A Generalized and Distributable Generative Model for Private Representation Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of learning data representations that are private yet informative, i.e., providing information about intended "ally" targets while obfuscating sensitive "adversary" attributes. We propose a novel framework, Exclusion-Inclusion Generative Adversarial Network (EIGAN), that generalizes adversarial private representation learning (PRL) approaches to generate data encodings that account for multiple (possibly overlapping) ally and adversary targets. Preserving privacy is even more difficult when the data is collected across multiple distributed nodes, which for privacy reasons may not wish to share their data even for PRL training. Thus, learning such data representations at each node in a distributed manner (i.e., without transmitting source data) is of particular importance. This motivates us to develop D-EIGAN, the first distributed PRL method, based on fractional parameter sharing that promotes differentially private parameter sharing and also accounts for communication resource limitations. We theoretically analyze the behavior of adversaries under the optimal EIGAN and D-EIGAN encoders and consider the impact of dependencies among ally and adversary tasks on the encoder performance. Our experiments on real-world and synthetic datasets demonstrate the advantages of EIGAN encodings in terms of accuracy, robustness, and scalability; in particular, we show that EIGAN outperforms the previous state-of-the-art by a significant accuracy margin (47% improvement). The experiments further reveal that D-EIGAN's performance is consistent with EIGAN under different node data distributions and is resilient to communication constraints.

1 Introduction

Training machine learning (ML) models frequently requires sharing data among multiple parties, e.g., cloud services aggregating data from multiple users to learn a global model. Such sharing naturally raises privacy concerns in terms of exposing sensitive attributes in datasets. Even when data is anonymized at the source, de-anonymization attacks are possible when coupled with auxiliary datasets, as was famously shown in the Netflix challenge [1].

A widely used technique for obfuscating sensitive attributes in data is differential privacy (DP) [2], a context-agnostic technique. DP introduces noise (e.g., Laplace) into a dataset to provide membership security. However, noise injection can impact ML training/inference significantly. This makes such context-agnostic privacy techniques unsuitable in scenarios where only a few attributes need to be obfuscated. For example, upon sharing patient vitals for preventive healthcare [3], both privacy (e.g., gender, ethnicity anonymization) and predictivity (e.g., accurate diagnosis) are important.

These drawbacks of context-agnostic privacy measures motivate private representation learning (PRL) [4], which exploits existing 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".

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Conventionally, the literature on PRL assumes the existence of a single sensitive attribute and a central dataset [4, 5, 6, 7]. 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, and ethnicity that prevent sharing of source data, and (iii) governed by regulations that vary from one region to another, e.g., while GDPR (in Europe) considers racial/ethnic origin as sensitive information, HIPAA (in USA) does not. 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 [8], e.g., remote health analytics across user devices at the network edge.

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 game-theoretic 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.

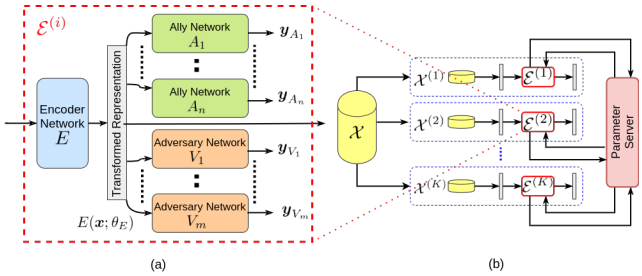

Figure 1: (a) Architecture of a single EIGAN node, consisting of an encoder, n ally, and m adversary networks. (b) D-EIGAN system for distributed EIGAN training, consisting of K different EIGAN nodes, each with their own subset of the full dataset. The nodes must coordinate their local encodings via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

Related work. Recent works in PRL [4, 5, 6] have proposed centralized architectures that jointly maximize the loss in predicting sensitive attributes while minimizing the loss of target task prediction. Specifically, [4] 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 [5], the problem was formulated as a non-zero-sum game between the three networks to minimize information leakage in encoded image representations. [6] experimentally outperform [5, 9, 10] 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. EIGAN also has computational advantages over [6] as it does not depend on matrix inversions.

