Think Locally, Act Globally: Federated Learning with Local and Global Representations

Paul Pu Liang[1*], Terrance Liu[1*], Liu Ziyin[2], Nicholas B. Allen[3], Randy P. Auerbach[4], David Brent[5], Ruslan Salakhutdinov[1], Louis-Philippe Morency[1]

[1]School of Computer Science, Carnegie Mellon University
[2]Department of Physics, University of Tokyo
[3]Department of Psychology, University of Oregon
[4]Department of Psychiatry, Columbia University
[5]Department of Psychiatry, University of Pittsburgh
{pliang,terrancl,morency}@cs.cmu.edu

July 15, 2020

Abstract

Federated learning is a method of training models on private data distributed over multiple devices. To keep device data private, the global model is trained by only communicating parameters and updates which poses scalability challenges for large models. To this end, we propose a new federated learning algorithm that jointly learns compact *local representations* on each device and a global model across all devices. As a result, the global model can be smaller since it only operates on local representations, reducing the number of communicated parameters. Theoretically, we provide a generalization analysis which shows that a combination of local and global models reduces both variance in the data as well as variance across device distributions. Empirically, we demonstrate that local models enable *communication-efficient* training while retaining performance. We also evaluate on the task of *personalized* mood prediction from real-world mobile data where privacy is key. Finally, local models handle *heterogeneous* data from new devices, and learn *fair* representations that obfuscate protected attributes such as race, age, and gender.

1 Introduction

Federated learning is an emerging research paradigm to train machine learning models on private data distributed in a potentially non-i.i.d. setting over multiple devices [38]. A key challenge involves keeping private all the data on each device by training a global model only via communication of parameter updates to each device. This relies on the global model being sufficiently compact so that the parameters and updates can be sent efficiently over existing communication channels such as wireless networks [44]. However, the recent demands in larger models pose a challenge for deploying federated learning on real-world tasks. In this paper, we propose a new federated learning algorithm, Local Global Federated Averaging (LG-FEDAVG), which jointly learns compact *local representations* on each device and a global model across all devices. We perform a generalization analysis of federated learning which shows that a combination of local and

[*]first two authors contributed equally.

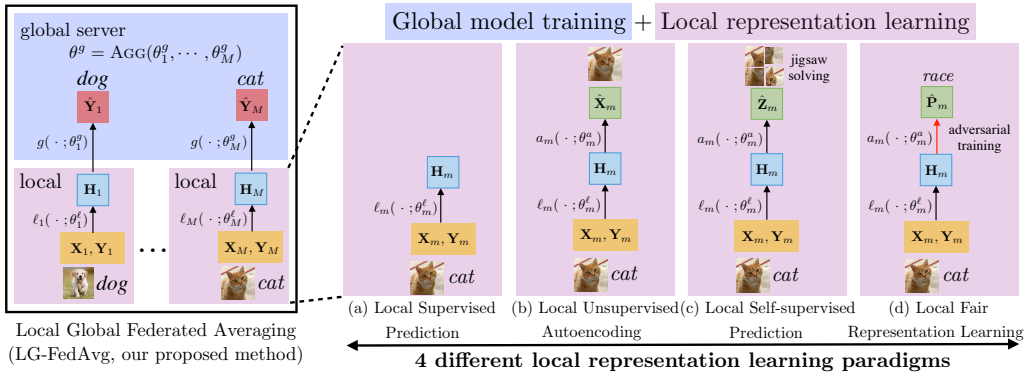

Figure 1: (a) Local Global Federated Averaging (LG-FEDAVG) allows for *efficient* global parameter updates (smaller number of global parameters θ^g), *flexibility* in design across local and global models, the ability to handle *heterogeneous* data, and *fair* representation learning. (a) through (c) show various approaches of training local models including supervised, unsupervised, and self-supervised learning (e.g. jigsaw solving [45]). (d) shows adversarial training against protected attributes $\mathbf{P}_m$. Blue represents the global server and purple represents the local devices. $(\mathbf{X}_m, \mathbf{Y}_m)$ represents data on device m, $\mathbf{H}_m$ are learned local representations via local models $\ell_m(\,\cdot\,;\theta^\ell_m) : \mathbf{x} \longrightarrow \mathbf{h}$ and (optionally) auxiliary models $a_m(\,\cdot\,;\theta^a_m) : \mathbf{h} \longrightarrow \mathbf{z}$. $g(\,\cdot\,;\theta^g) : \mathbf{h} \longrightarrow \mathbf{y}$ is the global model. AGG is an aggregation function over local updates to the global model (e.g. FEDAVG).

global models reduces both variance in the data as well as variance across device distributions, which is more optimal than either extreme. To support our theoretical analysis, we perform a wide range of experiments that suggest local representation learning is beneficial for the following reasons:

1) *Efficiency:* Having local models extract useful, lower-dimensional representations means that the global model now requires fewer number of parameters, thereby reducing the number of parameters and updates that need to be communicated to and from the global model as well as the bottleneck in terms of communication cost. Our proposed method also maintains performance on publicly available datasets spanning image recognition (MNIST, CIFAR) and multimodal learning (VQA).

2) *Heterogeneity:* Real-world data is often heterogeneous (coming from different sources). A new device could contain sources of data that have never been observed before during training, such as images of a different domain or different texting styles on personalized mobile devices. Local representations allow us to process new device data using specialized encoders depending on their source modalities [2] instead of using a single global model that might not generalize to new modalities and distributions [41]. We show that our model learns *personalized* mood predictors from real-world private mobile data and better deals with *heterogeneous* data never seen during training.

3) *Fairness:* Real-world data often contains sensitive attributes and recent work has shown that it is possible to recover these attributes from data representations without access to the data itself [4]. We show that local models can be modified to learn *fair* representations that obfuscate protected attributes such as race, age, and gender, a feature crucial to preserving the privacy of on-device data.

2 Related Work

Federated Learning aims to train models in massively distributed networks [38] at a large scale [5], over multiple sources of heterogeneous data [30], and over multiple learning objectives [50]. Recent methods aim to improve the efficiency of federated learning [7], perform learning in a one-shot setting [20], propose realistic benchmarks [8], and reduce the data mismatch between local and global data distributions [41]. While several specific algorithms have been proposed for heterogeneous data, LG-FEDAVG is a more *general* framework that can handle heterogeneous data from new devices, reduce communication complexity, and ensure fair representation learning. We compare with these existing baselines and show that LG-FEDAVG outperforms them in heterogeneous settings.

Distributed Learning is a related field with similarities and key differences: while both study the theory and practice involving the partition of data and aggregation of updates [3, 52], federated learning is additionally concerned with data that is private and distributed in a *non-i.i.d.* fashion. Recent work has improved communication efficiency by sparsifying the data and model [54], developing efficient gradient methods [55, 12], and compressing the updates [53]. These compression techniques are *complementary* to our approach and can be applied to our local and global models.

Representation Learning involves learning features from data for generative and discriminative tasks. A recent focus has been on learning *fair* representations [57], including using adversarial training [19] to learn representations that are not informative of private attributes [17, 9] such as demographics [15] and gender [56]. A related line of research is differential privacy which constraints statistical databases to limit the privacy impact on individuals whose information is in the database [13, 14]. Our approach extends recent advances in adapting federated learning for heterogeneous data and fairness. LG-FEDAVG is a *general* framework that can handle heterogeneous data from new devices, reduce communication complexity, and ensure fair representation learning.

3 Local Global Federated Averaging

At a high level, LG-FEDAVG combines local representation learning with global model learning in an *end-to-end manner*. Each local device learns to extract higher-level representations from raw data before a global model operates on the representations (rather than raw data) from all devices. An overview of LG-FEDAVG is shown in Figure 1. The local and global learning procedures are designed to be complementary: local representation learning aims to extract *high level, compact features* important for prediction, thereby allowing the global model to save parameters by operating only on *lower dimensional representations*. At the same time, the global model objective ensures that the global model must be able to classify data from *all* devices, thereby ensuring that the local representations are general enough instead of overfitting to the subset of data on each device. We begin by describing how local (§3.1) and global (§3.2) learning is performed. We then detail one example of adversarial local learning to learn fair local representations (Appendix B.1).

Notation: We use uppercase letters X to denote random variables and lowercase letters x to denote their values. Upper case boldface letters $\mathbf{X}$ denote datasets consisting of multiple vector data points $\mathbf{x}$ which we represent by lowercase boldface letters. In the standard federated learning setting, we assume that we have data $\mathbf{X}_m \in \mathbb{R}^{N_m \times d}$, $m \in [M]$ and their corresponding labels $\mathbf{Y}_m \in \mathbb{R}^{N_m \times c}$, $m \in [M]$ across M devices. N_m denotes the number of data points on device m, $N = \sum_m N_m$ is the total number of data points, d represents the input dimension and c represents the number of classes for classification ($c = 1$ for regression).

Algorithm 1 LG-FEDAVG. The M clients are indexed by m, η is the learning rate.

Server executes:

initialize global model with weights θ^g; initialize M local models with weights θ^ℓ_m.
for each round $t = 1, 2, \ldots$ **do**
  $m \leftarrow \max(C \cdot M, 1)$; $S_t \leftarrow$ (random set of m clients)
  for each client $m \in S_t$ **in parallel do**
    $\theta^{g(t+1)}_m \leftarrow \text{ClientUpdate}(m, \theta^{g(t)})$
  $\theta^{g(t+1)} \leftarrow \sum_{m=1}^{M} \frac{N_m}{N} \theta^{g(t+1)}_m$ // aggregate updates

过于naive，是否可以加权平均

ClientUpdate (m, θ^g_m): // run on client m
$\mathcal{B} \leftarrow$ (split local data $(\mathbf{X}_m, \mathbf{Y}_m)$ into batches)
for each local epoch **do**
  for batch $(\mathbf{X}, \mathbf{Y}) \in \mathcal{B}$ **do**
    $\mathbf{H} = \ell_m(\mathbf{X}; \theta^\ell_m)$, $\hat{\mathbf{Y}} = g(\mathbf{H}; \theta^g_m)$ // inference steps
    $\theta^\ell_m \leftarrow \theta^\ell_m - \eta_{\theta^\ell_m} \mathcal{L}^g_m(\theta^\ell_m, \theta^g_m)$ // update local model wrt global loss
    $\theta^g_m \leftarrow \theta^g_m - \eta_{\theta^g_m} \mathcal{L}^g_m(\theta^\ell_m, \theta^g_m)$ // update (local copy of) global model wrt global loss
return global parameters θ^g_m to server

Intuitively, each source of data captures a different view $p(X_m, Y_m)$ of the global data distribution $p(X, Y)$. In our experiments, we consider settings where the individual data points in $\mathbf{X}_m, \mathbf{Y}_m$ are sampled both i.i.d. and non i.i.d. with respect to $p(X, Y)$. During training, we use parenthesized superscripts (e.g. $\theta^{(t)}$) to represent iteration t.

3.1 Local Representation Learning

For each source of data $(\mathbf{X}_m, \mathbf{Y}_m)$, we learn a representation $\mathbf{H}_m$ which should: 1) be low-dimensional as compared to raw data $\mathbf{X}_m$, 2) capture important features in $\mathbf{X}_m$ that are useful towards the global model, and 3) not overfit to device data which may not align to the global data distribution. To be more concrete, we define features $\mathbf{z} \in \mathcal{Z}$ that should be captured using a good representation $\mathbf{h}$. In Figure 1(a) through 1(c) we summarize these local learning methods according to the choice of $\mathbf{z}$: (a) the labels $\mathbf{y}$ (supervised learning), (b) the data itself $\mathbf{x}$ (unsupervised autoencoder learning), or (c) some auxiliary labels $\mathbf{z}$ (self-supervised learning). For simplicity, we focus the description on supervised learning but describe extensions to local adversarial learning of fair representations (Figure 1(d)) and unsupervised learning in Appendix B.1.

本地模型学习低维表示

Each device consists of a local model $\ell_m : \mathbf{x} \rightarrow \mathbf{h}$ with parameters θ^ℓ_m which allow us to infer features $\mathbf{H}_m = \ell_m(\mathbf{X}_m; \theta^\ell_m)$ from local device data. These features should be useful in predicting the labels using a joint global model $g : \mathbf{h} \rightarrow \mathbf{y}$ with parameters θ^g over the features from all devices $\{\mathbf{H}_1, ..., \mathbf{H}_M\}$. The key difference is that the global model $g : \mathbf{h} \rightarrow \mathbf{y}$ now operates on lower-dimensional local representations $\mathbf{H}_m$. Therefore, g can be a much smaller model which we will show in our experiments (§5.2).

3.2 Global Aggregation

Learning this joint global model g across all devices requires the aggregation of global parameter updates from each device. At each iteration t of global model training, the server sends a copy of the global model parameters $\theta^{g(t)}$ to each device which we now label as $\theta^{g(t)}_m$ to represent the asynchronous updates made to

把全球模型发给本地服务器
训练后把全球模型发回服务器
服务器进行平均得到下一轮全局模型

each local copy. Each device runs their local model $\mathbf{H}_m = \ell_m(\mathbf{X}_m; \theta^\ell_m)$ to obtain local features and the global model $\hat{\mathbf{Y}}_m = g(\mathbf{H}_m; \theta^{g(t)}_m)$ to obtain predictions. We can compute the overall loss on device m:

$$\mathcal{L}^g_m(\theta^\ell_m, \theta^g_m) = \mathbb{E}_{\substack{\mathbf{x}\sim X_m \\ \mathbf{y}\sim Y_m|\mathbf{x}}}\left[-\log \sum_{\mathbf{h}} \left(p_{\theta^g_m}(\mathbf{y}|\mathbf{h})\, p_{\theta^\ell_m}(\mathbf{h}|\mathbf{x})\right)\right]. \quad (1)$$

概率越大 loss越小

The loss is a function of both the local and global model parameters (θ^ℓ_m and θ^g_m respectively) so both can be updated in an end-to-end manner. We argue that this synchronizes the training of the local models: while local models can flexibly fit the small amounts of data on their device, objective 1 acts as a regularizer to synchronize the local representations learned from all devices. Each local model cannot overfit to local data because otherwise, the joint global model would not be able to simultaneously predict from all local representations and the value of objective 1 would be high.

