Can we Generalize and Distribute Private Representation Learning?

Sheikh Shams Azam
School of ECE
Purdue Univeristy
IN, USA

Taejin Kim
Department of ECE
Carnegie Mellon University
CA, USA

Seyyedali Hosseinalipour
School of ECE
Purdue Univeristy
IN, USA

Carlee Joe-Wong
Department of ECE
Carnegie Mellon University
CA, USA

Saurabh Bagchi
School of ECE
Purdue Univeristy
IN, USA

Christopher Brinton
School of ECE
Purdue Univeristy
IN, USA

Abstract

We study the problem of learning representations that are private yet informative, i.e., provide information about intended "ally" targets while hiding sensitive "adversary" attributes. We propose Exclusion-Inclusion Generative Adversarial Network (EIGAN), a generalized private representation learning (PRL) architecture that accounts for multiple ally and adversary attributes unlike existing PRL solutions. While centrally-aggregated dataset is a prerequisite for most PRL techniques, data in real-world is often siloed across multiple distributed nodes unwilling to share the raw data because of privacy concerns. We address this practical constraint by developing D-EIGAN, the first distributed PRL method that learns representations at each node without transmitting the source data. We theoretically analyze the behavior of adversaries under the optimal EIGAN and D-EIGAN encoders and the impact of dependencies among ally and adversary tasks on the optimization objective. Our experiments on various datasets demonstrate the advantages of EIGAN in terms of performance, robustness, and scalability. In particular, EIGAN outperforms the previous state-of-the-art by a significant accuracy margin (47% improvement), and D-EIGAN's performance is consistently on par with EIGAN under different network settings.

1 INTRODUCTION

Training machine learning (ML) models often requires sharing data among multiple parties, e.g., cloud services aggregating data from multiple users to learn a global model. Such data sharing naturally raises concerns (Chakraborty et al., 2013; Saleheen et al., 2016) about exposing sensitive user attributes in datasets. It is thus imperative that both data aggregators and users engage in/propose procedures that minimize leakage of sensitive information.

A widely used technique for obfuscating sensitive attributes in data is context-agnostic noise injection (e.g. Laplace mechanism) (Dwork et al., 2006), that introduces additive noise into a dataset to provide membership security (Li et al., 2007). However, noise injection can impact ML training and inference significantly (Tossou and Dimitrakakis, 2017). This makes such context-agnostic techniques unsuitable in scenarios where only a few attributes need to be concealed. For example, upon sharing patient data for preventive healthcare (Henry et al., 2015; Azam et al., 2019), both privacy (e.g., gender anonymization) and predictivity (e.g., accurate diagnosis) are desirable.

These drawbacks of context-agnostic privacy measures motivate private representation learning (PRL) (Yang et al., 2018), which exploits the knowledge of sensitive attributes in a dataset. PRL considers privacy and predictivity as joint (and possibly competing) objectives, and learns a transformation on the data that balances the goals of (i) obfuscating sensitive attributes of interest to an "adversary" (adv.) while (ii) preserving predictivity on intended targets for an "ally" (Ganin et al., 2016).

Conventionally, the literature on PRL assumes the existence of a single sensitive attribute and a central

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

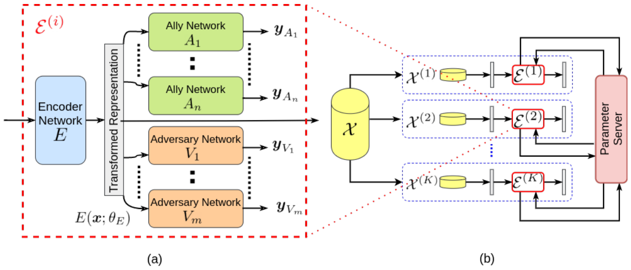


Figure 1: (a) Architecture of a single EIGAN node, consisting of an encoder, n ally, and m adversary networks. (b) D-EIGAN system for distributed EIGAN training, consisting of K different EIGAN nodes, each with their own subset of the full dataset. The nodes must coordinate their local encodings via a parameter server.

dataset (Yang et al., 2018; Roy and Boddeti, 2019; Sadeghi and Boddeti, 2020; Bertran et al., 2019). However, most real-world datasets have multiple sensitive attributes and are collected across multiple distributed nodes. Healthcare records, for example, are (i) spread across hospitals in different regions, (ii) consist of potentially multiple sensitive attributes, such as mental health, gender, ethnicity, etc., and (iii) may have varying notions of privacy that vary from one region to another, e.g., while in Europe racial/ethnic origin are considered as sensitive information (as per GDPR), in USA they are not (as per HIPAA). These challenges call for a *generalized and distributed* PRL methodology that takes into account multiple sensitive attributes, trains on data distributed across nodes, and learns representations that incorporate the privacy/predictivity goals of each node. Communication-efficiency is also a key objective in distributed learning, particularly when it is being deployed in network settings where nodes are restricted to communicate over limited-bandwidth links (Hosseinalipour et al., 2020; Chatterjee et al., 2020), e.g., remote health analytics across user devices (Dimitrov, 2016).

In this paper, we propose a novel PRL architecture called ***Exclusion-Inclusion Generative Adversarial Network (EIGAN)***, which addresses the aforementioned challenges. EIGAN is a generalized PRL technique designed to generate encodings "inclusive" of signals that are of utility to a set of allies, while "exclusive" of signals that can be used by adversaries to recover sensitive attributes. Further, to address the privacy vulnerabilities of pooling raw data, we develop ***D-EIGAN*** (for Distributed-EIGAN), where multiple EIGAN nodes train encoders on their local datasets and synchronize their model parameters periodically, as depicted in Fig. 1. D-EIGAN implements distributed training without noticeable model degradation compared to the centralized EIGAN, while accounting for realistic factors of communication constraints and non-i.i.d data distributions across nodes.

	Adult Dataset		Facescrub Dataset	
Objective	**Ally** (identity)	**Adversary** (gender)	**Ally** (income)	**Adversary** (gender)
Unencoded	0.85	0.85	0.98	0.99
Linear-ARL	**0.84**	0.67	-	-
Kernel-ARL	**0.84**	0.67	-	-
Bertran-PRL	0.82	0.67	**0.56**	0.68
EIGAN	**0.84**	0.67	**0.82**	0.68
% Improv.	Matches closed form solution	Controlled to be equal	47.01%	Controlled to be equal

Table 1: Performance comparison between EIGAN, Sadeghi and Boddeti (2020) (Linear-ARL, Kernel-ARL), and Bertran et al. (2019) (Bertran-PRL) on the Adult & FaceScrub datasets considered in those works. For the same adv. performance, EIGAN obtains a notable improvement over Bertran et al. (2019) (ally improvement of 47.01%). It also reaches the optimal closed form solution of Sadeghi and Boddeti (2020).

Related work. Recent works in PRL (Yang et al., 2018; Roy and Boddeti, 2019; Sadeghi and Boddeti, 2020; Raval et al., 2017; Wu et al., 2018) have proposed centralized architectures that jointly maximize the loss in predicting sensitive attributes while minimizing the loss of target task prediction. Specifically, Yang et al. (2018) proposed a three-network encoder-ally-adversary architecture and showed that the achievable tradeoff between the two objectives is better than that provided by DP. In Roy and Boddeti (2019), the problem was formulated as a non-zero-sum game between the three networks to minimize information leakage in encoded image representations. Sadeghi and Boddeti (2020) experimentally outperform Roy and Boddeti (2019); Louizos et al. (2016); Xie et al. (2017); Zemel et al. (2013) using a minimax optimization among three networks, and derive its closed-form solution when the networks are linear maps. We demonstrate that EIGAN converges to the optimal performance obtained by these closed form solutions. However, unlike the closed form solution in Sadeghi and Boddeti (2020), EIGAN can be extended to account for multiple ally/adv. attributes. Furthermore, EIGAN has computational advantage over Sadeghi and Boddeti (2020) as it does not depend on matrix inversions, and thus can work with higher dimensional data.

Other PRL works take an information-theoretic approach. Bertran et al. (2019) view PRL as minimization of the utility lost in the learned representation, subject to an upper bound on mutual information between the output representation and the sensitive attribute. Similarly, Tripathy et al. (2019) formulate the minimax problem in terms of KL-divergence. EIGAN, on the other hand, considers a cross-entropy PRL formulation, which promotes interpretability and training stability over multiple objectives (discussed in Section 2.1). Furthermore, our experiments show that

EIGAN significantly outperforms the state-of-the-art (Bertran et al., 2019) in the single ally/adversary case. *Distinct from all prior work in PRL, we consider multiple sensitive attributes and distributed learning.*

There are two other related directions in adversarial learning. One addresses privacy-preservation through synthetic data generation (Jordon et al., 2018; Xu et al., 2018), which differs from EIGAN's goal of learning a transformation. The other is fair representation learning (Oh et al., 2016; Edwards and Storkey, 2016; Kusner et al., 2017), which seeks to learn intrinsically fair representations that promote demographic parity on a single attribute (Madras et al., 2018).

Contributions. Our main contributions are:

1. We introduce EIGAN (Section 2.1), generalizing PRL to account for multiple target and sensitive attributes. We prove that EIGAN's encoder utility is maximized if the adversary outputs follow a uniform distribution, and consider the effect of correlations between ally and adversary objectives (Prop. 1).

2. To the best of our knowledge, D-EIGAN (Section 2.2) is the first technique for distributed training of PRL models. We show that when the nodes engaged in the training possess independent and identically distributed (i.i.d) datasets, the objective of D-EIGAN exhibits similar properties to EIGAN (Prop. 3).

3. Our experiments (Section 3) reveal that EIGAN significantly outperforms the state-of-the-art in PRL (Table 1, Fig. 5) and is robust to the choice of adversary architectures (Table 2). We also demonstrate that D-EIGAN matches the performance of EIGAN even as the number of nodes increases (Fig. 7), and is robust even when nodes have different objectives (Fig. 8). We further show the resilience of D-EIGAN to non-i.i.d data distributions across nodes, and under communication restrictions that require partial parameter sharing and delayed model aggregations in the system (Fig. 9).

2 EIGAN MODEL

Overview. Our PRL methodology consists of two phases: training and testing. In the training phase, EIGAN – knowing the sensitive/target labels of interest to adversary/ally on the train dataset – aims to learn the encoder by simulating allies and adversaries. Each of the allies, adversaries, and encoder independently maximize their own utilities by updating their local model parameters. The selfish maximization by each player naturally leads to the minimax optimization in (2). In the testing phase, the test data undergoes a transformation through the trained encoder. The transformed data is used for conventional training and inference by the actual allies and adversaries on their respective tasks of interest.

In Section 2.1, we present the EIGAN formulation for centralized model training, and derive properties of the solution. Then, we extend it to the distributed learning case, D-EIGAN, in Section 2.2. Refer to Appendix. A for the proofs of the propositions.

2.1 EIGAN: Centralized Model Architecture

We first consider a system consisting of n allies, indexed $A_1, ..., A_n$; and m adversaries, indexed $V_1, ..., V_m$. Ally A_i is characterized by model parameters θ_{A_i} and a set of target attributes/labels Y_{A_i} drawn from distribution $\mathcal{Y}_{A_i}$. A_i aims to associate each input sample with its corresponding target attribute in Y_{A_i}. Similarly, adversary V_j parameterized by θ_{V_j} wishes to associate input samples with a set of (known) sensitive attributes/labels Y_{V_j} following distribution $\mathcal{Y}_{V_j}$.

The goal of EIGAN is to learn an encoder E parameterized by θ_E that maximizes the performance of $A_1, ..., A_n$ while minimizing the performance of $V_1, ..., V_m$. The encoder uses a centrally-located dataset $\mathcal{X}$ consisting of N samples, where each sample is represented as a d-dimensional feature vector $\boldsymbol{x}_j \in \mathbb{R}^d$, $j = 1, ..., N$. We let $E(\boldsymbol{x}; \theta_E)$ denote the output of the encoder for a data sample $\boldsymbol{x}$ realized via the parameters θ_E. $E(\boldsymbol{x}; \theta_E) : \mathbb{R}^d \rightarrow \mathbb{R}^l$ is in general a non-linear differentiable function (e.g., a neural network), where l is the dimension of the representation output by the encoder, and typically $l \leq d$.