Other PRL works take an information-theoretic approach. [7] 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, [11] formulate the minimax problem in terms of KL-divergence. EIGAN, on the other hand, considers a log-loss PRL formulation, which promotes interpretability and training stability over multiple objectives (discussed in Sec. 2.1). Furthermore, our experiments show that EIGAN significantly outperforms the state-of-the-art [7] in the single ally/adversary case. *Distinct from all prior work in PRL, we consider multiple sensitive attributes and distributed learning.*

It is worth mentioning two other directions in adversarial learning related to PRL. One addresses privacy-preservation through synthetic data generation [12], which differs from EIGAN's goal of learning a data transformation. The other is fair representation learning [13, 14], which seeks representations that are intrinsically fair in the sense of promoting demographic parity on a single attribute [15].

Contributions. Our contributions are summarized below:

1. We introduce EIGAN (Sec. 2.1), generalizing PRL to account for multiple potentially overlapping 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 (Sec. 2.2) is the first technique for distributed training of PRL models. We show that when the nodes engaged in the training possess i.i.d. datasets, the objective of D-EIGAN exhibits similar properties to EIGAN (Prop. 3).
3. Our experiments (Sec. 3) reveal that EIGAN significantly outperforms the state-of-the-art in PRL (Tab. 1, Fig. 4) and is robust to the choice of adversary architectures (Tab. 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 Formulation and Model Learning

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 Sec. 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 Sec. 2.2. Refer to App. A for the proofs of the propositions.

2.1 EIGAN: Centralized Model Architecture

We first consider a system consisting of n allies, indexed by $A_1, ..., A_n$; and m adversaries, indexed by $V_1, ..., V_m$. Ally A_i is characterized by model parameters θ_{A_i} and a set of target attributes/labels Y_{A_i} following 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. $E(\boldsymbol{x}; \theta_E) : \mathbb{R}^d \to \mathbb{R}^l$ 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 a sample $\boldsymbol{x} \in \mathcal{X}$, 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 utility functions of the allies and adversaries as:

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

where $p_{A_i}\left(Y|E(\mathcal{X};\theta_E)\right)$ and $p_{V_j}\left(Y|E(\mathcal{X};\theta_E)\right)$ 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

	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, [6] (Linear-ARL, Kernel-ARL), and [7] (Bertran-PRL) on the Adult & FaceScrub datasets considered in those works. For the same adv. performance, EIGAN obtains a notable improvement over [7] (ally improvement of 47.01%). It also reaches the optimal closed form solution of [6].

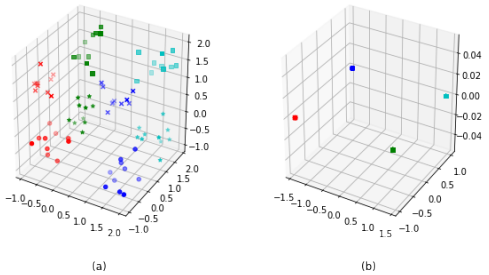

Figure 2: (a) Synthetic dataset with eight groups of points, two allies, and one adversary. The allies are interested in separating the color pairs (the two horizontal axes), and the adversary is interested in classifying shapes (the vertical axis). (b) EIGAN's encoding has collapsed the adversary dimension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

adversary). Specifically, we formulate the optimization problem:

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

where

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

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 [7], our analysis on the expected posterior distribution of the predictions in EIGAN map directly to interpretable metrics such as accuracy [16] and generalization error [17], 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. 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. This resembles the initial GAN result in [18] obtained for a two-network architecture. In App. D.3, 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 App. 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 log-loss functions. For the encoder, we define the log-loss $\mathcal{L}_E$ for a single training instance as a weighted combination of the predictive capability of the allies and adversaries as

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

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). For the full algorithm, see Alg. A1 in App. B.

