

A Study of Enhancing Federated Learning on Non-IID Data with Server Learning

Van Sy Mai, Richard J. La, and Tao Zhang, *Fellow, IEEE*

Abstract—Federated Learning (FL) has emerged as a means of distributed learning using local data stored at clients with a coordinating server. Recent studies showed that FL can suffer from poor performance and slower convergence when training data at the clients are not independent and identically distributed (IID). Here, we consider auxiliary server learning as a *complementary* approach to improving the performance of FL on non-IID data. Our analysis and experiments show that this approach can achieve significant improvements in both model accuracy and convergence time even when the dataset utilized by the server is small and its distribution differs from that of the clients’ aggregate data. Moreover, experimental results suggest that auxiliary server learning delivers benefits when employed together with other techniques proposed to mitigate the performance degradation of FL on non-IID data.

Impact Statement—Federated learning (FL) – a novel and promising distributed machine learning framework – has been shown to degrade in performance considerably when the data at the clients are not independent or have different distributions as is often the case in practice. For this reason, improving the performance of FL in such situations is crucial to its wide deployment. The incremental server learning approach described and analyzed in this paper is proven to enhance the performance significantly under some conditions, with the help of a small dataset accessible to the server. Thus, this approach, alone or together with other complementary approaches available in the literature, can help alleviate the shortcoming of FL in practice, thereby making FL more widely applicable.

Index Terms—Distribute Machine Learning, Federated Learning, Non-IID Data

I. INTRODUCTION

Federated Learning (FL) is a recent paradigm in which multiple clients collaborate under the coordination of a central server to train machine learning (ML) models [16]. A key advantage of FL is that clients need not send their local data to any central sever or share their data with each other. Performing learning where the data is generated is becoming necessary as a large and growing amount of data is created at the network edge and cannot all be forwarded to a central location due to many factors such as network capacity constraints, latency requirements, and data privacy concerns [5].

In its basic form, FL trains a global model for all clients based on the following high-level iterative procedure. At each global round: 1) the central server selects a subset of clients and shares the current global model with them, 2) each

selected client updates the model using only its local data and forwards the updated model to the server, and 3) the server aggregates the updated local models from the clients to update the global model. This process is repeated until certain convergence criteria are satisfied.

Background: Conventional FL techniques, such as the well-known Federated Averaging (FedAvg) algorithm [25], carry out model aggregation by averaging the model parameters received from the clients. This performs well when clients have access to independent and identically distributed (IID) training samples. In practice, however, the local data available to the clients often do not satisfy this IID assumption for different reasons. For instance, clients may collect data from different sources, using different tools, under different conditions, or only have access to partial or biased data, which can cause the distributions of the samples or features at different clients to differ considerably. Such divergences are also referred to as *drifts* or *shifts*, and can take different forms [16].

Large divergences can cause conventional FL techniques to suffer from poor model performance and slow convergence [7], [11], [15], [17], [21], [37]. For example, feature divergence, where the distributions of features differ at different clients, may cause local models to focus on different features or even use different feature representations. Non-IID training data can also cause clients to optimize their local models toward local optima that can differ significantly from global optima. This can further cause the weights of clients’ local models to diverge [24], [37]. As a result, simply averaging local models may not move the aggregated model toward a global optimum.

Recently, growing efforts have been devoted to improving FL performance for non-IID data [23]. The following are several representative categories of approaches.

- **Personalization:** Clients, individually or in groups, personalize their models to perform well on their local data [4], [7], [12], [13], [19], [20], [30]. Many practical applications, however, desire a common model for all clients. E.g., consider autonomous vehicles (AVs) in different regions learning to recognize stop signs. The snow-covered stop signs in northeast United States can look very different from those along the sunny southern country roads. Since cars can travel anywhere, they will benefit from a model that can work well everywhere.

- **Changing how clients learn or contribute:** Several approaches aim to better align the objectives of clients that can diverge due to non-IID training data, e.g., [27], [34]. Clients may use Batch Normalization to alleviate local model divergence caused by non-IID data [22]. Various methods have also been proposed to select a subset of clients to participate in each round to counterbalance distribution shifts [29], [36].

V. S. Mai and T. Zhang are with the National Institute of Standards and Technology (NIST), Gaithersburg, MD 20899, USA (email: {vansy.mai, tao.zhang}@nist.gov). R.J. La is with NIST and the University of Maryland, College Park, MD 20742, USA (email: hyongla@umd.edu). *Corresponding author: Van Sy Mai.*

U.S. Government work not protected by U.S. copyright.

- Changing how the server aggregates local models: This approach alters the aggregation method of local models based on, e.g., their distances to an estimated global model baseline [32], or additional client states or control variates [17].

- Lifelong learning techniques: These techniques treat the learning at each client as a separate task and learn these tasks sequentially using a single model without forgetting the previously learned tasks [16].