For $\boldsymbol{x} \in \mathcal{X}$, the encoded representation $E(\boldsymbol{x}; \theta_E)$ is what the allies $A_1, .., A_n$ and adversaries $V_1, .., V_m$ are provided with for their tasks, as depicted in Fig. 1(a). We quantify the utilities of the allies and adversaries as:

$$\begin{aligned} u_{A_i} &= \mathbb{E}_{Y \sim \mathcal{Y}_{A_i}} \left[\log \left(p_{A_i}(Y|E(\mathcal{X}; \theta_E))\right)\right], 1 \leq i \leq n, \\ u_{V_j} &= \mathbb{E}_{Y \sim \mathcal{Y}_{V_j}} \left[\log \left(p_{V_j}(Y|E(\mathcal{X}; \theta_E))\right)\right], 1 \leq j \leq m, \end{aligned} \tag{1}$$

where $p_{A_i}(Y|E(\mathcal{X}; \theta_E))$ and $p_{V_j}(Y|E(\mathcal{X}; \theta_E))$ denote the probabilities of successful inference of target labels $Y \sim \mathcal{Y}_{A_i}$ and sensitive labels $Y \sim \mathcal{Y}_{V_j}$ for ally A_i and adversary V_j, respectively, over the outputs that the encoder E provides for the dataset $\mathcal{X}$. This leads to our minimax game among three types of players, in which two (the encoder and allies) are colluding against the third (the adversary). Specifically, we formulate the optimization problem:

$$\min_{\theta_V = \{\theta_{V_j}\}_{j=1}^m} \quad \max_{\theta_E, \theta_A = \{\theta_{A_i}\}_{i=1}^n} U(\theta_E, \theta_A, \theta_V), \tag{2}$$

where

$$U(\theta_E, \theta_A, \theta_V) = \sum_{i=1}^{n} \alpha_{A_i} u_{A_i} - \sum_{j=1}^{m} \alpha_{V_j} u_{V_j}. \qquad (3)$$

Here, $\alpha_{A_i}, \alpha_{V_j} > 0$ denote normalized importance parameters placed on each objective such that $\sum_{i=1}^{n} \alpha_{A_i} + \sum_{j=1}^{m} \alpha_{V_j} = 1$. Similar to the encoder, we assume that the ally and adversary are non-linear, differentiable functions. The encoder in (2) seeks to maximize the achievable utility of the allies while minimizing those of the adversaries, operating in conjunction with the allies in the inner max layer of (2). The adversaries then operate on the encoder result in the outer min layer, where each adversary V_j aims to maximize its utility u_{V_j} by updating θ_{V_j}, as it cannot access other ally/adversary's parameters directly.

It is worth noting that, similar to the formulation based on mutual information in Bertran et al. (2019), our analysis on the expected posterior distribution of the predictions in EIGAN map directly to interpretable metrics such as accuracy (Bassily et al., 2018) and generalization error (Feder and Merhav, 1994), instead of the worst case guarantees provided by context-agnostic privacy frameworks such as DP.

Intuitively, the encoder will attempt to diminish the adversary predictions to a random guess, i.e., to a uniform distribution over its target labels (Abadi and Andersen, 2016). However, this may be difficult to achieve when the interests of the allies and adversaries are related, which makes the weights $\alpha_{A_i}, \alpha_{V_j}$ important to the minimax solution in (2) formalized in the proposition below:

Proposition 1. *Let $\mathcal{O}$ denote the set of all (i, j) pairs of allies A_i and adversaries V_j for which $Y_{A_i} \cap Y_{V_j} \neq \emptyset$, i.e., overlapping interests. Given a fixed encoder E in EIGAN architecture, if $\mathcal{O} = \emptyset$, the overall score in (2) is maximized when the adversaries' output predictions follow a uniform distribution. On the other hand, if $\mathcal{O} \neq \emptyset$, then for each overlapping label, the architecture proposed by (2) considers the utility of the attributes that have the higher importance weight, i.e., A_i if $\alpha_{A_i} > \alpha_{V_j}$ and V_j if $\alpha_{A_i} < \alpha_{V_j}$.*

Prop. 1 shows that given an encoded representation, if the allies and adversaries possess non-overlapping interests, then a uniform prediction distribution among the sensitive parameters of interest to the adversaries is adopted by the optimal solution. In Appendix. D.4, we consider an experiment with such overlapping interests and equal importance weights, and find that EIGAN is unable to balance the objectives.

In practice, coincidental overlaps between ally and adversary interests would be relatively rare, but could nonetheless occur. In such cases, EIGAN must balance predictivity and privacy, which leads to different ally and adversary outputs described in Prop. 1. We further analyze EIGAN's characteristics when there is a linear relationship between the target distribution of an ally and an adversary (see Prop. 2 in Appendix. A.2).

Model training. We train the encoder and the allies/adversaries in EIGAN by alternately updating their parameters using stochastic gradient descent (SGD) to minimize their cross-entropy (CE) loss. For the encoder, we define the CE-loss $\mathcal{L}_E$ for a single training instance as a weighted combination of the predictive capability of the allies and adversaries as

$$\mathcal{L}_E = \sum_{i=1}^{n} \underbrace{-\langle \boldsymbol{y}_{A_i}, \log \hat{\boldsymbol{y}}_{A_i} \rangle}_{\text{loss of ally } A_i,\ \mathcal{L}_{A_i}} - \alpha \cdot \sum_{j=1}^{m} \underbrace{-\langle \boldsymbol{y}_{V_j}, \log \hat{\boldsymbol{y}}_{V_j} \rangle}_{\text{loss of adversary } V_j,\ \mathcal{L}_{V_j}}, \qquad (4)$$

where $\langle ., . \rangle$ denotes inner product, and log is applied element-wise. $\boldsymbol{y}_{A_i}$ and $\boldsymbol{y}_{V_j}$ are the binary vector representations of the true class labels for ally A_i and adversary V_j, respectively, while $\hat{\boldsymbol{y}}_{A_i}$ and $\hat{\boldsymbol{y}}_{V_i}$ are the vectors of soft predictions (i.e., probabilities) for each class. Here, we have made the simplifications $\alpha_{A_i} = \alpha/n \; \forall i$ and $\alpha_{V_j} = (1-\alpha)/m \; \forall j$, where $\alpha \in (0, 1)$ is tuned to emphasize either predictivity (higher α) or privacy (lower α). It can be seen that the minimization of loss $\mathcal{L}_E$ is equivalent to the maximization of utility defined by (3). In each epoch, we average $\mathcal{L}_E$ over a minibatch of size J to obtain an estimate of (1), and update θ_E based on the gradient. Then, we update the θ_{A_i} and θ_{V_i} according to (4). See Alg. A1 in Appendix. B for details.

Loss consideration. Alternative objectives to (4) exist in PRL literature. In particular, recent works (Huang et al., 2018; Tripathy et al., 2019; Bertran et al., 2019) formulate the adversarial loss using KL divergence. We choose CE-loss over KL-divergence based on the fact that KL divergence fails to give meaningful value under disjoint distributions (Adler and Lunz, 2018). Also, our CE-loss formulation is unconstrained as opposed to KL-divergence formulation which is a Lagrangian dual of the constrained formulation (Bertran et al., 2019). As analyzed in Prop. 1, our formulation naturally pushes adversary prediction towards uniform distribution, however, the same does not hold for the constrained formulation. Our results in Table 1 and Fig. 5 validate our formulation choice. We show consistent improvements over the state-of-the-art (Bertran et al., 2019) that uses KL divergence.

Visual demonstration. Fig. 2 is a visual demonstration of EIGAN's trained representation on a synthetic

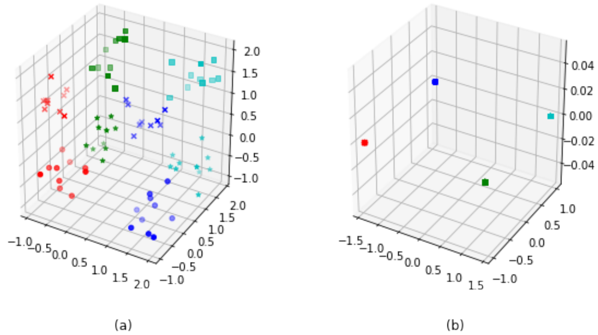


Figure 2: (a) Synthetic dataset with eight groups of points, two allies, and one adversary. The allies are interested in separating the color pairs (the two horizontal axes), and the adversary is interested in classifying shapes (the vertical axis). (b) EIGAN's encoding has collapsed the adversary dimension while preserving the allies.

dataset. There are two allies A_1 and A_2 which are each interested in separating data points along one of the horizontal axes, and an adversary V that is interested in separation along the vertical axis. We see in (b) that the EIGAN encoding collapses the data along the vertical axis while retaining separability in the other two dimensions. Other illustrations are given in Appendix. C.

2.2 D-EIGAN: Distributed EIGAN Model

The distributed setting for EIGAN (D-EIGAN) is depicted in Fig. 1(b). There are K nodes in the system, denoted $\mathcal{E}^{(1)}, ..., \mathcal{E}^{(K)}$, and a parameter server for model synchronization. Each node $\mathcal{E}^{(k)}$ has a set of allies, denoted $A_1^{(k)}, ..., A_{n(k)}^{(k)}$ with target label sets $Y_{A^{(k)}} = \{Y_{A_1^{(k)}}, ..., Y_{A_{n(k)}^{(k)}}\}$, a set of adversaries, denoted $V_1^{(k)}, ..., V_{m(k)}^{(k)}$ with target sets $Y_{V^{(k)}} = \{Y_{V_1^{(k)}}, ..., Y_{V_{m(k)}^{(k)}}\}$, and a subset $\mathcal{X}_k \subset \mathcal{X}$ of N_k datapoints from the overall dataset $\mathcal{X}$ of N samples. These local datasets are in general non-overlapping, and may differ in size. While the specific allies and adversaries may differ at each node, the goal is to train encoder models that maximize all allies' and minimizes all adversaries' performances, so that the encodings are meaningful throughout the system. Since sharing the raw datasets could potentially leak sensitive information, each node $\mathcal{E}^{(k)}$ will train its own local encoder $E^{(k)}(\boldsymbol{x}; \theta_{E^{(k)}})$, and the server in Fig. 1(b) will periodically aggregate the locally-trained models.

The utility function for node $\mathcal{E}^{(k)}$ is defined as

$$U^{(k)}(\theta_{E^{(k)}}, \theta_{A^{(k)}}, \theta_{V^{(k)}}) = \sum_{i=1}^{n} \alpha_{A_i^{(k)}} u_{A_i^{(k)}} - \sum_{j=1}^{m} \alpha_{V_j^{(k)}} u_{V_j^{(k)}}, \quad (5)$$

where $\theta_{A^{(k)}} = \{\theta_{A_i^{(k)}}\}_{i=1}^{n(k)}$ and $\theta_{V^{(k)}} = \{\theta_{V_j^{(k)}}\}_{j=1}^{m(k)}$ denote the sets of ally and adversary parameters at node $\mathcal{E}^{(k)}$, and $u_{A_i^{(k)}}, u_{V_j^{(k)}}$ denote the utility functions of $A_i^{(k)}, V_j^{(k)}$ defined analogously to (1). $\alpha_{A_i^{(k)}}, \alpha_{V_j^{(k)}} > 0$ denote the normalized importance parameters for node $\mathcal{E}^{(k)}$, where $\sum_{i=1}^{n} \alpha_{A_i^{(k)}} + \sum_{j=1}^{m} \alpha_{V_j^{(k)}} = 1$. This leads to the following minimax game for the distributed case:

$$\begin{aligned} \min_{\mathcal{S}_V} \max_{\mathcal{S}_E, \mathcal{S}_A} \; & \frac{1}{K} \sum_{k=1}^{K} U^{(k)}(\theta_{E^{(k)}}, \theta_{A^{(k)}}, \theta_{V^{(k)}}) \\ \text{s.t.} \quad & \theta_{E^{(k)}} = \theta_{E^{(k')}}, \; k \neq k', 1 \leq k, k' \leq K, \end{aligned} \quad (6)$$

where $\mathcal{S}_V = \{\theta_{V^{(k)}}\}_{k=1}^{K}, \mathcal{S}_E = \{\theta_{E^{(k)}}\}_{k=1}^{K}$, and $\mathcal{S}_A = \{\theta_{A^{(k)}}\}_{k=1}^{K}$. The constraint in (6) ensures that the optimal encoder is the same across all nodes, even though each node may have different allies and adversaries. In this way, an encoded datapoint $E^{(k)}(\boldsymbol{x}; \theta_E)$ at node k could be transferred to another node k' and applied to a task $A_i^{(k')}$ privately, e.g., for anonymized user data sharing during single sign-ons.