Alternative objectives to (4) can be found in PRL literature. In particular, recent works [19, 11, 7] formulate the adversarial loss using KL divergence. Our choice of log-loss over KL-divergence is based on the facts that: (i) KL divergence fails to give meaningful value under disjoint distributions [20], and thus is less interpretable, and (ii) log-loss is more stable under varying priors during training, which is important in the EIGAN scenario since training data is affected by mini-batches, non-i.i.d distributions, and the evolving representations. Our experimental results in Tab. 1 validate our choice of formulation in terms of obtaining improvements over the state-of-the-art that uses KL divergence (discussed further in Sec. 3.1).

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

2.2 D-EIGAN: Distributed Adversarial Learning

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

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

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:

Algorithm 1 D-EIGAN training

Notation:

θ_E denotes the global parameter vector

$\mathcal{Q}_k$ denotes uniformly random choice of indices at node $\mathcal{E}^{(k)}$

$\tilde{\theta}_{E^{(k)}}$ denotes the parameter vector recovered at the server for encoder $E^{(k)}$, with its qth element denoted $\tilde{\theta}_{E^{(k)}}(q)$.

ϕ denotes the fraction of parameters shared

δ denotes the number of epochs between aggregations

$(\cdot)_j$ denotes the value for the jth minibatch

$\mathcal{L}_{A_i^{(k)}}$ and $\mathcal{L}_{V_i^{(k)}}$ denote the loss of ally $A_i^{(k)}$ and adversary $V_i^{(k)}$ at node $\mathcal{E}^{(k)}$

η_E, η_A, and η_V denote the learning rates

Aggregation at Parameter Server:

Initialize parameter θ_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 θ_E from parameter server

for number of training epochs **do**

  After δ 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 of datapoints 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 δ epochs, upload $\phi \cdot |\theta_{E^{(k)}}|$ 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

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

where $\mathcal{S}_V = \{\theta_{V^{(k)}}\}_{k=1}^{K}, \mathcal{S}_E = \{\theta_{E^{(k)}}\}_{k=1}^{K}, \mathcal{S}_A = \{\theta_{A^{(k)}}\}_{k=1}^{K}$. 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, we learn both a global model (encoder) and individual (personalized) local models (allies and adversaries) [21], 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 log-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

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.

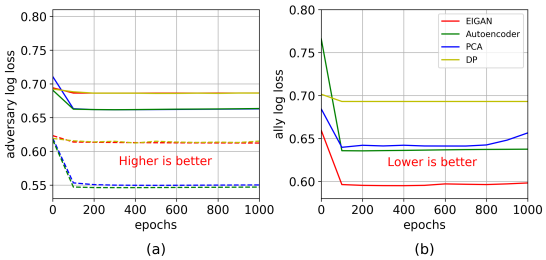

Figure 3: Predictivity and privacy comparison between EIGAN and baselines on MIMIC. (a) On the adversary prediction (gender, solid lines and race, dashed lines), EIGAN matches DP's performance (by tuning DP's noise). (b) On the ally prediction (survival), EIGAN achieves noticeable improvement over the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

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, uplink 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 [22]. Fractional parameter sharing, similar to pruning (both choose a subset of parameters), mimics the additive-noise DP mechanism [23] on model weights, reducing associated leakage [24] to any untrusted entity with access to the system. We study the effect of δ and ϕ on D-EIGAN performance in Sec. 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 can show that the importance of each objective is proportional to the number of nodes implementing it; see Prop. 4 in App. A.4.

3 Experimental Evaluation and 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 Sec. 3.1, and evaluate D-EIGAN compared to the centralized case and as the system characteristics change in Sec. 3.2.

Datasets. We consider four datasets: MNIST [25], MIMIC-III [26], Adult [27], and FaceScrub [28]. MNIST consists of 60,000 handwritten digits with labels 0-9. MIMIC has medical information for hospital admissions with attributes, such as patient vitals and medication; we obtain a dataset consisting of 58,976 admission data points 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.

