A Machine Learning Framework for Privacy-Aware Distributed Functional Compression over AWGN Channels

Yashas Malur Saidutta, Faramarz Fekri, and Afshin Abdi
Dept. of Electrical and Computer Engineering,
Georgia Institute of Technology,
Atlanta, Georgia, USA
Email: {yashas.saidutta, abdi}@gatech.edu, fekri@ece.gatech.edu

2022 IEEE Information Theory Workshop (ITW) | 978-1-6654-8341-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITW54588.2022.9965919

***Abstract*—In many diverse fields, distributed IoT devices perform collaborative inference by communicating with an edge router. Often sensory data contains sensitive attributes that should not be revealed to the router. To address this, we develop, to the best of our knowledge, the first privacy-aware machine learning framework for distributed functional compression over AWGN channels. The key feature of our approach to privacy is that we focus only on sensitive attributes of data rather than paying a high cost to protect everything. Employing a mutual information based privacy constraint, we first propose a novel approximate upper bound to protect sensitive attributes in the compressed representations of the sensory data. Next, in conjunction with the upper bound, we propose an adversarial lower bound to enhance the protection further. Thirdly, we propose novel decompositions to these bounds such distributed edge devices can ensure overall privacy by independently privatizing their components. This allows us to propose an enhanced privacy-aware algorithm that protects sensitive information during training and inference. Our experiments show that the privacy-utility trade-off from our proposed methods is significantly better than existing mechanisms.**

I. Introduction

Internet of Things (IoT) is set to revolutionize cyber-physical systems. A majority of the projected 75 billion IoT devices will be connected over wireless networks and collect close to two exabytes of data per day [1]. In many diverse areas like autonomous driving, chemical/nuclear power plant monitoring, environment monitoring, and augmented reality, distributed IoT devices collectively compute specific target functions without simple known forms like failure prediction, obstacle detection, etc. One way to implement such systems is to leverage Machine Learning. Traditionally cloud-based solutions send edge device data to the router for processing. However, such systems are fraught with privacy risks, and the transmission of uncompressed data can burden existing communication systems. Further, sometimes the training data is itself collected by the sensors and their privacy must also be protected. In this paper, we seek to address these problems in the wireless communication setting by processing sensory data to send compressed representations across AWGN channels. The key feature of our approach to privacy is that we focus only on sensitive attributes of data rather than paying a high cost to protect everything.

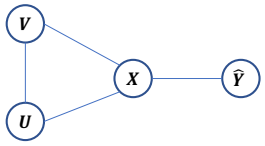


Fig. 1: Graphical Model indicating the relationship between the target variable ($\boldsymbol{V}$), the noisy received signal ($\hat{\boldsymbol{Y}}$), the source variable ($\boldsymbol{X}$), and the private attribute ($\boldsymbol{U}$).

We defer the formal definition of the problem to the next section; however, succinctly, it can be represented using the Generalized Information Bottleneck (GIB) framework in Fig. 1. Here, $\boldsymbol{X}$ is the observed random variable, $\boldsymbol{V}$ is the target function value that we are interested in communicating to the receiver, $\boldsymbol{U}$ is the sensitive attribute we are interested in hiding, and $\hat{\boldsymbol{Y}}$ is the noisy representation received by the receiver. The goal is to learn $\hat{\boldsymbol{Y}}$ that is most informative about $\boldsymbol{V}$, contains as little information as possible about $\boldsymbol{U}$, and is a compressed representation of $\boldsymbol{X}$, i.e., $I(\boldsymbol{X};\hat{\boldsymbol{Y}}) \leq R$. Here $I(\cdot,\cdot)$ represents the mutual information between the two random variables and R is the rate constraint. We call this framework the GIB framework because it is a generalized form of the Information Bottleneck framework [2] and also subsumes the privacy-funnel problem [3]. Note, $(\boldsymbol{U},\boldsymbol{V}) \leftrightarrow \boldsymbol{X}^N \leftrightarrow \hat{\boldsymbol{Y}}^N$ forms a Markov Chain. In this paper, by leveraging training samples, we propose a mechanism to learn the encoding and decoding functions at the sensor(s) and receiver, respectively, in a data-driven manner.

Related Works: *Information Bottleneck* framework [2] has been suggested for privacy, but it does not incorporate explicit privacy constraints and fails when the sensitive attribute is correlated with the target variable [4]–[7]. Alternatively, the *Privacy-Funnel* framework does not incorporate rate constraints [3], [8]–[12]. This problem formulation is also prevalent in machine learning areas like fairness [13], [14] and input obfuscation [15]–[17]. The *GIB* framework has seen very little work. Razeghi et al. proposed theoretical conditions to achieve perfect privacy [18]. Moyer et al.

proposed a GIB based solution for fairness in Machine Learning [19]. However, apart from [7], none of the works look at the distributed/split learning scenario where each sensor node is exposed to a subset of the input data's dimensions. In the split learning setting works rely on sending processed representations to the router (no explicit privacy/rate constraints) [20]–[22].