Distributed model training. While solving (6) in a distributed manner, D-EIGAN learns both a global model and personalized local models (allies and adversaries) (Smith et al., 2017), unlike standard Federated Learning (FL).

Our algorithm consists of two iterative steps. The first is *local update*: each $\mathcal{E}^{(k)}$ conducts a series of δ SGD iterations. For each minibatch in SGD, training proceeds as in the centralized case, with the encoder, allies', and adversaries' parameters updated via SGD to minimize the CE-losses $\mathcal{L}_E^{(k)}$, $\mathcal{L}_{A_i}^{(k)}$, and $\mathcal{L}_{V_j}^{(k)}$ defined as in (4) but in this case for each node. The second step is *global aggregation*, in which each $\mathcal{E}^{(k)}$ uploads its locally-trained encoder to the parameter server to construct a global version, after every δ SGD iterations. We introduce a sparsification technique here in which each node selects a fraction ϕ of its parameters at random to upload for each aggregation. Letting $\mathcal{Q}_k$ be the indices chosen by $\mathcal{E}^{(k)}$, then the vector recovered at the server is $\tilde{\theta}_{E^{(k)}}$, where $\tilde{\theta}_{E^{(k)}}(q) = \theta_{E^{(k)}}(q)$ if $q \in \mathcal{Q}_k$ and 0 otherwise. With this, the global aggregation becomes the weighted average $\theta_E = \sum_k \frac{N_k}{N} \tilde{\theta}_{E^{(k)}}$. Then, the server also selects a fraction ϕ of indices at random to synchronize each node k with on the downlink. Letting $\mathcal{Q}$ be these indices, each node k sets $\theta_{E^{(k)}}(q) = \theta_E(q)$ if $q \in \mathcal{Q}$, and makes no change to the qth parameter otherwise. The pseudo-code of the training procedure is given in Alg. 1.

The synchronization frequency δ and sparsification factor ϕ are directly related to the amount of data transferred through the system: as δ increases, up-

Algorithm 1 D-EIGAN training

```
Notation:
$\theta_E$: global parameter vector
$\mathcal{Q}_k$: uniformly random choice of indices at node $\mathcal{E}^{(k)}$
$\tilde{\theta}_{E^{(k)}}$: parameter vector recovered at the server for encoder $E^{(k)}$, with its $q$th element denoted $\tilde{\theta}_{E^{(k)}}(q)$
$\phi$: fraction of parameters shared
$\delta$: number of epochs between aggregations
$(\cdot)_j$: value for the $j$th minibatch
$\mathcal{L}_{A_i^{(k)}}$ and $\mathcal{L}_{V_i^{(k)}}$: loss of ally $A_i^{(k)}$ and adversary $V_i^{(k)}$
$\eta_E, \eta_A$, and $\eta_V$: learning rates
Aggregation at Parameter Server:
Initialize parameter $\theta_E$
for each update round do
    Update parameter vector: $\theta_E \leftarrow \sum_{k=1}^{K} \frac{N_k}{N} \tilde{\theta}_{E^{(k)}}$
end for
Local Training at Node $\mathcal{E}^{(k)}$:
Initialize $\{\theta_{A_i^{(k)}}\}_{i=1}^{n^{(k)}}$ and $\{\theta_{V_j^{(k)}}\}_{j=1}^{m^{(k)}}$
Download initial $\theta_E$ from parameter server
for number of training epochs do
    After $\delta$ epochs, update $\phi \cdot |\theta_{E^{(k)}}|$ chosen parameters from parameter server: $\theta_{E^{(k)}}(q) = \theta_E(q)$ if $q \in \mathcal{Q}$
    Sample a minibatch $J$ from local dataset $\mathcal{X}_k$
    Update encoder: $\theta_{E^{(k)}} \leftarrow \theta_{E^{(k)}} - \eta_E \cdot \nabla_{\theta_{E^{(k)}}} \mathcal{L}_{E^{(k)}}$
    Update ally/adversary parameters:
        $\theta_{A_i^{(k)}} \leftarrow \theta_{A_i^{(k)}} - \eta_A \cdot \nabla_{\theta_{A_i^{(k)}}} \mathcal{L}_{A_i^{(k)}}$,
        $\theta_{V_i^{(k)}} \leftarrow \theta_{V_i^{(k)}} - \eta_V \cdot \nabla_{\theta_{V_i^{(k)}}} \mathcal{L}_{V_i^{(k)}}$
    After $\delta$ epochs, upload $\phi|\theta_{E^{(k)}}|$ encoder parameters: $\tilde{\theta}_{E^{(k)}}(q) = \theta_{E^{(k)}}(q)$ if $q \in \mathcal{Q}_k$, else $\tilde{\theta}_{E^{(k)}}(q) = 0$
end for
```

link transfers to the server occur less frequently; as ϕ decreases, each uplink/downlink transmission requires fewer communication resources. This is an important consideration in networking applications where the nodes communicate over a resource-constrained channel (Wang et al., 2019; McMahan et al., 2017). Fractional parameter sharing, similar to pruning (both choose a subset of parameters), mimics the additive-noise DP mechanism (Huang et al., 2020) on model weights, reducing associated leakage (Fredrikson et al., 2015; Shokri et al., 2017) to any untrusted entity with access to the system. We study the effect of δ and ϕ on D-EIGAN performance in Section 3.2.

In D-EIGAN, the allies and adversaries may differ at each node, and each node trains an individual local encoder. Since the encoder parameters are globally synchronized, however, the local encoder implicitly trains using global union of allies/adversaries across nodes. In the case that the nodes have same objectives and i.i.d. datasets, we show that D-EIGAN yields the same properties as Prop. 1:

Proposition 3. *Given a set of fixed encoders in the D-EIGAN architecture, if all the nodes have the same number of allies and adversaries with the same sets of target labels $Y_{A^{(k)}} = Y_{A^{(k')}}$ and $Y_{V^{(k)}} = Y_{V^{(k')}}$, $1 \leq k, k' \leq K$, then Prop. 1 holds for all the allies and adversaries belonging to different nodes if the local datasets at each node are i.i.d.*

When the nodes have different objectives, we show that the importance of each objective is proportional to the number of nodes implementing it; see Prop. 4 in Appendix. A.4.

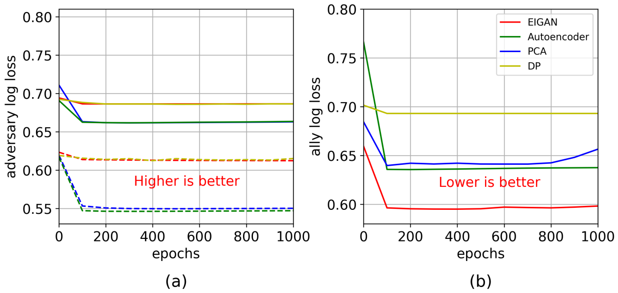
3 EXPERIMENTAL EVALUATION/DISCUSSION

We now turn to an experimental evaluation of our methodology. We analyze EIGAN's convergence characteristics and compare its performance with relevant baselines in Section 3.1, and evaluate D-EIGAN compared to the centralized case and as the system characteristics change in Section 3.2.

Datasets. We consider datasets: MNIST (LeCun and Cortes, 2010), MIMIC-III (Johnson et al., 2016), Adult (Dua and Graff, 2017), and FaceScrub (Ng and Winkler, 2014). MNIST consists of 60,000 handwritten digits with labels 0-9. MIMIC has medical information from hospitals with attributes, such as vitals and medication; we obtain a dataset consisting of 58,976 patients by joining multiple tables on patient IDs. Adult consists of 45,223 records extracted from the 1994 census data. Facescrub is a dataset comprising over 22,000 images of celebrities with identity and gender labels.

Objectives. In MIMIC, we consider survival (2-class) as the ally objective, and gender (2-class) and race (3-class) as adversary objectives. In the FaceScrub dataset, as in Bertran et al. (2019), the ally objective is user identity (200-class), and the adversary objective is gender (2-class). In MNIST, we consider whether a digit is even or odd (2-class) as the ally objective, and the label of the digit (10-class) as the adversary objective. In Adult, as in Sadeghi and Boddeti (2020), the ally objective is an annual income classification (more or less than 50K) and the adversary objective is gender. We also generate synthetic Gaussian datasets to analyze the effect of ally/adversary class overlap in some experiments.

Implementation. We use fully connected networks (FCNs) for the encoder, allies, and adversaries in the experiments on MIMIC and the synthetic datasets. The FCN encoder uses ReLU (Nair and Hinton, 2010) activation for the hidden layers and tanh activation for the final fully-connected layer, whereas the ally and adversaries use sigmoid activation in the final layer. We use dropout (Srivastava et al., 2014) and L2-regularization to prevent network overfitting. For



(a)

(b)

Figure 3: Predictivity and privacy comparison between EIGAN and baselines on MIMIC. (a) On the adversary prediction (gender, solid lines and race, dashed lines), EIGAN matches DP's performance (by tuning DP's noise). (b) On the ally prediction (survival), EIGAN achieves noticeable improvement over the baselines.

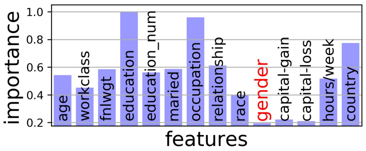


Figure 4: Feature importance derived from EIGAN encoder on Adult dataset results summarized in Table 1.

FaceScrub, we employ U-Net (Ronneberger et al., 2015) for the encoder and Xception-Net (Chollet, 2017) for the ally/adversary as in Bertran et al. (2019). For Adult, we employ linear FCN as in Sadeghi and Boddeti (2020). Unless otherwise stated, we set $\alpha = 0.5$ (i.e., equal privacy/predictivity importance). We train to minimize CE loss over 70/30 training/test splits on a system with 8 GB GPU and 64 GB RAM.

Baselines. We consider six baselines: principal component analysis (PCA) (Abdi and Williams, 2010), autoencoders (Kramer, 1991), differential privacy (DP) in the form of Laplace Mechanism as in Yang et al. (2018), and the methods in Sadeghi and Boddeti (2020), Bertran et al. (2019). Autoencoders and PCA preserve information content and do not have explicit privacy objectives; they are expected to give encoded data that has good predictivity. PCA chooses the number of components retaining 99% of the variance, and we train the autoencoder to transform data to the same dimensional space as PCA. As discussed in Section 1, DP is widely used for context-agnostic privacy. For DP, we employ the Laplace mechanism (Yang et al., 2018). Bertran et al. (2019) is the most recent state-of-the-art in adversarial PRL; in this case, we use their open-source implementation and compare on the setting described in their paper. We also compare against the closed form optimal solution of Sadeghi and Boddeti (2020) for linear maps on their Adult dataset use case, where Sadeghi and Boddeti (2020) outperforms Roy and Boddeti (2019); Louizos et al. (2016); Xie et al. (2017); Zemel et al. (2013).

Model	**Ally** (accuracy)	**Adv.** (accuracy)
Resnet152 Unencoded	0.99	0.99
Resnet152	0.85	0.45
ResNext101	0.86	0.42
Resnet101	0.88	0.64
Resnet50	0.87	0.56
WideResnet101	0.85	0.42
VGG19	0.77	0.42

Table 2: Accuracy of various architectures used to infer ally (even/ odd) and adversary (digits 0-9) objectives on MNIST encoded using ResNet152-trained EIGAN. We see that the ally accuracies are consistent across network architectures, and the adversary accuracies remain significantly below the performance on the unencoded data.