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 [7], 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 [6], the ally objective is an annual income classification (more or less than 50K) and

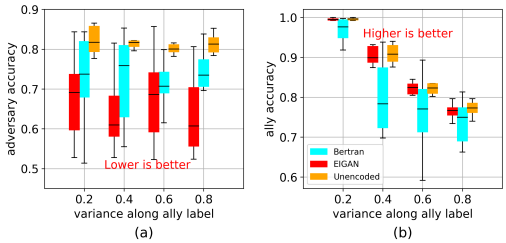

Figure 4: Effect of varying the ally class overlap (by changing the variances of synthetic Gaussian data) on the performance of EIGAN, [7], and the unencoded data. (a) and (b) plot the achieved accuracies of the adv. and ally objectives, respectively. EIGAN is able to consistently outperform both baselines on the adversary objective, and obtains performance close to the unencoded data for the ally.

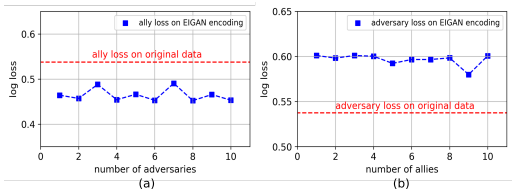

Figure 5: EIGAN's effect of the number of (a) adversaries, and (b) allies on the testing loss for MIMIC-III. The ally/adversary objectives are chosen as different attributes from the source. The achievable loss is reasonably constant and is not affected by addition of addition of more allies/adversaries.

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 employ fully connected networks (FCNs) for the encoder, allies, and adversaries in the experiments on MIMIC and the synthetic datasets. The FCN encoder uses ReLU 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 and L2-regularization to prevent network overfitting. For FaceScrub, we employ U-Net [29] for the encoder and Xception-Net [30] for the ally/adversary, as in [7]. For Adult, we employ linear FCN as in [6]. Unless otherwise stated, we set $\alpha = 0.5$ (i.e., equal privacy/predictivity importance). We train to minimize cross-entropy loss over 70/30 training/test splits on a system with 8GB GPU memory and 64 GB RAM.

Baselines. We consider six baselines: principal component analysis (PCA), autoencoders [31], differential privacy (DP), and the methods in [6], [7]. 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 Sec. 1, DP is widely used for context-agnostic privacy. For DP, we employ the Laplace mechanism [4]. [7] 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 [6] for linear maps on their Adult dataset use case.

All of our code and trained models are available at `https://github.com/aithiest/EIGAN`. For each experiments, we report log-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 [6, 7] 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 [6] and [7] on the Adult and Facescrub dataset settings considered in these works, respectively. Note that the linearity requirement in [6] impedes its usage on non-linear models like the U-Net and Xception-Net employed for Facescrub by [7]. For comparison, we adjust α in (4) to equalize the resulting adversary performances between the models. Tab. 1 gives the results: EIGAN matches the performance of [6]'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 [7].

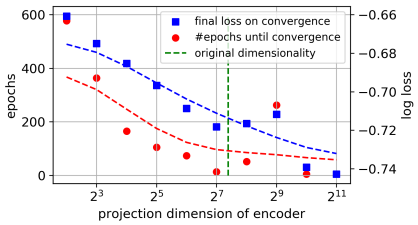

Figure 6: Effect of EIGAN's encoding dimension on the number of training epochs required to reach within 1% of training loss convergence (left axis) and the achieved final testing loss (right axis) for MIMIC-III. The achieved loss decreases as the dimension increases, emphasizing a tradeoff between model quality and the memory needed (dashed curves show moving average).

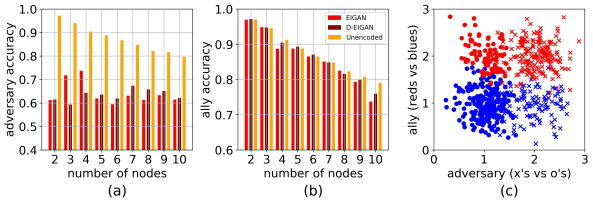

Figure 7: Comparison of (a) adversary and (b) ally performance as the number of nodes in the system is increased from $K = 2$ to 10, for D-EIGAN ($\phi, \delta = 1$), EIGAN, and unencoded. Node k's data, $k = 1, ..., K$ is generated from four Gaussians centered on a unit square, each with $\sigma^2 =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

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 [7, 19, 11].

