Baust, and M Jorge Cardoso. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321. ACM, 2015.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Aidmar Wainakh, Fabrizio Ventola, Till Müßig, Jens Keim, Carlos Garcia Cordero, Ephraim Zimmer, Tim Grube, Kristian Kersting, and Max Mühlhäuser. User-level label leakage from gradients in federated learning. In *PTES*, 2022.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 2020.

Zhenyu Wu, Zhangyang Wang, Zhaowen Wang, and Hailin Jin. Towards privacy-preserving visual recognition via adversarial training: A pilot study. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 606–624, 2018.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, 2019.

Algorithm 1 Task-agnostic privacy-preserving rep. learning for FL against attr. infer. attacks (**TAPPFL**)

Input: ρ: fraction of participating devices; $\mathcal{C} = \{C_i\}_{i=1}^{M}$: M total devices; B: batch size; E: #local epochs; T: #global rounds; lr_1, lr_2, lr_3: learning rates of the feature extractor NN, privacy protection NN, and utility preservation NN.
Output: $\{\Theta_i^T\}_{i=1}^{M}$, $\{\Psi_i^T\}_{i=1}^{M}$, $\{\Omega_i^T\}_{i=1}^{M}$
Initialization: $\{\Theta_i^0, \Psi_i^0, \Omega_i^0\}_{i=1}^{M}$. E.g., $\{\Theta_i^0\}_{i=1}^{M}$ are initialized via pretraining each feature extractor NN.
for global round $t = 0, 1, 2, \cdots, T-1$ **do**
  for each device $C_i \in \mathcal{C}$ **do**
    $\Theta_i^{t+1} \leftarrow$ **DeviceUpdate**(i, Θ_i^t)
  end for
  $\Theta^{t+1} \leftarrow$ **ServerUpdate**$(\{\Theta_i^{t+1}\}, \rho)$
  Set $\{\Theta_i^{t+1}\} \leftarrow \Theta^{t+1}$
end for
DeviceUpdate(i, Θ^t) :
$\Theta_i^t \leftarrow \Theta^t$
$CE\_loss = CE(\mathbf{u}^i, f_{\Theta_i^t}(\mathbf{x}^i))$
$JSD\_loss = -I^{(JSD)}(\mathbf{x}^i, f_{\Theta_i^t}(\mathbf{x}^i), \mathbf{u}^i)$
for local epoch $e = 1, 2, \cdots, E$ **do**
  $\mathcal{B} \leftarrow$ Split device C_i's data into mini-batches of size B
  for each min-batch $b \in \mathcal{B}$ **do**
    $\Psi_i^{t+1} \leftarrow \Psi_i^t - lr_1 \cdot \partial CE\_loss / \partial \Psi_i^t$
    $\Omega_i^{t+1} \leftarrow \Omega_i^t + lr_2 \cdot \partial JSD\_loss / \partial \Omega_i^t$
    $\Theta_i^{t+1} \leftarrow \Theta_i^t + lr_3 \cdot \partial\big(\lambda CE\_loss + (1-\lambda) JSD\_loss\big) / \partial \Theta_i^t$
  end for
end for
ServerUpdate$(\{\Theta_i^{t+1}\}, \rho)$:
$\mathcal{C}_K \leftarrow$ randomly select $K = \rho \cdot M$ devices
$\Theta^{t+1} \leftarrow \frac{1}{K} \sum_{k \in \mathcal{C}_K} \Theta_k^{t+1}$

A PROOFS

A.1 PROOF OF THEOREM 1

We first introduce the following definitions and lemmas that will be used to prove Theorem 1.