All of our code using PyTorch (Paszke et al., 2019) and trained models are available at `https://github.com/shams-sam/PrivacyGANs`. For each experiments, we report cross-entropy loss and/or accuracy from the testing step of PRL.

3.1 Centralized EIGAN

Performance comparison. We first compare the ally and adversary losses over training epochs between EIGAN, autoencoder, PCA, and DP on the MIMIC dataset in Fig. 3. Note that the recent baselines (Sadeghi and Boddeti, 2020; Bertran et al., 2019) cannot handle multiple adversary objectives. It is observed in (a) that EIGAN is able to match the adversary losses of DP, while in (b) the EIGAN ally loss matches that of PCA and autoencoder while outperforming DP by a significant margin. Thus, EIGAN is capable of achieving private representations while simultaneously maintaining the predictivity of the encoded representations.

Next, we compare EIGAN with Sadeghi and Boddeti (2020) and Bertran et al. (2019) on the Adult and Facescrub dataset settings considered in these works, respectively. Note that the linearity requirement in Sadeghi and Boddeti (2020) impedes its usage on nonlinear models like the U-Net and Xception-Net employed for Facescrub by Bertran et al. (2019). For comparison, we adjust α in (4) to equalize the resulting adversary performances between the models. Table 1 gives the results: EIGAN matches the performance of Sadeghi and Boddeti (2020)'s optimal closed-form solution on Adult. On the Facescrub dataset, it displays a 47% improvement in the ally's task of identity recognition when compared to Bertran et al. (2019). This validates our choice of optimization using cross-entropy loss in (4) for PRL over the technique of optimization using KL divergence that is common in recent PRL literature (Bertran et al., 2019; Huang et al., 2018; Tripathy et al., 2019).

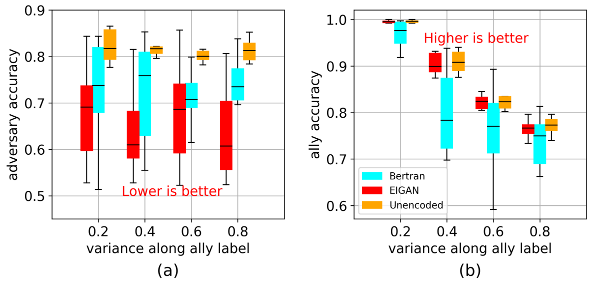


Figure 5: Effect of varying the ally class overlap (by changing the variances of synthetic Gaussian data) on the performance of EIGAN, Bertran et al. (2019), and the unencoded data. (a) and (b) plot the achieved accuracies of the adv. and ally objectives, respectively. EIGAN is able to consistently outperform both baselines on the adversary objective, and obtains performance close to the unencoded data for the ally.

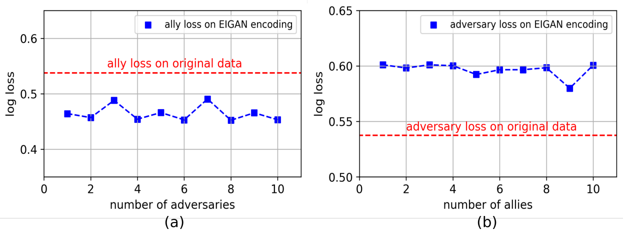


Figure 6: EIGAN's effect of the number of (a) adversaries, and (b) allies on the testing loss for MIMIC-III. The ally/adversary objectives are chosen as different attributes from the source. The achievable loss is reasonably constant and is not affected by addition of more allies/adversaries.

Robustness of learned representation. In Fig. 4 we consider the importance placed by EIGAN encoder on input features of Adult dataset for learning the private representations. It can be observed that the importance of gender and it's correlated features is very low. This implies that the learnt representations minimize the signals w.r.t adversary's interest, i.e., gender. We next consider the robustness of EIGAN's learned representation to ally and adversary architectures that deviate from the one used for training. Table 2 shows the performance of varying architectures (ResNet (He et al., 2016), ResNext (Xie et al., 2017), etc.) for allies and adversaries applied to the data encoded using EIGAN trained with ResNet152 adversary on MNIST. We see that the representations learned by EIGAN are able to obfuscate adversary targets from the other networks. Adversary accuracy remains significantly below the performance on the unencoded data, validating the robustness to differences between simulated and actual adversaries.

Varying ally/adversary overlap. Next, we consider the effect of class overlap for the ally/adv. objectives on model performance. To do this, we generate a 2D dataset consisting of four Gaussians with means at $(x, y) = (1, 1), (1, 2), (2, 1), (2, 2)$, each corresponding to one class. The variance of these Gaussian-distributed classes is adjusted to achieve varying degrees of overlap. Fig. 7(c) shows an instance of this dataset: the ally is interested in differentiating color, while the adversary wants to differentiate shape. Fig. 5 shows the effect of the ally label variance on the resulting accuracies for EIGAN, the method in Bertran et al. (2019), and the unencoded data. As the ally variance increases, we observe that (a) the accuracy of the adversary for EIGAN remains consistently lower than that of the others, while (b) the accuracy on the ally objective for EIGAN remains higher than that of Bertran et al. (2019) and is comparable to the unencoded case. Similar results are seen on changing the adversary variance (see Appendix. D.1).

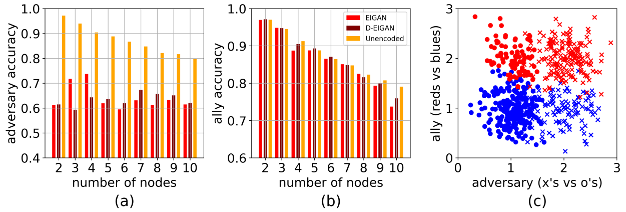


Figure 7: Comparison of (a) adversary and (b) ally performance as the number of nodes in the system is increased from $K = 2$ to 10, for D-EIGAN $(\phi, \delta = 1)$, EIGAN, and unencoded. Node k's data, $k = 1, ..., K$ is generated from four Gaussians centered on a unit square, each with $\sigma^2 = 0.1k$, i.e. increasing variance. (c) visualizes the ally (reds vs. blues) and adv. (x's vs. o's) objectives for node $k = 3$. As expected, the ally performs worse with higher K, but D-EIGAN is able to match EIGAN's performance.

Varying system dimensions. We also consider the impact of the encoding dimension l and the number of allies/adversaries on EIGAN's performance using MIMIC. We summarize our key findings here: (i) We observe (in Fig. 6) that the final test loss obtained by an adversary (ally) under varying number of allies (adversaries) stays reasonably constant. (ii) We find (in Appendix. D.5) that as encoding dimension l is increased, EIGAN converges faster (fewer epochs), and is able to achieve a lower testing loss, even as l exceeds the input dimension. Thus, encodings are robust to the number of objectives that are included in EIGAN.

3.2 Distributed EIGAN (D-EIGAN)

Varying number of nodes. For the distributed case, we first study the effect of increasing the number of training nodes K. We use synthetic Gaussian data and generate non-i.i.d. data distributions across the nodes by increasing the variance of the Gaussians at each subsequent node k (Fig. 7(c) shows the distribution for $k = 3$). Fig. 7(a)&(b) show the resulting ally and adversary accuracies obtained when trained on D-EIGAN, on EIGAN, and on the unencoded data. As

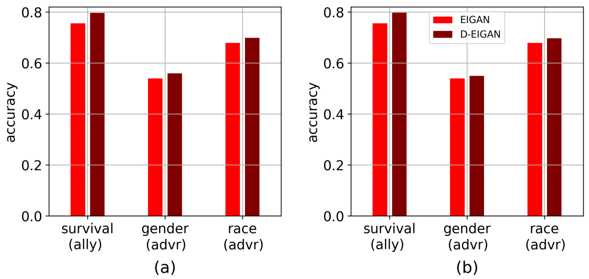


Figure 8: Performance of ally and adversary objectives trained on D-EIGAN ($K = 10, \phi = 0.8, \delta = 2$, non-i.i.d) for MIMIC in the cases of (a) all nodes having all three objectives and (b) each node having the ally but only one of the adversaries. The distribution of objectives across the nodes does not affect the resulting accuracies.

K increases, the ally performance degrades in each case, due to the higher variance for each class exhibited in the overall dataset $\mathcal{X}$. Overall, we see that D-EIGAN matches the performance of the centrally-trained EIGAN in both metrics, which shows that distributed learning can yield a comparable solution when all parameters ($\phi = 1$) are synchronized frequently ($\delta = 1$). See Appendix. E.1 for results on i.i.d. data.

Varying objectives across nodes. Next, we study the effect of varying ally and adversary objectives across nodes. For this, we consider the MIMIC dataset and allocate the dataset across $K = 10$ nodes randomly so that each has a different distribution of patient data. In Fig. 8, we show the accuracies achieved by D-EIGAN on the one ally and two adversary objectives for two cases: (a) when each node has all three objectives, and (b) when each node has the ally objective, but half have one adversary objective and half have the other. The EIGAN performance on the full dataset is included for comparison. The dataset is distributed in a non-i.i.d manner across nodes by non-uniform random sampling. We see that D-EIGAN in (a) only has a slight improvement over (b) in the case of the gender adversary, which indicates that D-EIGAN is robust to varying node objectives, even though the aggregation period has increased ($\delta = 2$) and the fraction of parameters shared has decreased ($\phi = 0.8$) from Fig. 7. The implication of this is that once a data sample is encoded at a node via D-EIGAN, it can be transferred to another node with different objectives and securely applied to ally tasks there, e.g., referring to the healthcare use case in Section 1, if a patient moves to a different hospital with different health regulations. Similar conclusions are drawn when the data is i.i.d across nodes (see Appendix. E.2).

Varying synchronization parameters. Finally, we consider the impact of the aggregation period δ and the sparsification factor ϕ on D-EIGAN. This has implications for the communication resources between the nodes and the server required for training, as discussed in Section 2.2. For this experiment, we use the setting from the experiment in Fig. 8(a), i.e., with non-i.i.d data and all nodes having all three objectives. In Fig. 9, we show the performance of D-EIGAN as (a) δ increases and (b) ϕ increases (EIGAN shown for comparison). In (a), we see that D-EIGAN is robust to the number of training epochs between aggregations, implying that it can be increased to limit the frequency of transmissions to/from the server. In (b), we similarly observe generally robust performance as the fraction of sharing changes, though surprisingly, the performance noticeably *decreases* once ϕ reaches 1 and all are shared. A similar effect was observed by Sattler et al. (2019), that in the case of distributed model training over non-i.i.d datasets, sparsification actually can *enhance* performance because it minimizes the effect of data bias at each node on the global model. Indeed, in the i.i.d case, we do not observe this effect (see Appendix. E.3). Thus, we conclude that D-EIGAN is well suited for communication-constrained environments.

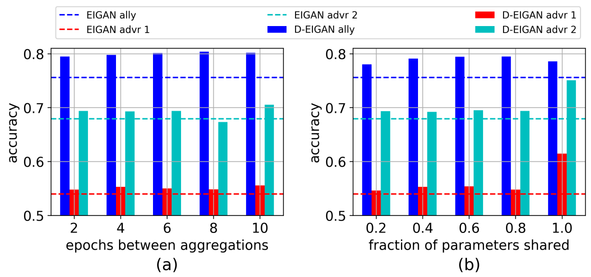


Figure 9: Effect of (a) aggregation frequency δ ($\phi = 0.8$) and (b) sparsification factor ϕ ($\delta = 2$) on ally/adv. performance on D-EIGAN for the non-i.i.d case in Fig. 8(a). The robust performance shows that D-EIGAN can be applied in communication-constrained environments.

4 CONCLUSION

We developed the first methodology for generalized and distributable PRL. EIGAN accounts for the presence of multiple allies and adversaries with potentially overlapping objectives, and D-EIGAN addresses privacy concerns and resource constraints in scenarios with decentralized data. We proved that for an optimal encoding, the adversary's output from EIGAN follows a uniform distribution, and that dependencies between ally and adversary interests requires careful balancing of objectives in encoder optimization. Our experiments showed that EIGAN outperforms six baselines in jointly optimizing predictivity and privacy on different datasets and system settings. They also showed that D-EIGAN achieves comparable performance to EIGAN with different numbers of training nodes and as the training parameters vary to account for communication constraints.