如果本地特征过拟合全球模型的p会很小 loss会很大

The local model parameters θ^ℓ_m are updated by gradient-based methods in a straightforward manner. The global model parameters on device m, $\theta^{g(t)}_m$, are also asynchronously updated to $\theta^{g(t+1)}_m$ using gradient-based methods. After these local updates, each device now returns the updated global parameters $\theta^{g(t+1)}_m$ back to the server which aggregates these updates using a weighted average of the fraction of data points in each device, $\theta^{g(t+1)} = \sum_{m=1}^{M} \frac{N_m}{N} \theta^{g(t+1)}_m$ [38]. The overall training procedure is shown in Algorithm 1. Communication only happens when training the global model, which as we will show in our experiments, can be small given good local representations.

3.3 Inference at Test Time

Given a new test sample $\mathbf{x}'$, how do we know which trained local model ℓ^*_m fits $\mathbf{x}'$ best? We consider two settings: (1) **Local Test** where we know which device the test data belongs to (e.g. training a personalized text completer). Using that local model works best for best match between train and test distributions. (2) **New Test** where it is possible to have a new device during testing with new data distributions. To address the new device, we view each local model as trained on a different view of the global data distribution. We pass $\mathbf{x}'$ through all trained local models ℓ^*_m and *ensemble* the outputs.

4 Theoretical Analysis

In this section, we provide a theoretical analysis of using local and global models for federated learning. We show that 1) purely local models do not suffer from *device variance* but suffer from *data variance*, 2) the opposite holds true for purely global models, and 3) having both local and global models achieves a balance between both desiderata. All detailed proofs can be found in Appendix A. The link between the analysis and LG-FedAvg for deep networks is discussed at the end of the section and verified via comprehensive experiments in § 5.1.

We assume a student-teacher setting [25, 21], where the goal is to train a network $f_{\hat{\mathbf{u}}}$ with weights $\hat{\mathbf{u}} \in \mathbb{R}^d$ on a task whose target is produced by a teacher network $f_{\mathbf{u}}$ (i.e. the *realizability assumption*). We assume that the targets are generated from a linear model. While simple in setup, its behavior is rich and has proved insightful in the understanding of nonlinear models as well [40, 39, 21]. To adapt this setting for federated learning, we assume that all device share some underlying structure (e.g. natural syntactic and semantic structures in text) while also displaying personalization across users (e.g. personalized vocabularies and writing styles). Mathematically, this involves a global feature vector $\mathbf{v}$ that represents shared features across devices as well

假设每个设备是由一个共享v和个性化r组成

as local features $\mathbf{r}_m$ that represent differences across devices. The labels on device m are generated by a local teacher with weights $\mathbf{u}_m = \mathbf{v} + \mathbf{r}_m \in \mathbb{R}^d$. We assume that each local feature $\mathbf{r}_m \sim \mathcal{N}(0, \rho^2 I)$ is a different independent draw from a d-dimensional Gaussian with diagonal covariances of ρ^2. ρ^2 represents *device variance*: with higher ρ^2, the local features differ more representing more personalized targets across devices. We also assume that the training targets are corrupted by noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ to account for naturally-occurring *data variance* [42]. Under this model, the training targets on model m are given by $\tilde{f}_{\mathbf{u}_m}(\mathbf{x}) = f_{\mathbf{u}_m}(\mathbf{x}) + \epsilon$. For simplicity, we assume each device contains N data points.

Given N datapoints $\{\mathbf{x}_i\}_{i=1,\dots,N}$, learning local and global parameters $\hat{\mathbf{u}}_m, \hat{\mathbf{v}}$ involve optimizing the following training objectives:

$$\hat{\mathbf{u}}_m = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \left(f_{\mathbf{w}}(\mathbf{x}_i) - \tilde{f}_{\mathbf{u}_m}(\mathbf{x}_i) \right)^2, \hat{\mathbf{v}} = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{m=1}^{M} \sum_{i=1}^{N} \left(f_{\mathbf{w}}(\mathbf{x}_i) - \tilde{f}_{\mathbf{u}_m}(\mathbf{x}_i) \right)^2. \tag{2}$$

We denote the overall model as $f(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m)$ using both local and global models. The *local empirical generalization error* on device m is defined as

$$\hat{\mathcal{E}}_m = \frac{1}{N} \sum_{i=1}^{N} \left(f(\mathbf{x}_i; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) - f_{\mathbf{u}_m}(\mathbf{x}_i) \right)^2. \tag{3}$$

The total *empirical* generalization error is defined as the mean of all local errors $\hat{\mathcal{E}} = \frac{1}{M} \sum_{m=1}^{M} \hat{\mathcal{E}}_m$ and the true *generalization error* is defined as the expectation taken over the randomness present in the data, devices, and noise:

$$\mathcal{E} = \mathbb{E}_{\mathbf{x}, \mathbf{r}_m, \epsilon}[\hat{\mathcal{E}}_m] = \mathbb{E}_{\mathbf{x}, \mathbf{r}_m, \epsilon} \left[\left(f(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) - \tilde{f}_{\mathbf{u}_m}(\mathbf{x}) \right)^2 \right] \tag{4}$$

which can further be manipulated to obtain a *bias-variance decomposition* for federated learning.

Theorem 1. *The generalization loss for federated learning can be decomposed as*

$$\mathcal{E} = \mathbb{E}_{\mathbf{x}, \mathbf{r}_m, \epsilon}[\hat{\mathcal{E}}_m] = Var[\hat{f}] + b^2 \tag{5}$$

with variance $Var[\hat{f}] = \mathbb{E}_{\mathbf{x}, \mathbf{r}_m} \left[Var_\epsilon[\hat{f} | \mathbf{x}, \mathbf{r}_m] \right]$ *and bias* $b^2 = \mathbb{E} \left[\left(f_{\mathbf{u}_m} - \mathbb{E}_\epsilon f(\hat{\mathbf{v}}, \hat{\mathbf{u}}_m) \right)^2 \right]$.

Using only local models results in an unbiased estimator of $\mathbf{u}_m$. The bias term arises when learning global parameters since federated learning couples the estimation of both local and global parameters. The variance term comes from both the variance of both local and global parameter estimates.

As a simplified version, LG-FedAvg can be seen as a ensemble of local and global models, i.e. $f(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) = \alpha f_{\hat{\mathbf{u}}_m}(\mathbf{x}) + (1 - \alpha) f_{\hat{\mathbf{v}}}(\mathbf{x})$. In this case, one can show that:

Proposition 1. *Let* $\mathbb{E}_{\mathbf{r}_m, \epsilon}[f_{\hat{\mathbf{v}}}] = f_{\mathbf{v}}$, $\mathbb{E}_\epsilon[f_{\hat{\mathbf{u}}_m}] = f_{\mathbf{u}_m}$, *and let* $f(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) = \alpha f_{\hat{\mathbf{u}}_m}(\mathbf{x}) + (1 - \alpha) f_{\hat{\mathbf{v}}}(\mathbf{x})$, *then equation 5 can be written as*

$$\mathcal{E} = (1 - \alpha)^2 \delta^2 + \mathrm{Var}[\hat{f}], \tag{6}$$

where $\delta^2 = \mathbb{E}_{\mathbf{x}, \mathbf{r}_m}[(f_{\mathbf{u}} - \mathbb{E}_\epsilon[f_{\mathbf{v}}])^2 | \mathbf{r}_m]]$ *measures the discrepancy between the local and global features as a result of local variations across devices.*

For the linear setting we are considering, the above result can be further expanded as:

$$\mathcal{E} = (1 - \alpha)^2 \left(\frac{M - 1}{M} \right) \rho^2 + \mathrm{Var}[\hat{f}]. \tag{7}$$

Analysis of local and global baselines: We first analyze two baselines for federated learning. The first method learns local models on each device [50]: $f_\ell(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) = f(\mathbf{x}; \hat{\mathbf{u}}_m) = \hat{\mathbf{u}}_m^\top \mathbf{x}$.

Proposition 2. *The generalization error $\mathcal{E}(f_\ell)$ of local model $f_\ell(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m)$ is $\frac{d}{N}\sigma^2$.*

This shows that local models only control data variance at a rate of d/N since they are only updated using local device data which may be limited in number and vary highly (both in quality and quantity) across devices. However, local models *do not suffer from device variance*.

The second method updates a joint global model (i.e. vanilla federated learning; [38]), which is equivalent to setting $\alpha = 0$, i.e. $f_g(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) = f(\mathbf{x}; \hat{\mathbf{v}}) = \hat{\mathbf{v}}^\top \mathbf{x}$. Its generalization error $\mathcal{E}(f_g)$ is:

Proposition 3. *The generalization error $\mathcal{E}(f_g)$ of the global model $f_g(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m)$ is $\frac{M-1}{M}\rho^2 + \frac{d}{MN}\sigma^2$.*

Global models can control for data variance (σ^2) at a rate of $d/(MN)$, decreasing with the total number of datapoints across *all* devices (since global parameters are updated using data across all devices), which is better than the rate for local models. However, it suffers from an extra $O(\rho^2)$ term representing device variance so one global model is unable to account for very different devices.

Analysis of LG-FEDAVG: Given that the above baselines achieve different generalization errors, one should be able to interpolate between the two methods to find the optimal tradeoff point. Therefore, our method defines an α–interpolation between the global and local models, $f_\alpha(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) = \alpha f_\ell(\mathbf{x}; \hat{\mathbf{u}}_m) + (1-\alpha) f_g(\mathbf{x}; \hat{\mathbf{v}})$, where $\alpha \in [0, 1]$. The generalization error, $\mathcal{E}(f_\alpha)$ is:

Theorem 2. *The generalization error $\mathcal{E}(f_\alpha)$ is $\alpha^2 \frac{d}{N}\sigma^2 + (1-\alpha)^2 \frac{M-1}{M}\rho^2 + (1-\alpha^2)\frac{d}{MN}\sigma^2$.*

Corollary 1. *The optimal α^* minimizing $\mathcal{E}(f_\alpha)$ is $\alpha^* = \frac{\rho^2}{\rho^2 + \frac{d}{N}\sigma^2}$. When $\rho^2,\ \sigma^2 > 0$, we have that $\mathcal{E}(f_{\alpha^*}) < \mathcal{E}(f_\ell)$ and $\mathcal{E}(f_{\alpha^*}) < \mathcal{E}(f_g)$, a generalization error better than local or global extremes.*

This shows that using an ensemble of local and global models *reduces both data variance and device variance*. When ρ^2 is large (high device variance), one should prioritize local models that better model the local data distributions (larger α^*). Conversely, when σ^2 is large (high data variance), one should prioritize a global model (smaller α^*).

While our theory holds true for linear models, we believe that it provides accurate insight into the practical generalization abilities of LG-FEDAVG, where we use deep networks and treat α as the *split* of the layers of between the local and global models. Empirically, we compare Figure 2(a)-(b) (test error for linear models using *α-interpolation* on synthetic data) with Figure 2(d) (test accuracy for deep networks using *α-split* on real-world mobile data). The close similarity implies that our theoretical analysis captures the correct relationship between local and global models.

5 Experiments

We evaluate how our method 1) verifies our theory under different data and device variances, 2) *efficiently* reduces parameters while retaining performance, 3) learns *personalized* models and handles data from *heterogeneous* sources, two settings with particularly high device variance, and 4) learns fair local representations that obfuscate private attributes. Code is included in the supplementary. Implementation details and sensitivity reports across hyperparameters are provided in Appendix C.

5.1 Verifying Theoretical Analysis

Synthetic Data: We first experiment on synthetic data to verify our theoretical analysis on ensembling local and global models. Data on device m is generated by $\mathbf{x} \sim \mathcal{U}[-1.0, 1.0]$ and teacher weights $\mathbf{u}_m = \mathbf{v} + \mathbf{r}_m$ are sampled as $\mathbf{v} \sim \mathcal{U}[0.0, 1.0]$, $\mathbf{r}_m \sim \mathcal{N}(\mathbf{0}_d, \rho^2 \mathbf{I}_d)$, where ρ^2 represents device variance. Labels are observed with

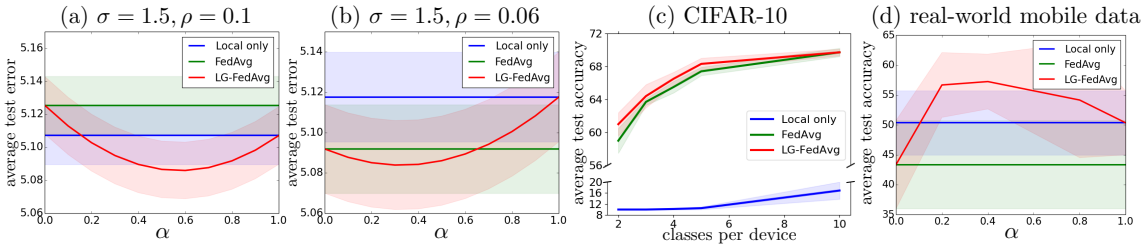

Figure 2: Test error (with shaded std dev) on synthetic data when local models perform better (plot (a): $\sigma = 1.5, \rho = 0.1$) and when global models perform better (plot (b): $\sigma = 1.5, \rho = 0.06$). For both settings, using an α-interpolation of local and global models performs better than either extremes. (c): We also verify these theoretical findings on increasing device variance when splitting CIFAR-10 (fewer classes per device), where LG-FEDAVG consistently outperforms local only and FEDAVG. (d): On predicting personalized moods from real-world private mobile data, an α-split across local and global models outperforms either extremes.

Table 1: Comparison of federated learning methods on CIFAR-10 with non-iid split over devices. We report accuracy under both local test and new test settings as well as the total number of parameters communicated across training iterations. Best results in **bold**. LG-FEDAVG outperforms FEDAVG and MTL under local test and achieves similar performance under new test while using around 50% of the total parameters, and outperforms all using the same number of parameters. Mean and standard deviation are computed over 10 runs.