Contributions of this paper are:

1) We propose a novel approximate upper bound on the mutual information between the noisy representation and the private attribute.
2) We also propose using an adversarial lower bound, which gives complementary benefits in conjunction with the above.
3) We decompose both the upper and the adversarial bounds proposed so that its components can be computed independently at the nodes without information from other nodes.
4) We combine this with the Partially Synchronous Block Coordinate Descent (PSBCD) training algorithm proposed in [7] to obtain an algorithm capable of maintaining privacy during training and inference.

II. Problem Formulation

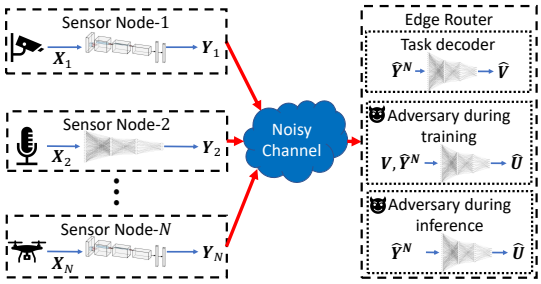


Fig. 2: Distributed Functional Compression with Adversary.

Figure 2 shows the distributed setup under consideration. Multiple sensor nodes observe different possibly correlated random variables $\boldsymbol{X}_n$, where $n \in \{1,\dots,N\}$ indexes the sensor nodes. We also denote $\boldsymbol{X}^N := [\boldsymbol{X}_1,\dots,\boldsymbol{X}_N]^T$. The edge router is interested in approximating a specific function of these random variables, i.e., $\boldsymbol{v} := \mathcal{F}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_N)$. To facilitate this, each ensor node-n encodes its observations using some encoding function $g_e^{(n)}(\cdot)$ and transmits the encoding $\boldsymbol{Y}_n \in \mathbb{R}^{K_n}$ across orthogonal AWGN channels, i.e., $\hat{\boldsymbol{Y}}_n = \boldsymbol{Y}_n + \boldsymbol{Z}_n$, where $\boldsymbol{Z}_n \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_{z_n}^2 \boldsymbol{I}_{K_n}\right)$ and $\sigma_{z_n}^2$ is the noise power. The edge router concatenates the received noisy encodings (denoted as $\hat{\boldsymbol{Y}}^N$) and attempts to recover the value of $\boldsymbol{V}$ as $\hat{\boldsymbol{V}} := g_d(\hat{\boldsymbol{Y}}^N)$. Additionally, the encoding should be such that it should remove any information about some common sensitive attribute $\boldsymbol{U}$, which an inferential adversary [23] at the edge router is interested in infering. We can accomplish this by minimizing $I(\boldsymbol{U};\hat{\boldsymbol{Y}}^N)$. Finally, each encoder has to obey the rate constraint $I(\boldsymbol{X}_n;\hat{\boldsymbol{Y}}_n) \le R_n$ (In the wireless communication case, this is equivalent to a power constraint encoder/transmitter). Finally, we can write the optimization problem as

$$\min_{g_e^{(1)},\dots,g_e^{(N)},g_d} \mathbb{E}_{\boldsymbol{X}^N,\boldsymbol{V}}\left[\mathcal{D}_{\boldsymbol{V}}\left(\boldsymbol{v},\hat{\boldsymbol{v}}\right)\right] + \sum_{n=1}^{N} \lambda_n I(\boldsymbol{X}_n;\hat{\boldsymbol{Y}}_n) + \beta I\left(\boldsymbol{U};\hat{\boldsymbol{Y}}^N\right). \quad (1)$$

Here, β is a factor to determine the trade-off between privacy and the ability to reconstruct $\boldsymbol{v}$ (utility) and λ_n are Lagrange multipliers.

However, the above formulation only looks at privacy during inference. In many situations the training sensory data itself is collected by the sensors in a distributed manner and the sensitive attributes of the training data has to be protected too. Let us define our training dataset as $\{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_N,\boldsymbol{u},\boldsymbol{v})^{(b)}\}_{b=1}^{B}$ where, b indexes the samples, and B is the size of the dataset. During training, the sensors have access to $\{(\boldsymbol{x}_n,\boldsymbol{u},\boldsymbol{v})^{(b)}\}_{b=1}^{B}$ and the edge router has access to $\{(\boldsymbol{v})^{(b)}\}_{b=1}^{B}$. During inference, the sensor nodes only observe $\boldsymbol{x}_n$. In this setup, since the edge router has access to the target function values corresponding to the training data, to ensure privacy during training time, we have to minimize the mutual information $I(\boldsymbol{U};(\hat{\boldsymbol{Y}},\boldsymbol{V}))$. For privacy during inference we have to ensure that $I(\boldsymbol{U};\hat{\boldsymbol{Y}})$ is small. Since $I(\boldsymbol{U};(\hat{\boldsymbol{Y}},\boldsymbol{V})) \ge I(\boldsymbol{U};\hat{\boldsymbol{Y}})$, minimizing the former also ensures privacy during inference. Thus, we can write the new optimization objective as

$$\min_{g_e^{(1)},\dots,g_e^{(N)},g_d} \mathbb{E}_{\boldsymbol{X}^N,\boldsymbol{V}}\left[\mathcal{D}_{\boldsymbol{V}}\left(\boldsymbol{v},\hat{\boldsymbol{v}}\right)\right] + \sum_{n=1}^{N} \lambda_n I(\boldsymbol{X}_n;\hat{\boldsymbol{Y}}_n) + \beta I\left(\boldsymbol{U};\left(\hat{\boldsymbol{Y}}^N,\boldsymbol{V}\right)\right). \quad (2)$$

Note 1: An inferential adversary is one who attempts to predict the value of the sensitive attribute ($\boldsymbol{U}$) using the noisy encoded representation ($\hat{\boldsymbol{Y}}^N$) [23]. It is known that any measure of privacy loss in the inferential adversary setup is upper bounded by a factor of mutual information between the sensitive attribute and the noisy compressed representation, thus making it a general measure of privacy [3].