Acknowledgements

This research was partially supported by the Northrop Grumman Cybersecurity Research Consortium (NGCRC), the Office of Naval Research (ONR) under grant N00014-21-1-2472, and the National Science Foundation (NSF) under grant CNS-2106891.

References

Martín Abadi and David G Andersen. Learning to Protect Communications with Adversarial Neural Cryptography. *arXiv preprint*, *arXiv:1610.06918*, 2016.

Hervé Abdi and Lynne J Williams. Principal Component Analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 2, 2010.

Jonas Adler and Sebastian Lunz. Banach Wasserstein GAN. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

Sheikh Shams Azam, Manoj Raju, Venkatesh Pagidimarri, and Vamsi Chandra Kasivajjala. CASCADENET: An LSTM based Deep Learning Model for Automated ICD-10 Coding. In *Future of Information and Communication Conference (FICC)*, 2019.

Raef Bassily, Shay Moran, Ido Nachum, Jonathan Shafer, and Amir Yehudayoff. Learners that use little information. In *Algorithmic Learning Theory*, 2018.

Martin Bertran, Natalia Martinez, Afroditi Papadaki, Qiang Qiu, Miguel Rodrigues, Galen Reeves, and Guillermo Sapiro. Adversarially Learned Representations for Information Obfuscation and Inference. In *International Conference on Machine Learning (ICML)*, 2019.

Supriyo Chakraborty, Kasturi Rangan Raghavan, Matthew P Johnson, and Mani B Srivastava. A Framework for Context-aware Privacy of Sensor Data on Mobile Systems. In *Mobile Computing Systems and Applications*, 2013.

Baibhab Chatterjee, Dong-Hyun Seo, Shramana Chakraborty, Shitij Avlani, Xiaofan Jiang, Heng Zhang, Mustafa Abdallah, Nithin Raghunathan, Charilaos Mousoulis, Ali Shakouri, Saurabh Bagchi, and Shreyas Sen. Context-Aware Collaborative Intelligence with Spatio-Temporal In-Sensor-Analytics for Efficient Communication in a Large-Area IoT Testbed. *IEEE Internet of Things Journal*, 2020.

François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Dimiter V Dimitrov. Medical Internet of Things and Big Data in Healthcare. *Healthcare Informatics Research*, 22(3):156, 2016.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017. URL http://archive.ics.uci.edu/ml.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, 2006.

Harrison Edwards and Amos Storkey. Censoring Representations with an Adversary. In *International Conference on Learning Representations (ICLR)*, 2016.

M. Feder and N. Merhav. Relations between Entropy and Error Probability. *IEEE Transactions on Information Theory*, Vol. 40, 1994.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that exploit Confidence Information and Basic Countermeasures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial Training of Neural Networks. *Journal of Machine Learning Research (JMLR)*, Vol. 17, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Katharine E Henry, David N Hager, Peter J Pronovost, and Suchi Saria. A Targeted Real-Time Early Warning Score (TREWScore) for Septic Shock. *Science Translational Medicine*, Vol. 7, 2015.

Seyyedali Hosseinalipour, Sheikh Shams Azam, Christopher G Brinton, Nicolo Michelusi, Vaneet Aggarwal, David J Love, and Huaiyu Dai. Multi-Stage Hybrid Federated Learning over Large-Scale Wireless Fog Networks. *arXiv preprint 2007.09511*, 2020.

Chong Huang, Peter Kairouz, and Lalitha Sankar. Generative Adversarial Privacy: A Data-Driven Approach to Information-Theoretic Privacy. In *Asilomar Conference on Signals, Systems, and Computers*, 2018.

Yangsibo Huang, Yushan Su, Sachin Ravi, Zhao Song, Sanjeev Arora, et al. Privacy-Preserving Learning via Deep Net Pruning. *arXiv preprint*, *2003.01876*, 2020.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, et al. MIMIC-III, A Freely Accessible Critical Care Database. *Scientific Data*, 3, 2016.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees. In *International Conference on Learning Representations (ICLR)*, 2018.

Mark A Kramer. Non-Linear Principal Component Analysis using Auto-Associative Neural Networks. *AIChE Journal*, Vol. 37, 1991.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual Fairness. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Yann LeCun and Corinna Cortes. MNIST Handwritten Digit Database, 2010. URL http://yann.lecun.com/exdb/mnist/.

Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE International Conference on Data Engineering*, 2007.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The Variational Fair Autoencoder. In *International Conference on Learning Representations (ICLR)*, 2016.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning Adversarially Fair and Transferable Representations. In *International Conference on Machine Learning (ICML)*, 2018.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Vinod Nair and Geoffrey E Hinton. Rectified Linear Units improve Restricted Boltzmann Machines. In *International Conference on Machine Learning (ICML)*, 2010.

H. Ng and S. Winkler. A Data-Driven Approach to Cleaning Large Face Datasets. In *IEEE International Conference on Image Processing (ICIP)*, 2014.

Seong Joon Oh, Rodrigo Benenson, Mario Fritz, and Bernt Schiele. Faceless Person Recognition: Privacy Implications in Social Media. In *European Conference on Computer Vision (ECCV)*, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Nisarg Raval, Ashwin Machanavajjhala, and Landon P Cox. Protecting Visual Secrets using Adversarial Nets. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

Proteek Chandan Roy and Vishnu Naresh Boddeti. Mitigating Information Leakage in Image Representations: A Maximum Entropy Approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Bashir Sadeghi and Vishnu Naresh Boddeti. Imparting fairness to pre-trained biased representations. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

Nazir Saleheen, Supriyo Chakraborty, Nasir Ali, Md Mahbubur Rahman, Syed Monowar Hossain, Rummana Bari, Eugene Buder, Mani Srivastava, and Santosh Kumar. mSieve: Differential Behavioral Privacy in Time Series of Mobile Sensor Data. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and Communication-Efficient Federated Learning from Non-Iid Data. *IEEE Transactions on Neural Networks and Learning Systems (NNLS)*, Vol. 31, 2019.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks against Machine Learning Models. In *IEEE Symposium on Security and Privacy*, 2017.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated Multi-task Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMLR)*, Vol. 15, 2014.

Aristide Charles Yedia Tossou and Christos Dimitrakakis. Achieving Privacy in the Adversarial

Multi-armed Bandit. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Ardhendu Tripathy, Ye Wang, and Prakash Ishwar. Privacy-Preserving Adversarial Networks. In *IEEE Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, et al. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE Journal on Selected Areas in Communications (JSAC)*, 2019.

Zhenyu Wu, Zhangyang Wang, Zhaowen Wang, and Hailin Jin. Towards Privacy-preserving Visual Recognition via Adversarial training: A pilot study. In *European Conference on Computer Vision (ECCV)*, 2018.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable Invariance through Adversarial Feature Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated Residual Transformations for Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. Diversity-Promoting GAN: A Cross-Entropy Based Generative Adversarial Network for Diversified Text Generation. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

Tsung-Yen Yang, Christopher Brinton, Prateek Mittal, Mung Chiang, and Andrew Lan. Learning Informative and Private Representations via Generative Adversarial Networks. In *IEEE International Conference on Big Data*, 2018.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning Fair Representations. In *International Conference on Machine Learning (ICML)*, 2013.

Supplementary Material: Can we Generalize and Distribute Private Representation Learning?

A Propositions and Proofs

A.1 Proof of Proposition 1

Suppose $\hat{Y}_{A_i} = p_{_{A_i}}(Y|E(\mathcal{X}))$ and $\hat{Y}_{V_j} = p_{_{V_j}}(Y|E(\mathcal{X}))$, where $p_{_{A_i}}(Y|E(\mathcal{X}))$ and $p_{_{V_j}}(Y|E(\mathcal{X}))$ denote the posterior probabilities of successful inference of target labels $Y \sim \mathcal{Y}_{A_i}$ and sensitive labels $Y \sim \mathcal{Y}_{V_j}$ for ally A_i and adversary V_j, respectively, given the outputs encoder E provides for the dataset $\mathcal{X}$. Then, the utilities in (1) can be expressed as

$$u_{_{A_i}} = \mathbb{E}_{Y\sim\mathcal{Y}_{A_i}}\left[\log \hat{Y}_{A_i}\right]; u_{_{V_j}} = \mathbb{E}_{Y\sim\mathcal{Y}_{V_j}}\left[\log \hat{Y}_{V_j}\right], \tag{7}$$

where $1 \leq i \leq n$ and $1 \leq j \leq m$. Let $H_Q = \mathbb{H}(P, Q)$ denote the cross-entropy of Q with respect to P defined as $H_Q = \mathbb{H}(P, Q) = \mathbb{E}_{x\sim P}\left[-\log Q\right]$, then (7) can be re-stated as:

$$\begin{aligned} u_{_{A_i}} &= -H_{A_i} = -\mathbb{H}(Y \sim \mathcal{Y}_{A_i}, \hat{Y}_{A_i}), \quad 1 \leq i \leq n, \\ u_{_{V_j}} &= -H_{V_j} = -\mathbb{H}(Y \sim \mathcal{Y}_{V_j}, \hat{Y}_{V_j}), \quad 1 \leq j \leq m. \end{aligned} \tag{8}$$

The maximization of ally utilities u_{A_i} and minimization of adversary utilities u_{V_j} $\forall i, j$ in the optimization objective (3) can be re-written as minimization of its negative given by,

$$U' = -\sum_{i=1}^{n} \alpha_{A_i} u_{A_i} + \sum_{j=1}^{m} \alpha_{V_j} u_{V_j} = \sum_{i=1}^{n} \alpha_{A_i} H_{A_i} - \sum_{j=1}^{m} \alpha_{V_j} H_{V_j}. \tag{9}$$

Through (9), it can be observed that the minimization occurs when entropy of allies $\sum_{i=1}^{n} \alpha_{A_i} H_{A_i}$ is minimized while that of adversaries $\sum_{j=1}^{m} \alpha_{V_j} H_{V_j}$ is maximized. Using the definition of entropy, each of the allies and adversaries has a global optimum and can be optimized separately if their labels are non-overlapping. Note that ally and adversary entropies are non-negative, and given a fixed encoder E, the sum of ally entropies is minimized when individual entropies are minimized. For each ally, individual entropy H_{A_i} is minimized when $\hat{Y}_{A_i}$ takes the value of 1 $\forall i$ as every ally label is then predicted correctly. Similarly for adversaries, each individual entropy H_{V_j} is maximized when $\hat{Y}_{V_j} = 1/|Y_{V_j}|$ is the uniform distribution. Thus, it can be seen that, at the optimal solution, the adversaries' output follows a uniform distribution, as it minimizes the overall entropy in (9), or equivalently maximizes the utility in (3).

Given that $(A_i, V_j) \in \mathcal{O}$ is the set of all (i, j) pairs of allies A_i and adversaries V_j for which $Y_{A_i} \cap Y_{V_j} \neq \emptyset$, the ally and adversary objectives in (9) are overlapping if $\mathcal{O} \neq \emptyset$. Given that the encoder is fixed, for allies/adversaries not included in $\mathcal{O}$, the associated utilities can be independently optimized. We are thus left with the maximization of the following:

$$U_{\mathcal{O}} = \sum_{(A_i, V_j)\in\mathcal{O}} \alpha_{A_i} \cdot u_{A_i} - \alpha_{V_j} \cdot u_{V_j}. \tag{10}$$

For the kth element in $\mathcal{O}$, $(A_{i(k)}, V_{j(k)})$, we have $Y_{A_{i(k)}}(c) = Y_{V_{j(k)}}(c)$ $\forall c \in \mathcal{C}_k$ $\forall k$;, where $\mathcal{C}_k$ is the set of indices of elements in $Y_{A_{i(k)}} \cap Y_{V_{j(k)}} \neq \emptyset$. Separating the indices c for which the ally/adversary try to predict the same label (i.e. $u_{A_i}(c) = u_{V_j}(c)$), we can express (10) as follows:

$$U_{\mathcal{O}} = \sum_{k}\Bigg(\underbrace{\sum_{c\in\mathcal{C}_k} (\alpha_{A_i} - \alpha_{V_j}) u_{A_i}(c)}_{\text{utility w.r.t. overlapping labels, } U_{\mathcal{O}+}} + \underbrace{\sum_{c\notin\mathcal{C}_k} \alpha_{A_i} u_{A_i}(c) - \alpha_{V_j} u_{V_j}(c)}_{\text{utility w.r.t. non-overlapping labels,} U_{\mathcal{O}-}} \Bigg). \tag{11}$$

The utilities in (11) reward only one of the two discriminators $(A_i, V_j) \in \mathcal{O}$ predicting on overlapping label $c \in \mathcal{C}$ if $\alpha_{A_i} \neq \alpha_{V_j}$. If $\alpha_{A_i} = \alpha_{V_j}$ for $(A_i, V_j) \in \mathcal{O}$, then $U_{\mathcal{O}^+} = 0$, and no optimization occurs w.r.t. the overlapping labels in $Y \sim \mathcal{Y}_{A_i}$.

