Exploiting Shared Representations for Personalized Federated Learning

Liam Collins [1] **Hamed Hassani** [2] **Aryan Mokhtari** [1] **Sanjay Shakkottai** [1]

Abstract

Deep neural networks have shown the ability to extract universal feature representations from data such as images and text that have been useful for a variety of learning tasks. However, the fruits of representation learning have yet to be fully-realized in federated settings. Although data in federated settings is often non-i.i.d. across clients, the success of centralized deep learning suggests that data often shares a global *feature representation*, while the statistical heterogeneity across clients or tasks is concentrated in the *labels*. Based on this intuition, we propose a novel federated learning framework and algorithm for learning a shared data representation across clients and unique local heads for each client. Our algorithm harnesses the distributed computational power across clients to perform many local-updates with respect to the low-dimensional local parameters for every update of the representation. We prove that this method obtains linear convergence to the ground-truth representation with near-optimal sample complexity in a linear setting, demonstrating that it can efficiently reduce the problem dimension for each client. Further, we provide extensive experimental results demonstrating the improvement of our method over alternative personalized federated learning approaches in heterogeneous settings.

1. Introduction

Many of the most heralded successes of modern machine learning have come in *centralized* settings, wherein a single model is trained on a large amount of centrally-stored data. The growing number of data-gathering devices, however, calls for a distributed architecture to train models. Federated learning aims at addressing this issue by providing a platform in which a group of clients collaborate to learn effective models for each client by leveraging the local computational power, memory, and data of all clients (McMahan et al., 2017). The task of coordinating between the clients is fulfilled by a central server that combines the models received from the clients at each round and broadcasts the updated information to them. Importantly, the server and clients are restricted to methods that satisfy communication and privacy constraints, preventing them from directly applying centralized techniques.

However, one of the most important challenges in federated learning is the issue of *data heterogeneity*, where the underlying data distribution of client tasks could be substantially different from each other. In such settings, if the server and clients learn a single shared model (e.g., by minimizing average loss), the resulting model could perform poorly for many of the clients in the network (and also not generalize well across diverse data (Jiang et al., 2019)). In fact, for some clients, it might be better to simply use their own local data (even if it is small) to train a local model; see Figure 1. Finally, the (federated) trained model may not generalize well to unseen clients that have not participated in the training process. These issues raise this question:

> "*How can we exploit the data and computational power of all clients in data heterogeneous settings to learn a personalized model for each client?*"

PFL的核心问题

We address this question by taking advantage of the common representation among clients. Specifically, we view the data heterogeneous federated learning problem as n parallel learning tasks that they possibly have some common structure, and *our goal is to learn and exploit this common representation to improve the quality of each client's model*. Indeed, this would be in line with our understanding from centralized learning, where we have witnessed success in training multiple tasks simultaneously by leveraging a common (low-dimensional) representation in popular machine learning tasks (e.g., image classification, next-word prediction) (Bengio et al., 2013; LeCun et al., 2015).

Main Contributions. We introduce a novel federated learning framework and an associated algorithm for data heterogeneous settings. Next, we present our main contributions.

(i) **FedRep Algorithm.** Federated Representation Learn-

[1] University of Texas at Austin [2] University of Pennsylvania. Correspondence to: Liam Collins <liamc@utexas.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

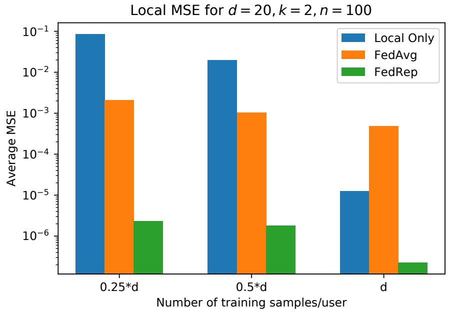

Figure 1. Local only training suffers in small-training data regimes, whereas training a single global model (FedAvg) cannot overcome client heterogeneity even when the number of training samples is large. FedRep exploits a common representation of the clients to achieve small error in all cases.

ing (FedRep) leverages the full quantity of data stored across clients to learn a global low-dimensional representation using gradient-based updates. Further, it enables each client to compute a personalized, low-dimensional classifier, which we term as the client's local head, that accounts for the unique labeling of each client's local data.

(ii) **Convergence Rate.** We show that FedRep converges to the optimal representation at a *exponentially fast rate* with near-optimal sample complexity in the case that each client aims to solve a linear regression problem with a two-layer linear neural network. Our analysis further implies that we only need $\mathcal{O}(\kappa^2 k^3 (\log(n) + \frac{\kappa^2 k d}{n}))$ samples per client, where n is the number of clients, d is the dimension of the data, k is the representation dimension and κ is the condition number of the ground-truth client-representation matrix.

(iii) **Empirical Results.** Through a combination of synthetic and real datasets (CIFAR10, CIFAR100, FEMNIST, Sent140) we show the benefits of FedRep in: (a) leveraging many local updates, (b) robustness to different levels of heterogeneity, and (c) generalization to new clients. We consider several important baselines including FedAvg (McMahan et al., 2017), Fed-MTL (Smith et al., 2017), LG-FedAvg (Liang et al., 2020), and Per-FedAvg (Fallah et al., 2020). Our experiments indicate that FedRep outpeforms these baselines in heterogeneous settings that share a global representation.

Benefits of FedRep. Next, we list benefits of FedRep over standard federated learning (that learns a single model).

(I) More local updates. By reducing the problem dimension, each client can make many local updates at each communication round, which is beneficial in learning its own individual head. This is unlike standard federated learning where multiple local updates in a heterogeneous setting moves each client *away* from the best averaged representation, and thus *hurts* performance.

(II) Gains of cooperation. Denote d to be the data dimension and n the number of clients. From our sample complexity bounds, it follows that with FedRep, the sample complexity per client scales as $\Theta(\log(n) + {}^{d}\!/\!_{n})$. On the other hand, local learning (without any collaboration) has a sample complexity that scale as $\Theta(d)$. Thus, if $1 \ll n \ll e^{\Theta(d)}$ (see Section 4.2 for details), we expect benefits of collaboration through federation. When d is large (as is typical in practice), $e^{\Theta(d)}$ is exponentially larger, and federation helps each client. *To the best of our knowledge, this is the first sample-complexity-based result for heterogeneous federated learning that demonstrates the benefit of cooperation.*

(III) Generalization to new clients. For a new client, since a ready-made representation is available, the client only needs to learn a head with a low-dimensional representation of dimension k. Thus, its sample complexity scales only as $\Theta(k\log({}^{1}\!/\!_{\epsilon}))$ to have no more than ϵ error in accuracy.