Note 2: The GIB objective is written as

$$\min -I(\boldsymbol{V};\hat{\boldsymbol{Y}}^N) + \sum_{n=1}^{N} \lambda_n I(\boldsymbol{X}_n;\hat{\boldsymbol{Y}}_n) + \beta I\left(\boldsymbol{U};\left(\boldsymbol{V},\hat{\boldsymbol{Y}}^N\right)\right). \quad (3)$$

We can show that $-I(\boldsymbol{V};\hat{\boldsymbol{Y}}^N) \le -\mathbb{E}_{\boldsymbol{V},\hat{\boldsymbol{Y}}^N}\left[\log q_{\boldsymbol{V}|\hat{\boldsymbol{Y}}^N}(\boldsymbol{v}|\hat{\boldsymbol{y}}^N)\right] - H(\boldsymbol{V})$. By modeling $q_{\boldsymbol{V}|\hat{\boldsymbol{Y}}^N}(\boldsymbol{v}|\hat{\boldsymbol{y}}^N) := \frac{1}{Z_1}\exp\left(-\mathcal{D}_{\boldsymbol{V}}\left(\boldsymbol{v},g_d(\hat{\boldsymbol{y}}^N)\right)\right)$, where Z_1 is a normalization constant, we can show that (2) is an upper bound on (3).

III. Methodology

We parametrize all $g_e^{(n)}(\cdot)$ and $g_d(\cdot)$ as neural networks with parameters $\boldsymbol{\Phi}_n$ and $\boldsymbol{\Theta}$ resepctively. Since, we do not know the distributions $p(\boldsymbol{x},\hat{\boldsymbol{y}}^N)$ and $p(\boldsymbol{u},\boldsymbol{v},\hat{\boldsymbol{y}}^N)$ in closed form (the distribution depends on the neural network parameters

making finding the closed form expressionx difficult), we cannot compute the mutual information terms $I(\boldsymbol{X}_n;\hat{\boldsymbol{Y}}_n)$ and $I(\boldsymbol{U};(\boldsymbol{V},\hat{\boldsymbol{Y}}^N))$. Thus we resort to variational approximations.

We upper bound $I(\boldsymbol{X}_n;\hat{\boldsymbol{Y}}_n)$ as

$$I(\boldsymbol{X}_n;\hat{\boldsymbol{Y}}_n) \leq -H(\boldsymbol{Z}_n) - \mathbb{E}_{\hat{\boldsymbol{Y}}^n}\left[\log q_{\hat{\boldsymbol{Y}}_n}(\hat{\boldsymbol{y}}_n)\right]. \quad (4)$$

Here, $q_{\hat{\boldsymbol{Y}}_n}(\hat{\boldsymbol{y}}_n)$ is the variational approximation of $p(\hat{\boldsymbol{y}}_n)$. Note that $H(\hat{\boldsymbol{Y}}_n|\boldsymbol{X}) = H(\hat{\boldsymbol{Y}}_n|\boldsymbol{Y}_n) = H(\boldsymbol{Z}_n)$. This inturn follows because $\boldsymbol{Y}_n$ is a deterministic function of $\boldsymbol{X}_n$.

We now focus on upper bounding the last term in (2). We can use the chain rule of mutual information to write

$$I(\boldsymbol{U};(\boldsymbol{V},\hat{\boldsymbol{Y}}^N)) = I(\boldsymbol{U};\hat{\boldsymbol{Y}}^N) + I(\boldsymbol{U};\boldsymbol{V}|\hat{\boldsymbol{Y}}^N). \quad (5)$$