Motivation: Our main observation is that most existing studies do not consider the server as a learner with access to own data. In practice, however, the server can and often have access to some training data that are often more representative than individual client data. For example, the server may (i) receive data from sensors and testing devices that do not participate in the learning process, (ii) acquire some raw data directly from the FL clients, or (iii) collect synthetic data obtained from simulation, emulation, and digital twins. The goal of our study is to understand how server learning (SL) can enhance the performance of FL on non-IID data.

Consider the example of AVs which need ML models to recognize objects. Today, two main sources of data are used to train and test such models. First, test vehicles are used to scout selected areas to collect real-world data. Note that this typically can be carried out in ways that do not impose privacy concerns. However, it may require a large fleet of test vehicles, take years to accomplish, incur heavy costs, and yet still fail to collect enough data to cover the vast range of possible learning needs [35]. Therefore, the AV industry is increasingly relying on a second source of data – synthetic data – that is typically generated in the cloud to emulate driving environments and scenarios and is used to extend model training and testing scopes. Furthermore, a small fleet of vehicles (test vehicles and/or production AVs) may still be used to collect for the server to complement the data that the FL clients can collect.

It has been shown that sharing a common IID dataset among clients may improve FL performance on non-IID data [16], [20], [37]. But, this method, which we refer to as *FL with data sharing* or simply *data sharing*, increases clients’ workload, making it less suitable for resource-constrained clients. Moreover, it is often impractical to share data directly among clients due to privacy concerns, network bandwidth constraints, and latency requirements. We will show that comparable or better performance can be achieved by having the server learn from the dataset rather than sharing it among clients.

Several recent works have also considered using server data in FL. For example, [28] uses server data for meta-gradient computation in personalized FL. The studies in [8], [14] consider a semisupervised learning setting where the server has labelled IID data while clients have unlabelled data. Studies more closely related to ours include hybrid training [31], mixed FL [1], and FL with server learning [23]. However, [31] analyzes only the case where client data and server data are both IID and requires full client participation in every communication round. Similarly, [1] assumes IID client data and considers server’s role as a regularizer. The work in [23] considers non-IID client data but uses IID data for server. Additionally, these studies only focus on the FedAvg algorithm – they do not consider SL in conjunction with

other methods proposed to mitigate the adverse effects of non-IID data at clients. Thus, the primary focus of our study and analysis is fundamentally different from those in prior works. Here, we build upon [23] to examine the benefits of incremental SL to enhance FL on non-IID data. We provide both analytical results (including convergence analysis and mathematical proofs) and much more extensive experimental results than presented in [23], to demonstrate that SL can be beneficial in many cases, even with non-IID server data.

Contributions: We consider SL as a *complementary approach* to improving the performance of FL, with an emphasis on handling non-IID data. The server collects a small amount of data, learns from it, and distills the knowledge into the global model *incrementally* during the FL process. Although the idea of utilizing SL itself may not be novel, to the best of our knowledge, our work is the first that examines its benefits on non-IID data via a rigorous analysis. Moreover, we will illustrate that SL can be employed with other FL algorithms as well. Our contributions can be summarized as follows:

- Our analysis and experimental studies show that, FedAvg with SL, which we call Federated Learning with Server Learning (FSL), can significantly improve the performance of FedAvg in both final accuracy and convergence time, provided that server data is more representative of the aggregate data than individual clients’ data. In particular, SL significantly accelerates the learning process when far from convergence. Also, only a *small amount of data* (compared to clients’ aggregate data) is needed at the server for improvements even when its distribution deviates from that of the clients’ aggregate data.

- SL is simple and can be tuned fairly easily. As SL adds only a local learning component to the server, it neither affects clients workload nor increases their per-round communication overhead. Practically, it requires tuning only one additional parameter, namely the weight given to the server loss function.

- Our experiments show that FSL outperforms the data sharing method [37], suggesting that better performance can be achieved without sharing common datasets among clients especially under privacy- and resource-constrained settings. In addition, SL also improves other FL algorithms, including SCAFFOLD [17], FedDyn [9] and FedDC [10] when implemented with suitable changes (see Appendix B).

The rest of the paper is organized as follows. The problem formulation and FSL are described in Section II. Main convergence results are presented in Section III, followed by experimental evaluations in Section IV. We conclude in Section V and provide proofs in Appendix A and additional experimental results in Appendix C and Supplementary Material.

Notation: For any integer $n > 0$, let $[n] := \{1, \dots, n\}$. For a finite set \mathcal{D} , $|\mathcal{D}|$ denotes its cardinality. For any $x \in \mathbb{R}^n$, $\|x\|$ denotes its 2-norm. We denote by $\langle x, y \rangle$ the inner product of x and y . A function $f : D \rightarrow \mathbb{R}$ is L -smooth if $f(x) - f(y) - \langle \nabla f(y), x - y \rangle \leq \frac{L}{2} \|x - y\|^2, \forall x, y \in D$. Both $\mathbb{E}[X]$ and $\mathbb{E}X$ denote the expected value of random variable X .

II. PROBLEM FORMULATION AND OUR APPROACH

We first present our problem formulation in connection with the idea of data sharing, and then delineate our approach.