A.2 Proposition 2

Proposition 2. *Assume that the number of labels of interest is the same among all the allies and adversaries. For any adversary V_j, the distribution of its prediction over its set of labels of interest does not follow a uniform distribution if sufficient weight is given to the ally utilities (i.e., $\alpha_{A_i}, \forall A_i$, is sufficiently large) and the distribution of prediction of one ally A_i, can be defined as a linear combination of the distribution of predictions of V_j and that of other allies/adversaries.*

Proof. Without loss of generality, consider a system with one ally network with a scalar output $\hat{Y}_A$ and m adversary networks with scalar outputs $\hat{Y}_{V_j}$ for $1 \leq j \leq m$. The true distribution of each predicted output is $\mathcal{Y}_A$ for the ally and $\mathcal{Y}_{V_j}$ for the adversaries, and Y_A and Y_{V_j} are the actual labels drawn from those distributions respectively. The true values and predictions between that of the ally and the adversaries have the relation, $Y_A = \sum_{j=1}^{m} w_j Y_{V_j}$, and $\hat{Y}_A = \sum_{j=1}^{m} w_j \hat{Y}_{V_j}$ where w_j is scaling weight. The cross entropy of the entire system is given by $U = \alpha_A Y_A \log(\hat{Y}_A) - \sum_{j=1}^{m} \alpha_{V_j} Y_{V_j} \log(\hat{Y}_{V_j})$. Optimizing for the output of a specific adversary V_n, we obtain:

$$\hat{Y}_{V_n} = \frac{\sum_{j \neq n} w_j \hat{Y}_{V_j}}{\alpha_A Y_A w_n} \left(\frac{1}{\alpha_{V_n} Y_{V_n}} - \frac{1}{\alpha_A Y_A} \right)^{-1}. \tag{12}$$

Notably, $\hat{Y}_{V_n}$ only returns a non-uniform distribution when $\alpha_{V_n} Y_{V_n} < \alpha_A Y_A$. If the weight α_A is not large enough to maintain the inequality, the value of $\hat{Y}_{V_n}$ cannot be obtained via (12) and will have a uniform distribution. If $\alpha_{V_n} Y_{V_n} = \alpha_A Y_A$, then the cross entropy $U = 0$ and no optimization occurs. □

A.3 Proof of Proposition 3

Given that the global encoder is the average of the local encoders in the federated learning procedure for a single synchronization across K local nodes, the maximization of the expectation in (6) can be described as the maximization of ally utilities and minimization of adversary utilities given by:

$$U = \frac{1}{K} \sum_{k=1}^{K} \left(\sum_{i=1}^{n_{(k)}} \alpha_{A_i^{(k)}} u_{A_i^{(k)}} - \sum_{j=1}^{m_{(k)}} \alpha_{V_j^{(k)}} u_{V_j^{(k)}} \right). \tag{13}$$

In (13), $A_i^{(k)}$ and $V_i^{(k)}$ refer to the i^{th} ally or adversary of the k^{th} local node. Since data at each node is i.i.d, the distributions $\mathcal{Y}$ are the same at each node, and thus each node has the same objective function. Using the result of Prop. 1 and assuming that $A_i^{(k_1)}, V_j^{(k_1)} = A_i^{(k_2)}, V_j^{(k_2)}$ $\forall i, j, k_1, k_2$ (i.e., the ally and adversary labels are same across all nodes), the output of the adversaries at each node follow a uniform distribution.

The ally and adversary objectives in (13) are overlapping if $\mathcal{O} \neq \emptyset$ given that $(A_i, V_j) \in \mathcal{O}$ is the set of all A_i, V_j pairs for which $Y_{A_i} = Y_{V_j}$. Since each of the local nodes have the same overlapping ally/adversary labels with potentially different weights $\alpha_{A_i^{(k)}}$ and $\alpha_{V_j^{(k)}}$, their utilities can be expressed using entropy as in (8). The final optimization of the distributed system can be expressed as the minimization of following:

$$U_{\mathcal{O}} = \sum_{(A_i, V_j) \in \mathcal{O}} \left(\sum_{k=1}^{K} (\alpha_{A_i^{(k)}} - \alpha_{V_j^{(k)}}) \cdot u_{A_i^k} \right). \tag{14}$$

The entropy values given in (14) reward only one of the two discriminators predicting label Y_{A_i} if $\sum_{k=1}^{K} \alpha_{A_i^{(k)}} \neq \sum_{k=1}^{k} \alpha_{V_i^{(k)}}$. If $\sum_{k=1}^{K} \alpha_{A_i^{(k)}} = \sum_{k=1}^{K} \alpha_{V_i^{(k)}}$, these two networks have no contribution to $U_{\mathcal{O}}$, and no optimization occurs..

A.4 Proposition 4

Proposition 4. *If the allies and adversaries located at the K nodes of D-EIGAN have non-overlapping target sets, i.e., $Y_{A^{(k)}} \neq Y_{A^{(k')}}$ and $Y_{V^{(k)}} \neq Y_{V^{(k')}}$, $1 \leq k, k' \leq K$, then individual encoders under D-EIGAN consider the union of these local allies, $\bigcup_{k=1}^{K} Y_{A^{(k)}}$, and adversaries ,$\bigcup_{k=1}^{K} Y_{V^{(k)}}$ for optimization as a result of the global aggregation step. The weights $\alpha_{A_i^{(k)}}$ and $\alpha_{V_i^{(k)}}$ associated with the allies/adversaries are scaled by the ratio of the number of nodes that implement them locally to the total number of nodes.*

Proof. Without loss of generality, consider a two network D-EIGAN. Let node 1 have 2 allies and 1 adversary with objectives: Y_{A_c}, Y_{A_1}, and Y_{V_1}, and node 2 have 2 allies and 1 adversary with objectives: Y_{A_c}, Y_{A_2} and Y_{V_2}. Here, objective Y_{A_c} is common among them, while the rest are different. Utilities of individual nodes can be calculated using (3):

$$U^{(1)} = \alpha_{A_c} \cdot u_{A_c} + \alpha_{A_1} \cdot u_{A_1} - \alpha_{V_1} \cdot u_{V_1}, \tag{15}$$

$$U^{(2)} = \alpha_{A_c} \cdot u_{A_c} + \alpha_{A_2} \cdot u_{A_2} - \alpha_{V_2} \cdot u_{V_2}. \tag{16}$$

Under federated training, the equivalent loss function that is optimized by the D-EIGAN can be calculated using (6):

$$U = \alpha_{A_c} \cdot u_{A_c} + \frac{\alpha_{A_1}}{2} \cdot u_{A_1} - \frac{\alpha_{V_1}}{2} \cdot u_{V_1} + \frac{\alpha_{A_2}}{2} \cdot u_{A_2} - \frac{\alpha_{V_2}}{2} \cdot u_{V_2}, \tag{17}$$

which shows that the overall objective under D-EIGAN considers all the objectives, but the associated weights are lower for non-common allies/adversaries. In contrast to a D-EIGAN where all allies and adversaries are common across nodes, the difference is the weights associated with objectives. □

B Pseudocode of EIGAN

In this section, Algorithm A1 presents the step-by-step implementation of centralized EIGAN training. It is an iterative mini-batch stochastic gradient descent procedure in which we update the weights for the encoder and allies/adversaries alternately until convergence. The learning rates for the encoder, allies and adversaries are controlled using parameters η_E, η_A, and η_V.

Algorithm A1 EIGAN training

```
Notations:
$(\cdot)_j$ denotes the value for the $j$th minibatch
$\mathcal{L}_{A_i}$ denotes the loss of ally $A_i$
$\mathcal{L}_{V_i}$ denotes the loss of the adversary $V_i$
$\eta_E, \eta_A, \eta_V$: learning rates of the encoders, allies and adversaries
Training:
initialize $\alpha$ used in loss function (4)
initialize $\theta_{A_i}$'s and $\theta_{V_j}$'s and $\theta_E$ to start the training
for number of training epochs do
    Sample a minibatch set $J$ of data points
    Compute encoder loss using (4): $\mathcal{L}_E = \frac{1}{|J|} \sum_{j \in J} (\mathcal{L}_E)_j$
    Update encoder parameters: $\theta_E \leftarrow \theta_E - \eta_E \cdot \nabla_{\theta_E} \mathcal{L}_E$
    Compute allies/adversaries losses using (4):
        $\mathcal{L}_{A_i} = -\frac{1}{|J|} \sum_{j \in J} (\mathcal{L}_{A_i})_j, \quad \mathcal{L}_{V_i} = -\frac{1}{|J|} \sum_{j \in J} (\mathcal{L}_{V_i})_j$
    Update local allies/adversaries parameters:
        $\theta_{A_i} \leftarrow \theta_{A_i} - \eta_A \cdot \nabla_{\theta_{A_i}} \mathcal{L}_{A_i}, \quad \theta_{V_i} \leftarrow \theta_{V_i} - \eta_V \cdot \nabla_{\theta_{V_i}} \mathcal{L}_{V_i}$
end for
```

C Additional Proof of Concept Visualizations

In this section, we include additional proof of concept visualizations beyond those presented in Fig. 2 from Sec. 2.1.

The first experiment uses a synthetic dataset comprising 4 sets of Gaussian distributed points in 2-D around the means (-0.5, -0.5), (-0.5, 1.5), (1.5, -1.5) and (1.5, 1.5) as shown in Fig. 10(a). We implement EIGAN with the ally objective to distinguish between reds and blues and adversary objective to segregate x's and o's. This is the simplest case we consider, as there is a single ally and single adversary, each with binary labels. Decision boundaries are linear. We thus use a logistic regression classifier as it has a convex loss function. The encoder is a neural network with a single hidden layer and output dimension $l = 2$. The learnt representation in Fig. 10(b) is intuitive: it maintains linear separability among ally classes, i.e., reds vs blues, but ensures a collapse of adversary classes.

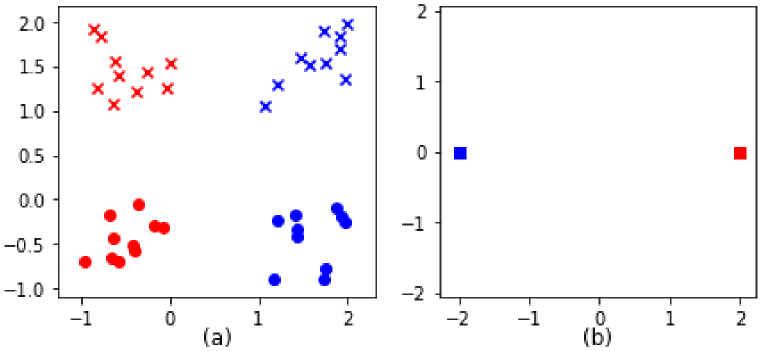


Figure 10: (a) Quadrant dataset with four groups of points, one ally, and one adversary. The points are linearly separable with regard to the ally's (classifying reds/blues) and an adversary's (classifying x's/o's) objectives. (b) EIGAN learns a representation that collapses the axes along the adversary's objective while enhancing separation along the ally's.

Next we consider a dataset with non-linear decision boundary as shown in Fig. 11 (a). The ally is interested in a decision boundary between the red and the blue circle, while the adversary is interested in the upper vs. lower semicircle, i.e., x's vs o's. The same encoder is used as in the previous experiment. We use a neural network with a single hidden layer as the ally and adversary because the ally's decision boundary is not linearly separable. Fig. 11(b) shows the learnt representation, which achieves a separability in the encoded space that is qualitatively similar to the representation learnt in Fig. 10(b).