Related Work. A variety of recent works have studied personalization in federated learning using, for example, local fine-tuning (Wang et al., 2019; Yu et al., 2020), meta-learning (Chen et al., 2018; Khodak et al., 2019; Jiang et al., 2019; Fallah et al., 2020), additive mixtures of local and global models (Hanzely & Richtárik, 2020; Deng et al., 2020; Mansour et al., 2020), and multi-task learning (Smith et al., 2017). In all of these methods, each client's subproblem is still full-dimensional - there is no notion of learning a dimensionality-reduced set of local parameters. More recently, Liang et al. (2020) also proposed a representation learning method for federated learning, but their method attempts to learn many local representations and a single global head as opposed to a single global representation and many local heads. Earlier, Arivazhagan et al. (2019) presented an algorithm to learn local heads and a global network body, but their local procedure jointly updates the head and body (using the same number of updates), and they did not provide any theoretical justification for their proposed method. Meanwhile, another line of work has studied federated learning in heterogeneous settings (Karimireddy et al., 2020; Wang et al., 2020; Pathak & Wainwright, 2020; Haddadpour et al., 2020; Reddi et al., 2020; Reisizadeh et al., 2020; Mitra et al., 2021), and the optimization-based insights from these works may be used to supplement our formulation and algorithm.

2. Problem Formulation

The generic form of federated learning with n clients is

$$\min_{(q_1,\ldots,q_n)\in\mathcal{Q}_n} \frac{1}{n}\sum_{i=1}^{n} f_i(q_i), \qquad (1)$$

where f_i and q_i are the error function and learning model for the i-th client, respectively, and $\mathcal{Q}_n$ is the space of feasible sets of n models. We consider a supervised setting in which the data for the i-th client is generated by a distribution $(\mathbf{x}_i, y_i) \sim \mathcal{D}_i$. The learning model $q_i : \mathbb{R}^d \to \mathcal{Y}$ maps inputs $\mathbf{x}_i \in \mathbb{R}^d$ to predicted labels $q_i(\mathbf{x}_i) \in \mathcal{Y}$, which we would like to resemble the true labels y_i. The error f_i is in the form of an expected risk over $\mathcal{D}_i$, namely $f_i(q_i) := \mathbb{E}_{(\mathbf{x}_i,y_i)\sim\mathcal{D}_i}[\ell(q_i(\mathbf{x}_i), y_i)]$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function that penalizes the distance of $q_i(\mathbf{x}_i)$ from y_i.

In order to minimize f_i, the i-th client accesses a dataset of M_i labelled samples $\{(\mathbf{x}_i^j, y_i^j)\}_{j=1}^{M_i}$ from $\mathcal{D}_i$ for training. Federated learning addresses settings in which the M_i's are typically small relative to the problem dimension while the number of clients n is large. Thus, clients may not be able to obtain solutions q_i with small expected risk by training completely locally on *only* their M_i local samples. Instead, federated learning enables the clients to cooperate, by exchanging messages with a central server, in order to learn models using the cumulative data of all the clients.

Standard approaches to federated learning aim at learning a *single* shared model $q = q_1 = \cdots = q_n$ that performs well on average across the clients (McMahan et al., 2017; Li et al., 2018). In this way, the clients aim to solve a special version of Problem (1), which is to minimize $(1/n)\sum_i f_i(q)$ over the choice of the shared model q. However, this approach may yield a solution that performs poorly in heterogeneous settings where the data distributions $\mathcal{D}_i$ vary across the clients. Indeed, in the presence of data heterogeneity, the error functions f_i will have different forms and their minimizers are not the same. Hence, learning a shared model q may not provide good solution to Problem (1). This necessities the search for more personalized solutions $\{q_i\}$ that can be learned in a federated manner using the clients' data.

Learning a Common Representation. We are motivated by insights from centralized machine learning that suggest that heterogeneous data distributed across tasks may share a common representation despite having different labels (Bengio et al., 2013; LeCun et al., 2015); e.g., shared features across many types of images, or across word-prediction tasks. Using this common (low-dimensional) representation, the labels for each client can be simply learned using a linear classifier or a shallow neural network.

Formally, we consider a setting consisting of a global representation $q_\phi : \mathbb{R}^d \to \mathbb{R}^k$, which is a function parameterized by $\phi \in \Phi$ that maps data points to a lower space of dimension k, and client-specific heads $q_{h_i} : \mathbb{R}^k \to \mathcal{Y}$, which are functions parameterized by $h_i \in \mathcal{H}$ for $i \in [n]$ that map from the low-dimensional representation space to the label space. The model for the i-th client is the composition of the client's local parameters and the representation: $q_i(\mathbf{x}) = (q_{h_i} \circ q_\phi)(\mathbf{x})$. Critically, $k \ll d$, meaning that the number of parameters that must be learned locally by each client may be small. Thus, we can assume that any client's optimal classifier for any *fixed representation* is easy to compute, which motivates the following re-written global objective:

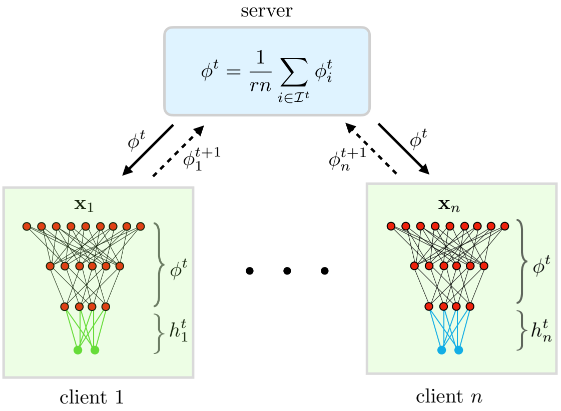

Figure 2. Federated representation learning structure where clients and the server aim at learning a global representation ϕ together, while each client i learns its unique head h_i locally.

本地模型

全球模型

$$\min_{\phi\in\Phi} \frac{1}{n}\sum_{i=1}^{n} \min_{h_i\in\mathcal{H}} f_i(h_i, \phi), \tag{2}$$

where we have used the shorthand $f_i(h_i, \phi) := f_i(q_{h_i} \circ q_\phi)$ for ease of notation. In our proposed scheme, clients cooperate to learn the global model using all clients' data, while they use their local information to learn their personalized head. We discuss this in detail in Section 3.