Definition 1 (Total variance (TV) distance)**.** *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two distributions over the same sample space Γ, the TV distance between $\mathcal{D}_1$ and $\mathcal{D}_2$ is defined as: $d_{TV}(\mathcal{D}_1, \mathcal{D}_2) = \max_{E \subseteq \Gamma} |\mathcal{D}_1(E) - \mathcal{D}_2(E)|$.*

Definition 2 (1-Wasserstein distance)**.** *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two distributions over the same sample space Γ, the 1-Wasserstein distance between $\mathcal{D}_1$ and $\mathcal{D}_2$ is defined as $W_1(\mathcal{D}_1, \mathcal{D}_2) = \max_{\|f\|_L \leq 1} | \int_\Gamma f d\mathcal{D}_1 - \int_\Gamma f d\mathcal{D}_2 |$, where $\| \cdot \|_L$ is the Lipschitz norm of a real-valued function.*

Definition 3 (Pushforward distribution)**.** *Let $\mathcal{D}$ be a distribution over a sample space and g be a function of the same space. Then we call $g(\mathcal{D})$ the pushforward distribution of $\mathcal{D}$.*

Lemma 1 (Contraction of the 1-Wasserstein distance)**.** *Let g be a function defined on a space and L be constant such that $\|g\|_L \leq C_L$. For any distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ over this space, $W_1(g(\mathcal{D}_1), g(\mathcal{D}_2)) \leq C_L \cdot W_1(\mathcal{D}_1, \mathcal{D}_2)$.*

Lemma 2 (1-Wasserstein distance on Bernoulli random variables)**.** *Let y_1 and y_2 be two Bernoulli random variables with distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. Then, $W_1(\mathcal{D}_1, \mathcal{D}_2) = |Pr(y_1 = 1) - Pr(y_2 = 1)|$.*

Lemma 3 (Relationship between the 1-Wasserstein distance and the TV distance (Gibbs & Su, 2002))**.** *Let g be a function defined on a norm-bounded space $\mathcal{Z}$, where $\max_{\mathbf{r} \in \mathcal{Z}} \|\mathbf{r}\| \leq R$, and $\mathcal{D}_1$ and $\mathcal{D}_1$ are two distributions over the space $\mathcal{Z}$. Then $W_1(g(\mathcal{D}_1), g(\mathcal{D}_2)) \leq 2R \cdot d_{TV}(g(\mathcal{D}_1), g(\mathcal{D}_2))$.*

We now prove Theorem 1, which is restated as below:

Theorem 1. *Let $\mathbf{r}^i$ be the representation with a bounded norm R (i.e., $\max_{\mathbf{r}^i \in \mathcal{R}^i} \|\mathbf{r}^i\| \leq R$) learnt by the feature extractor f_{Θ_i} for device C_i's data $\mathbf{x}^i$, and $\mathcal{A}$ be the set of all binary attribute inference classifiers. Assume the primary task classifier c is C_L-Lipschitz, i.e., $\|c\|_L \leq C_L$. Then, the device C_i's utility loss (i.e., classification error) err_i can be bounded as:*

$$err_i = CE_{\mathbf{u}^i=0}(\mathbf{y}^i, c(\mathbf{r}^i)) + CE_{\mathbf{u}^i=1}(\mathbf{y}^i, c(\mathbf{r}^i)) \geq \Delta_{\mathbf{y}^i|\mathbf{u}^i} - 2R \cdot C_L \cdot Adv_{\mathcal{D}^i}(\mathcal{A}), \tag{8}$$

where $CE_{\mathbf{u}^i=a}(\mathbf{y}^i, c(\mathbf{r}^i))$ is the conditional cross-entropy error of predicting $\mathbf{y}^i$ using $\mathbf{r}^i$ given the attribute $\mathbf{u}^i = a \in \{0, 1\}$; $\Delta_{y^i|\mathbf{u}^i} = |Pr_{\mathcal{D}^i}(y^i = 1|\mathbf{u}^i = 0) - Pr_{\mathcal{D}^i}(y^i = 1|\mathbf{u}^i = 1)|$ is a device-dependent constant.