Robustness of learned representation. We next consider the robustness of EIGAN's learned representation to ally and adversary architectures that deviate from the one used for training. Tab. 2 shows the performance of varying architectures (ResNet [32], ResNext [33], 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/adversary 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. 4 shows the effect of the ally label variance on the resulting accuracies for EIGAN, the method in [7], 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 [7] and is comparable to the unencoded case. Similar results are observed when changing the adversary variance (see App. D.1).

Varying system dimensions. We also considered 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. 5) that the final testing loss obtained by an adversary (ally) under varying number of allies (adversaries) stays reasonably constant. (ii) We find (in Fig. 6) that as encoding dimension l is increased, the training requires fewer epochs to converge, and is able to achieve a lower testing loss, even as l exceeds the input dimension. Thus, EIGAN encodings are robust to the number of objectives that are included in the system.

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 (centralized) EIGAN, and on the unencoded data. As 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 App. E.1 for similar observations on i.i.d. data across nodes.

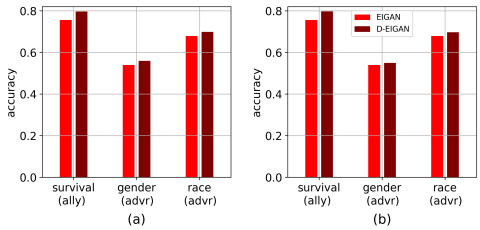

Figure 8: Performance of ally and adversary objectives trained on D-EIGAN ($K = 10, \phi = 0.8, \delta = 2$, non-i.i.d) for MIMIC in the cases of (a) all nodes having all three objectives and (b) each node having the ally but only one of the adversaries. The distribution of objectives across the nodes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

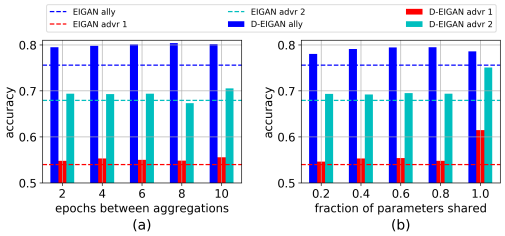

Figure 9: Effect of (a) aggregation frequency δ ($\phi = 0.8$) and (b) sparsification factor ϕ ($\delta = 2$) on ally/adv. performance on D-EIGAN for the non-i.i.d case in Fig. 8(a). The robust performance shows that D-EIGAN can be applied in communication-constrained environments.

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 Sec. 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 App. 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 Sec. 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 and 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 [34], 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 App. E.3). Thus, we conclude that D-EIGAN is well suited for 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.

References

[1] Arvind Narayanan and Vitaly Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *IEEE Symposium on Security and Privacy*, 2008.

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

[3] Sheikh Shams Azam, Manoj Raju, Venkatesh Pagidimarri, and Vamsi Chandra Kasivajjala. CASCADENET: An LSTM based Deep Learning Model for Automated ICD-10 Coding. In *FICC*, 2019.

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

[5] Proteek Chandan Roy and Vishnu Naresh Boddeti. Mitigating Information Leakage in Image Representations: A Maximum Entropy Approach. In *IEEE CVPR*, 2019.

[6] Bashir Sadeghi and Vishnu Naresh Boddeti. Imparting fairness to pre-trained biased representations. In *IEEE CVPR Workshops*, 2020.

[7] Martin Bertran, Natalia Martinez, Afroditi Papadaki, Qiang Qiu, Miguel Rodrigues, Galen Reeves, and Guillermo Sapiro. Adversarially Learned Representations for Information Obfuscation and Inference. In *ICML*, 2019.

[8] 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.

[9] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The Variational Fair Autoencoder. *arXiv preprint 1511.00830*, 2015.

[10] Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable Invariance through Adversarial Feature Learning. In *NeurIPS*, 2017.