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
CIFAR-10	FEDAVG [38]	58.99 ± 1.50	58.99 ± 1.50	1800	0	12.7×10^9
	Local only [50]	87.93 ± 2.14	10.03 ± 0.06	0	0	0
	MTL [50]	89.68 ± 0.75	10.06 ± 0.11	1800	0	12.0×10^9
	LG-FEDAVG (ours)	**91.07 ± 0.50**	57.95 ± 1.48	1200	100	$\mathbf{8.5 \times 10^9}$
	LG-FEDAVG (ours)	**91.77 ± 0.56**	**60.79 ± 1.45**	1800	100	12.7×10^9

noise, $y = \mathbf{u}_m^\top \mathbf{x} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where σ^2 represents data variance. We plot the average test error when local models perform better due to higher device variance (Figure 2(a), $\sigma = 1.5, \rho = 0.1$) and when global models perform better due to lower device variance (Figure 2(b), $\sigma = 1.5, \rho = 0.06$). In Appendix C.1, we discuss the performance of LG-FEDAVG under several other variance settings. For all settings, using an α-interpolation of both local and global models performs either close to the optimal extremes or better than either extreme.

CIFAR-10: Next, we verify our theory on deep networks over complex image classification problems using CIFAR-10. We focus on a highly *non-i.i.d.* setting and follow the experimental design in [38] by assigning examples from at most $s \in \{2, 3, 4, 5, 10\}$ classes to each device. The value of s simulates device variance: $s = 2$ represents highest device variance while $s = 10$ represents an i.i.d. split of labels to devices. From Figure 2(c), we observe that LG-FEDAVG consistently outperforms local only and FEDAVG. The performance gap is higher as device variance increases, which supports our theory that local models deal with high device variance.

5.2 Model Performance & Communication Efficiency

CIFAR-10: We compare our approach with existing federated learning methods with respect to model performance and communication. We randomly assign each device to examples of two classes (highly unbalanced). We consider two settings during testing: 1) *Local Test*, where we know which device the data belongs to (i.e. new predictions on an existing device) and choose that particular trained local model. For this

Table 2: Comparison of FEDAVG and LG-FEDAVG methods on VQA on non-i.i.d. device split setting. LG-FEDAVG achieves strong performance while using fewer communicated parameters.

Data	Method	Local Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Communicated (↓)
VQA	FEDAVG [38]	40.02	47	0	13.97×10^{10}
	LG-FEDAVG (ours)	**40.94**	32	17	$\mathbf{9.99 \times 10^{10}}$

setting, we split each device's data into train, validation, and test data, similar to [50]. 2) *New Test*, in which we do not know which device the data belongs to (i.e. new predictions on new devices) [38], so we use an ensemble approach by averaging all trained local model logits before choosing the most likely class [6]. For ensembling, all local model weights are sent to the server *only once* and averaged. We include this parameter exchange step and still show substantial communication improvement. We find that averaging model weights performs similarly (0.5% for CIFAR) to averaging model outputs since deep ReLU nets are mostly linear). We choose LeNet-5 [28] as our base model and we compare 4 methods: 1) FEDAVG [38] which is the traditional federated learning approach, 2) Local only [50] as an extreme setting with only local models, 3) MTL [50] which trains local models with parameter sharing in a multi-task fashion, and 4) LG-FEDAVG which is our proposed method with local and global models.

The results in Table 1 show that **LG-FEDAVG gives strong performance with low communication cost**. For CIFAR local test, LG-FEDAVG significantly outperforms FEDAVG since local models allow us to better model the local device data distribution. For new test, LG-FEDAVG achieves similar performance to FEDAVG while using around 50% of the total parameters, and better performance with the same number of parameters. LG-FEDAVG also outperforms using local models only and local models trained with multitask learning (MTL). This shows that our end-to-end training strategy for local and global models is particularly suitable for large neural networks. We show MNIST results and sensitivity analysis wrt data and model splits in Appendix C.2 and find similar observations.

Visual Question Answering (VQA): VQA is a large-scale multimodal benchmark with 0.25M images, 0.76M questions, and 10M answers [1]. We split the dataset in a non-i.i.d. manner and evaluate the accuracy under the local test setting. We use LSTM [23] and ResNet-18 [22] unimodal encoders as our local models and a global model which performs early fusion [32] of text and image features for answer prediction. In Table 2, we observe that LG-FEDAVG reaches a goal accuracy of 40% while requiring lower communication costs (more VQA results and details in Appendix C.2.3).

5.3 Learning Personalized Mood Predictors from Mobile Data

Data: We designed and collected a new dataset, Mobile Assessment for the Prediction of Suicide (MAPS), to determine real-time indicators of suicide risk in adolescents aged 13 – 18 years. This study monitors 100 adolescents including individuals who have recently attempted suicide, individuals who experience suicidal ideation, and psychiatric controls. Across a duration of 6 months, data was collected from each participant's smartphone using a keyboard logger which tracks all typed words. Participants were asked to rate their mood for the previous day on a scale ranging from 1 – 100, with higher scores indicating a better mood. All users have given consent for their mobile device data to be collected and shared with us for research purposes. MAPS is a realistic federated learning benchmark since it contains real-world data with privacy concerns and high device variance due to highly personalized use of mobile phones. We used a preliminary preprocessed version containing 572 samples across 14 participants. We discretize the scores into 5 bins for 5-way classification. We use a random 80/10/10 split for training/validation/testing, conduct all experiments 10 times, and report the average accuracy and standard deviation (details in Appendix C.3).

Models: To assess how mobile text data can be used to make personalized mood predictions, we train a MLP

Table 3: What happens when FEDAVG trained on 100 devices of normal MNIST sees a device with rotated MNIST? Catastrophic forgetting, unless one fine-tunes again on training devices and incur high communication cost. LG-FEDAVG relieves catastrophic forgetting by using local models to perform well on both online rotated and regular MNIST, with ($C = 0.1$) and without ($C = 0.0$) fine-tuning.

Data	Method	C	i.i.d. device data		non-i.i.d. device data	
			Normal (↑)	Rotated (↑)	Normal (↑)	Rotated (↑)
MNIST	FEDAVG [38]	0.0	32.0 ± 6.2	91.8 ± 3.0	35.7 ± 4.3	93.6 ± 0.3
	LG-FEDAVG (ours)	0.0	**96.6 ± 0.9**	**92.9 ± 2.7**	**96.3 ± 0.3**	**94.1 ± 0.7**
MNIST	FEDAVG [38]	0.1	97.4 ± 0.3	89.3 ± 0.8	96.9 ± 0.5	89.6 ± 0.6
	FEDPROX [30]	0.1	94.8 ± 1.1	87.2 ± 0.7	97.9 ± 0.1	91.6 ± 0.2
	LG-FEDAVG (ours)	0.1	**97.7 ± 0.8**	**93.2 ± 1.3**	**98.2 ± 0.7**	**93.9 ± 1.4**

classifier on top of a Bi-LSTM encoder on word embeddings. In addition to local only and FEDAVG, we test LG-FEDAVG across different splits of local and global model layers (i.e. $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$) while keeping the total parameter count constant.

Results: From Figure 2(d), consistent with our theoretical findings, an α-split across local and global models leverages both personalized representations per devices as well as statistical strength sharing through data across all devices, **outperforming either local or global extremes**.

5.4 Heterogeneous Data in an Online Setting

Data: We test whether LG-FEDAVG can handle heterogeneous data from a new source introduced during testing. We split MNIST across 100 devices in both an i.i.d. and non-i.i.d. setting, then introduce a new device with $3,000$ training and 500 test MNIST examples but *rotated* 90 *degrees*. This simulates a drastic change in the data distribution.

Models: We consider 3 methods: 1) FEDAVG, 2) FEDPROX [30]: a method designed specifically for heterogeneous data by regularizing the local updates to reduce overfitting to local devices, and 3) LG-FEDAVG: train on the original 100 devices, and when a new device comes, learn local representations before fine-tuning the global model. We hypothesize that good local models can "unrotate" images from the new device to better match the data distribution seen by the global model. When learning on the new device, we also retrain on a fraction C of the original devices: $C = 0.0$ implies no fine-tuning and $C = 0.1$ implies some fine-tuning ($C = 1.0$ implies retraining on all data which is impractical).

Results: We report results in Table 3 and observe that: 1) **FEDAVG suffers from catastrophic forgetting** [48] without fine-tuning ($C = 0.0$), in which the global model can perform well on the new device's rotated MNIST (92%) but completely forgets how to classify regular MNIST (32%). Only after fine-tuning ($C = 0.1$) does the performance on both regular and rotated MNIST improve, but this requires more communication over the 100 devices. 2) **LG-FEDAVG with local models relieves catastrophic forgetting**. Augmenting local models indeed helps to improve online performance on rotated MNIST (93%) while allowing the global model to retain performance on regular MNIST (97%), outperforming both FEDAVG and FEDPROX. We believe LG-FEDAVG achieves these results by learning a strong local representation that requires fewer updates from the trained global model.

5.5 Learning Fair Representations

Data: We examine whether local models can be trained to protect private attributes from the global model. We use the UCI adult dataset [24] to predict whether an individual makes more than 50K per year based on

Table 4: Enforcing independence with respect to protected attributes *race* and *gender* on income prediction with the UCI dataset. LG-FEDAVG+Adv uses local models with adversarial (adv) training to remove information about protected attributes, at the expense of a small drop in classifier (class) accuracy of around 2 – 4%.

Data	Method	i.i.d. device data			non-i.i.d. device data		
		Class Acc (↑)	Class AUC (↑)	Adv AUC (↓)	Class Acc (↑)	Class AUC (↑)	Adv AUC (↓)
UCI	FEDAVG [38]	83.7 ± 3.1	89.4 ± 1.9	65.5 ± 1.6	83.7 ± 1.8	88.7 ± 1.2	64.1 ± 2.1
	LG-FEDAVG	84.3 ± 2.4	89.0 ± 2.2	63.3 ± 3.7	81.1 ± 1.6	84.4 ± 2.4	62.7 ± 2.5
	LG-FEDAVG+Adv	82.1 ± 1.0	85.7 ± 1.7	**50.1 ± 1.3**	80.1 ± 2.0	84.1 ± 2.3	49.8 ± 2.2

their personal attributes. However, we want our models to be invariant to the sensitive attributes of *race* and *gender* instead of picking up on correlations that could exacerbate biases.

Models: We adapt adversarial learning to remove protected attributes from local models (see Appendix B.1. Specifically, we aim to learn fair local representations from which a fully trained adversarial network should *not* be able to predict the protected attributes. We report three methods: 1) FEDAVG with only a global model and global adversary both updated using FEDAVG. The global model is not trained with the adversarial loss since it is simply not possible: once local device data passes through the global model, privacy is potentially violated. 2) LG-FEDAVG without penalizing the adversarial network, and 3) LG-FEDAVG+Adv which jointly trains local, global, and adversary models to learn fair local representations before global prediction.

Results: We report results according to: 1) classifier binary accuracy, 2) classifier ROC AUC score, and 3) adversary ROC AUC score. The classifier metrics should be as close to 100% as possible while the adversary should be as close to 50% as possible. From Table 4, LG-FEDAVG+Adv **learns fair local representations that are unable to predict protected attributes** (~ 50% adversary AUC) with **only a small drop in global accuracy** (~ 4%). In order to ensure that poor adversary AUC was indeed due to fair representations instead of a poorly trained adversary, we train a post-fit classifier from local representations to protected attributes and achieve similar random results.

6 Conclusion

We proposed LG-FEDAVG combining *local representation learning* with federated training of global models. Our theoretical analysis shows that an ensemble of local and global models reduces both data variance and device variance. On a suite of real-world datasets, LG-FEDAVG achieves strong performance while reducing communication costs, learns personalized models, better deals with heterogeneous data, and effectively learns fair representations that obfuscate protected attributes.

Acknowledgements

PPL and LM were partially supported by the National Science Foundation (Awards #1750439, #1722822) and National Institutes of Health. RS was supported in part by NSF IIS1763562, Office of Naval Research N000141812861, and Google focused award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of National Science Foundation or National Institutes of Health, and no official endorsement should be inferred. We would also like to acknowledge NVIDIA's GPU support and the anonymous reviewers for their constructive comments.

Broader Impacts

Federated learning provides tools for large-scale distributed training at unprecedented scales but at the same time requires more research on its implications to society and policy.

Broader applications: By 2025, it is estimated that there will be more than to 75 billion IoT (Internet of Things) devices all connected to the internet and sharing data with each other [51]. Organizing, processing, and learning from device data will use federated learning techniques. It has already been shown to be a promising approach for applications such as learning the social activities of mobile phone users, early forecasting of health events like heart attack risks from wearable devices, and localization of pedestrians for autonomous vehicles [29]. The societal impacts revolving around more invasive federated learning technologies have to be taken into account as we design future systems that increasingly leverage distributed mobile data.

Applications in mental health: Suicide is the second leading cause of death among adolescents. In addition to deaths, 16% of high school students report seriously considering suicide each year, and 8% make one or more suicide attempts (CDC, 2015). Despite these alarming statistics, there is little consensus concerning imminent risk for suicide [18, 26]. Given the impact of suicide on society, there is an urgent need to better understand the behavior markers related to suicidal ideation.

"Just-in-time" adaptive interventions delivered via mobile health applications provide a platform of exciting developments in low-intensity, high-impact interventions [43]. The ability to intervene precisely during an acute risk for suicide could dramatically reduce the loss of life. To realize this goal, we need accurate and timely methods that predict when interventions are most needed. Federated learning is particularly useful in monitoring (with participants' permission) mobile data to assess mental health and provide early interventions. Our data collection, experimental study, and computational approaches provide a step towards data intensive longitudinal monitoring of human behavior. However, one must take care to summarize behaviors from mobile data without identifying the user through personal (e.g., personally identifiable information) or protected attributes (e.g., race, gender). This form of anonymity is critical when implementing these suicide detection technologies in real-world scenarios. Our goal is to be highly predictive of STBs while remaining as privacy-preserving as possible. We outline some of the potential privacy and security concerns below and show some possibilities brought about from the flexibility of our local models.