2.1. Comparison with Standard Federated Learning

To formally demonstrate the advantage of our formulation over the standard (single-model) federated learning formulation in heterogeneous settings with a shared representation, we study a linear representation setting with quadratic loss. As we will see below, standard federated learning *cannot recover the underlying representation in the face of heterogeneity*, while our formulation does indeed recover it.

Consider a setting in which the functions f_i are quadratic losses, the representation q_ϕ is a projection onto a k-dimensional subspace of $\mathbb{R}^d$ given by matrix $\mathbf{B} \in \mathbb{R}^{d\times k}$, and the i-th client's local head q_{h_i} is a vector $\mathbf{w}_i \in \mathbb{R}^k$. In this setting, we model the local data of clients $\{\mathcal{D}_i\}_i$ such that $y_i = \mathbf{w}_i^{*\top}\mathbf{B}^{*\top}\mathbf{x}_i$ for some ground-truth representation $\mathbf{B}^* \in \mathbb{R}^{d\times k}$ and local heads $\mathbf{w}_i^* \in \mathbb{R}^k$. This setting will be described in detail in Section 4. In particular, one can show that the expected error over the data distribution $\mathcal{D}_i$ has the following form: $f_i(\mathbf{w}_i, \mathbf{B}) := \frac{1}{2}\|\mathbf{B}\mathbf{w}_i - \mathbf{B}^*\mathbf{w}_i^*\|_2^2$.

Consequently, Problem (2) becomes

$$\min_{\mathbf{B}\in\mathbb{R}^{d\times k},\mathbf{w}_i,\ldots,\mathbf{w}_n\in\mathbb{R}^k} \frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{B}\mathbf{w}_i-\mathbf{B}^*\mathbf{w}_i^*\|_2^2. \qquad (3)$$

In contrast, standard federated learning methods, which aim to learn a shared model $(\mathbf{B},\mathbf{w})$ for all the clients, solve

$$\min_{\mathbf{B}\in\mathbb{R}^{d\times k},\mathbf{w}\in\mathbb{R}^k} \frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{B}\mathbf{w}-\mathbf{B}^*\mathbf{w}_i^*\|_2^2. \qquad (4)$$

Let $(\hat{\mathbf{B}},\{\hat{\mathbf{w}}_i\}_i)$ denote a global minimizer of (3). We thus have $\hat{\mathbf{B}}\hat{\mathbf{w}}_i=\mathbf{B}^*\mathbf{w}_i^*$ for all $i\in[n]$. Also, it is not hard to see that $(\mathbf{B}^\diamond,\mathbf{w}^\diamond)$ is a global minimizer of (4) if and only if $\mathbf{B}^\diamond\mathbf{w}^\diamond=\mathbf{B}^*(\frac{1}{n}\sum_{i=1}^n\mathbf{w}_i^*)$. Thus, our formulation finds an exact solution with zero global error, whereas standard federated learning has global error of $\frac{1}{2n}\sum_{i=1}^n\|\frac{1}{n}\mathbf{B}^*\sum_{i'=1}^n(\mathbf{w}_{i'}^*-\mathbf{w}_i^*)\|_2^2$, which grows with the heterogeneity of the $\mathbf{w}_i^*$. Moreover, since solving our formulation provides n matrix equations, we can fully recover the column space of $\mathbf{B}^*$ as long as $\mathbf{w}_i^*$'s span $\mathbb{R}^k$. In contrast, solving (4) yields only one matrix equation, so there is no hope to recover the column space of $\mathbf{B}^*$ for any $k>1$.

3. FedRep Algorithm

FedRep solves Problem (2) by distributing the computation across clients. The server and clients aim to learn the parameters of the global representation together, while the i-th client aims to learn its unique local head locally (see Figure 2). To do so, FedRep alternates between client updates and a server update on each communication round.

Client Update. On each round, a constant fraction $r\in(0,1]$ of the clients are selected to execute a client update. In the client update, client i makes τ local gradient-based updates to solve for its optimal head given the current global representation ϕ^t communicated by the server. Namely, for $s=1,\ldots,\tau$, client i updates its head as follows:

$$h_i^{t,s+1}=\texttt{GRD}(f_i(h_i^{t,s},\phi^t),h_i^{t,s},\alpha),$$

where $\texttt{GRD}(f,h,\alpha)$ is generic notation for an update of the variable h using a gradient of function f with respect to h and the step size α. For example, $\texttt{GRD}(f_i(h_i^{t,s},\phi^t),h_i^{t,s},\alpha)$ can be a step of gradient descent, stochastic gradient descent (SGD), SGD with momentum, etc. The key is that client i makes many such local updates, i.e., τ is large, to find the optimal head based on its local data, given the most recent representation ϕ^t received from the server.

Server Update. Once the local updates with respect to the head h_i finish, the client participates in the server update by taking one local gradient-based update with respect to the current representation, i.e., computing

$$\phi_i^{t+1}\leftarrow\texttt{GRD}(f_i(h_i^{t,\tau},\phi^t),\phi^t,\alpha).$$

It then sends ϕ_i^{t+1} to the server, which averages the local updates to compute the next representation ϕ^{t+1}. The entire procedure is outlined in Algorithm 1.

Algorithm 1 FedRep

```
**Parameters:** Participation rate $r$, step sizes $\alpha, \eta$; number of local updates $\tau$; number of communication rounds $T$.
Initialize $\phi^0, h_1^0, \ldots, h_n^0$
for $t = 1, 2, \ldots, T$ do
    Server receives a batch of clients $\mathcal{I}^t$ of size $rn$
    Server sends current representation $\phi^t$ to these clients
    for each client $i$ in $\mathcal{I}^t$ do
        Client $i$ initializes $h_i^t \leftarrow h_i^{t-1,\tau}$
        Client $i$ makes $\tau$ updates to its head $h_i^t$:
        for $s = 1$ to $\tau$ do
            $h_i^{t,s} \leftarrow \texttt{GRD}(f_i(h_i^{t,s}, \phi^t), h_i^{t,s}, \alpha)$
        end for
        Client $i$ locally updates the representation as:
            $\phi_i^{t+1} \leftarrow \texttt{GRD}(f_i(h_i^{t,\tau}, \phi^t), \phi^t, \alpha)$
        Client $i$ sends updated representation $\phi_i^{t+1}$ to server
    end for
    for each client $i$ not in $\mathcal{I}^t$, do
        Set $h_i^{t,\tau} \leftarrow h_i^{t-1,\tau}$
    end for
    Server computes the new representation as
        $\phi^{t+1} = \frac{1}{rn}\sum_{i\in\mathcal{I}^t}\phi_i^{t+1}$
end for
```