Let us focus on the second term in the RHS of (5). We can write $0 \leq I\left(\boldsymbol{U};\boldsymbol{V}|\hat{\boldsymbol{Y}}^N\right) \leq C_1 - I(\boldsymbol{V};\hat{\boldsymbol{Y}}^N)$, where $C_1 := I((\boldsymbol{U},\boldsymbol{X}^N);\boldsymbol{V})$ is a constant, and the upper bound follows from both chain rule and $I((\boldsymbol{U},\hat{\boldsymbol{Y}}^N);\boldsymbol{V}) \leq I((\boldsymbol{U},\boldsymbol{X}^N);\boldsymbol{V})$ which follows from Data Processing Inequality [24] for our problem setup. Remember, in our setup, we are trying to minimize an upper bound of $-I(\boldsymbol{V};\boldsymbol{Y}^N)$ in (2), i.e., come up with a $\hat{\boldsymbol{Y}}^N$ that can be used to reliably reconstruct the value of the function $\boldsymbol{V} := \mathcal{F}(\boldsymbol{X}_1,\ldots,\boldsymbol{X}_N)$. So as $I(\boldsymbol{V};\boldsymbol{Y}^N)$ increases, $I\left(\boldsymbol{U};\boldsymbol{V}|\hat{\boldsymbol{Y}}^N\right)$ is sandwiched between 0 and an upper bound that is reducing. In fact, for a perfect system where we can perfectly predict the value of $\boldsymbol{V}$ using $\hat{\boldsymbol{Y}}^N$, $I\left(\boldsymbol{U};\boldsymbol{V}|\hat{\boldsymbol{Y}}^N\right) = 0$. Thus this term is small as long as the system can predict the functional value with reasonably good accuracy using $\hat{\boldsymbol{Y}}_N$. Hence, we ignore this term. To bound the first term in the RHS of (5), we can express it as

$$I\left(\boldsymbol{U};\hat{\boldsymbol{Y}}^N\right) \leq I\left(\boldsymbol{X}^N;\hat{\boldsymbol{Y}}^N\right) - I\left(\boldsymbol{V};\hat{\boldsymbol{Y}}^N|\boldsymbol{U}\right). \quad (6)$$

This is because $\boldsymbol{U} \perp\!\!\!\perp \hat{\boldsymbol{Y}}^N \mid \boldsymbol{X}^N$ and $-I(\hat{\boldsymbol{Y}}^N;\boldsymbol{X}^N|\boldsymbol{U}) \leq -I(\hat{\boldsymbol{Y}}^N;\boldsymbol{V}|\boldsymbol{U})$ in our problem setup (The independence holds because $(\boldsymbol{U},\boldsymbol{V}) \leftrightarrow \boldsymbol{X}^N \leftrightarrow \hat{\boldsymbol{Y}}^N$ forms a Markov Chain, and the inequality follows from the Data Processing Inequality). We can bound the first term in RHS of (6) similar to (4). We can bound the second term in RHS of (6) as

$$-I\left(\boldsymbol{V};\hat{\boldsymbol{Y}}^N|\boldsymbol{U}\right) \leq -C_2 - \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{U},\boldsymbol{V},\hat{\boldsymbol{Y}}_n}\left[\log q_{\boldsymbol{V}|\boldsymbol{U},\hat{\boldsymbol{Y}}_n}(\boldsymbol{v}|\boldsymbol{u},\hat{\boldsymbol{y}}_n)\right], \quad (7)$$

where $C_2 := H(\boldsymbol{V}|\boldsymbol{U})$ is a constant and $q_{\boldsymbol{V}|\boldsymbol{U},\hat{\boldsymbol{Y}}_n}(\boldsymbol{v}|\boldsymbol{u},\hat{\boldsymbol{y}}_n)$ is the variational approximation of the distribution $p(\boldsymbol{v}|\boldsymbol{u},\hat{\boldsymbol{y}}_n)$. The inequality follows because $H(\boldsymbol{V}|\hat{\boldsymbol{Y}}^N,\boldsymbol{U}) \leq \frac{1}{N}\sum_{n=1}^{N} H(\boldsymbol{V}|\hat{\boldsymbol{Y}}_n,\boldsymbol{U})$ and cross entropy upper bounds entropy. We construct $q_{\boldsymbol{V}|\boldsymbol{U},\hat{\boldsymbol{Y}}_n}(\boldsymbol{v}|\boldsymbol{u},\hat{\boldsymbol{y}}_n)$ as $\frac{1}{Z_2}\exp\left(\mathcal{D}_{\mathcal{V}}(\boldsymbol{v},h_n(\boldsymbol{u},\hat{\boldsymbol{y}}_n;\boldsymbol{\Psi}_n))\right)$, where $h_n(\cdot;\boldsymbol{\Psi}_n)$ is a Neural Network with parameters $\boldsymbol{\Psi}_n$.

Thus, combining all these approximations we can write the objective for training as

$$\begin{aligned}\min_{\boldsymbol{\Phi}_1,\ldots,\boldsymbol{\Phi}_N,\boldsymbol{\Theta},\boldsymbol{\Psi}_1,\ldots,\boldsymbol{\Psi}_N} & \mathbb{E}_{\boldsymbol{V},\hat{\boldsymbol{Y}}}\left[\mathcal{D}_{\boldsymbol{V}}(\boldsymbol{v},g_d(\hat{\boldsymbol{y}};\boldsymbol{\Theta}))\right] \\ & - (\lambda_n+\beta)\sum_{n=1}^{N}\mathbb{E}_{\hat{\boldsymbol{Y}}_n}\left[\log q_{\hat{\boldsymbol{Y}}_n}(\hat{\boldsymbol{y}}_n)\right] \\ & + \beta\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{X}_n,\boldsymbol{U},\hat{\boldsymbol{Y}}_n}\left[\mathcal{D}_{\boldsymbol{V}}(\boldsymbol{v},h(\boldsymbol{u},\hat{\boldsymbol{y}}_n;\boldsymbol{\Psi}_n))\right]. \quad (8)\end{aligned}$$

The attractive nature of this decomposition is that the privacy term is represented as a sum. Thus, the required components can be computed independently at the respective nodes, i.e., the term $\mathcal{D}_{\boldsymbol{V}}(\boldsymbol{v},h_n(\boldsymbol{u},\hat{\boldsymbol{y}}_n;\boldsymbol{\Psi}_n))$ only depends on the encoded values from the encoder at sensor node-n.