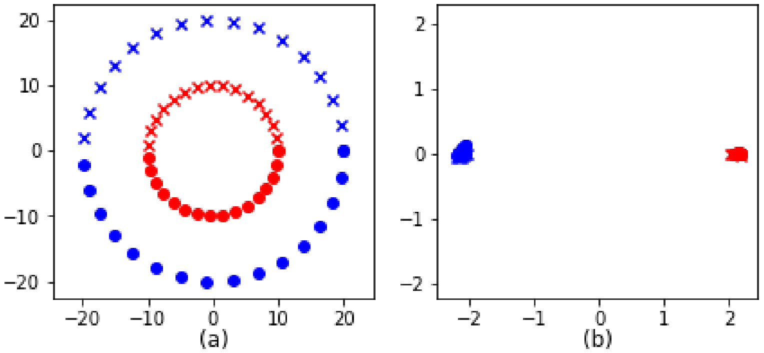


Figure 11: (a) Circle dataset with the same objectives as Figure 10 but ally classes (reds vs blues) are not linearly separable. (b) EIGAN learns a similar transformation, making the ally's classification task linearly separable.

For the final experiment, we extend EIGAN from single ally and single adversary to multiple allies and adversaries. We consider two different cases: EIGAN with (i) 2 ally and 1 adversary objective, and (ii) 1 ally and 2 adversaries presented in Fig. 12. Case (i) is what was presented in Fig. 2 of Sec. 2.1. In the case (ii), we have 8 set of Gaussian distributed points, one in each octant as shown in Fig. 12(a). The ally wants to separate reds vs blues, and the adversaries want to separate along the other axes, i.e., top vs bottom (adversary 1) and squares and stars vs x's

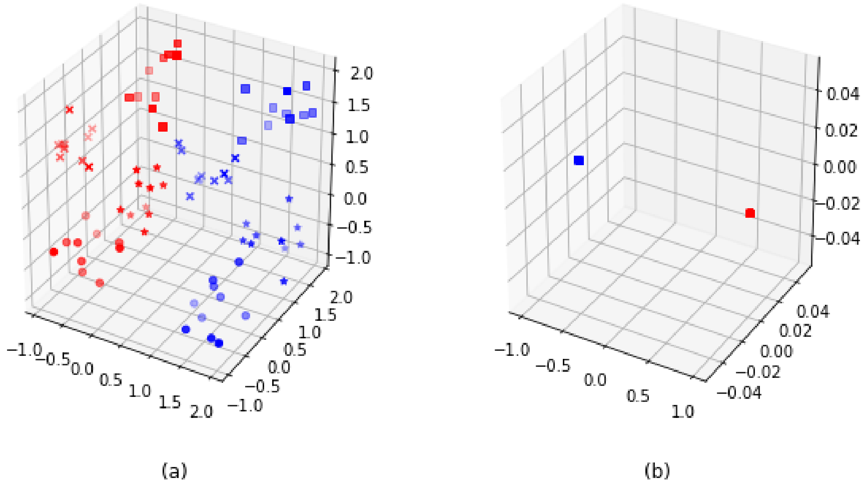


Figure 12: (a) Octant dataset with eight groups of points, one ally, and two adversaries. The ally is interested in classifying reds/blues while the adversaries are interested in separation along other axes. (b) EIGAN collapses the two adversary dimensions while maintaining separability for the ally.

and o's (adversary 2). The learnt representation only preserves ally's dimension of variation, i.e. reds vs blues. All the other dimensions are collapsed.

D Additional EIGAN Experiments

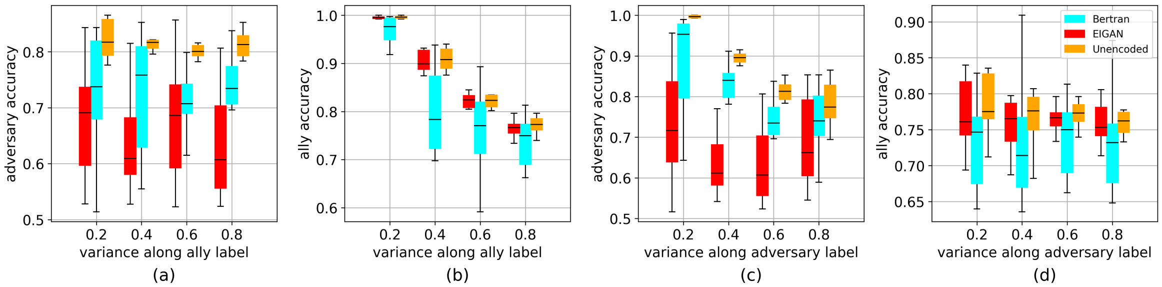


Figure 13: Effect of change in ally overlap (a-b) and adversary overlap (c-d) on the performance of EIGAN, and its comparison with the unencoded data as well as the method in Bertran et al. Bertran et al. (2019). EIGAN is able to consistently outperform both baselines on the adversary objective, and obtains performance close to the unencoded data for the ally.

D.1 Comparison with the method in Bertran et al. (2019)

Fig. 5 from Sec 3.1 presented a comparison of Bertran et al. (2019) with EIGAN on synthetic Gaussian data. Fig. 13 is the extended version, presenting additionally the comparison of (c) adversary and (d) ally performance as we alter the class overlap between adversary labels. Consistent with the conclusions presented in Sec 3.1 for the ally variation, EIGAN outperforms Bertran et al. consistently as the adversary exhibits more variance. The p-values of the improvements EIGAN makes over the method in Bertran et al. (2019) are below 0.002 in all 16 cases of comparisons between boxplot distributions.

D.2 Comparison on MIMIC-III

Here we present the extended results of Fig. 3 from Sec. 3.1 which compared EIGAN against baselines on the MIMIC dataset. Fig. 14(c) shows the loss progression of encoder and adversary as the EIGAN training proceeds. It can be observed that increase/decrease in encoder loss is corresponding to the decrease/increase in adversary loss during the same epoch, consistent with the definition of the encoder loss in (4). The magnitude of the

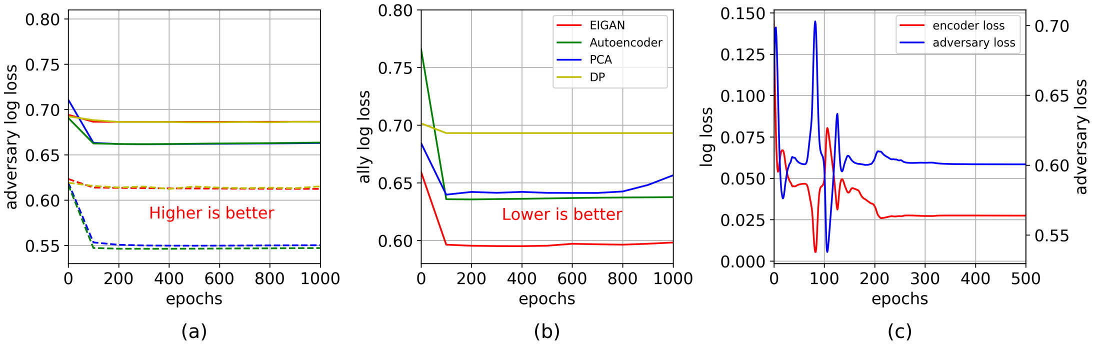


Figure 14: Predictivity and privacy comparison between EIGAN and the baselines across one ally and two adversaries on the MIMIC-III dataset. (a) On the adversary objectives (gender prediction, solid lines and race prediction, dashed lines) EIGAN matches DP's performance (by design of the experiment, as determined by the selection of the DP ϵ parameter). Hence, the red and the khaki colored curves overlap. (b) On the ally objective (survival prediction), EIGAN achieves noticeable improvement over the baselines. (c) EIGAN training converges after initial oscillations corresponding to the minimax game.

oscillations decreases as we progress through the training and eventually the networks (i.e., the players in the game) reach a steady state.

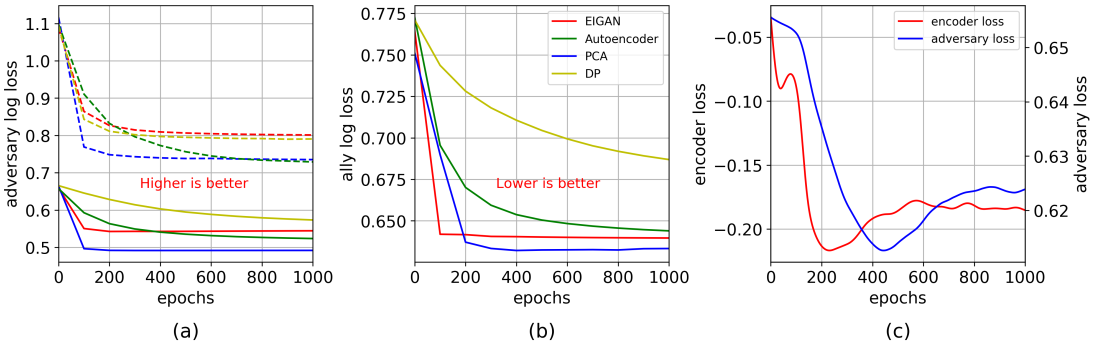


Figure 15: Predictivity and privacy comparison between EIGAN and the baselines across one ally and two adversaries on the Titanic dataset. (a) On one of the adversary objectives (gender prediction, solid lines) EIGAN matches DP's performance (by design of the experiment, as determined by the selection of the DP ϵ parameter), but in this case it does not match the other adversary prediction (passenger class prediction, dashed lines), which could be matched for another value of ϵ. (b) On the ally objective (survival prediction), EIGAN achieves marginal improvement over the the baseline Autoencoder. (c) EIGAN training converges after initial oscillations corresponding to the minimax game.

D.3 Comparison on Titanic dataset

For completeness, we also evaluate EIGAN algorithm on another dataset, Titanic, which consists of data listing the details of roughly 800 of the passengers that were onboard the Titanic ship. This experiment aims at understanding the convergence behaviour of EIGAN under limited training data.

Algorithm	**Ally** (Survival)	**Adversary 1** (Gender)	**Adversary 2** (P-Class)
Autoencoder	**0.6333**	0.4918	0.7351
PCA	0.6439	0.5236	0.7289
DP	0.6869	**0.5733**	0.7904
EIGAN	**0.6396**	0.5444	**0.8011**

Table 3: Comparison of log-loss achieved on the test set between the algorithms for the Titanic dataset. EIGAN matches autoencoder on the ally and performs slightly better than DP on adversary 2, while slightly worse on adversary 1.

Similar to result on MIMIC-III from Fig. 14, Fig. 15 (a) shows that while EIGAN is able to perform as well or nearly as well as any of the baselines on adversary obfuscation, (b) it obtains the best predictivity on ally objective. (c) shows that the training reaches a steady-state.

Table 3 summarizes the loss-values of the trained allies/adversaries on encoded data using different techniques. It can be seen that while EIGAN is able to match DP's performance on adversary 2, it performs marginally worse than it on adversary 1, while having a considerable gain on the corresponding ally.

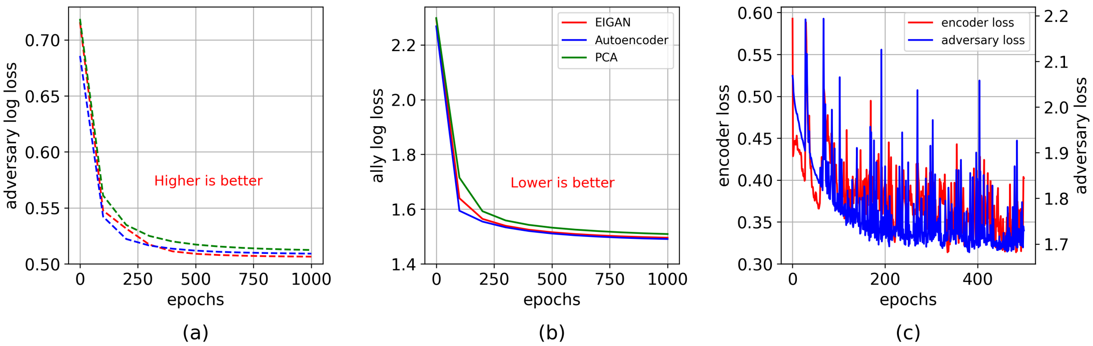


Figure 16: Comparison across one ally and two adversaries on the MNIST dataset. The (a) adversary objective (odd-even prediction, a binary classification with virtually identical trends) converge to roughly the same loss for each algorithm, and (b) ally objective (digit prediction, 10-class classification). With dependencies (in particular, partial overlaps) between the ally and adversary objectives, EIGAN training in (c) is unable to fully converge, consistent with Prop. 2.