4. Low-Dimensional Linear Representation

In this section, we analyze an instance of Problem (2) with quadratic loss functions and linear models, as discussed in Section 2.1. Here, each client's problem is to solve a linear regression with a two-layer linear neural network. In particular, each client i attempts to find a shared global projection onto a low-dimension subspace $\mathbf{B}\in\mathbb{R}^{d\times k}$ and a unique regressor $\mathbf{w}_i\in\mathbb{R}^k$ that together accurately map its samples $\mathbf{x}_i\in\mathbb{R}^d$ to labels $y_i\in\mathbb{R}$. The matrix $\mathbf{B}$ corresponds to the representation ϕ, and $\mathbf{w}_i$ corresponds to local head h_i for the i-th client. We thus have $(q_{h_i}\circ q_\phi)(\mathbf{x}_i)=\mathbf{w}_i^\top\mathbf{B}^\top\mathbf{x}_i$. Hence, the loss function for client i is given by:

$$f_i(\mathbf{w}_i,\mathbf{B}):=\tfrac{1}{2}\mathbb{E}_{(\mathbf{x}_i,y_i)\sim\mathcal{D}_i}\left[(y_i-\mathbf{w}_i^\top\mathbf{B}^\top\mathbf{x}_i)^2\right] \qquad (5)$$

meaning that the global objective is:

$$\min_{\substack{\mathbf{B}\in\mathbb{R}^{d\times k}\\ \mathbf{W}\in\mathbb{R}^{n\times k}}} F(\mathbf{B},\mathbf{W}):=\frac{1}{2n}\sum_{i=1}^{n}\mathbb{E}_{(\mathbf{x}_i,y_i)}\left[(y_i-\mathbf{w}_i^\top\mathbf{B}^\top\mathbf{x}_i)^2\right], \qquad (6)$$

where $\mathbf{W}=[\mathbf{w}_1^\top,\ldots,\mathbf{w}_n^\top]\in\mathbb{R}^{n\times k}$ is the concatenation of client-specific heads. To evaluate the ability of FedRep to learn an accurate representation, we model the local datasets

$\{\mathcal{D}_i\}_i$ such that, for $i = 1 \dots, n$

$$y_i = \mathbf{w}_i^{*\top} \mathbf{B}^{*\top} \mathbf{x}_i,$$

for some ground-truth representation $\mathbf{B}^* \in \mathbb{R}^{d \times k}$ and local heads $\mathbf{w}_i^* \in \mathbb{R}^k$–i.e. a standard regression setting. In other words, all of the clients' optimal solutions live in the same k-dimensional subspace of $\mathbb{R}^d$, where k is assumed to be small. Moreover, we make the following standard assumption on the samples $\mathbf{x}_i$.

Assumption 1 (Sub-gaussian design). *The samples $\mathbf{x}_i \in \mathbb{R}^d$ are i.i.d. with mean $\mathbf{0}$, covariance $\mathbf{I}_d$, and are $\mathbf{I}_d$-sub-gaussian, i.e. $\mathbb{E}[e^{\mathbf{v}^\top \mathbf{x}_i}] \leq e^{\|\mathbf{v}\|_2^2/2}$ for all $\mathbf{v} \in \mathbb{R}^d$.*

4.1. FedRep

We next discuss how FedRep tries to recover the optimal representation in this setting. First, the server and clients execute the Method of Moments to learn an initial representation. Then, client and server updates are executed in an alternating fashion as follows.

Client Update. As in Algorithm 1, rn clients are selected on round t to update their current local head $\mathbf{w}_i^t$ and the global representation $\mathbf{B}^t$. Each selected client i samples a fresh batch $\{\mathbf{x}_i^{t,j}, y_i^{t,j}\}_{j=1}^m$ of m samples according to its local data distribution $\mathcal{D}_i$ to use for updating both its head and representation on each round t that it is selected. That is, within the round, client i considers the batch loss

$$\hat{f}_i^t(\mathbf{w}_i^t, \mathbf{B}^t) := \frac{1}{2m} \sum_{j=1}^m (y_i^{t,j} - \mathbf{w}_i^{t\top} \mathbf{B}^{t\top} \mathbf{x}_i^{t,j})^2. \quad (7)$$

Since $\hat{f}_i^t$ is strongly convex with respect to $\mathbf{w}_i^t$, the client can find an update for a local head that is ϵ-close to the global minimizer of (7) after at most $\log(1/\epsilon)$ local gradient updates. Alternatively, since the function is also quadratic, the client can solve for the optimal $\mathbf{w}$ directly in only $\mathcal{O}(mk^2 + k^3)$ operations. Thus, to simplify the analysis we assume each selected client obtains $\mathbf{w}_i^{t+1} = \text{argmin}_{\mathbf{w}} \hat{f}_i^t(\mathbf{w}, \mathbf{B}^t)$ during each round of local updates.

Server Update. After updating its head, client i updates the global representation with one step of gradient descent using the same m samples and sends the update to the server, as outlined in Algorithm 2. Then, the server computes the new representation by averaging over received representations.

Algorithm 2 FedRep for linear regression

```
Input: Step size $\eta$; number of rounds $T$, participation rate $r$.
Initialization: Each client $i \in [n]$ sends $\mathbf{Z}_i := \frac{1}{m}\sum_{j=1}^m (y_i^{0,j})^2 \mathbf{x}_i^{0,j}(\mathbf{x}_i^{0,j})^\top$ to server, server computes
        $\mathbf{U}\mathbf{D}\mathbf{U}^\top \leftarrow$ rank-$k$ SVD$(\frac{1}{n}\sum_{i=1}^n \mathbf{Z}_i)$
Server initializes $\mathbf{B}^0 \leftarrow \mathbf{U}$
for $t = 1, 2, \dots, T$ do
  Server receives a subset $\mathcal{I}^t$ of clients of size $rn$
  Server sends current representation $\mathbf{B}^t$ to these clients
  for $i \in \mathcal{I}^t$ do
    Client update:
    Client $i$ samples a fresh batch of $m$ samples
    Client $i$ updates $\mathbf{w}_i$:
        $\mathbf{w}_i^{t+1} \leftarrow \text{argmin}_{\mathbf{w}} \hat{f}_i^t(\mathbf{w}, \mathbf{B}^t)$
    Client $i$ updates representation:
        $\mathbf{B}_i^{t+1} \leftarrow \mathbf{B}^t - \eta \nabla_{\mathbf{B}} \hat{f}_i^t(\mathbf{w}_i^{t+1}, \mathbf{B}^t)$
    Client $i$ sends $\mathbf{B}_i^{t+1}$ to the server
  end for
  Server update: $\mathbf{B}^{t+1} \leftarrow \frac{1}{rn}\sum_{i \in \mathcal{I}^t} \mathbf{B}_i^{t+1}$
end for
```