Proof. We denote $\mathcal{D}^i_{\mathbf{u}^i=a}$ as the conditional distribution of $\mathcal{D}^i$ given $\mathbf{u}^i = a$, and cf_i as the (binary) composition function of $c \circ f_{\Theta_i}$. As c is binary task classifier on the learnt representations, it follows that the pushforward $cf_i(\mathcal{D}^i_{\mathbf{u}^i=a})$ induces two distributions over $\{0, 1\}$ with $a = \{0, 1\}$. By leveraging the triangle inequalities of the 1-Wasserstein distance, we have

$$\begin{aligned}
&W_1(\mathcal{D}^i_{\mathbf{y}^i|\mathbf{u}^i=0}, \mathcal{D}^i_{\mathbf{y}^i|\mathbf{u}^i=1}) \\
&\leq W_1(\mathcal{D}^i_{\mathbf{y}^i|\mathbf{u}^i=0}, cf_i(\mathcal{D}^i_{\mathbf{u}^i=0})) + W_1(cf_i(\mathcal{D}^i_{\mathbf{u}^i=0}), cf_i(\mathcal{D}^i_{\mathbf{u}^i=1})) + W_1(cf_i(\mathcal{D}^i_{\mathbf{u}^i=1}), \mathcal{D}^i_{\mathbf{y}^i|\mathbf{u}^i=1})
\end{aligned} \tag{9}$$

Using Lemma 2 on Bernoulli random variables $\mathbf{y}^i|\mathbf{u}^i = a$, we have

$$W_1(\mathcal{D}^i_{\mathbf{y}^i|\mathbf{u}^i=0}, \mathcal{D}^i_{\mathbf{y}^i|\mathbf{u}^i=1}) = |\Pr_{\mathcal{D}^i}(\mathbf{y}^i = 1|\mathbf{u}^i = 0) - \Pr_{\mathcal{D}^i}(\mathbf{y}^i = 1|\mathbf{u}^i = 1)| = \Delta_{\mathbf{y}^i|\mathbf{u}^i}. \tag{10}$$

Using Lemma 1 on the contraction of the 1-Wasserstein distance and that $\|c\|_L \leq C_L$, we have

$$W_1(cf_i(\mathcal{D}^i_{\mathbf{u}^i=0}), cf_i(\mathcal{D}^i_{\mathbf{u}^i=1})) \leq C_L \cdot W_1(f_i(\mathcal{D}^i_{\mathbf{u}^i=0}), f_i(\mathcal{D}^i_{\mathbf{u}^i=1})). \tag{11}$$

Using Lemma 3 with $\max_{i,\mathbf{r}^i} \|\mathbf{r}^i\| \leq R$, we have

$$W_1(f_i(\mathcal{D}^i_{\mathbf{u}^i=0}), f_i(\mathcal{D}^i_{\mathbf{u}^i=1})) \leq 2R \cdot d_{TV}(f_i(\mathcal{D}^i_{\mathbf{u}^i=0}), f_i(\mathcal{D}^i_{\mathbf{u}^i=1})). \tag{12}$$

We further show $d_{TV}(f_i(\mathcal{D}^i_{\mathbf{u}^i=0}), f_i(\mathcal{D}^i_{\mathbf{u}^i=1})) = \text{Adv}_{\mathcal{D}^i}(\mathcal{A})$, as proven in (Liao et al., 2021). Specifically,

$$\begin{aligned}
d_{TV}(f_i(\mathcal{D}^i_{\mathbf{u}^i=0}), f_i(\mathcal{D}^i_{\mathbf{u}^i=1})) &= \max_E |\Pr_{f_i(\mathcal{D}^i_{\mathbf{u}^i=0})}(E) - \Pr_{f_i(\mathcal{D}^i_{\mathbf{u}^i=1})}(E)| \\
&= \max_{A_E \in \mathcal{A}} |\Pr_{\mathbf{r}^i \sim f_i(\mathcal{D}^i_{\mathbf{u}^i=0})}(A_E(\mathbf{r}^i) = 1) - \Pr_{\mathbf{r}^i \sim f_i(\mathcal{D}^i_{\mathbf{u}^i=1})}(A_E(\mathbf{r}^i) = 1)| \\
&= \max_{A_E \in \mathcal{A}} |\Pr(A_E(\mathbf{r}^i) = 1|\mathbf{u}^i = 0) - \Pr(A_E(\mathbf{r}^i) = 1|\mathbf{u}^i = 1)| \\
&= \text{Adv}_{\mathcal{D}^i}(\mathcal{A}),
\end{aligned} \tag{13}$$

where the first equation uses the definition of TV distance, and $A_E(\cdot)$ is the characteristic function of the event E in the second equation.