[11] Ardhendu Tripathy, Ye Wang, and Prakash Ishwar. Privacy-Preserving Adversarial Networks. In *IEEE Allerton*, 2019.

[12] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees. In *ICLR*, 2018.

[13] Harrison Edwards and Amos Storkey. Censoring Representations with an Adversary. In *ICLR*, 2016.

[14] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual Fairness. In *NeurIPS*, 2017.

[15] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning Adversarially Fair and Transferable Representations. In *ICML*, 2018.

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

[17] M. Feder and N. Merhav. Relations between Entropy and Error Probability. *IEEE Trans. Inf. Theory*, Vol. 40, 1994.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, et al. Generative Adversarial Nets. In *NeurIPS*, 2014.

[19] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Generative Adversarial Privacy. *arXiv preprint*, *arXiv:1807.05306*, 2018.

[20] Jonas Adler and Sebastian Lunz. Banach Wasserstein GAN. In *NeurIPS*, 2018.

[21] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated Multi-task Learning. In *NeurIPS*, 2017.

[22] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, et al. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE JSAC*, 2019.

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

[24] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that exploit Confidence Information and Basic Countermeasures. In *ACM SIGSAC CCS*, 2015.

[25] Yann LeCun and Corinna Cortes. MNIST Handwritten Digit Database, 2010.

[26] 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.

[27] Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017.

[28] H. Ng and S. Winkler. A Data-Driven Approach to Cleaning Large Face Datasets. In *IEEE ICIP*, 2014.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015.

[30] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *IEEE CVPR*, 2017.

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

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE CVPR*, 2016.

[33] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated Residual Transformations for Deep Neural Networks. In *IEEE CVPR*, 2017.

[34] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and Communication-Efficient Federated Learning from Non-Iid Data. *IEEE Trans. Neural. Net. Learn. Syst.*, Vol. 31, 2019.

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 $H_Q = \mathbb{H}(P, Q) = \mathbb{E}_{x\sim P}[-\log Q]$, 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., α_{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 ith ally or adversary of the kth 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

Algorithm A1 EIGAN training

Notations:

$(\cdot)_j$ denotes the value for the jth minibatch

$\mathcal{L}_{A_i}$ denotes the loss of ally A_i

$\mathcal{L}_{V_i}$ denotes the loss of the adversary V_i

η_E, η_A, η_V : learning rates of the encoders, allies and adversaries

Training:

initialize α used in loss function (4)

initialize θ_{A_i}'s and θ_{V_j}'s and θ_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 Visualizations

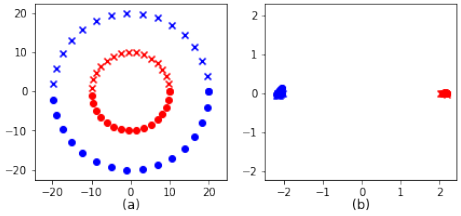

Figure A10: (a) Circle dataset with 4 types of points, one ally and one adversary. Ally classes (reds vs blues) are not linearly separable unlike adversary classes (x's vs o's). (b) EIGAN learns a transformation that makes the ally's classes linearly separable.

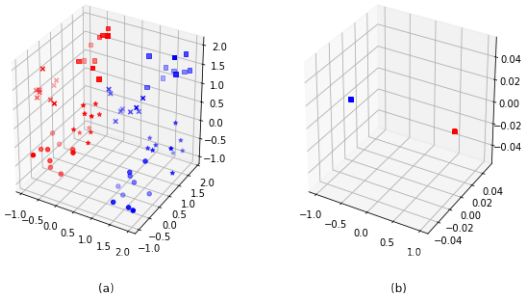

Figure A11: (a) Octant dataset with eight groups of points, one ally, and two adversaries. The ally is interested in classifying reds/blues while the adversaries are interested in separation along other axes. (b) EIGAN collapses the two adversary dimensions while maintaining separability for the ally.