Let us analyze the upper bound on the privacy constraint we variationally approximated, $I(\boldsymbol{U};\hat{\boldsymbol{Y}}^N) \leq I(\boldsymbol{X}^N;\hat{\boldsymbol{Y}}^N) - C_2 + H(\boldsymbol{V}|\boldsymbol{U},\hat{\boldsymbol{Y}}^N)$. The first term imposes a rate constraint on the information between $\boldsymbol{X}^N$ and $\hat{\boldsymbol{Y}}^N$, and the last term tries to minimize $H(\boldsymbol{V}|\boldsymbol{U},\hat{\boldsymbol{Y}}^N)$. To minimize the last term, $\hat{\boldsymbol{Y}}^N$ is **encouraged** to have information about $\boldsymbol{V}$ that is not present in $\boldsymbol{U}$. Thus, we call the training objective in (8) as **Encouragement Objective (Distributed)** or **ECO(D)**.

However, when the rate constraints $R_1,\ldots,R_n$ are high, $\hat{\boldsymbol{Y}}^N$ can carry both information about $\boldsymbol{U}$ and information orthogonal to $\boldsymbol{U}$. In such a situation, we need a term that actively enforces the removal of information about $\boldsymbol{U}$ in $\hat{\boldsymbol{Y}}^N$. This is done using an adversarial lower bound. We can write an adversarial lower bound for $I(\boldsymbol{U};(\hat{\boldsymbol{Y}},\boldsymbol{V}))$ as

$$\begin{aligned}& I\left(\boldsymbol{U};\left(\boldsymbol{V},\hat{\boldsymbol{Y}}^N\right)\right) \\ & \geq H(\boldsymbol{U}) + \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{U},\boldsymbol{V},\hat{\boldsymbol{Y}}_n}\left[\log q_{(\boldsymbol{U}|\boldsymbol{V},\hat{\boldsymbol{Y}}_n)}(\boldsymbol{u}|\boldsymbol{v},\hat{\boldsymbol{y}}_n)\right], \quad (9)\end{aligned}$$

where, $q_{(\boldsymbol{U}|\boldsymbol{V},\hat{\boldsymbol{Y}}_n)}(\boldsymbol{u}|\boldsymbol{v},\hat{\boldsymbol{y}}_n)$ is the variational approximation of the distribution $p(\boldsymbol{u}|\boldsymbol{v},\hat{\boldsymbol{y}}_n)$. Further, $q_{(\boldsymbol{U}|\boldsymbol{V},\hat{\boldsymbol{Y}}_n)} := \frac{1}{Z_3}\exp\left(-\mathcal{D}_{\boldsymbol{U}}(\boldsymbol{u},e_n(\hat{\boldsymbol{y}}_n;\boldsymbol{\zeta}_n))\right)$, where $e_n(\hat{\boldsymbol{y}}_n;\boldsymbol{\zeta}_n)$ is an adversarial neural network attempting to predict the value of $\boldsymbol{u}$ using $\hat{\boldsymbol{y}}_n$, $\boldsymbol{v}$ and $\boldsymbol{\zeta}_n$ is its parameters. This allows us to setup a min-max optimization of the form

$$\begin{aligned}\min_{\boldsymbol{\Phi}_1,\ldots,\boldsymbol{\Phi}_N,\boldsymbol{\Theta},\boldsymbol{\Psi}_1,\ldots,\boldsymbol{\Psi}_N}\max_{\boldsymbol{\zeta}_1,\ldots,\boldsymbol{\zeta}_N} & \mathbb{E}_{\boldsymbol{V},\hat{\boldsymbol{Y}}}\left[\mathcal{D}_{\boldsymbol{V}}(\boldsymbol{v},g_d(\hat{\boldsymbol{y}}^N;\boldsymbol{\Theta}))\right] \\ & - (\lambda+\beta)\sum_{n=1}^{N}\mathbb{E}_{\hat{\boldsymbol{Y}}_n}\left[\log q_{\hat{\boldsymbol{Y}}_n}(\hat{\boldsymbol{y}}_n)\right] \\ & + \beta\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{X}_n,\boldsymbol{U},\hat{\boldsymbol{Y}}_n}\left[\mathcal{D}_{\boldsymbol{V}}(\boldsymbol{v},h(\boldsymbol{u},\hat{\boldsymbol{y}}_n;\boldsymbol{\Psi}_n))\right] \\ & - \gamma\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{U},\hat{\boldsymbol{Y}}_n}\left[\mathcal{D}_{\boldsymbol{U}}(\boldsymbol{u},e(\hat{\boldsymbol{y}}_n;\boldsymbol{\zeta}_n))\right]. \quad (10)\end{aligned}$$

We call this the **Enforcement-Encouragement Objective (Distributed)** or **EEO(D)**. Like ECO(D), the adversarial components are decomposed into a sum whose individual components are computed at the sensor nodes.