Privacy: There are privacy risks associated with making predictions from mobile data. Although federated learning only keeps data private on each device without sending it to other locations, the presence of one's data during distributed model training will likely affect model predictions. Therefore it is crucial to obtain user consent before collecting device data. In our experiments with real-world mobile data, all participants have given consent for their mobile device data to be collected and shared with us for research purposes. All data was anonymized and stripped of all personal (e.g., personally identifiable information) and protected attributes (e.g., race, gender).

Security: Communicating model updates throughout the training process could possibly reveal sensitive information, either to a third-party, or to the central server. Federated learning is also particularly sensitive to external security attacks from adversaries [36]. Recent methods to increase the security of federated learning systems come at the cost of reduced performance or efficiency. We believe that our proposed local-global models make federated learning more interpretable and flexible since local models can be appropriately adjusted to be more secure. However, there is a lot more work to be done in these directions, starting by accurately quantifying the trade-offs between security, privacy and performance in federated

learning [29].

Social biases: We acknowledge that there is a risk of exposure bias due to imbalanced datasets, especially when personal mobile data is involved. Models trained on biased data have been shown to amplify the underlying social biases especially when they correlate with the prediction targets [34]. Our experiment showcased one example of maintaining fairness via adversarial training, but leaves room for future work in exploring other methods tailored for specific scenarios (e.g. debiasing words [4], sentences [31], and images [46]). These methods can be easily applied to local representations before input into the global model. Future research should also focus on quantifying the trade-offs between bias and performance [58].

Overall, we believe that our proposed approach can help quantify the tradeoffs between local and global models regarding performance, communication, privacy, security, and fairness. Its flexibility also offers several exciting directions of future work in ensuring privacy and fairness of local representations. We hope that this brings about future opportunities for large-scale real-time analytics in healthcare and transportation using federated learning.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *ICCV*, 2015.

[2] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2):423?443, February 2019.

[3] Tal Ben-Nun and Torsten Hoefler. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *CoRR*, abs/1802.09941, 2018.

[4] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *NIPS*, 2016.

[5] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konecný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. *CoRR*, abs/1902.01046, 2019.

[6] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[7] Sebastian Caldas, Jakub Konecný, H. Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *CoRR*, abs/1812.07210, 2018.

[8] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konecný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. *CoRR*, abs/1812.01097, 2018.

[9] L. Elisa Celis and Vijay Keswani. Improved adversarial learning for fair classification. *CoRR*, abs/1901.10443, 2019.

[10] Irene Chen, Fredrik D Johansson, and David Sontag. Why is my classifier discriminatory? In *NIPS*. 2018.

[11] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2285–2294, Lille, France, 07–09 Jul 2015. PMLR.

[12] Ilias Diakonikolas, Elena Grigorescu, Jerry Li, Abhiram Natarajan, Krzysztof Onak, and Ludwig Schmidt. Communication-efficient distributed learning of discrete distributions. In *NIPS*. 2017.

[13] Cynthia Dwork. Differential privacy. In *ICALP*, 2006.

[14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014.

[15] Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. *CoRR*, abs/1808.06640, 2018.

[16] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ?04, page 109?117, New York, NY, USA, 2004. Association for Computing Machinery.

[17] Rui Feng, Yang Yang, Yuehan Lyu, Chenhao Tan, Yizhou Sun, and Chunping Wang. Learning fair representations via an adversarial framework. *CoRR*, abs/1904.13341, 2019.

[18] Joseph C Franklin, Jessica D Ribeiro, Kathryn R Fox, Kate H Bentley, Evan M Kleiman, Xieyining Huang, Katherine M Musacchio, Adam C Jaroszewski, Bernard P Chang, and Matthew K Nock. Risk factors for suicidal thoughts and behaviors: a meta-analysis of 50 years of research. *Psychological bulletin*, 143(2):187, 2017.

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

[20] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *CoRR*, abs/1902.11175, 2019.

[21] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[24] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 202–207. AAAI Press, 1996.

[25] Anders Krogh and John A Hertz. Generalization in a linear perceptron in the presence of noise. *Journal of Physics A: Mathematical and General*, 25(5):1135, 1992.

[26] Matthew Michael Large, Daniel Thomas Chung, Michael Davidson, Mark Weiser, and Christopher James Ryan. In-patient suicide: selection of people at risk, failure of protection and the possibility of causation. *BJPsych open*, 3(3):102–105, 2017.

[27] Roy L. Lassiter. The association of income and education for males by region, race, and age. *Southern Economic Journal*, 32(1):15–22, 1965.

[28] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[29] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*, 2019.

[30] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *CoRR*, abs/1812.06127, 2018.

[31] Paul Pu Liang, Irene Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. Towards debiasing sentence representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2020.

[32] Paul Pu Liang, Yao Chong Lim, Yao-Hung Hubert Tsai, Ruslan Salakhutdinov, and Louis-Philippe Morency. Strong and simple baselines for multimodal utterance embeddings. In *Proceedings of NAACL*. Association for Computational Linguistics, 2019.

[33] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. 2018.

[34] Kirsten Lloyd. Bias amplification in artificial intelligence systems. *CoRR*, abs/1809.07842, 2018.

[35] Gilles Louppe, Michael Kagan, and Kyle Cranmer. Learning to pivot with adversarial networks. In *NIPS*. 2017.

[36] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

[37] Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[38] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2016.

[39] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. *arXiv preprint arXiv:1902.06015*, 2019.

[40] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

[41] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. *CoRR*, abs/1902.00146, 2019.

[42] Douglas C. Montgomery, Elizabeth A. Peck, and Geoffrey G. Vining. *Introduction to Linear Regression Analysis (4th ed.)*. Wiley & Sons, 2006.

[43] Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 52(6):446–462, 2018.

[44] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A performance evaluation of federated learning algorithms. In *DIDL*, 2018.

[45] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *CoRR*, abs/1603.09246, 2016.

[46] Jahna Otterbacher, Alessandro Checco, Gianluca Demartini, and Paul Clough. Investigating user perception of gender bias in image search: The role of sexism. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR ?18, page 933?936, New York, NY, USA, 2018. Association for Computing Machinery.

[47] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *CoRR*, abs/1802.05668, 2018.

[48] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, 2018.

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[50] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS?17, page 4427?4437, Red Hook, NY, USA, 2017. Curran Associates Inc.

[51] John A. Stankovic, Tu Le, Abdeltawab M. Hendawi, and Yuan Tian. Hardware/software security patches for internet of trillions of things. *CoRR*, abs/1903.05266, 2019.

[52] Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. Distributed mean estimation with limited communication. In *ICML*, 2017.

[53] Yusuke Tsuzuku, Hiroto Imachi, and Takuya Akiba. Variance-based gradient compression for efficient distributed deep learning. *CoRR*, abs/1802.06058, 2018.

[54] Jialei Wang, Mladen Kolar, Nathan Srebro, and Tong Zhang. Efficient distributed learning with sparsity. In *ICML*, 2017.

[55] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *CoRR*, abs/1808.07576, 2018.

[56] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Adversarial removal of gender from deep image representations. *CoRR*, abs/1811.08489, 2018.

[57] Richard Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. Learning fair representations. In *ICML*, 2013.

[58] Han Zhao and Geoffrey J. Gordon. Inherent tradeoffs in learning fair representation. *CoRR*, abs/1906.08386, 2019.

[59] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 629–634, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[60] Shen-Yi Zhao, Hao Gao, and Wu-Jun Li. Quantized epoch-sgd for communication-efficient distributed learning. *CoRR*, abs/1901.03040, 2019.

[61] Liu Ziyin, Blair Chen, Ru Wang, Paul Pu Liang, Ruslan Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Learning not to learn in the presence of noisy labels, 2020.

Appendix

A Theoretical Analysis

In this section, we restate our main results and provide details proofs for them. We start with the short comment that, for the federated linear regression problem we are solving, gradient descent converges to the same solution as the commonly used ridgeless linear regression solution, as is shown in [21]. This means that $\hat{\mathbf{u}}$ converges to $\mathbf{u}$ and $\hat{\mathbf{v}}$ converges to $\mathbf{v}$ in expectation. We also assume in our analysis that $\frac{d}{N}$ is small and can be summarized in the big-O notation; however, this does not mean that the term $\frac{d}{N}\sigma^2$ is small because the noise present in the dataset might be a function of N. For example, in the well-studied label noise literature, the noise rate σ^2 is often proportional to N, cancelling out the N term in the denominator [61]. In practice, this happens when the dataset has a fixed probability of wrong labels.

Theorem 1. *The generalization loss for federated learning can be decomposed as*

$$\mathcal{E} = \mathbb{E}_{\mathbf{x},\mathbf{r}_m,\epsilon}[\hat{\mathcal{E}}_m] = Var[\hat{f}] + b^2 \tag{8}$$

where $Var[\hat{f}] = \mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[Var_\epsilon[\hat{f}|\mathbf{x},\mathbf{r}_m]\right]$ *is the variance, and* $b^2 = \mathbb{E}\left[\left(f_{\mathbf{u}_m} - \mathbb{E}_\epsilon f(\hat{\mathbf{v}},\hat{\mathbf{u}}_m)\right)^2\right]$ *is the bias.*

Proof. The *generalization error* can be derived by taking expectation over the random variables:

$$\mathcal{E} = \mathbb{E}_{\mathbf{x},\mathbf{r}_m,\epsilon}[\hat{\mathcal{E}}_m] = \mathbb{E}_{\mathbf{x},\mathbf{r}_m,\epsilon}\left[\left(f(\mathbf{x};\hat{\mathbf{v}},\hat{\mathbf{u}}_m) - \tilde{f}_{\mathbf{u}_m}(\mathbf{x})\right)^2\right]. \tag{9}$$

which can further be manipulated to obtain a *bias-variance decomposition* for federated learning:

$$\begin{aligned}
\mathcal{E} &= \mathbb{E}_{\mathbf{x},\mathbf{r}_m,\epsilon}[\hat{\mathcal{E}}_m] && (10)\\
&= \mathbb{E}_{\mathbf{x},\mathbf{r}_m,\epsilon}\left[\left(f(\mathbf{x};\hat{\mathbf{v}},\hat{\mathbf{u}}_m) - f_{\mathbf{u}_m}(\mathbf{x})\right)^2\right] && (11)\\
&= \mathbb{E}\left[\left(f(\hat{\mathbf{v}},\hat{\mathbf{u}}_m) - \mathbb{E}f(\hat{\mathbf{v}},\hat{\mathbf{u}}_m) - [f_{\mathbf{u}_m} - \mathbb{E}f(\hat{\mathbf{v}},\hat{\mathbf{u}}_m)]\right)^2\right] && (12)\\
&= \mathbb{E}\left[\left(f(\hat{\mathbf{v}},\hat{\mathbf{u}}_m) - \mathbb{E}f(\hat{\mathbf{v}},\hat{\mathbf{u}}_m)\right)^2\right] + \mathbb{E}\left[\left(f_{\mathbf{u}_m} - \mathbb{E}f(\hat{\mathbf{v}},\hat{\mathbf{u}}_m)\right)^2\right] && (13)\\
&= \underbrace{\mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[\mathrm{Var}_\epsilon[\hat{f}|\mathbf{x},\mathbf{r}_m]\right]}_{\mathrm{Var}[\hat{f}]\text{: variance of model}} + \underbrace{\mathbb{E}\left[\left(f_{\mathbf{u}_m} - \mathbb{E}_\epsilon f(\hat{\mathbf{v}},\hat{\mathbf{u}}_m)\right)^2\right]}_{b^2\text{: bias of model}} + \underbrace{0}_{\text{cross term}} && (14)
\end{aligned}$$

□

where we have omitted $\mathbf{x}_i$ in the input to $f(\cdot)$ for notational conciseness. Using only local models results in an unbiased estimator of $\mathbf{u}_m$. The bias term arises when learning global model parameters since federated learning couples the estimation of both local and global parameters. The variance term comes from both the variance of both local and global parameter estimates.