With Equations 11-13, we have $W_1(cf_i(\mathcal{D}^i_{\mathbf{u}^i=0}), cf_i(\mathcal{D}^i_{\mathbf{u}^i=1})) \leq 2R \cdot C_L \cdot \text{Adv}_{\mathcal{D}^i}(\mathcal{A})$. Furthermore, using Lemma 2 on Bernoulli random variables $\mathbf{y}^i$ and $cf_i(\mathbf{x}^i)$, we have

$$\begin{aligned}
W_1(\mathcal{D}^i_{\mathbf{y}^i|\mathbf{u}^i=a}, cf_i(\mathcal{D}^i_{\mathbf{u}^i=a})) &= |\Pr_{\mathcal{D}^i}(\mathbf{y}^i = 1|\mathbf{u}^i = a) - \Pr_{\mathcal{D}^i}(cf_i(\mathbf{x}^i) = 1|\mathbf{u}^i = a))| \\
&= |\mathbb{E}_{\mathcal{D}^i}[\mathbf{y}^i|\mathbf{u}^i = a] - \mathbb{E}_{\mathcal{D}^i}[cf_i(\mathbf{x}^i)|\mathbf{u}^i = a]| \\
&\leq \mathbb{E}_{\mathcal{D}^i}[|\mathbf{y}^i - cf_i(\mathbf{x}^i)||\mathbf{u}^i = a] \\
&= \Pr_{\mathcal{D}^i}(\mathbf{y}^i \neq cf_i(\mathbf{x}^i)|\mathbf{u}^i = a) \\
&\leq CE_{\mathbf{u}^i=a}(\mathbf{y}^i, cf_i(\mathbf{x}^i)),
\end{aligned} \tag{14}$$

where we use the fact that cross-entropy loss is an upper bound of the binary loss in the last inequality.

Finally, by combining Equation 11-Equation 14, we have:

$$\Delta_{\mathbf{y}^i|\mathbf{u}^i} \leq CE_{\mathbf{u}^i=0}(\mathbf{y}^i, cf_i(\mathbf{x}^i)) + 2R \cdot C_L \cdot \text{Adv}_{\mathcal{D}^i}(\mathcal{A}) + CE_{\mathbf{u}^i=1}(\mathbf{y}^i, cf_i(\mathbf{x}^i)) \tag{15}$$

Hence, $\text{err}_i = CE_{\mathbf{u}^i=0}(\mathbf{y}^i, c(\mathbf{r}^i)) + CE_{\mathbf{u}^i=1}(\mathbf{y}^i, c(\mathbf{r}^i)) \geq \Delta_{\mathbf{y}^i|\mathbf{u}^i} - 2R \cdot C_L \cdot \text{Adv}_{\mathcal{D}^i}(\mathcal{A})$,.

□

A.2 Proof of Theorem 2

The following lemma about the inverse binary entropy will be useful in the proof of Theorem 2:

Lemma 4 ((Calabro, 2009) Theorem 2.2)**.** *Let $H_2^{-1}(p)$ be the inverse binary entropy function for $p \in [0, 1]$, then $H_2^{-1}(p) \geq \frac{p}{2\log_2(\frac{6}{p})}$.*

Lemma 5 (Data processing inequality)**.** *Given random variables X, Y, and Z that form a Markov chain in the order $X \to Y \to Z$, then the mutual information between X and Y is greater than or equal to the mutual information between X and Z. That is $I(X;Y) \geq I(X;Z)$.*

With the above lemma, we are ready to prove Theorem 2 as below.

Theorem 2. *Let Θ_*^i (resp. $\mathbf{r}_*^i$) be the learnt optimal feature extractor parameters (resp. optimal representations) by Equation (6) for device C_i's data. Define $H_*^i = H(\mathbf{u}^i|\mathbf{r}_*^i)$. Then, for any attribute inference adversary $\mathcal{A} = \{A : \mathbf{r}^i \to \mathbf{u}^i\}$, $Pr(A(\mathbf{r}_*^i) = \mathbf{u}^i) \leq 1 - \frac{H_*^i}{2\log_2(\frac{6}{H_*^i})}$.*