4.2. Analysis

As mentioned earlier, in FedRep, each client i perform an alternating minimization-descent method to solve its nonconvex objective in (7). This means the global loss over all clients at round t is given by

$$\frac{1}{n}\sum_{i=1}^n \hat{f}_i^t(\mathbf{w}_i^t, \mathbf{B}^t) := \frac{1}{2mn}\sum_{i=1}^n \sum_{j=1}^m (y_i^{t,j} - \mathbf{w}_i^{t\top} \mathbf{B}^{t\top} \mathbf{x}_i^{t,j})^2. \quad (8)$$

This objective has many global minima, including all pairs of matrices $(\mathbf{Q}^{-1}\mathbf{W}^*, \mathbf{B}^*\mathbf{Q}^\top)$ where $\mathbf{Q} \in \mathbb{R}^{k \times k}$ is invertible, eliminating the possibility of exactly recovering the ground-truth factors $(\mathbf{W}^*, \mathbf{B}^*)$. Instead, the ultimate goal of the server is to recover the ground-truth *representation*, i.e., the column space of $\mathbf{B}^*$. To evaluate how closely the column space is recovered, we define the distance between subspaces as follows.

Definition 1. *The principal angle distance between the column spaces of $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{d \times k}$ is given by*

$$\text{dist}(\mathbf{B}_1, \mathbf{B}_2) := \|\hat{\mathbf{B}}_{1,\perp}^\top \hat{\mathbf{B}}_2\|_2, \quad (9)$$

where $\hat{\mathbf{B}}_{1,\perp}$ and $\hat{\mathbf{B}}_2$ are orthonormal matrices satisfying span$(\hat{\mathbf{B}}_{1,\perp})$ = span$(\mathbf{B}_1)^\perp$ and span$(\hat{\mathbf{B}}_2)$ = span$(\mathbf{B}_2)$.

The principal angle distance is a standard metric for measuring the distance between subspaces (e.g. (Jain et al., 2013)). Next, we make two regularity assumptions.

Assumption 2 (Client diversity). *Let $\bar{\sigma}_{\min,*}$ be the minimum singular value of any matrix $\overline{\mathbf{W}} \in \mathbb{R}^{rn \times k}$ with rows being an rn-sized subset of ground-truth client-specific parameters $\{\mathbf{w}_1^*, \dots, \mathbf{w}_n^*\}$. Then $\bar{\sigma}_{\min,*} > 0$.*

Assumption 2 states that if we select any rn clients, their optimal solutions span $\mathbb{R}^k$. Indeed, this assumption is weak as we expect the number of participating clients rn to be substantially larger than k. Note that if we do not have client solutions that span $\mathbb{R}^k$, recovering $\mathbf{B}^*$ would be impossible because the samples $(\mathbf{x}_i^j, y_i^j)$ may never contain any information about one or more features of $\mathbf{B}^*$.

Assumption 3 (Client normalization)**.** *The ground-truth client-specific parameters satisfy* $\|\mathbf{w}_i^*\|_2 = \sqrt{k}$ *for all* $i \in [n]$*, and* $\mathbf{B}^*$ *has orthonormal columns.*

Assumption 2 ensures that the ground-truth matrix $\mathbf{W}^*\mathbf{B}^{*\top}$ is row-wise *incoherent*, i.e. its row norms have similar magnitudes. We define this formally in Appendix B. Incoherence of the ground-truth matrices is a key property required for efficient matrix completion and other sensing problems with sparse measurements (Chi et al., 2019). Since our measurement matrices are row-wise sparse, we require the row-wise incoherence of the ground truth. Note that Assumption 3 can be relaxed to allow $\|\mathbf{w}_i^*\|_2 \leq O(\sqrt{k})$, as the exact normalization is only for simplicity of analysis.

Our main result shows that the iterates $\{\mathbf{B}^t\}_t$ generated by FedRep in this setting linearly converge to the optimal representation $\mathbf{B}_*$ in principal angle distance.

Theorem 1. *Define* $E_0 := 1 - \mathit{dist}^2(\hat{\mathbf{B}}^0, \hat{\mathbf{B}}^*)$ *and* $\bar{\sigma}_{\max,*} := \max_{\mathcal{I}\in[n],|\mathcal{I}|=rn} \sigma_{\max}(\frac{1}{\sqrt{rn}}\mathbf{W}_{\mathcal{I}}^*)$ *and* $\bar{\sigma}_{\min,*} := \min_{\mathcal{I}\in[n],|\mathcal{I}|=rn} \sigma_{\min}(\frac{1}{\sqrt{rn}}\mathbf{W}_{\mathcal{I}}^*)$*, i.e. the maximum and minimum singular values of any matrix that can be obtained by taking* rn *rows of* $\frac{1}{\sqrt{rn}}\mathbf{W}^*$*. Let* $\kappa := \bar{\sigma}_{\max,*}/\bar{\sigma}_{\max,*}$*. Suppose that* $m \geq c(\kappa^4 k^3 \log(rn)/E_0^2 + \kappa^4 k^2 d/(E_0^2 rn))$ *for some absolute constant* c*. Then for any* t *and any* $\eta \leq 1/(4\bar{\sigma}_{\max,*}^2)$*, we have*

$$\operatorname{dist}(\hat{\mathbf{B}}^T, \hat{\mathbf{B}}^*) \leq \left(1 - \eta E_0 \bar{\sigma}_{\min,*}^2/2\right)^{T/2} \operatorname{dist}(\hat{\mathbf{B}}^0, \hat{\mathbf{B}}^*), \tag{10}$$

with probability at least $1 - Te^{-100\min(k^2\log(rn),d)}$.

From Assumption 2, we have that $\bar{\sigma}_{\min,*}^2 > 0$, so the RHS of (10) strictly decreases with T for appropriate step size. Considering the complexity of m and the fact that the algorithm converges exponentially fast, the total number of samples required per client to reach an ϵ-accurate solution in principal angle distance is $\Theta\left(m \log\left({}^1\!/\!_\epsilon\right)\right)$, which is

$$\Theta\left(\left[\kappa^4 k^3 \left(\log(rn) + {}^{\kappa^4 kd}\!/\!_{rn}\right)\right] \log\left({}^1\!/\!_\epsilon\right)\right). \tag{11}$$

Next, a few remarks about this sample complexity follow.