Partially Synchronous Block Coordinate Descent Algorithm: Once the upper and the adversarial bound are decomposed into a sum of components that can be independently computed at the sensor nodes, we can use the PSBCD algorithm proposed in [7]. The algorithm was proposed as a communication-efficient distributed learning mechanism for distributed functional compression where the

sensory training data is never revealed to the edge router. All communication to the edge router during training is in terms of $\hat{\boldsymbol{Y}}$, i.e., the noisy compressed representation for the samples in the training set. This is attractive from a privacy standpoint. This implies if we remove information about the sensitive attributes from the compressed representation $\boldsymbol{y}_n$ at the individual sensors before transmission during training, we should be able to protect the training data.

Privacy Guarantees: Given the Mutual Information between $\boldsymbol{U}$ and $\hat{\boldsymbol{Y}}$, Proposition 1 in [25] provides an upper bound on the **maximum accuracy achievable by any adversary**. However, in reality, this bound is pretty loose. Instead, we can numerically solve for the largest probability of accuracy (of an inferential adversary) that satisfies Fano's inequality. This provides a tighter upper bound on the maximum accuracy achievable by any adversary [24]. Similarly we can compute train-time privacy guarantees using $I(\boldsymbol{U};(\boldsymbol{V},\hat{\boldsymbol{Y}}^N))$.

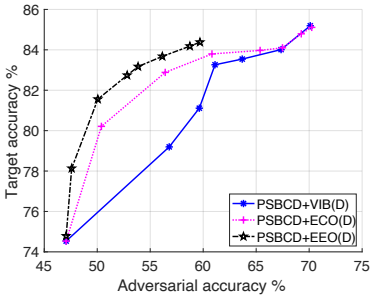
IV. Experimental Evaluations

A. Experimental Settings

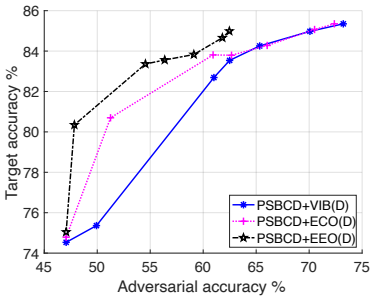
We use the adult income dataset [26], popular in the fairness machine learning community. The objective is to predict whether a person has an annual income greater than fifty thousand dollars based on various attributes [26]. In our experiments, we chose marital status as the sensitive attribute. The input attributes are divided into N subsets such that their union is equal to the set of all attributes. Each set is made available to one of the sensor nodes. For example, for $N = 2$, the first six attributes are observed by node-1 and the next seven by node-2. Following the suggestions of [27], all neural networks have two hidden layers with 100 neurons each. Privacy is evaluated by a post-hoc adversary trained after the system's complete training. The post-hoc adversary has $[100, 100, 75, 50, 25]$ neurons per layer in that order. The accuracy of predicting the income level is **target accuracy**, and the accuracy of the post-hoc adversary in predicting the sensitive marital status is the **adversarial accuracy**. *The goal is to have high target accuracy and low adversarial accuracy*. We use the Adam Optimizer with an initial learning rate of 10^{-3}, decaying by 0.5 upon validation loss saturation [28]. The training-validation-test split is 70-10-20. All values reported here correspond to ten repetitions over the test set. We used the NPEET toolbox for estimating mutual information to compute the privacy guarantees [29].

B. Experimental Results

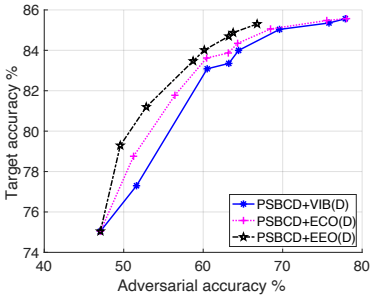
Figure 3 shows the privacy-utility trade-offs based on a simulated adversary for $N = 2, 4$, and 6 nodes. Any privacy-utility trade-off curve closer to the top left is better because this ensures a higher target accuracy (accuracy of predicting the variable of interest, the income level) for the same post-hoc adversarial accuracy (accuracy of predicting the sensitive marital attribute value). All algorithms use the PSBCD algorithm. We compare with the existing work of [7], which protects privacy by sending the compressed representations. We find that both our privacy-aware training objective. The



(a) Simulated adversary based privacy-utility trade-off for $N = 2$ nodes.



(b) Simulated adversary based privacy-utility trade-off for $N = 4$ nodes.

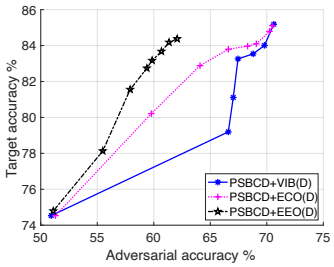


(c) Simulated adversary based privacy-utility trade-off for $N = 6$ nodes.