As a simplified version, LG-FedAvg can be seen as a ensemble of local and global models, i.e. $f(\mathbf{x};\hat{\mathbf{v}},\hat{\mathbf{u}}_m) = \alpha f_{\hat{\mathbf{u}}_m}(\mathbf{x}) + (1-\alpha)f_{\hat{\mathbf{v}}}(\mathbf{x})$. In this case, one can show that:

Proposition 0. *Let* $\mathbb{E}_{\mathbf{r}_m,\epsilon}[f_{\hat{\mathbf{v}}}] = f_{\mathbf{v}}$, $\mathbb{E}_\epsilon[f_{\hat{\mathbf{u}}_m}] = f_{\mathbf{u}_m}$, *and let* $f(\mathbf{x};\hat{\mathbf{v}},\hat{\mathbf{u}}_m) = \alpha f_{\hat{\mathbf{u}}_m}(\mathbf{x}) + (1-\alpha)f_{\hat{\mathbf{v}}}(\mathbf{x})$, *then equation 8 can be written as*

$$\mathcal{E} = (1-\alpha)^2\delta^2 + \mathrm{Var}[\hat{f}] \tag{15}$$

where $\delta^2 = \mathbb{E}_{\mathbf{x},\mathbf{r}_m}[(f_{\mathbf{u}} - \mathbb{E}_\epsilon[f_{\mathbf{v}}])^2|\mathbf{r}_m]]$ *measures the discrepancy between the local features and the global features, and thus measures the local variations across devices.*

Proof. We plug in to get

$$\mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[\left(f_{\mathbf{u}_m} - \mathbb{E}_\epsilon f(\hat{\mathbf{v}}, \hat{\mathbf{u}}_m)\right)^2\right] = \mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[\left(f_{\mathbf{u}_m} - \alpha f_{\mathbf{u}_m} - \mathbb{E}_\epsilon[(1-\alpha)f_{\hat{\mathbf{v}}}]\right)^2\right] \tag{16}$$

$$= (1-\alpha)^2 \mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[(f_{\mathbf{u}_m} - \mathbb{E}_\epsilon[f_{\hat{\mathbf{v}}}|\mathbf{r}_m])^2\right] \tag{17}$$

□

When using linear models, we can further expand this result as follows:

Corollary 0. *Let* $f_{\hat{\mathbf{v}}}(\mathbf{x}) = \hat{\mathbf{v}}^\top \mathbf{x}$, $f_{\hat{\mathbf{u}}_m}(\mathbf{x}) = \hat{\mathbf{u}_m}^\top \mathbf{x}$ *be learned through gradient descent algorithm, then* $\delta^2 = (\frac{M-1}{M})\rho^2$, *and* $\mathrm{Var}[\hat{f}]$ *can be expanded as follows:*

$$\mathcal{E} = (1-\alpha)^2 \left(\frac{M-1}{M}\right)\rho^2 + (1-\alpha)^2 \,\mathrm{Var}[f_{\hat{\mathbf{v}}}] + \alpha^2 \,\mathrm{Var}[f_{\hat{\mathbf{u}}}] + 2\alpha(1-\alpha)\,\mathrm{Cov}[f_{\hat{\mathbf{v}}}, f_{\hat{\mathbf{u}}}] \tag{18}$$

Proof. We first show that

$$\mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[(f_{\mathbf{u}_m} - \mathbb{E}_\epsilon[f_{\hat{\mathbf{v}}}|\mathbf{r}_m])^2\right] = \left(\frac{M-1}{M}\right)\rho^2.$$

We expand to get:

$$\mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[(f_{\mathbf{u}_m} - \mathbb{E}_\epsilon[f_{\hat{\mathbf{v}}}|\mathbf{r}_m])^2\right] = \mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[(\mathbf{u}_m^\top \mathbf{x} - \mathbb{E}_\epsilon[\hat{\mathbf{v}}^\top \mathbf{x}|\mathbf{r}_m])^2\right] \tag{19}$$

$$= \mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[\left(\mathbf{u}_m^\top \mathbf{x} - \mathbb{E}_\epsilon[\frac{1}{M}\sum_{j=1}^{M} \hat{\mathbf{u}}_j^\top \mathbf{x}|\mathbf{r}_m]\right)^2\right] \tag{20}$$

$$= \mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[\left(\mathbf{u}_m^\top \mathbf{x} - \frac{1}{M}\sum_{j=1}^{M} \mathbf{u}_j^\top \mathbf{x}\right)^2\right] \tag{21}$$

$$= \mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[\frac{1}{M^2}\left(\sum_{j\neq m}^{M} \mathbf{u}_m^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right)^2\right] \tag{22}$$

$$= \mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[\frac{1}{M^2}\left(\sum_{j\neq m}^{M} \mathbf{r}_m^\top \mathbf{x} - \mathbf{r}_j^\top \mathbf{x}\right)^2\right] \tag{23}$$

$$= \mathbb{E}_{\mathbf{x},\mathbf{r}_m}\left[\frac{1}{M^2}\left(\sum_{j\neq m}^{M} \mathbf{r}_m^\top \mathbf{x} - \mathbf{r}_j^\top \mathbf{x}\right)^2\right] \tag{24}$$

$$= \mathrm{Tr}\left\{\left[\frac{1}{M^2}\mathbb{E}_{\mathbf{r}_m}\left(\sum_{j\neq m}^{M} \mathbf{r}_m^\top - \mathbf{r}_j^\top\right)^2\right]\mathbb{E}_{\mathbf{x}}[\mathbf{x}\mathbf{x}^\top]\right\} \tag{25}$$

$$= \frac{M(M-1)}{M^2}\rho^2 \tag{26}$$

$$= \left(\frac{M-1}{M}\right)\rho^2 \tag{27}$$

where we have used the fact that the expectation over squared $\mathbf{x}$, $\mathbb{E}_{\mathbf{x}}[\mathbf{x}\mathbf{x}^\top]$, is the identity matrix.

Furthermore, by using the fact that $f(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) = (1-\alpha)f_{\hat{\mathbf{u}}_m}(\mathbf{x}) + \alpha f_{\hat{\mathbf{v}}}(\mathbf{x})$, we have that

$$\mathrm{Var}[\hat{f}] = (1-\alpha)^2 \,\mathrm{Var}[f_{\hat{\mathbf{v}}}] + \alpha^2 \,\mathrm{Var}[f_{\hat{\mathbf{u}}}] + 2\alpha(1-\alpha)\,\mathrm{Cov}[f_{\hat{\mathbf{v}}}, f_{\hat{\mathbf{u}}}] \tag{28}$$

□

Using these results, we can compute the generalization error of several federated learning baselines as well as LG-FEDAVG.

A.1 Analysis of Federated Learning Baselines

We begin by analyzing two baselines for federated learning.

The first method learns local models on each device [50]: $f_\ell(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) = f(\mathbf{x}; \hat{\mathbf{u}}_m) = \hat{\mathbf{u}}_m^\top \mathbf{x}$.
Proposition 1. *The generalization error $\mathcal{E}(f_\ell)$ of local model $f_\ell(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m)$ is $\frac{d}{N}\sigma^2$.*

Proof. Similarly, set $\alpha = 1$ in equation 18 of corollary 0 and we obtain that $\mathcal{E}(f_\ell) = \text{Var}[f_{\hat{\mathbf{u}}}] = \frac{d}{N}\sigma^2$. □

This shows that local models only control data variance at a rate of d/N since they are only updated using local device data which may be limited in number and vary highly (both in quality and quantity) across devices. However, local models do not suffer from device variance.

The second method updates a joint global model [38], which is equivalent to setting $\alpha = 0$ in our previous analysis, i.e. $f_g(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) = f(\mathbf{x}; \hat{\mathbf{v}}) = \hat{\mathbf{v}}^\top \mathbf{x}$. We can compute its generalization error:
Proposition 2. *The generalization error $\mathcal{E}(f_g)$ of the global model $f_g(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m)$ is $\frac{M-1}{M}\rho^2 + \frac{d}{MN}\sigma^2$.*

Proof. Set $\alpha = 0$ in equation 18 of corollary 0, we obtain $\mathcal{E}(f_g) = \frac{M-1}{M}\rho^2 + \text{Var}[f_{\hat{\mathbf{v}}}] = \frac{M-1}{M}\rho^2 + \frac{d}{MN}\sigma^2$. □

Therefore, global models are able to control for data variance (σ^2) at a rate of $d/(MN)$, decreasing with the total number of datapoints across *all* devices (since global parameters are updated using data across all devices), which is better than the rate for local models. However, it suffers from an extra $O(\rho^2)$ term representing device variance so one global model is unable to account for very different devices.

A.2 Analysis of LG-FEDAVG

Given that the above baselines achieve different generalization errors, one should be able to interpolate between the two methods to find the optimal tradeoff point. Our method therefore defines an α–interpolation between the local and global models:

$$f_\alpha(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m) = \alpha f_\ell(\mathbf{x}; \hat{\mathbf{u}}_m) + (1-\alpha) f_g(\mathbf{x}; \hat{\mathbf{v}}). \tag{29}$$

where $\alpha \in [0, 1]$. The following theorem gives the generalization of this model.
Theorem 2. *The generalization error of $f_\alpha(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m)$ is $\alpha^2 \frac{d}{N}\sigma^2 + (1-\alpha)^2 \frac{M-1}{M}\rho^2 + (1-\alpha^2)\frac{d}{MN}\sigma^2$.*

Proof. The proof follows by computing each term in equation 18 of corollary 0

$$\mathcal{E}(f_\alpha) = (1-\alpha)^2 \left(\frac{M-1}{M}\right)\rho^2 + (1-\alpha)^2 \,\text{Var}[f_{\hat{\mathbf{v}}}] + \alpha^2 \,\text{Var}[f_{\hat{\mathbf{u}}}] + 2\alpha(1-\alpha)\,\text{Cov}[f_{\hat{\mathbf{v}}}, f_{\hat{\mathbf{u}}}] \tag{30}$$

We need to find $\text{Cov}[f_{\hat{\mathbf{v}}}, f_{\hat{\mathbf{u}}}]$:

$$\text{Cov}[f_{\hat{\mathbf{v}}}, f_{\hat{\mathbf{u}}}] = \mathbb{E}\left[(f_{\hat{\mathbf{v}}} - \mathbb{E}f_{\hat{\mathbf{v}}})(f_{\hat{\mathbf{u}}} - \mathbb{E}f_{\hat{\mathbf{u}}})\right] \tag{31}$$

$$= \mathbb{E}\left[\left(\frac{1}{M}\sum_{j=1}^{M}\hat{\mathbf{u}}_j - \mathbf{v}\right)^{\top}(\hat{\mathbf{u}}_m - \mathbf{u}_m)\right] \tag{32}$$

$$= \mathbb{E}\left[\frac{1}{M}\sum_{j=1}^{M}\hat{\mathbf{r}}_j^{\top}\hat{\mathbf{r}}_m\right] \tag{33}$$

$$= \frac{d}{MN}\sigma^2 \tag{34}$$

where, as in the previous theorem, the expectation over squared $\mathbf{x}$, $\mathbb{E}_{\mathbf{x}}[\mathbf{x}\mathbf{x}^{\top}]$, is the identity matrix. Likewise, we obtain $\text{Var}[f_{\hat{\mathbf{u}}}] = \frac{d}{N}\sigma^2$, and $\text{Var}[f_{\hat{\mathbf{v}}}] = \frac{d}{MN}\sigma^2$. Combining everything above, we get that the generalization error is

$$\mathcal{E}(f_\alpha) = \alpha^2\frac{d}{N}\sigma^2 + (1-\alpha)^2\frac{M-1}{M}\rho^2 + (1-\alpha)^2\frac{d}{MN}\sigma^2 + 2\alpha(1-\alpha)\frac{d}{MN}\sigma^2 \tag{35}$$

$$= \alpha^2\frac{d}{N}\sigma^2 + (1-\alpha)^2\frac{M-1}{M}\rho^2 + (1-\alpha^2)\frac{d}{MN}\sigma^2 \tag{36}$$

□

This can be solved to find the optimal α^* that minimizes $f_\alpha(\mathbf{x}; \hat{\mathbf{v}}, \hat{\mathbf{u}}_m)$.

Corollary 1. *The optimal α^* that minimizes $\mathcal{E}(f_\alpha)$ is*

$$\alpha^* = \frac{\rho^2}{\rho^2 + \frac{d}{N}\sigma^2}. \tag{37}$$

Moreover, when $\sigma^2, \rho^2 \neq 0$, we have that $\mathcal{E}(f_{\alpha^*}) < \mathcal{E}(f_\ell)$ and $\mathcal{E}(f_{\alpha^*}) < \mathcal{E}(f_g)$.

Proof. Taking derivative w.r.t to α, and setting to 0 gives an exact expression for α^*:

$$\alpha^* = \frac{\frac{M-1}{M}\rho^2}{\frac{M-1}{M}\rho^2 + \frac{M-1}{M}\frac{d}{N}\sigma^2} = \frac{\rho^2}{\rho^2 + \frac{d}{N}\sigma^2}. \tag{38}$$

□

This shows that using an ensemble of local and global models reduces both data variance and device variance. When ρ^2 is large (high device variance), one should lean towards using local models that better model the local data distributions (larger α^*). Conversely, when σ^2 is large (high data variance), one should lean towards using a global model whose parameters are updated using more data across all devices (smaller α^*).

B Fair Representation Learning

B.1 Fair Training of Local Models

In this section we detail one extension of local representation learning to remove information that might be indicative of protected attributes. The data on each device is now a triple $(\mathbf{X}_m, \mathbf{Y}_m, \mathbf{P}_m)$ drawn non-i.i.d. from a joint distribution $p(X, Y, P)$ where $\mathbf{p} \in \mathcal{P}$ are some protected attributes which the model should not

predict. For example, although there exist correlations between race and income [27] which could help in income prediction [10], it would be undesirable for our models to rely on these correlations since these would exacerbate racial biases.

To learn fair local representations, we use adversarial training [35] to remove protected attributes (Figure 1 (d)). More formally, we aim to learn a local model ℓ_m such that the distribution of $\ell_m(\mathbf{x}; \theta^\ell_m)$ conditional on $\mathbf{h}$ is invariant with respect to protected attributes $\mathbf{p}$:

$$p(\ell_m(\mathbf{x}; \theta^\ell_m) = \mathbf{h}|\mathbf{p}) = p(\ell_m(\mathbf{x}; \theta^\ell_m) = \mathbf{h}|\mathbf{p}') \tag{39}$$

for all $\mathbf{p}, \mathbf{p}' \in \mathcal{P}$ and outputs $\mathbf{h} \in \mathcal{H}$ of $\ell_m(\,\cdot\,; \theta^\ell_m)$, thereby implying that $\ell_m(\mathbf{x}; \theta^\ell_m)$ and $\mathbf{p}$ are independent. [35] showed that we can use adversarial networks in order to constrain model ℓ_m to satisfy Equation (39). ℓ_m is pit against an auxiliary adversarial model $a_m = p_{\theta^a_m}(\mathbf{p}|f(\mathbf{x}; \theta^\ell_m) = \mathbf{h})$ with parameters θ^a_m and loss $\mathcal{L}^a_m(\theta^\ell_m, \theta^a_m)$. The adversarial network a_m is trained to predict $\mathbf{p}$ as much as possible given local representations $\mathbf{h}$. If $p(\ell_m(\mathbf{x}; \theta^\ell_m) = \mathbf{h}|\mathbf{p})$ varies with $\mathbf{p}$, then the corresponding correlation can be captured by adversary a_m. On the other hand, if $p(\ell_m(\mathbf{x}; \theta^\ell_m) = \mathbf{h}|\mathbf{p})$ is indeed invariant with respect to $\mathbf{p}$, then adversary a_m should perform randomly. Therefore, we train ℓ_m to both minimize the global prediction loss $\mathcal{L}^g_m(\theta^\ell_m, \theta^a_m)$ and to maximize the adversarial loss $\mathcal{L}^a_m(\theta^\ell_m, \theta^a_m)$. In practice, the local model ℓ_m, (local copy of the) global model g, and adversarial model a_m are updated by solving for the minimax solution:

$$\hat{\theta}^\ell_m, \hat{\theta}^g_m, \hat{\theta}^a_m = \arg \min_{\{\theta^\ell_m, \theta^g_m\}} \max_{\theta^a_m} \left[\mathcal{L}^g_m(\theta^\ell_m, \theta^g_m) - \mathcal{L}^a_m(\theta^\ell_m, \theta^a_m)\right]. \tag{40}$$

$\mathcal{L}^g_m$ and $\mathcal{L}^a_m$ are computed using the expected value of the log-likelihood through inference networks ℓ_m, g, and a_m. We optimize equation (40) by treating it as a coordinate descent problem and alternately solving for $\hat{\theta}^\ell_m, \hat{\theta}^g_m, \hat{\theta}^a_m$ using gradient-based methods (details in Appendix A). Proposition 3 in Appendix A further shows that adversarial training learns an optimal local model ℓ_m that is invariant with respect to $\mathbf{p}$ under local device data distribution $p(X_m, Y_m, P_m)$. To the best of our knowledge, we are the first to extend this analysis, both theoretically and empirically, to the federated learning setting. Key to this analysis is the separation of local and global models which allows learning of *fair intermediate representations* $\mathbf{h}$.