D Additional EIGAN Experiments

D.1 Comparison with the Method in [7]

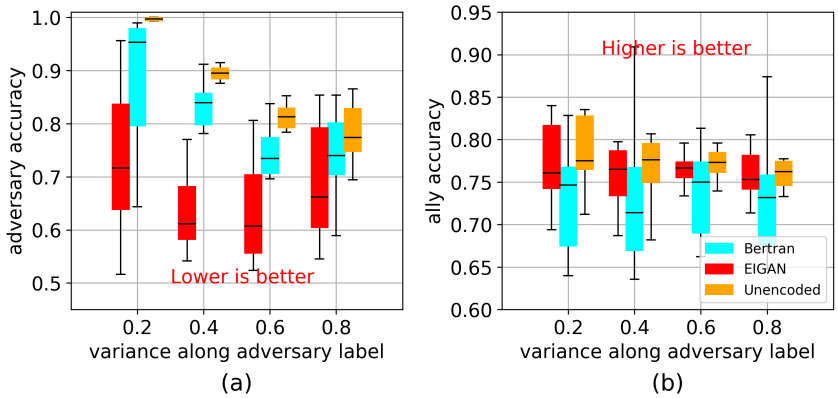

Figure A12: Effect of change in adversary overlap (a-b) on the performance of EIGAN, and its comparison with the unencoded data as well as the method in Bertran et al. [7]. EIGAN is able to consistently outperform both baselines on the adversary objective, and obtains performance close to the unencoded data for the ally.

D.2 Comparison on MIMIC-III

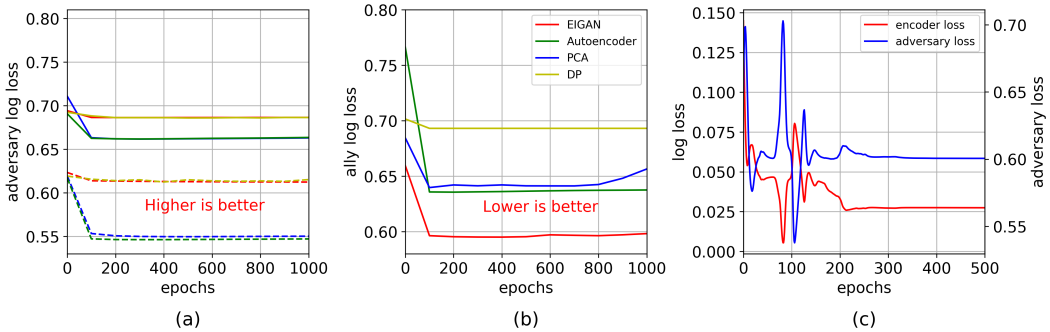

Figure A13: Predictivity and privacy comparison between EIGAN and the baselines across one ally and two adversaries on the MIMIC-III dataset. (a) On the adversary objectives (gender prediction, solid lines and race prediction, dashed lines) EIGAN matches DP's performance (by design of the experiment, 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.

D.3 Comparison on MNIST

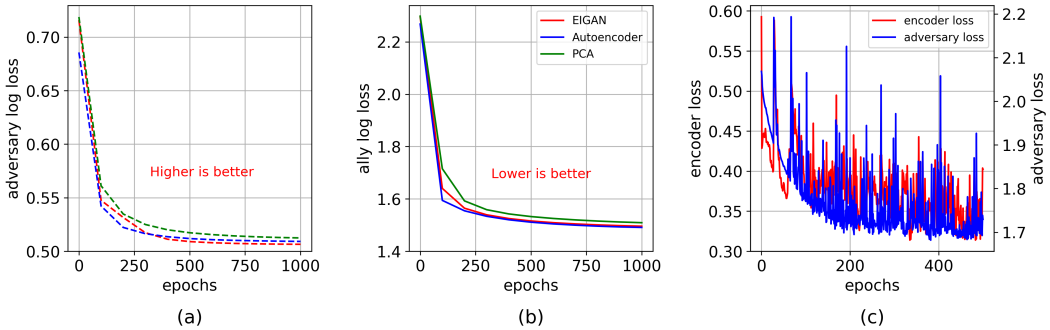

Figure A14: Comparison across one ally and two adversaries on the MNIST dataset. The (a) adversary objective (odd-even prediction, a binary classification with virtually identical trends) converge to roughly the same loss for each algorithm, and (b) ally objective (digit prediction, 10-class classification). 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.