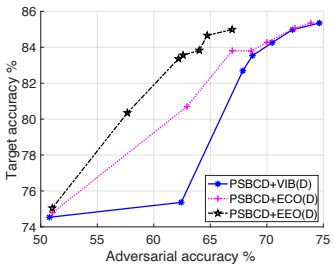
Fig. 3: Privacy-Utility (PU) trade-off over a simulated adversary during inference.

EEO(D) (Distributed Encouragement Enforcement Objective from (10)) and ECO(D) (Distributed Encouragement Objective from (8)) outperform [7] and amongst them, EEO(D) is the best. Even though we do not include the plots, the same ordering is maintained for privacy guarantees.

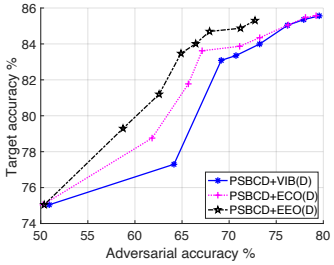
Having established that this system maintains privacy during inference, we now study privacy during training in Fig. 4. The compressed representation is transmitted to the edge router multiple times, during training for every training sample. These transmissions correspond to different training iterations. The post-hoc adversary accumulates all the corresponding noisy received signals for all training samples and leverages all of them along with $\boldsymbol{V}$ to predict the value of $\boldsymbol{U}$. Thus, the performance of the post-hoc adversary captures the **total** privacy leakage during training. We find that the ordering of the methods is identical to the inference time ordering, with PSBCD+EEO(D) performing the best. However, note that the actual adversarial accuracy is higher than during inference time, indicating more leakage during training. This is to be



(a) Simulated adversary based privacy-utility trade-off for $N = 2$ nodes.



(b) Simulated adversary based privacy-utility trade-off for $N = 4$ nodes.



(c) Simulated adversary based privacy-utility trade-off for $N = 6$ nodes.

Fig. 4: PU trade-off over a simulated adversary during training.

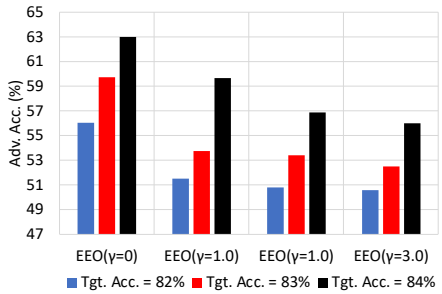
expected because the system has access to the function value $\boldsymbol{V}$ during inference.

Next, we look at the effect of γ on EEO(D). Here, γ refers to the weight given to the train-time adversary during training in (10). Figure 5a shows that as γ increases, the encoder learns an encoding better at fooling the post-hoc adversary. However, as seen in Fig. 5b, this is at the cost of a higher channel capacity requirement. Note that transmission power increases exponentially w.r.t. capacity.

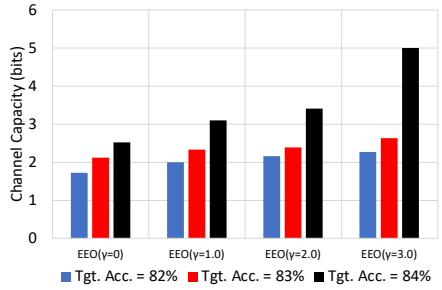
Finally, in Fig. 6, we simulated the special case of $N = 1$ as it allows us to compare with the work of [19]. Here PSBCD is not applicable, and hence we only look at privacy during inference. The privacy constraint here is $I(\boldsymbol{U}; \hat{\boldsymbol{Y}})$. Thus, our adversarial bound has to be modified, i.e., the network $e(\cdot)$ only has $\hat{\boldsymbol{y}}$ as input. We find that our proposed methods significantly outperform both VIB and the work of [19]. The performance improvement over [19] is because the encouragement term encourages $\hat{\boldsymbol{Y}}$ to learn all information about $\boldsymbol{X}$ not present in $\boldsymbol{U}$, whereas our setup prioritizes information relevant to $\boldsymbol{V}$.

V. Conclusion

In this paper, we develop, to the best of our knowledge, the first privacy-aware machine learning framework for distributed



(a) Adversarial accuracies for different γ values in EEO(D) at particular target accuracy.



(b) Channel capacity in bits for different γ values in EEO(D) at particular target accuracy.

Fig. 5: Studying the effect of γ on EEO(D) in distributed functional compression for $N = 4$.

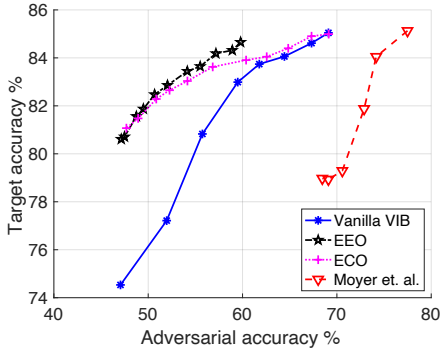


Fig. 6: Privacy-Utility trade-off of Variational Information Bottleneck (VIB) [6], ECO, EEO, and Moyer et al. [19] using a simulated adversary for $N = 1$.

functional compression over AWGN channels. The key feature of our approach to privacy is to focus only on sensitive attributes of data rather rather than paying a high cost to protect everything. To achieve this, we propose a novel approximate upper bound for the mutual information between the received representation and the sensitive attributes. Next, we suggest the use of an adversarial lower bound and combining it with the approximate upper bound for complementary privacy benefits. We also proposed novel decompositions to these bounds that allow distributed edge devices to ensure overall privacy by independently privatizing their components. This makes it amenable to be combined with distributed training algorithms that can protect sensitive attributes in the training data. Our experiments showed that our methods significantly outperformed existing methods.