In Figure 1, we also illustrate several other choices of local representation learning using auxiliary local models a_m for (b) unsupervised autoencoding training, where a_m reconstructs $\mathbf{x}$ given $\mathbf{h}$, and (c) self-supervised learning (e.g. jigsaw solving [45]), where a_m predicts auxiliary features $\mathbf{z}$ given $\mathbf{h}$. However, when using local supervised learning, a_m and g are similar classification branches (Figure 1 (a)) both supervised by the target labels. LG-FedAvg can therefore be trained without a_m to directly learn local representations $\mathbf{h}$ for the global model g to make predictions (equation 1). Thus, LG-FedAvg for supervised learning does not incur additional computational complexity while reducing communicated global parameters.

B.2 Theoretical Analysis of Local Fair Representation Learning

In this section we provide some details of the theoretical proofs and implementation of fair representation learning in our federated learning framework. The setting is adapted from [35]. First recall our dual objective across the local model ℓ_m, (local copy of the) global model g and auxiliary adversarial model a_m:

$$E(\theta^\ell_m, \theta^g_m, \theta^a_m) = \mathcal{L}^g_m(\theta^\ell_m, \theta^g_m) - \mathcal{L}^a_m(\theta^\ell_m, \theta^a_m). \tag{41}$$

This implies that the global models should be trained in an adversarial manner since the path of inference when training the (local copy) of the global model also involves the local representation $\mathbf{h}$ and protected attributes $\mathbf{p}$ (refer to Figure 3). When training the global model we optimize for the dual objective across

the local model ℓ_m, global model g, and adversarial model a_m by finding the minimax solution $\hat{\theta}^{\ell}_m, \hat{\theta}^{g}_m, \hat{\theta}^{g}_m$, defined as

$$\hat{\theta}^{\ell}_m, \hat{\theta}^{g}_m, \hat{\theta}^{a}_m = \arg \min_{\{\theta^{\ell}_m, \theta^{g}_m\}} \max_{\theta^{a}_m} E(\theta^{\ell}_m, \theta^{g}_m, \theta^{a}_m). \tag{42}$$

To do so, we can iteratively solve for $\hat{\theta}^{\ell}_m, \hat{\theta}^{g}_m, \hat{\theta}^{a}_m$ in an alternating fashion. In other words, initialize $\hat{\theta}^{\ell(0)}_m, \hat{\theta}^{g(0)}_m, \hat{\theta}^{a(0)}_m$ and repeat until convergence:

$$\hat{\theta}^{\ell(t+1)}_m = \arg \min_{\theta^{\ell}_m} E(\theta^{\ell}_m, \theta^{g(t)}_m, \theta^{a(t)}_m), \tag{43}$$

$$\hat{\theta}^{g(t+1)}_m = \arg \min_{\theta^{g}_m} E(\theta^{\ell(t+1)}_m, \theta^{g}_m, \theta^{a(t)}_m), \tag{44}$$

$$\hat{\theta}^{a(t+1)}_m = \arg \max_{\theta^{a}_m} E(\theta^{\ell(t+1)}_m, \theta^{g(t+1)}_m, \theta^{a}_m). \tag{45}$$

$\mathcal{L}^{g}_m$ and $\mathcal{L}^{a}_m$ are computed using the expected value of the log likelihood through the inference networks ℓ_m, g, and a_m and the optimization procedure involves using gradient descent and iteratively solving for $\hat{\theta}^{\ell}_m, \hat{\theta}^{g}_m, \hat{\theta}^{a}_m$ until convergence. Suppose we define the local data distribution $p(X_m, Y_m, P_m)$, then Proposition 3 shows that this adversarial training procedure learns an optimal local model ℓ_m that is at the same time pivotal (invariant) with respect to $\mathbf{p}$ under $p(X_m, Y_m, P_m)$.

Proposition 3 (Optimality of ℓ_m, adapted from Proposition 1 in [35])**.** *Suppose we compute losses $\mathcal{L}^{g}_m$ and $\mathcal{L}^{a}_m$ using the expected log likelihood through the inference networks ℓ_m, g, and a_m,*

$$\mathcal{L}^{g}_m(\theta^{\ell}_m, \theta^{g}_m) = \mathbb{E}_{\mathbf{x}\sim X_m}\mathbb{E}_{\mathbf{y}\sim Y_m|\mathbf{x}}\left[-\log\left(\sum_{\mathbf{h}} p_{\theta^{g}_m}(\mathbf{y}|\mathbf{h})\, p_{\theta^{\ell}_m}(\mathbf{h}|\mathbf{x})\right)\right], \tag{46}$$

$$\mathcal{L}^{a}_m(\theta^{\ell}_m, \theta^{a}_m) = \mathbb{E}_{\mathbf{h}\sim f(X_m;\theta^{\ell}_m)}\mathbb{E}_{\mathbf{p}\sim P_m|\mathbf{h}}[-\log p_{\theta^{a}_m}(\mathbf{p}|\mathbf{h})]. \tag{47}$$

Then, if there is a minimax solution $(\hat{\theta}^{\ell}_m, \hat{\theta}^{g}_m, \hat{\theta}^{a}_m)$ for equation (42) such that $E(\hat{\theta}^{\ell}_m, \hat{\theta}^{g}_m, \hat{\theta}^{a}_m) = H(Y_m|X_m) - H(P_m)$, then $\ell_m(\,\cdot\,; \hat{\theta}^{\ell}_m)$ is both an optimal classifier and a pivotal quantity.

Proof. For fixed θ^{ℓ}_m, the adversary a_m is optimal at

$$\hat{\theta}_{\mathbf{r}_m} = \arg \max_{\theta^{a}_m} E(\theta^{\ell}_m, \theta^{g}_m, \theta^{a}_m) = \arg \min_{\theta^{a}_m} \mathcal{L}^{a}_m(\theta^{\ell}_m, \theta^{a}_m), \tag{48}$$

in which case $p_{\hat{\theta}_{a_m}}(\mathbf{p}|f(X_m; \theta^{\ell}_m) = \mathbf{h}) = p(\mathbf{p}|\ell_m(X_m; \theta^{\ell}_m) = \mathbf{h})$ for all $\mathbf{p}$ and all $\mathbf{h}$, and $\mathcal{L}^{a}_m$ reduces to the expected entropy $\mathbb{E}_{\mathbf{h}\sim\ell_m(X_m;\theta^{\ell}_m)}\left[H\left(P_m|\ell_m(X_m; \theta^{\ell}_m) = \mathbf{h}\right)\right]$ of the conditional distribution of the protected variables $\mathbf{p}$.

This expectation corresponds to the conditional entropy of the random variables P_m and $f(X_m; \theta^{\ell}_m)$ and can be written as $H(P_m|\ell_m(X_m; \theta^{\ell}_m))$. Accordingly, the value function E can be restated as a function depending only on θ^{ℓ}_m and θ^{g}_m:

$$E'(\theta^{\ell}_m, \theta^{g}_m) = \mathcal{L}^{g}_m(\theta^{\ell}_m, \theta^{g}_m) - H(P_m|\ell_m(X_m; \theta^{\ell}_m)). \tag{49}$$

By our choice of the objective function we know that

$$\mathcal{L}^{g}_m(\theta^{\ell}_m, \theta^{g}_m) = \mathbb{E}_{\mathbf{x}\sim X_m}\mathbb{E}_{\mathbf{y}\sim Y_m|\mathbf{x}}[-\log p(\mathbf{y}|\mathbf{x})] \geq H(Y_m|X_m) \tag{50}$$

which implies that we have the lower bound

$$H(Y_m|X_m) - H(P_m) \leq \mathcal{L}^{g}_m(\theta^{\ell}_m, \theta^{g}_m) - H(P_m|\ell_m(X_m; \theta^{\ell}_m)) \tag{51}$$

where the equality holds at $\hat{\theta}^{\ell}_m, \hat{\theta}^{g}_m = \arg\min_{\{\theta^{\ell}_m, \theta^{g}_m\}} E'(\theta^{\ell}_m, \theta^{g}_m)$ when:

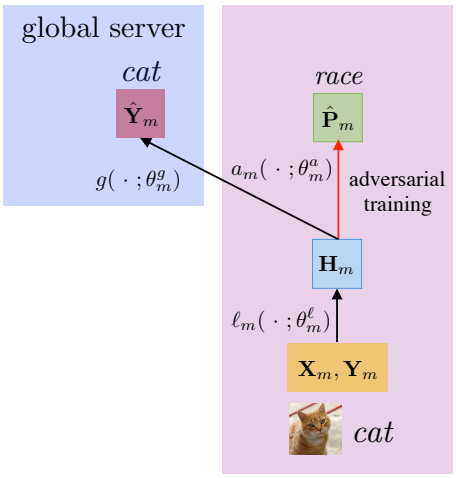

Figure 3: A closer look at the inference paths involved in adversarial training. The local models ℓ_m, (local copy of the) global model g and adversarial model a_m are trained jointly for the global prediction objective and adversarial objective. Refer to equation (42) for the dual optimization objective over local and global model and adversary parameters respectively.

1. $\mathbb{E}_{\mathbf{x}\sim X_m}\mathbb{E}_{\mathbf{y}\sim Y_m|\mathbf{x}}[-\log p(\mathbf{y}|\mathbf{x})] \geq H(Y_m|X_m)$, which implies that $\hat{\theta}^\ell_m$ and $\hat{\theta}^g_m$ perfectly minimize the negative log-likelihood of $Y_m|X_m$ under ℓ_m, which happens when $\hat{\theta}^\ell_m$ and $\hat{\theta}^g_m$ are the parameters of an optimal classifier from X_m to Y_m (through an intermediate representation H). In this case, $\mathcal{L}^g_m$ reduces to its minimum value $H(Y_m|X_m)$.
2. $\hat{\theta}^\ell_m$ maximizes the conditional entropy $H(P_m|\ell_m(X_m;\theta^\ell_m))$, since $H(P_m|\ell_m(X_m;\theta^\ell_m)) \leq H(P_m)$ from the properties of entropy.

By assumption, the lower bound is active which implies that $H(P_m|\ell_m(X_m;\theta^\ell_m)) = H(P_m)$ because of the second condition. This in turn implies that P_m and $\ell_m(X_m;\theta^\ell_m)$ are independent variables by the properties of (conditional) entropy. Therefore, the optimal classifier $\ell_m(\cdot\,;\hat{\theta}^\ell_m)$ is also a pivotal quantity with respect to the protected attributes $\mathbf{p}$ under local data distribution $p(X_m, Y_m, P_m)$. □

In practice, we optimize for the following dual objectives over local models ℓ_m, (local copy of the) global model g, and adversarial model a_m respectively:

$$E(\theta^\ell_m, \theta^g_m, \theta^a_m) = \mathcal{L}^g_m(\theta^\ell_m, \theta^g_m) - \lambda\mathcal{L}^a_m(\theta^\ell_m, \theta^a_m). \tag{52}$$

where λ is a hyperparameter that controls the tradeoff between the prediction model and the adversary model.

C Experimental Details and Extra Results

Here we provide all the details regarding experimental setup, dataset preprocessing, model architectures, model training, and performance evaluation. Our anonymized code is attached in the supplementary material. All experiments are conducted on a single machine with 4 GeForce GTX TITAN X GPUs.

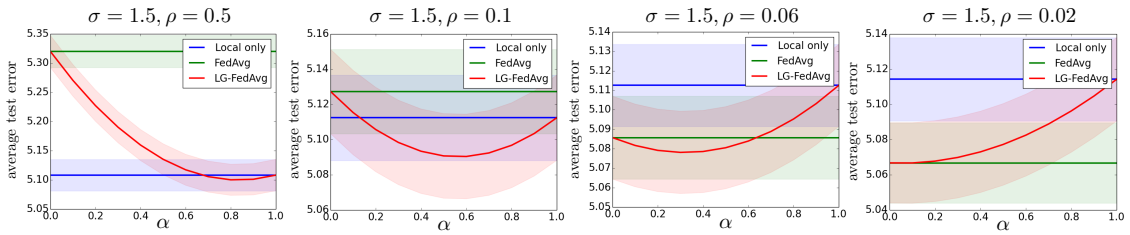

Figure 4: Average test error under four settings: 1) when local models perform close to optimal (far left, $\sigma = 1.5, \rho = 0.5$), 2) when local models perform better (middle left, $\sigma = 1.5, \rho = 0.1$), 3) when global models perform better (middle right, $\sigma = 1.5, \rho = 0.06$), and 4) when global models perform close to optimal (far right, $\sigma = 1.5, \rho = 0.02$). For all settings, using an α-interpolation of both local and global models performs either close to the optimal extremes (cases 1 and 4) or better than either extremes (cases 2 and 3).