When and whom does federation help? Observe that for a single client with no collaboration, the sample complexity scales as $\Theta(d)$. With FedRep, however, the sample complexity scales as $\Theta(\log(n) + {}^d\!/\!_n)$. Thus, so long as $\log(n) + {}^d\!/\!_n \ll d$, federation helps. This holds in several settings, for instance when $1 \ll n \ll e^{\Theta(d)}$. In practical scenarios, d (the data dimension) is large, and thus $e^{\Theta(d)}$ is exponentially larger; thus collaboration helps *each individual client.* Furthermore, from the point of view of a new client who enters the system later, it has a representation available for free, and this new client's sample complexity for adapting to its task is only $k\log({}^1\!/\!_\epsilon)$. Thus, both the overall system benefits (a representation has been learned, which is useful for the new client because it now only needs to learn a head), and each individual client that did take part in the federated training also benefits.

Connection to matrix sensing. The problem in (6) is an instance of matrix sensing; see the proof in Appendix B for more details. Considering this connection, our theoretical results also contribute to the theoretical study of matrix sensing. Although matrix sensing is a well-studied problem, our setting presents two new analytical challenges: (i) due to row-wise sparsity in the measurements, the sensing operator does not satisfy the commonly-used Restricted Isometry Property (RIP) within an efficient number of samples, i.e., it does not efficiently concentrate to an identity operation on all rank-k matrices, and (ii) FedRep executes a novel non-symmetric procedure. We further discuss these challenges in Appendix B.5. To the best of our knowledge, Theorem 1 provides the first convergence result for an alternating minimization-descent procedure to solve a matrix sensing problem. It is also the first result to show sample-efficient linear convergence of any solution to a matrix sensing with rank-one, row-wise sparse measurements. The state-of-the-art result for the closest matrix sensing setting to ours is given by Zhong et al. (2015) for rank-1, independent Gaussian measurements, which our result matches up to an $\mathcal{O}(\kappa^2)$ factor. However, our setting is more challenging as we have rank-1 *and* row-wise sparse measurements, and dependence on κ^4 has been previously observed in settings with sparse measurements, e.g. matrix completion (Jain et al., 2013).

New users and dimensionality reduction. Theorem 1 is related to works studying representation learning in the context of multi-task learning. Tripuraneni et al. (2020) and Du et al. (2020) provided upper bounds on the generalization error resulting from learning a low-dimensional representation of tasks assumed to share a common representation. They show that, if the common representation is learned, then excess risk bound on a new task is $O(\frac{\mathcal{C}(\Phi)}{nm} + \frac{k}{m_{\text{new}}})$, where $\mathcal{C}(\Phi)$ is the complexity of the representation class Φ and m_{new} is the number of labelled samples from the new task that the learner can use for fine-tuning. Since the number of test samples must exceed only $O(k)$, where k is assumed to small, these works demonstrate the dimensionality-reducing benefits of representation learning. Our work complements these results by showing how to provably and efficiently learn the representation in the linear case.

Remark on initialization. Theorem 1 requires that the initial principal angle distance $\operatorname{dist}(\hat{\mathbf{B}}^0, \hat{\mathbf{B}}^*)$ is larger than

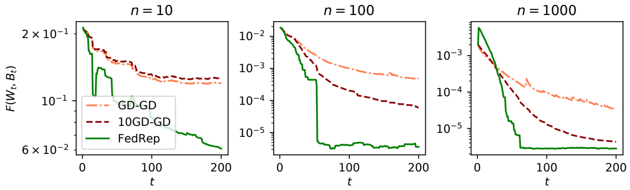

Figure 3. Comparison of (principal angle) distances between the ground-truth and estimated representations by FedRep and alternating gradient descent algorithms for different numbers of clients n. In all plots, $d = 10$, $k = 2$, $m = 5$, and $r = 0.1$.

a constant $c > 0$. This can be achieved by the Method of Moments without increasing the sample complexity for each client up to log factors (Tripuraneni et al., 2020). In turn, each user must send the server a polynomial of their data, namely $\sum_{j=1}^{m} (y_i^j)^2 \mathbf{x}_i^j (\mathbf{x}_i^j)^\top$ at the start of the learning procedure, which does not compromise privacy. We discuss the details of this in Appendix B.

5. Experiments

We focus on three points in our experiments: (i) the effect of many local updates for the local head in FedRep (ii) the quality of the global representation learned by FedRep and (iii) the applicability of FedRep to a wide range of datasets. Full experimental details are provided in Appendix A.

5.1. Synthetic Data

We start by experimenting with an instance of the multi-linear regression problem analyzed in Section 4. Consistent with this formulation, we generate synthetic samples $\mathbf{x}_i^j \sim \mathcal{N}(0, \mathbf{I}_d)$ and labels $y_i^j \sim \mathcal{N}(\mathbf{w}_i^{*\top} \hat{\mathbf{B}}^{*\top} \mathbf{x}_i^j, 10^{-3})$ (here we include an additive Gaussian noise). The ground-truth heads $\mathbf{w}_i^* \in \mathbb{R}^k$ for clients $i \in [n]$ and the ground-truth representation $\hat{\mathbf{B}}^* \in \mathbb{R}^{d \times k}$ are generated randomly by sampling and normalizing Gaussian matrices.

Benefit of finding the optimal head. We first demonstrate that the convergence of FedRep improves with larger number of clients n, making it highly applicable to federated settings. Further, we give evidence showing that this improvement is augmented by the minimization step in FedRep, since methods that replace the minimization step in FedRep with 1 and 10 steps of gradient descent (GD-GD and 10GD-GD, respectively) do not scale properly with n. In Figure 3, we plot convergence trajectories for FedRep, GD-GD, and 10GD-GD for four different values of n and fixed m, d, k and r. As we observe in Figure 3, by increasing the number of nodes n, clients converge to the true representation faster. Also, running more local updates for finding the local head accelerates the convergence speed of FedRep. In particular, FedRep which exactly finds the optimal local head at each round has the fastest rate compared to GD-GD and 10GD-GD that only run 1 and 10 local updates, respectively, to learn the head.

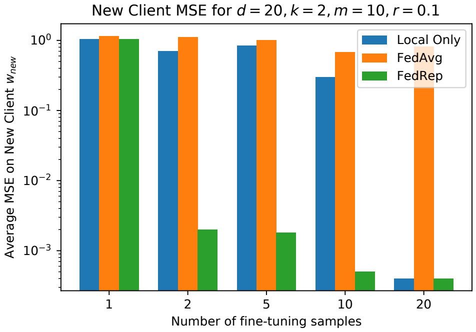

Figure 4. MSE on new clients sharing the representation after fine-tuning using various numbers of samples from the new client.