E Additional D-EIGAN Experiments

E.1 Varying Number of Nodes

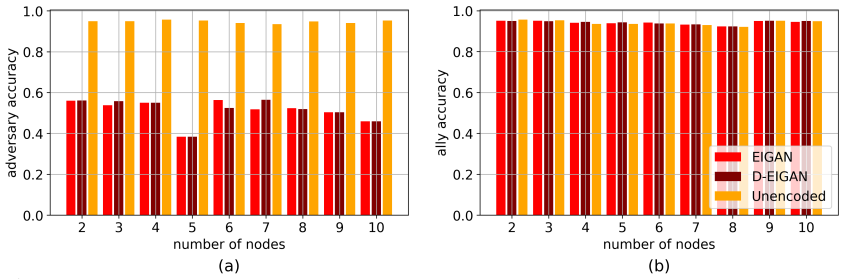

Figure A15: Comparison of (a) adversary and (b) ally performance using synthetic Gaussian data while increasing the number of nodes and sharing all the model weights ($\phi = 1$) after every minibatch ($\delta = 1$) during federated training. The data is i.i.d. across the nodes, which is obtained by 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.2 Varying Objectives across Nodes

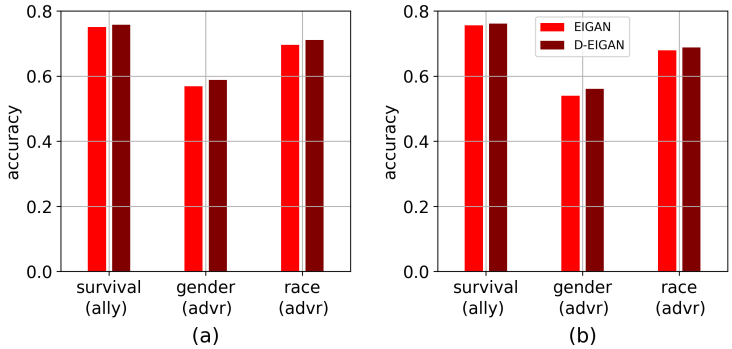

Figure A16: Comparison of D-EIGAN ($K = 2$ nodes) with centralized EIGAN. Survival is the ally objective, and gender and race are the chosen adversary objectives for the experiment. (a) Training of distributed EIGAN involves same adversary objectives, i.e., obfuscating gender and race across the both the nodes. (b) Each node has a different adversary objective, while they share the same ally objective.

E.3 Varying Synchronization Parameters

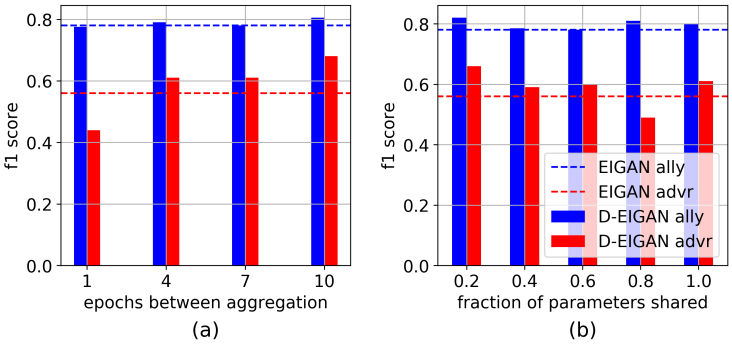

Figure A17: Effect of varying (a) frequency of sync (δ, number of epochs between parameter sharing) and (b) fraction of parameters uploaded/downloaded (ϕ) in D-EIGAN consisting of $K = 2$ nodes with i.i.d data using MIMIC-III. The results show that as ϕ is varied, the performance of the system on hiding the sensitive variable is not affected considerably (in contrast to Fig. 9 with non-i.i.d data).