A note on hyperparameters used: For MNIST and CIFAR experiments, we would like to emphasize that our initial set of hyperparameters were taken directly from the default set of hyperparamters in https://github.com/shaoxiongji/federated-learning for fair comparison across all baselines and our approach. They were **NOT** manually tuned for LG-FEDAVG to perform better.

For synthetic data, there are no hyperparamters involved.

For experiments on mobile data, since it is a new dataset, we start by training the best vanilla federated learning model using FEDAVG and use the exact same set of hyperparameters for LG-FEDAVG.

For experiments on fairness, we again use all the default hyperparameters as obtained from the tutorial https://blog.godatadriven.com/fairness-in-ml and associated code https://github.com/equialgo/fairness-in-ml.

C.1 Synthetic Data

We set $d = 20, M = 100$, number of train samples per device as 2000 and the number of test samples per device as 1000. Data on device m is generated by $\mathbf{x} \sim \mathcal{U}[-1.0, 1.0]$ and teacher weights $\mathbf{u}_m = \mathbf{v} + \mathbf{r}_m$ are sampled as $\mathbf{v} \sim \mathcal{U}[0.0, 1.0]$, $\mathbf{r}_m \sim \mathcal{N}(\mathbf{0}_d, \rho^2 \mathbf{I}_d)$, where ρ^2 represents device variance. Labels are observed with noise, $y = \mathbf{u}_m^\top \mathbf{x} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where σ^2 represents data variance. We plot the average test error when local models perform better due to higher device variance (Figure 2 left, $\sigma = 1.5, \rho = 0.1$) and when global models perform better due to lower device variance (Figure 2 right, $\sigma = 1.5, \rho = 0.06$). For both settings, using an interpolation of local and global models performs better than either extremes, which supports our analysis.

We also provide several other results demonstrating the effects of data and device variance on the performance local and/or global models. In particular, we fix data variance $\sigma = 1.5$ and gradually decrease device variance $\rho \in \{0.5, 0.1, 0.06, 0.02\}$. This results in 4 cases: 1) when local models perform close to optimal (Figure 4 far left, $\sigma = 1.5, \rho = 0.5$), 2) when local models perform better (Figure 4 middle left, $\sigma = 1.5, \rho = 0.1$), 3) when global models perform better (Figure 4 middle right, $\sigma = 1.5, \rho = 0.06$), and 4) when global models perform close to optimal (Figure 4 far right, $\sigma = 1.5, \rho = 0.02$). For all settings, using an α-interpolation of both local and global models performs either close to the optimal extremes (cases 1 and 4) or better than either extremes (cases 2 and 3).

C.2 Model Performance and Communication Efficiency

C.2.1 MNIST

Details: In all our experiments, we train with number of local epochs $E = 1$ and local minibatch size $B = 10$. We set $C = 0.1$. Images were normalized prior to training and testing. In our experiments, we take the last two layers to form our global model, reducing the number of parameters to 15.79% (99, 978/633, 226). Table 5 shows the of hyperparameters used. The dataset can be found here: http://yann.lecun.com/exdb/mnist/. We train LG-FEDAVG with global updates until we reach a goal accuracy (97.5% for MNIST) before training for additional rounds to jointly update local and global models. Our results are averaged over 10 runs. # FedAvg and LG Rounds are rounded to the nearest multiple of 5, which we use to calculate the number of parameters communicated. Standard deviations are also reported.

Extra results: In this section we provide results on MNIST in comparison with the baselines, see Table 6. Although MNIST is a slightly smaller dataset, we find that both local and global models help in maintaining performance while using fewer communication parameters.

C.2.2 CIFAR10

Details: We train with number of local epochs $E = 1$ and local minibatch size $B = 50$. We set $C = 0.1$. Images are randomly cropped to size 32, randomly flipped horizontally with probability $p = 0.5$, resized to 224×224, and normalized. For our model architecture, we chose Lenet-5. We use the two convolutional layers for the global model in our LG-FEDAVG method to minimize the number of parameters. We therefore reduce the number of parameters to 4.48% (2872/64102). Table 5 shows a table of additional hyperparameters used. The dataset can be found here: https://www.cs.toronto.edu/~kriz/cifar.html. We train LG-FEDAVG with global updates until we reach a goal accuracy (57% for CIFAR-10) before training for additional rounds to jointly update local and global models. Our results are averaged over 10 runs and we report standard deviations. # FedAvg and LG Rounds are rounded to the nearest multiple of 5, which we use to calculate the number of parameters communicated.

Extra results: In this section we provide more results and also show a sensitivity analysis to various hyperparameters especially regarding the data splits across devices and the local-global model split in our method. See Table 8. Our results are especially strong here: across different data splits (different number of users per device), LG-FEDAVG consistently performs better on local test and new test while using fewer parameters.

C.2.3 VQA

Details: We adapt the baseline model from [1] without *norm I* image channel embeddings. We also substitute the VGGNet [49] used in the original baseline model with a pre-trained ResNet-18 [22]. Finally we use the deep LSTM [23] embedding, which is an LSTM that consists of two hidden layers. For the LG-FEDAVG method, the global model uses the two final fully connected layers of the image and question channels, as well as the the additional two fully connected layers following the fusion via element-wise multiplication. The global model reduces the number of parameters to 9.53% (5149200/54042572). We use 50 devices and set number of local epochs $E = 1$, local minibatch size $B = 100$, fraction of devices sampled per round $C = 0.1$. To train and evaluate our models, we use the data from the following: https:

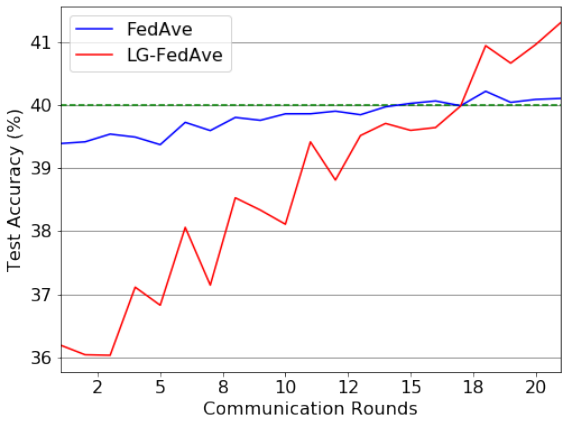

Figure 5: Test accuracy on VQA across 20 rounds (dotted green line marks the goal accuracy of 40% used in Table 2). LG-FEDAVG reaches an accuracy of 41.30% compared to 40.22% for FEDAVG while using only 9.53% of the parameters.

//visualqa.org/download.html. Table 10 shows a table of hyperparameters used, which strictly follows the baseline model from [1]. We first trained the best FEDAVG model and used the exact same hyperparamters to train LG-FEDAVG as well.

Extra results: In Figure 5, we plot the convergence of test accuracy across communication rounds. LG-FEDAVG outperforms FEDAVG after 20 rounds while requiring only 9.53% of parameters in FEDAVG and continues to improve.

C.2.4 Other Baselines

The MTL baseline [50] is implemented by training local models each with parameters $\mathbf{w}_m$ and adding a joint regularization term $\mathcal{R}(\mathbf{W}, \Omega) = \lambda_1 \text{tr}(\mathbf{W}\Omega\mathbf{W}^\top) + \lambda_2 \|\mathbf{W}\|_F^2$ where λ_1 and λ_2 are hyperparmeters, and $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_M]$ is a matrix whose m-th column is the weight vector for the m-th device. We choose $\Omega = (\mathbf{I}_M - \frac{1}{M}\mathbf{1}\mathbf{1}^\top)^2$ which implements mean-regularized multitask learning [16], which assumes that all the tasks form one cluster, and that the weight vectors are close to their mean. For CIFAR-10, computing the full $\mathbf{W}$ requires storing a matrix of size $p \times M$. Optimizing and storing this matrix becomes infeasible as the number of users or model size increases. Even for our experimental setting, where $M = 100$ and $p \approx 64,000$ (number of parameters in each local model), running MTL require the following relaxations. First, we reduce M to 10, reducing the size of $\mathbf{W}$ and therefore the number of parameters communicated per round.

C.3 Learning Personalized Mood Predictors from Mobile Data

Dataset details: We designed and collected a new dataset called Mobile Assessment for the Prediction of Suicide (MAPS). MAPS was designed to elucidate real-time indicators of suicide risk in adolescents ages 13 – 18 years. Current adolescent suicide ideators and recent suicide attempters along with aged-matched psychiatric controls with no lifetime suicidal thoughts and behaviors completed baseline clinical assessments (i.e., lifetime mental disorders, current psychiatric symptoms). Following the baseline clinical characterization, a smartphone app, the Effortless Assessment of Risk States (EARS), was installed onto adolescents' phones,

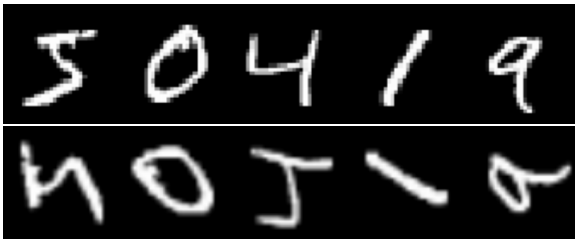

Figure 6: Sample MNIST images used for training (top) and their rotated counterparts used to test the impact of heterogeneous data on a trained federated model in an online setting (bottom).

and passive sensor data were acquired for 6-months. Notably, during EARS installation, a keyboard logger is configured on adolescents' phones, which then tracks all words typed into the phone as well as they app used during this period. Each day during the 6-month follow-up, participants also were asked to rate their mood on the previous day on a scale ranging from $1-100$, with higher scores indicating a better mood.

All users have given consent for their mobile device data to be collected and shared with us for research purposes.

MAPS is a realistic federated learning benchmark since it contains real-world data with privacy concerns and high device variance due to highly personalized use of mobile phones. We used a preliminary preprocessed version containing 572 samples across 14 participants. We discretize the scores into 5 bins for 5-way classification. We use a random 80/10/10 split for training/validation/testing, conduct all experiments 10 times, and report the average accuracy and standard deviation.

Model details: To assess how mobile text data can be used to make personalized mood predictions, we train a MLP classifier on top of a Bi-LSTM encoder. The Bi-LSTM has 128 hidden units and the MLP has two hidden layers, each with size 512. We conduct our experiments over 10 iterations. Within each iteration, we use a random 80/10/10 split for training/validation/testing. We train and validate our model 5 times on this split and select the model that performs best on the validation set. We use the test accuracy of this best-performing model as the test accuracy for the iteration. We report the average accuracy and standard deviation over all 10 iterations in Figure 2(d). In addition to local only and FEDAVG results, we plot the performance of LG-FEDAVG across different splits of local and global models (i.e. $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$). Consistent with our theoretical findings, an α-split across local and global models leverages both personalized representations per devices as well as statistical strength sharing through data across all devices, outperforming either local or global extremes.

C.4 Heterogeneous Data in an Online Setting

Details: Our experiments for the rotated MNIST follow the same settings and hyperparameter selection as our normal MNIST experiments (section C.2). However, we include an additional device, which randomly samples 3000 and 500 images from the train and test sets respectively and rotates them by a fixed 90 degrees. We show some samples of the rotated MNIST images we used in Figure 6, where the top row shows the normal MNIST images used during training and the bottom row shows the rotated MNIST images on the new test device.

C.5 Learning Fair Representations

Details: For method 1, FEDAVG, we train the global model and global adversary for 50 outer epochs, within which the number of local epochs $E = 10$. For methods 2 and 3 involving local models, we begin by pre-training the local models and local adversaries for 10 epochs before joint local and global training for 10 epochs. Table 11 shows the table of all hyperparameters used. Experiments were run 10 times with the same hyperparameters but different random seeds. We aimed to keep the local, global, and adversary models as similar as possible between the three baselines for fair comparison. Apart from the number of local and global epochs all hyperparameters were kept the same from the tutorial https://blog.godatadriven.com/fairness-in-ml and associated code https://github.com/equialgo/fairness-in-ml. The data can be found at https://archive.ics.uci.edu/ml/datasets/Adult.

D Discussion and Future Work

We believe that LG-FEDAVG is a general approach that offers several extensions for future work.

Firstly, combining LG-FEDAVG with existing work on compressing the number of parameters and gradient updates could further improve the efficiency of federated learning. For example, existing work in sparsifying the data and model [54], developing efficient gradient-based methods [55, 12], and compressing the updates [53, 60, 33] can all be applied to our local and global models. In particular, sparsifying the model through techniques such as distillation [47] and hashing [11] could help to store the local models on devices with small memory and computational power.

Secondly, depending on the test time scenario (i.e. local test vs new test), there is a trade off between the ideal size of local models and the global model. If we know which device the test data belongs to, then having a more accurate local model would allow us to perform better prediction at test time. However, if we do not know which device the test data belongs to, it is important to use a more accurate global model to learn the true data distribution across all devices. Therefore, another step for future work would be dynamically learn the number of layers spread across the local and the global models, in a manner similar to learning dynamic computation steps in neural networks [5]. Different devices which contain different data distributions could use different local models which are dynamically learnt rather than hand-designed by the user. Techniques in neural architecture search could also be helpful for this purpose.

Finally, learning fair representations is of utmost importance as our machine learning systems are deployed in real-life settings such as healthcare, law, and policy-making. In addition to the adversarial training method we described in this paper, there are a variety of methods for learning fair representations that can also be incorporated into our flexible local models. For example, recent work has shown that pre-trained word and sentence representations encode and exacerbate gender, race, and religious biases [4, 37, 59]. Incorporating these debiasing methods for text data would be an important step towards learning *fair* and *unbiased* local representations in federated learning.

Table 5: Table of hyperparameters for MNIST experiments.