REFERENCES

[1] 2022. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[3] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the information bottleneck to the privacy funnel," in *2014 IEEE Information Theory Workshop (ITW 2014)*. IEEE, 2014, pp. 501–505.

[4] S. Kariyappa, O. Dia, and M. K. Qureshi, "Enabling inference privacy with adaptive noise injection," *arXiv preprint arXiv:2104.02261*, 2021.

[5] A. A. Atashin, B. Razeghi, D. Gündüz, and S. Voloshynovskiy, "Variational leakage: The role of information complexity in privacy leakage," *arXiv preprint arXiv:2106.02818*, 2021.

[6] Y. M. Saidutta, A. Abdi, and F. Fekri, "Analog joint source-channel coding for distributed functional compression using deep neural networks," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 2429–2434.

[7] ——, "A machine learning framework for distributed functional compression over wireless channels in IoT," *arXiv preprint arXiv:2201.09483*, 2022.

[8] A. Zamani, T. J. Oechtering, and M. Skoglund, "Data disclosure mechanism design with non-zero leakage," in *2020 IEEE Information Theory Workshop (ITW)*. IEEE, 2021, pp. 1–5.

[9] S. Asoodeh, F. Alajaji, and T. Linder, "Notes on information-theoretic privacy," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2014, pp. 1272–1278.

[10] S. Asoodeh, M. Diaz, F. Alajaji, and T. Linder, "Information extraction under privacy constraints," *Information*, vol. 7, no. 1, p. 15, 2016.

[11] A. Zamani, T. J. Oechtering, and M. Skoglund, "Data disclosure with non-zero leakage and non-invertible leakage matrix," *arXiv preprint arXiv:2107.07484*, 2021.

[12] Y. Bu, T. Wang, and G. W. Wornell, "Sdp methods for sensitivity-constrained privacy funnel and information bottleneck problems," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 49–54.

[13] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International conference on machine learning*. PMLR, 2013, pp. 325–333.

[14] A. Jaiswal, D. Moyer, G. Ver Steeg, W. AbdAlmageed, and P. Natarajan, "Invariant representations through adversarial forgetting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4272–4279.

[15] S. A. Osia, A. Taheri, A. S. Shamsabadi, K. Katevas, H. Haddadi, and H. R. Rabiee, "Deep private-feature extraction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 1, pp. 54–66, 2018.

[16] M. Samragh, H. Hosseini, A. Triastcyn, K. Azarian, J. Soriaga, and F. Koushanfar, "Unsupervised information obfuscation for split inference of neural networks," *arXiv preprint arXiv:2104.11413*, 2021.

[17] C. Huang, P. Kairouz, and L. Sankar, "Generative adversarial privacy: A data-driven approach to information-theoretic privacy," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 2162–2166.

[18] B. Razeghi, F. P. Calmon, D. Gündüz, and S. Voloshynovskiy, "On perfect obfuscation: Local information geometry analysis," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2020, pp. 1–6.

[19] D. Moyer, S. Gao, R. Brekelmans, G. V. Steeg, and A. Galstyan, "Invariant representations without adversarial training," *arXiv preprint arXiv:1805.09458*, 2018.

[20] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed comparison of communication efficiency of split learning and federated learning," *arXiv preprint arXiv:1909.09145*, 2019.

[21] Y. Koda, J. Park, M. Bennis, K. Yamamoto, T. Nishio, M. Morikura, and K. Nakashima, "Communication-efficient multimodal split learning for mmwave received power prediction," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1284–1288, 2020.

[22] M. Krouka, A. Elgabli, C. B. Issaid, and M. Bennis, "Communication-efficient split learning based on analog communication and over the air aggregation," *arXiv preprint arXiv:2106.00999*, 2021.

[23] F. du Pin Calmon and N. Fawaz, "Privacy against statistical inference," in *2012 50th annual Allerton conference on communication, control, and computing (Allerton)*. IEEE, 2012, pp. 1401–1408.

[24] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[25] S. Salamatian, F. P. Calmon, N. Fawaz, A. Makhdoumi, and M. Medard, "Privacy-utility tradeoff and privacy funnel," 1 2020.

[26] R. Kohavi *et al.*, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid." in *Kdd*, vol. 96, 1996, pp. 202–207.

[27] A. Jaiswal, Y. Wu, W. AbdAlmageed, and P. Natarajan, "Unsupervised adversarial invariance," *arXiv preprint arXiv:1809.10083*, 2018.

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

[29] G. Ver Steeg, "Non-parametric entropy estimation toolbox (NPEET)," 2000. [Online]. Available: https://github.com/gregversteeg/NPEET