Generalization to new clients. Next, we evaluate the effectiveness of the representation learned by FedRep in reducing the sample complexity for a new client which has not participated in training. We first train FedRep and FedAvg on a fixed set of $n = 100$ clients as in Figure 1, where $(d, k) = (20, 2)$. The new client has access to m_{new} labelled local samples. It will use the representation $\hat{\mathbf{B}}^* \in \mathbb{R}^{d \times k}$ learned from the training clients, and learns a personalized head using this representation and its local training samples. For both FedRep and FedAvg, we solve for the optimal head given these samples and the representation learned during training. We compare the MSE of the resulting model on the new client's test data to that of a model trained by only using the m_{new} labelled samples from the new client (Local Only) in Figure 4. The large error for FedAvg demonstrates that it does not learn the ground-truth representation. Meanwhile, the representation learned by FedRep allows an accurate model to be found for the new client as long as $m_{\text{new}} \geq k$, which drastically improves over the complexity for Local Only ($m_{\text{new}} = \Omega(d)$).

5.2. Real Data

We next investigate whether these insights apply to nonlinear models and real datasets.

Datasets and Models. We use four real datasets: CIFAR10 and CIFAR100 (Krizhevsky et al., 2009), FEMNIST (Caldas et al., 2018; Cohen et al., 2017) and Sent140 (Caldas et al., 2018). The first three are image datasets and the last is a text dataset for which the goal is to classify the sentiment of a tweet as positive or negative. We control the heterogeneity of CIFAR10 and CIFAR100 by assigning different numbers S of classes per client, from among 10 and 100 total classes, respectively. Each client is assigned the same number of training samples, namely $50,000/n$.

For FEMNIST, we restrict the dataset to 10 handwritten letters and assign samples to clients according to a log-normal distribution as in (Li et al., 2019). We consider a partition of $n = 150$ clients with an average of 148 samples/client. For Sent140, we use the natural assignment of tweets to their author, and use $n = 183$ clients with an average of 72 samples per client. We use 5-layer CNNs for the CIFAR datasets, a 2-layer MLP for FEMNIST, and an RNN for Sent140 (details provided in Appendix A).

Baselines. We compare against a variety of personalized federated learning techniques as well as methods for learning a single global model and their fine-tuned analogues. Among the personalized methods, FedPer (Arivazhagan et al., 2019) is most similar to ours, as it also learns a global representation and personalized heads, but makes simultaneous local updates for both sets of parameters, therefore makes the same number of local updates for the head and the representation on each local round. Fed-MTL (Smith et al., 2017) learns local models and a regularizer to encode relationships among the clients, PerFedAvg (Fallah et al., 2020) leverages meta-learning to learn a single model that performs well after adaptation on each task, and LG-FedAvg (Liang et al., 2020) learns local representations and a global head. APFL (Deng et al., 2020) interpolates between local and global models, and L2GD (Hanzely & Richtárik, 2020) and Ditto (Li et al., 2020) learn local models that are encouraged to be close together by global regularization. For global FL methods, we consider FedAvg (McMahan et al., 2017), SCAFFOLD (Karimireddy et al., 2020), and FedProx (Li et al., 2018). To obtain fine-tuning results, we first train the global model for the full training period, then each client then fine-tunes only the head on its local training data for 10 epochs of SGD before computing the final test accuracy.

Implementation. In each experiment we sample a ratio $r = 0.1$ of all the clients on every round. We initialize all models randomly and train for $T = 100$ communication rounds for the CIFAR datasets, $T = 50$ for Sent140, and $T = 200$ for FEMNIST. In each case, FedRep executes ten local epochs of SGD with momentum to train the local head, followed by one or five epochs for the representation, in each local update (depending on the dataset). All other methods use the same number of local epochs as FedRep does for updating the representation. Accuracies are computed by taking the average local accuracies for all users over the final 10 rounds of communication, except for the fine-tuning methods. These accuracies are computed after locally training the head of the fully-trained global model for ten epochs for each client.

Benefit of more local updates. As mentioned in Section 1, a key advantage of our formulation is that it enables clients to run many local updates without causing divergence from the global optimal solution. We demonstrate an example of this in Figure 5. Here, there are $n = 100$ clients where each has $S = 2$ classes of images. For FedAvg, we observe running more local updates does not necessarily improve the performance. In contrast, FedRep's performance is monotonically non-decreasing with E, i.e., *FedRep requires less tuning of* E *and is never hurt by more local computation.*

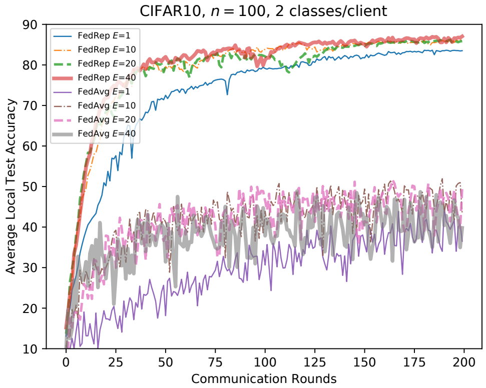

Figure 5. CIFAR10 local test errors for different numbers of local epochs E for FedRep (for the heads) and FedAvg.

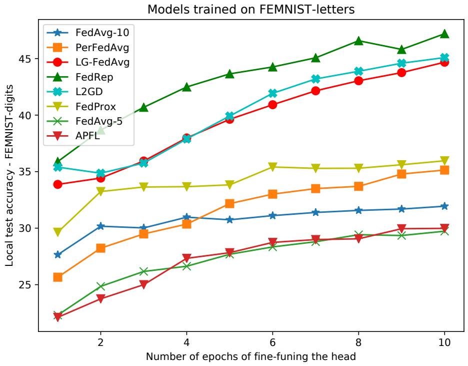

Figure 6. Test accuracy on handwritten digits from FEMNIST after fine-tuning the head of models trained on FEMNIST-letters.

Robustness to varying levels of heterogeneity, number of clients and number of samples per client. We show the average local test errors for all of the algorithms for a variety of settings in Table 1. In all cases, FedRep is either the top-performing method or is very close to the top-performing method. Recall that for the CIFAR datasets, the number of training samples per client is equal to $50,000/n$, so the column with 1000 users has the smallest number of samples per client.

Generalization to new clients. We also evaluate the strength of the representation learned by FedRep in terms of adaptation for new users. To do so, we first train FedRep, Fe-

Table 1. Average test accuracies on various partitions of CIFAR10, CIFAR100, Sent140 and FEMNIST with participation rate $r = 0.1$.