Proof. With loss of generality, we only prove the privacy guarantees for the device C_i. For ease of description, we set $\mathbf{r}^i = \mathbf{r}_*^i$ and $H^i = H_*^i$. Let s^i be an indicator that takes value 1 if and only if $\mathcal{A}(\mathbf{r}^i) \neq \mathbf{u}^i$, and 0 otherwise, i.e., $s^i = 1[\mathcal{A}(\mathbf{r}^i) \neq \mathbf{u}^i]$. Now consider the joint entropy $H(\mathcal{A}(\mathbf{r}^i), \mathbf{u}^i, s^i)$ of $\mathcal{A}(\mathbf{r}^i)$, $\mathbf{u}^i$, and s^i. By decomposing it, we have

$$H(s^i, \mathbf{u}^i|\mathcal{A}(\mathbf{r}^i)) = H(\mathbf{u}^i|\mathcal{A}(\mathbf{r}^i)) + H(s^i|\mathbf{u}^i, \mathcal{A}(\mathbf{r}^i)) = H(s^i|\mathcal{A}(\mathbf{r}^i)) + H(\mathbf{u}^i|s^i, \mathcal{A}(\mathbf{r}^i)), \tag{16}$$

Note that $H(s^i|\mathbf{u}^i, \mathcal{A}(\mathbf{r}^i)) = 0$ as when $\mathbf{u}^i$ and $\mathcal{A}(\mathbf{r}^i)$ are known, S_i is also known. Similarly,

$$H(\mathbf{u}^i|s^i, \mathcal{A}(\mathbf{r}^i)) = Pr(s^i = 1)H(\mathbf{u}^i|s^i = 1, \mathcal{A}(\mathbf{r}^i)) + Pr(s^i = 0)H(\mathbf{u}^i|s^i = 0, \mathcal{A}(\mathbf{r}^i)) = 0 + 0 = 0,$$

because when we know s^i's value and $\mathcal{A}(\mathbf{r}^i)$, we also actually knows $\mathbf{u}^i$.

Thus, Equation 16 reduces to $H(\mathbf{u}^i|\mathcal{A}(\mathbf{r}^i)) = H(s^i|\mathcal{A}(\mathbf{r}^i))$. As conditioning does not increase entropy, i.e., $H(s^i|\mathcal{A}(\mathbf{r}^i)) \leq H(s^i)$, we further have

$$H(\mathbf{u}^i|\mathcal{A}(\mathbf{r}^i)) \leq H(s^i). \tag{17}$$

On the other hand, using mutual information and entropy properties, we have $I(\mathbf{u}^i; \mathcal{A}(\mathbf{r}^i)) = H(\mathbf{u}^i) - H(\mathbf{u}^i|\mathcal{A}(\mathbf{r}^i))$ and $I(\mathbf{u}^i; \mathbf{r}^i) = H(\mathbf{u}^i) - H(\mathbf{u}^i|\mathbf{r}^i)$. Hence,

$$I(\mathbf{u}^i; \mathcal{A}(\mathbf{r}^i)) + H(\mathbf{u}^i|\mathcal{A}(\mathbf{r}^i)) = I(\mathbf{u}^i; \mathbf{r}^i) + H(\mathbf{u}^i|\mathbf{r}^i). \tag{18}$$

Notice $\mathcal{A}(\mathbf{r}^i)$ is a random variable such that $u_i \perp \mathcal{A}(\mathbf{r}^i)|\mathbf{z}^i$. Hence, we have the Markov chain $u_i \to \mathbf{z}^i \to \mathcal{A}(\mathbf{r}^i)$. Based on the data processing inequality in Lemma 5, we know $I(\mathbf{u}^i; \mathcal{A}(\mathbf{r}^i)) \leq I(\mathbf{u}^i; \mathbf{r}^i)$. Combining with Equation 18, we have

$$H(\mathbf{u}^i|\mathcal{A}(\mathbf{r}^i)) \geq H(\mathbf{u}^i|\mathbf{r}^i) = H^i. \tag{19}$$

Combing Equations 17 and 19, we have $H(s^i) = H_2(Pr(s^i = 1)) \geq H(\mathbf{u}^i|\mathbf{r}^i)$, which implies

$$Pr(\mathcal{A}(\mathbf{r}^i) \neq \mathbf{u}^i) = Pr(s^i = 1) \geq H_2^{-1}(H(\mathbf{u}^i|\mathbf{r}^i)) = H_2^{-1}(H^i), \tag{20}$$

where $H_2(t) = -t\log_2 t - (1-t)\log_2(1-t)$.