D.4 Comparison on MNIST

We conduct an additional experiment on the MNIST dataset of handwritten digits to validate the findings in Prop. 1&2 when dependencies exist between the ally and adversary objectives. In this case, we use digit recognition (0-9) as the ally objective and even vs odd as adversary objective, which exhibits a clear dependence because if someone could recover the digit (ally objective), then inferring odd-vs-even (adversary objective) becomes trivial. Formally, referring to the propositions, we have $\mathcal{Y}_{\text{odd}} = \mathcal{Y}_1 + \mathcal{Y}_3 + \cdots + \mathcal{Y}_9$ and $\mathcal{Y}_{\text{even}} = 1 - \mathcal{Y}_{\text{odd}}$ where $\mathcal{Y}_{(\cdot)}$ is the true probability distribution on the labels and thus can be added. Similarly, $\hat{Y}_{\text{odd}} = \hat{Y}_1 + \hat{Y}_3 + \cdots + \hat{Y}_9$ and $\hat{Y}_{\text{even}} = 1 - \hat{Y}_{\text{odd}}$, where $\hat{Y}_{(\cdot)}$ are probabilities of correct predictions. Prop. 2 follows when we substitute these in (12), i.e. the adversary is not forced to a follow uniform distribution if sufficient weight is given to the ally.

Fig. 16 shows the result of this experiment, where the weights of the allies and adversaries are set equal. (a) shows that the adversary is not able to achieve any separation from the Autoencoder or PCA. Observing (c), we realize that the training process does not reach a steady state-convergence point, consistent with the propositions.

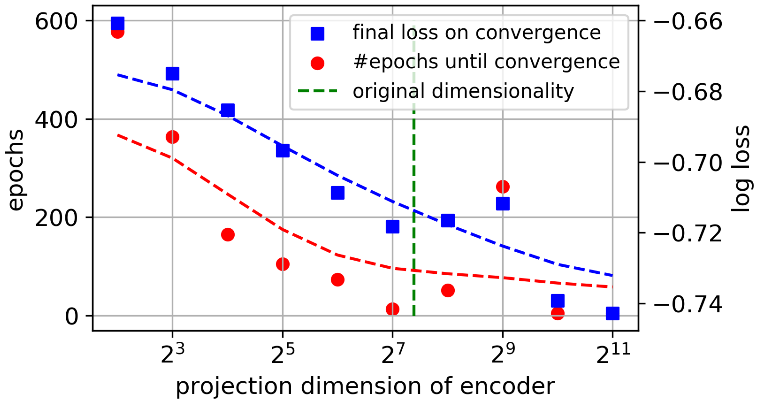


Figure 17: Effect of EIGAN's encoding dimension space on the number of training epochs required to reach within 1% of training loss convergence (left axis) and the achieved final testing loss (right axis) for MIMIC-III. The achieved loss decreases sharply as the dimension increases, emphasizing a tradeoff between model quality and the memory needed for the encoded data. In fact, beyond the right end of the X-axis value, the model runs out of memory on our high performance machine. (Dashed curves are fit using weighted moving averages.)

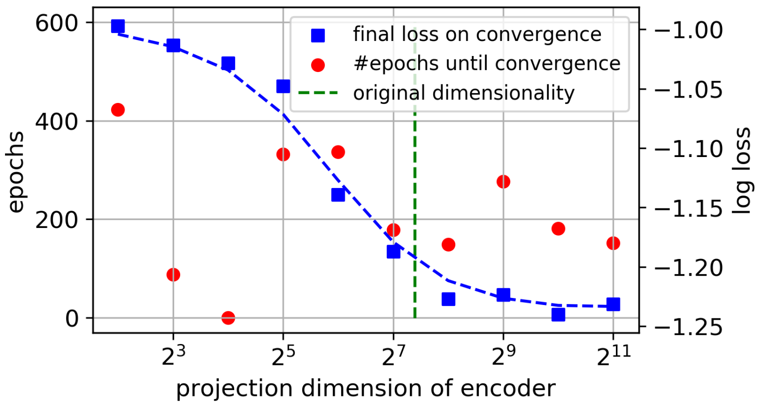


Figure 18: Effect of EIGAN's encoding dimension space on the number of training epochs required to reach within 1% of training loss convergence (left axis) and the achieved final testing loss (right axis) for the Titanic dataset. The achieved loss decreases sharply as the dimension increases, emphasizing a tradeoff between model quality and required memory. (The dashed curve is fit using a weighted moving average.)

D.5 Varying Encoder Dimensionality

Under *varying system dimensions*, in Sec. 3.1, we also discussed the effect of varying the encoder dimensionality. Fig. 17 depicts the results for the MIMIC-III dataset while Fig. 18 depicts the result of a similar experiment on Titanic dataset. In the two experiments, as the encoder output dimension l is increased, we observe that the training mostly requires fewer epochs to converge and is able to achieve a lower encoder testing loss. This could be explained by the fact that larger networks (i.e. more number of trainable parameters) have more degrees of freedom in training. Interestingly, while there is some variation, the test loss continues to decrease beyond d, the original dimension of the data samples, i.e., when $l \geq d$. The relevant consideration with EIGAN, then, appears to be the tradeoff between encoding quality, as measured by the encoding space dimension, and the memory required for training the encoder, which increases with the dimension of the encoder.

E Additional D-EIGAN Experiments

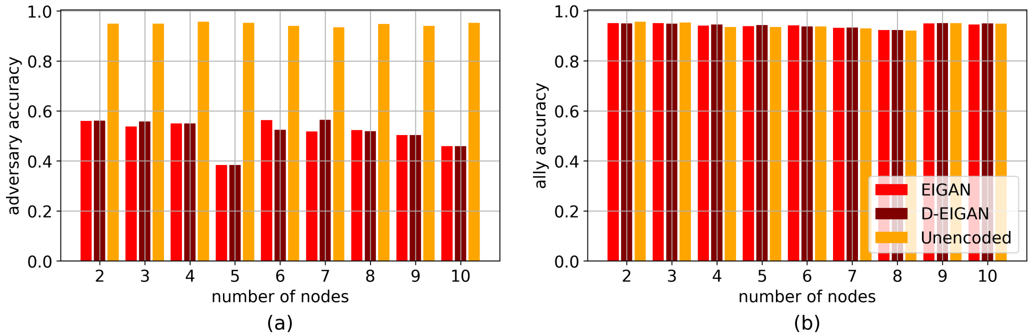


Figure 19: Comparison of (a) adversary and (b) ally performance using synthetic Gaussian data while increasing the number of nodes and sharing all the model weights ($\phi = 1$) after every minibatch ($\delta = 1$) during federated training. The distribution of data is i.i.d. across the nodes, which is obtained by generating Gaussian data with constant mean and variance across nodes. It can be observed that EIGAN and D-EIGAN converge to similar performances regardless of the number of nodes.

E.1 Varying Number of Nodes

Fig. 7 from Sec. 3.2 presented the effect of varying the number of nodes on D-EIGAN performance when the nodes have non-i.i.d data distributions. Fig. 19 shows the result of the experiment when the nodes instead have i.i.d data. We observe that the performance of the ally and adversary remains reasonably constant (and similar to EIGAN) as we increase the number of nodes under D-EIGAN. From the two experiments, we can conclude that D-EIGAN can readily extend to scenarios where data is distributed over larger number of nodes without sacrificing the performance on ally and adversary objectives.

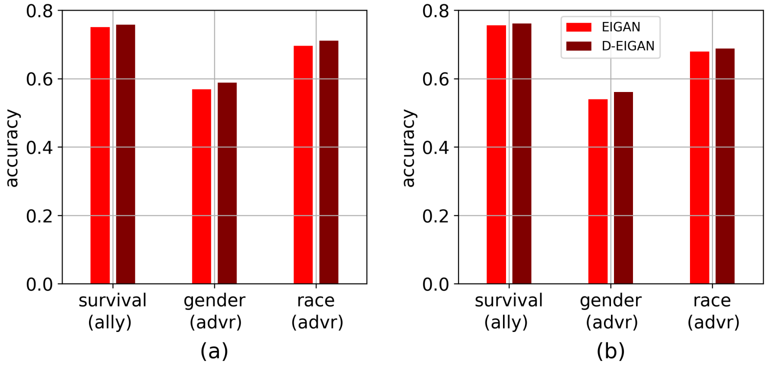


Figure 20: Comparison of distributed ($K = 2$ nodes) EIGAN with centralized EIGAN. Survival is the ally objective, and gender and race are the chosen adversary objectives for the experiment. (a) Training of distributed EIGAN involves same adversary objectives, i.e., obfuscating gender and race across the both the nodes. (b) Each node has a different adversary objective, while they share the same ally objective.

E.2 Varying Objectives across Nodes

Fig. 20 is an addition to results presented in Fig. 8 from Sec. 3.2. In this experiment we consider the same dataset with the same set of ally and adversary objectives, but in this case over i.i.d datasets on 2 nodes instead of non-i.i.d datasets on 10 nodes. Survival is the ally objective while, gender and race are adversary objectives. We consider two scenarios: in (a) gender and race are common adversaries across the two nodes, while in (b) gender is the adversary on one node and race is the adversary on the other node (i.e. different objectives on different nodes). We observe the performance of D-EIGAN is comparable to that of EIGAN in the case where the data is centralized. This observed behavior, i.e., that a privacy and/or predictivity objective at one node is adopted across all the encoders, is consistent with Prop. 4.

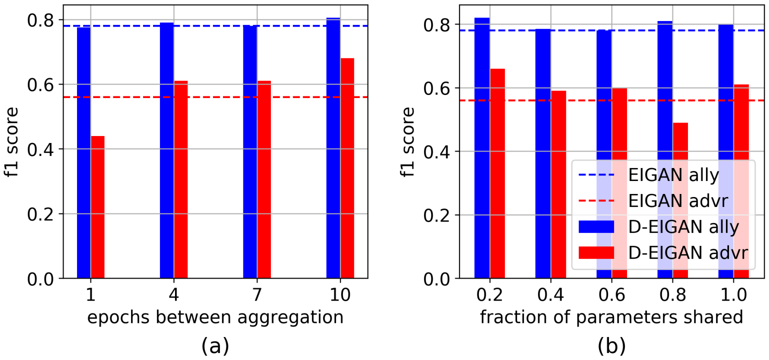


Figure 21: Effect of varying (a) frequency of sync (δ, measured in terms of number of epochs between parameter sharing) and (b) fraction of parameters uploaded/downloaded (ϕ) on a distributed implementation consisting of $K = 2$ nodes. The results shows that as the frequency of sync/fraction of parameters shared increases, the performance of the system on hiding the sensitive variable is increased considerably, while there is little effect on the ally convergence.

E.3 Varying Synchronization parameters

This section extends the results discussed in Sec. 3.2 under *varying synchronization parameters*. To understand the effect of fractional parameter sharing, we evaluate it on i.i.d. datasets over 2 nodes using the synthetic Gaussian dataset in Sec. 3.2. The data has unbalanced classes, so we compare f1-score instead of accuracy. The result is shown in Fig. 21: We see that unlike the trend on the non-i.i.d. case, there is no visible benefit of sharing only a fraction of parameters, as seen in Fig. 21(b). Similarly, in (a) it can be observed that performance over the adversary degrades as the frequency of sync is decreased, i.e., number of epochs between aggregation in increased. Hence, the properties observed upon having non-i.i.d data distributions in Sec. 3.2 do not hold when handling i.i.d data. This is because the reduction in model bias is not desirable in the case of i.i.d.