Model	Parameter	Value
FEDAVG	Input dim	784
	Layers	[512, 256, 256, 128]
	Output dim	10
	Loss	cross entropy
	Batchsize	10
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.05
	Momentum	0.5
	Global epochs	800
LOCAL ONLY	Input dim	784
	Layers	[512, 256, 256, 128]
	Output dim	10
	Loss	cross entropy
	Batchsize	10
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.05
	Momentum	0.5
	Global epochs	200
LG-FEDAVG, Local model	Input dim	784
	Layers	[512, 256, 256, 128]
	Output dim	10
	Loss	cross entropy
	Batchsize	10
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.05
	Momentum	0.5
	Global epochs	100
LG-FEDAVG, Global model	Layers kept	2
	Input dim	256
	Layers	[128]
	Output dim	10
	Loss	cross entropy
	Batchsize	10
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.05
	Momentum	0.5
	Global epochs	400

Table 6: Comparison of federated learning methods on MNIST with non-iid splits over devices. We report accuracy under both local test and new test settings as well as the total number of parameters communicated across training iterations. Best results in **bold**. LG-FEDAVG outperforms FEDAVG under local test and achieves similar performance under new test while using around 50% of the total communicated parameters, across different device splits (2 − 10 classes per device). Mean and standard deviation are computed over 10 runs.

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
2 classes/device	FEDAVG [38]	98.20 ± 0.05	**98.20 ± 0.05**	800	0	5.6×10^{10}
	Local only [50]	98.72 ± 0.35	30.41 ± 7.88	0	0	0
	LG-FEDAVG (ours)	**98.77 ± 0.09**	97.72 ± 0.08	400	100	$\mathbf{2.9 \times 10^{10}}$
	LG-FEDAVG (ours)	98.71 ± 0.08	97.94 ± 0.06	500	100	3.6×10^{10}
	LG-FEDAVG (ours)	98.70 ± 0.01	98.03 ± 0.05	600	100	4.3×10^{10}
	LG-FEDAVG (ours)	98.63 ± 0.09	98.07 ± 0.03	700	100	5.0×10^{10}
	LG-FEDAVG (ours)	98.54 ± 0.05	**98.17 ± 0.05**	800	100	5.7×10^{10}

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
3 classes/device	FEDAVG [38]	98.20 ± 0.02	**98.20 ± 0.02**	800	0	5.6×10^{10}
	Local only [50]	97.55 ± 0.30	36.11 ± 10.13	0	0	0
	LG-FEDAVG (ours)	**98.55 ± 0.09**	97.92 ± 0.08	400	100	$\mathbf{2.9 \times 10^{10}}$
	LG-FEDAVG (ours)	98.38 ± 0.04	98.03 ± 0.08	500	100	3.6×10^{10}
	LG-FEDAVG (ours)	98.44 ± 0.06	98.10 ± 0.03	600	100	4.3×10^{10}
	LG-FEDAVG (ours)	98.37 ± 0.08	98.14 ± 0.02	700	100	5.0×10^{10}
	LG-FEDAVG (ours)	98.34 ± 0.10	**98.18 ± 0.05**	800	100	5.7×10^{10}

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
4 classes/device	FEDAVG [38]	98.21 ± 0.05	**98.21 ± 0.05**	800	0	5.6×10^{10}
	Local only [50]	96.53 ± 0.41	43.15 ± 16.41	0	0	0
	LG-FEDAVG (ours)	98.32 ± 0.08	97.98 ± 0.08	400	100	$\mathbf{2.9 \times 10^{10}}$
	LG-FEDAVG (ours)	98.28 ± 0.07	98.00 ± 0.05	500	100	3.6×10^{10}
	LG-FEDAVG (ours)	98.33 ± 0.05	98.12 ± 0.05	600	100	4.3×10^{10}
	LG-FEDAVG (ours)	98.30 ± 0.06	98.12 ± 0.03	700	100	5.0×10^{10}
	LG-FEDAVG (ours)	**98.34 ± 0.06**	**98.20 ± 0.06**	800	100	5.7×10^{10}

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
5 classes/device	FEDAVG [38]	98.13 ± 0.05	**98.13 ± 0.05**	800	0	5.6×10^{10}
	Local only [50]	95.47 ± 0.31	58.69 ± 4.11	0	0	0
	LG-FEDAVG (ours)	98.18 ± 0.06	97.82 ± 0.10	400	100	$\mathbf{2.9 \times 10^{10}}$
	LG-FEDAVG (ours)	**98.26 ± 0.07**	98.01 ± 0.07	500	100	3.6×10^{10}
	LG-FEDAVG (ours)	98.23 ± 0.04	98.06 ± 0.05	600	100	4.3×10^{10}
	LG-FEDAVG (ours)	98.23 ± 0.04	98.10 ± 0.05	700	100	5.0×10^{10}
	LG-FEDAVG (ours)	98.21 ± 0.07	**98.12 ± 0.07**	800	100	5.7×10^{10}

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
10 classes/device (iid)	FEDAVG [38]	97.93 ± 0.08	**97.93 ± 0.08**	800	0	5.6×10^{10}
	Local only [50]	88.03 ± 0.37	86.24 ± 0.87	0	0	0
	LG-FEDAVG (ours)	97.59 ± 0.08	97.61 ± 0.08	400	100	$\mathbf{2.9 \times 10^{10}}$
	LG-FEDAVG (ours)	97.78 ± 0.13	97.82 ± 0.14	500	100	3.6×10^{10}
	LG-FEDAVG (ours)	97.84 ± 0.10	97.86 ± 0.08	600	100	4.3×10^{10}
	LG-FEDAVG (ours)	97.85 ± 0.09	97.88 ± 0.09	700	100	5.0×10^{10}
	LG-FEDAVG (ours)	**97.91 ± 0.10**	**97.93 ± 0.07**	800	100	5.7×10^{10}

Table 7: Table of hyperparameters for CIFAR-10 experiments.

Model	Parameter	Value
FEDAVG	Loss	cross entropy
	Batchsize	50
	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.5
	Learning rate decay	0.005
	Global epochs	1800
LOCAL ONLY	Loss	cross entropy
	Batchsize	50
	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.5
	Learning rate decay	0.005
	Global epochs	200
LG-FEDAVG, Local model	Loss	cross entropy
	Batchsize	50
	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.5
	Learning rate decay	0.005
	Global epochs	100
LG-FEDAVG, Global model	Loss	cross entropy
	Batchsize	50
	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.5
	Learning rate decay	0.005
	Global epochs	1200

Table 8: Comparison of federated learning methods on CIFAR-10 with non-iid split over devices. We report accuracy under both local test and new test settings as well as the total number of parameters communicated across training iterations. Best results in **bold**. LG-FEDAVG outperforms FEDAVG and under local test and achieves similar performance under new test while using around 50% of the total communicated parameters, across different device splits ($2-10$ classes per device). Mean and standard deviation are computed over 10 runs.

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
2 classes/device	FEDAVG [38]	58.99 ± 1.50	58.99 ± 1.50	1800	0	12.7×10^9
	Local only [50]	87.93 ± 2.14	10.03 ± 0.06	0	0	0
	LG-FEDAVG (ours)	90.20 ± 0.79	56.52 ± 1.59	1000	100	$\mathbf{7.1 \times 10^9}$
	LG-FEDAVG (ours)	90.77 ± 0.50	57.95 ± 1.48	1200	100	8.5×10^9
	LG-FEDAVG (ours)	91.07 ± 0.62	59.28 ± 1.70	1400	100	9.9×10^9
	LG-FEDAVG (ours)	91.45 ± 0.77	59.96 ± 1.61	1600	100	11.3×10^9
	LG-FEDAVG (ours)	**91.77 ± 0.56**	**60.79 ± 1.45**	1800	100	12.7×10^9

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
3 classes/device	FEDAVG [38]	63.68 ± 0.35	63.68 ± 0.350	1800	0	12.7×10^9
	Local only [50]	79.79 ± 1.05	10.00 ± 0.00	0	0	0
	LG-FEDAVG (ours)	86.01 ± 0.55	61.78 ± 0.61	1000	100	$\mathbf{7.1 \times 10^9}$
	LG-FEDAVG (ours)	85.13 ± 0.76	63.01 ± 0.58	1200	100	8.5×10^9
	LG-FEDAVG (ours)	86.69 ± 0.58	63.57 ± 0.31	1400	100	9.9×10^9
	LG-FEDAVG (ours)	86.70 ± 0.49	63.39 ± 1.68	1600	100	11.3×10^9
	LG-FEDAVG (ours)	**87.26 ± 0.63**	**64.79 ± 0.55**	1800	100	12.7×10^9

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
4 classes/device	FEDAVG [38]	65.54 ± 0.66	65.54 ± 0.66	1800	0	12.7×10^9
	Local only [50]	78.01 ± 0.46	10.22 ± 0.29	0	0	0
	LG-FEDAVG (ours)	82.56 ± 0.54	63.62 ± 0.64	1000	100	$\mathbf{7.1 \times 10^9}$
	LG-FEDAVG (ours)	83.02 ± 0.47	64.40 ± 0.45	1200	100	8.5×10^9
	LG-FEDAVG (ours)	83.61 ± 0.26	65.41 ± 0.71	1400	100	9.9×10^9
	LG-FEDAVG (ours)	83.78 ± 0.56	65.99 ± 0.55	1600	100	11.3×10^9
	LG-FEDAVG (ours)	**84.14 ± 0.42**	**66.48 ± 0.74**	1800	100	12.7×10^9

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
5 classes/device	FEDAVG [38]	67.21 ± 0.45	67.21 ± 0.45	1800	0	12.7×10^9
	Local only [50]	73.42 ± 0.56	10.51 ± 0.49	0	0	0
	LG-FEDAVG (ours)	80.97 ± 0.62	65.34 ± 1.00	1000	100	$\mathbf{7.1 \times 10^9}$
	LG-FEDAVG (ours)	81.50 ± 0.52	66.32 ± 0.48	1200	100	8.5×10^9
	LG-FEDAVG (ours)	81.92 ± 0.55	67.26 ± 0.44	1400	100	9.9×10^9
	LG-FEDAVG (ours)	82.29 ± 0.38	67.61 ± 0.61	1600	100	11.3×10^9
	LG-FEDAVG (ours)	**82.51 ± 0.50**	**68.32 ± 0.56**	1800	100	12.7×10^9

Data	Method	Local Test Acc. (↑)	New Test Acc. (↑)	FedAvg Rounds	LG Rounds	Params Comm. (↓)
10 classes/device (iid)	FEDAVG [38]	67.74 ± 0.45	67.74 ± 0.45	1800	0	12.7×10^9
	Local only [50]	45.54 ± 0.31	16.90 ± 3.09	0	0	0
	LG-FEDAVG (ours)	68.09 ± 0.66	67.93 ± 0.61	1000	100	$\mathbf{7.1 \times 10^9}$
	LG-FEDAVG (ours)	68.97 ± 0.55	68.90 ± 0.54	1200	100	8.5×10^9
	LG-FEDAVG (ours)	69.36 ± 0.37	69.16 ± 0.30	1400	100	9.9×10^9
	LG-FEDAVG (ours)	69.64 ± 0.38	69.52 ± 0.44	1600	100	11.3×10^9
	LG-FEDAVG (ours)	**69.89 ± 0.48**	**69.76 ± 0.49**	1800	100	12.7×10^9

Table 9: Table of hyperparameters for VQA experiments.

Model	Parameter	Value
FEDAVG	Loss	cross entropy
	Batchsize	100
	Optimizer	SGD
	Learning rate	0.01
	Momentum	0.9
	Learning rate decay	0.0005
	Global epochs	100
LG-FEDAVG, Local model	Loss	cross entropy
	Batchsize	100
	Optimizer	SGD
	Learning rate	0.01
	Momentum	0.9
	Learning rate decay	0.0005
	Global epochs	100
LG-FEDAVG, Global model	Loss	cross entropy
	Batchsize	100
	Optimizer	SGD
	Learning rate	0.01
	Momentum	0.9
	Learning rate decay	0.0005
	Global epochs	100

Table 10: Table of hyperparameters for MAPS experiments.

Model	Parameter	Value
FEDAVG	BiLSTM encoder hidden units	128
	MLP hidden layers	[512, 512]
	Loss	cross entropy
	Max tokens per batch	2000
	Optimizer	adam
	Learning rate	5e-3
	Learning rate shrink	0.5
Local only	BiLSTM encoder hidden units	128
	MLP hidden layers	[512, 512]
	Loss	cross entropy
	Max tokens per batch	2000
	Optimizer	adam
	Learning rate	5e-3
	Learning rate shrink	0.5
Global	BiLSTM encoder hidden units	128
	MLP hidden layers	[512, 512]
	Loss	cross entropy
	Max tokens per batch	2000
	Optimizer	adam
	Learning rate	5e-3
	Learning rate shrink	0.5

Table 11: Table of hyperparameters for experiments on learning fair representations on the UCI adult dataset.

Model	Parameter	Value
FEDAVG, Global model	Input dim	93
	Layers	[32,32,32]
	Output dim	1
	Loss	cross entropy
	Dropout	0.2
	Batchsize	32
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.5
	Global epochs	50
FEDAVG, Global Adversary	Input dim	32
	Layers	[32,32,32]
	Output dim	2
	Loss	cross entropy
	Dropout	0.2
	Batchsize	32
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.5
	Global epochs	50
LG-FEDAVG, Local model LG-FEDAVG + Ave, Local model	Input dim	93
	Layers	[32,32,32]
	Output dim	1
	Loss	cross entropy
	Dropout	0.2
	Batchsize	32
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.5
	Local epochs	10
LG-FEDAVG + Ave, Local adversary	Input dim	93
	Layers	[32,32,32]
	Output dim	2
	Loss	cross entropy
	Dropout	0.2
	Batchsize	32
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.5
	Local epochs	10
LG-FEDAVG, Global model LG-FEDAVG + Ave, Global model	Input dim	93
	Layers	[32,32,32]
	Output dim	2
	Loss	cross entropy
	Dropout	0.2
	Batchsize	32
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.5
	Global epochs	10