	CIFAR10			CIFAR100		Sent140	FEMNIST
(# clients n, # classes per client S)	$(100, 2)$	$(100, 5)$	$(1000, 2)$	$(100, 5)$	$(100, 20)$	$(183, 2)$	$(150, 3)$
Local Only	**89.79**	70.68	78.30	75.29	41.29	69.88	60.86
FedAvg (McMahan et al., 2017)	42.65	51.78	44.31	23.94	31.97	52.75	51.64
FedAvg+FT	87.65	73.68	82.04	**79.34**	55.44	71.92	72.41
FedProx (Li et al., 2018)	39.92	50.99	21.93	20.17	28.52	52.33	18.89
FedProx+FT	85.81	72.75	75.41	78.52	55.09	71.21	53.54
SCAFFOLD (Karimireddy et al., 2020)	37.72	47.33	33.79	20.32	22.52	51.31	17.65
SCAFFOLD+FT	86.35	68.23	78.24	78.88	44.34	71.49	52.11
Fed-MTL (Smith et al., 2017)	80.46	58.31	76.53	71.47	41.25	71.20	54.11
PerFedAvg (Fallah et al., 2020)	82.27	67.20	67.36	72.05	52.49	68.45	71.51
LG-Fed (Liang et al., 2020)	84.14	63.02	77.48	72.44	38.76	70.37	62.08
L2GD (Hanzely & Richtárik, 2020)	81.04	59.98	71.96	72.13	42.84	70.67	66.18
APFL (Deng et al., 2020)	83.77	72.29	82.39	78.20	55.44	69.87	70.74
Ditto (Li et al., 2020)	85.39	70.34	80.36	78.91	**56.34**	71.04	68.28
FedPer (Arivazhagan et al., 2019)	87.13	73.84	81.73	76.00	55.68	72.12	76.91
FedRep (Ours)	**87.70**	**75.68**	**83.27**	79.15	56.10	**72.41**	**78.56**

dAvg, PerFedAvg, LG-FedAvg, APFL, L2GD and FedProx in the usual setting on the partition of FEMNIST containing images of 10 handwritten letters (FEMNIST-letters). Then, we encounter clients with data from a different partition of the FEMNIST dataset, containing images of handwritten digits. We assume we have access to a dataset of 500 samples at this new client to fine tune the head. Using these, with each of the algorithms, we fine tune the head over multiple epochs while keeping the representation fixed. In Figure 6, we repeatedly sweep over the same 500 samples over multiple epochs to further refine the head, and plot the corresponding local test accuracy. As is apparent, FedRep has significantly better performance than these baselines.

6. Discussion

We have introduced a novel representation learning framework and algorithm for federated learning along with both theoretical and empirical evidence for its utility in federated settings. The FedRep framework is general and simple enough to easily apply to a broad range of federated learning problems, from linear regression to image classification to sentiment analysis, as we have shown here. Still, it is powerful enough to achieve significant improvement in local accuracy over a variety of personalized federated learning baselines. One interesting observation is that fine-tuning global federated learning methods, especially FedAvg, also tends to perform very well. We plan to further investigate this phenomenon in future work. Indeed, more complex extensions of the FedRep framework may be proposed to improve performance relative to fine-tuning methods.

7. Acknowledgements

The research of Liam Collins is supported through ARO Grant W911NF-11-1-0265 and NSF Grant 2019844. The research of Sanjay Shakkottai is supported by ONR Grant N00014-19-1-2566 and NSF Grant 2019844. The research of Aryan Mokhtari is supported in part by NSF Grant 2007668, ARO Grant W911NF2110226, and the Machine Learning Laboratory at UT Austin. The research of Hamed Hassani is supported by NSF Grants 1837253, 1943064, and AFOSR Grant FA9550-20-1-0111.

References

Arivazhagan, M. G., Aggarwal, V., Singh, A. K., and Choudhary, S. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Chen, F., Luo, M., Dong, Z., Li, Z., and He, X. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

Chi, Y., Lu, Y. M., and Chen, Y. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Deng, Y., Kamani, M. M., and Mahdavi, M. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. Few-shot learning via learning the representation, provably, 2020.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning: A meta-learning approach, 2020.

Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. *arXiv preprint arXiv:2007.01154*, 2020.

Hanzely, F. and Richtárik, P. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

Jain, P., Netrapalli, P., and Sanghavi, S. Low-rank matrix completion using alternating minimization. *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing - STOC '13*, 2013.

Jiang, Y., Konečnỳ, J., Rush, K., and Kannan, S. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Khodak, M., Balcan, M.-F. F., and Talwalkar, A. S. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, pp. 5915–5926, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Feddane: A federated newton-type method. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 1227–1231. IEEE, 2019.

Li, T., Hu, S., Beirami, A., and Smith, V. Ditto: Fair and robust federated learning through personalization. *arXiv: 2012.04221*, 2020.

Liang, P. P., Liu, T., Ziyin, L., Salakhutdinov, R., and Morency, L.-P. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Mitra, A., Jaafar, R., Pappas, G. J., and Hassani, H. Achieving linear convergence in federated learning under objective and systems heterogeneity. *arXiv preprint arXiv:2102.07053*, 2021.

Park, D., Kyrillidis, A., Caramanis, C., and Sanghavi, S. Finding low-rank solutions via nonconvex matrix factorization, efficiently and provably. *SIAM Journal on Imaging Sciences*, 11(4):2165–2204, 2018.

Pathak, R. and Wainwright, M. J. Fedsplit: An algorithmic framework for fast federated optimization. *arXiv preprint arXiv:2005.05238*, 2020.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Reisizadeh, A., Tziotis, I., Hassani, H., Mokhtari, A., and Pedarsani, R. Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity. *arXiv preprint arXiv:2012.14453*, 2020.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. In *Advances in neural information processing systems*, pp. 4424–4434, 2017.

Tripuraneni, N., Jin, C., and Jordan, M. I. Provable meta-learning of linear representations, 2020.

Tu, S., Boczar, R., Simchowitz, M., Soltanolkotabi, M., and Recht, B. Low-rank solutions of linear matrix equations via procrustes flow. In *International Conference on Machine Learning*, pp. 964–973. PMLR, 2016.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*, 2020.

Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., and Ramage, D. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

Yu, T., Bagdasaryan, E., and Shmatikov, V. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

Zheng, Q. and Lafferty, J. Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*, 2016.

Zhong, K., Jain, P., and Dhillon, I. S. Efficient matrix sensing using rank-1 gaussian measurements. In *International conference on algorithmic learning theory*, pp. 3–18. Springer, 2015.