Finally, by applying Lemma 4, we have

$$Pr(\mathcal{A}(\mathbf{r}^i) \neq \mathbf{u}^i) \geq \frac{H^i}{2\log_2(\frac{6}{H^i})}.$$

Hence the attribute privacy leakage is bounded by $Pr(\mathcal{A}(\mathbf{r}^i) = \mathbf{u}^i) \leq 1 - \frac{H^i}{2\log_2(\frac{6}{H^i})}$.

□

B DATASETS AND NETWORK ARCHITECTURES

B.1 DETAILED DATASET DESCRIPTIONS

CIFAR-10 dataset (Krizhevsky, 2009). The CIFAR-10 (Canadian Institute For Advanced Research) dataset contains 60,000 colored images of 32x32 resolution, which is split into the training set with 50,000 images, and the testing set with 10,000 images. It is obtained from the *torchvision.datasets* module, which provides a wide variety of built-in datasets. The dataset consists of images belonging to 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. There are 6,000 images per class.

For this dataset, the primary FL task has been established in accurately predicting the label of the image. The attribute to protect has been generated by the author, creating a binary attribute that is 1 if the image belongs to an animal and 0 otherwise.

Loans dataset (Hardt et al., 2016). This dataset is originally extracted from the loan-level Public Use Databases. The Federal Housing Finance Agency publishes these databases yearly, containing information about the Enterprises single family and multifamily mortgage acquisitions. Specifically, the database used in this project is a single-family dataset and has a variety of features related to the person asking for a mortgage loan. All the attributes in the dataset are numerical, so no preprocessing from this side was required. On the other hand, in order to create a balanced classification problem, some of the features were modified to have a similar number of observations belonging to all classes. We use 80% data for training and 20% for testing.

The utility under this scope was measured in the system accurately predicting the affordability category of the person asking for a loan. This attribute is named *Affordability*, and has three possible values: 0 if the person belongs to a mid-income family and asking for a loan in a low-income area, 1 if the person belongs to a low-income family and asking for a loan in a low-income area, and 2 if the person belongs to a low-income family and is asking for a loan not in a low-income area. The private attribute was set to be binary *Race*, being White (0) or Not White (1).

Adult Income dataset (Dua & Graff, 2017). This is a well-known dataset available in the UCI Machine Learning Repository. The dataset contains 32,561 observations each with 15 features, some of them numerical, other strings. Those attributes are not numerical were converted into categorical using an encoder. Again, we use the 80%-20% train-test split.

The primary classification task is predicting if a person has an income above $50,000, labeled as 1, or below, which is labeled as 0. The private attributes to predict are the *Gender*, which is binary, and the *Marital Status*, which has seven possible labels: 0 if Divorced, 1 if AF-spouse, 2 if Civil-spouse, 3 if Spouse absent, 4 if Never married, 5 if Separated, and 6 if Widowed.

Table 3: Network architectures for the used datasets

Feature Extractor	Privacy Protection Network	Utility Preservation Network
CIFAR-10		
2xconv3-64 MaxPool	3xconv3-256 MaxPool	conv3-16 MaxPool
2xconv3-128 MaxPool	3xconv3-512 MaxPool	conv3-32 MaxPool
	3xconv3-512 MaxPool	2xconv3-128 MaxPool
	2xlinear-4096	3xconv3-256 MaxPool
	linear-#labels	3xconv3-512 MaxPool
		3xconv3-512 MaxPool
		linear-4096
		linear-512
		linear-#labels
Loans and Adult Income		
linear-64	linear-64	linear-16
linear-128	linear-128	linear-32
	linear-4	2xlinear-128
	linear-#labels	3xlinear-256
		6xlinear-512
		linear-4096
		linear-512
		linear-#labels

B.2 NETWORK ARCHITECTURES

The used network architectures for the three neural networks are in Table 3.