A. Problem Formulation

Suppose $\mathcal{D} = \{s_i\}_{i=1}^n$ is the set of training samples on which we wish to learn a model by minimizing the following empirical loss:

$$\min_{x \in \mathbb{R}^d} F(x) \triangleq \frac{1}{n} \sum_{i \in [n]} \ell(x, s_i), \quad (1)$$

where $x \in \mathbb{R}^d$ is the vector of model parameters, and $\ell(x, s_i)$ is the loss for sample s_i under model x .

In FL, the goal remains the same, which is to minimize the total loss, but training data are distributed at multiple clients. Suppose that there are N clients and the dataset \mathcal{D} is partitioned into $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N\}$, where \mathcal{D}_i is the local dataset at client i . For each $i \in [N]$, define $n_i := |\mathcal{D}_i|$ and $f_i(x) := \frac{1}{n_i} \sum_{s \in \mathcal{D}_i} \ell(x, s)$ to be the loss function of client i over its own dataset \mathcal{D}_i under model x . Then, problem (1) can be reformulated as follows with $p_i = \frac{n_i}{n}$ for all $i \in [N]$:

$$\min_{x \in \mathbb{R}^d} F(x) = \sum_{i \in [N]} p_i f_i(x). \quad (2)$$

Suppose that the server has access to a dataset \mathcal{D}_0 with $n_0 = |\mathcal{D}_0|$. In [37], a subset of IID samples in \mathcal{D}_0 is shared with *all* clients and not utilized by the server. Each client i follows the FedAvg algorithm using the augmented dataset $\mathcal{D}'_i = \mathcal{D}_i \cup \mathcal{D}_0$.¹ As a result, the problem in (1) is modified as follows to reflect the change in clients' datasets:

$$\min_{x \in \mathbb{R}^d} F'(x) = \frac{1}{n+Nn_0} \left(\sum_{s \in \mathcal{D}} \ell(x, s) + N \sum_{s' \in \mathcal{D}_0} \ell(x, s') \right)$$

Similar to (2), we can express this problem as follows:

$$\min_{x \in \mathbb{R}^d} F'(x) = \sum_{i \in [N]} p'_i f'_i(x), \quad (3)$$

where $f'_i(x) = \frac{1}{n_i+n_0} \sum_{s \in \mathcal{D}'_i} \ell(x, s)$ is the modified loss of client i , and $p'_i = \frac{n_i+n_0}{n+Nn_0}$ is the corresponding weight.

Let $f_0(x) := \frac{1}{n_0} \sum_{s \in \mathcal{D}_0} \ell(x, s)$ be the loss for the samples in \mathcal{D}_0 . Thus, the new loss function F' can be rewritten as

$$F' = \frac{n}{n+Nn_0} \left(F + \frac{Nn_0}{n} f_0 \right). \quad (4)$$

This implies that the above data sharing method alters the global objective by adding the loss function f_0 for the shared samples with a weight of $\frac{Nn_0}{n}$. It also suggests that the quality of the solution obtained from (3), relative to the original problem in (2), depends on how similar F and f_0 are. More importantly, it shows that sharing the samples in \mathcal{D}_0 with clients may be unnecessary; instead, the server can learn from \mathcal{D}_0 and combine its learned model with clients' models in a federated fashion. Having the server learn, rather than sharing training samples among the clients, avoids practical issues such as extra communication overheads, long and unpredictable network delays, and privacy concerns. It also allows us to choose the weight for f_0 , which we denote by γ , to be different from $\frac{Nn_0}{n}$, based on the quality of \mathcal{D}_0 . This leads to the following (centralized) optimization problem:

$$\min_{x \in \mathbb{R}^d} F(x) + \gamma f_0(x) = \left(\sum_{i \in [N]} p_i f_i(x) \right) + \gamma f_0(x) \quad (5)$$

This formulation can also be used for the case with expected losses $f_i(x) = \mathbb{E}_{z \sim \mathcal{D}_i} [f_i(x; z)]$ where \mathcal{D}_i is the corresponding data distribution and $p = (p_i; i \in [N])$ is a probability vector.

¹For simplicity, we either assume that $\mathcal{D}_i \cap \mathcal{D}_0 = \emptyset$ or consider any dataset as a multiset. Thus, we can write $|\mathcal{D}'_i| = |\mathcal{D}_i| + |\mathcal{D}_0|$.

B. Proposed Approach

We assume that the server has access to dataset \mathcal{D}_0 and will augment FL with what the server learns over \mathcal{D}_0 . There are several ways to incorporate SL into FL. One is to treat the server as a regular client that participates in every round of FL process [33]: During each global round, the server updates the current global model using \mathcal{D}_0 and then aggregates it with the updated models reported by the clients. We call this approach *non-incremental SL*. One issue with non-incremental SL is that the weight for the server would be very small when $n_0 \ll n$, which means that the server's contributions, based on its learning from \mathcal{D}_0 , to the global model will be minor. Moreover, this approach fails to exploit the good quality of \mathcal{D}_0 , especially when its distribution is close to that of \mathcal{D} .

These observations motivate us to consider an *incremental learning* scheme where the server performs additional learning over dataset \mathcal{D}_0 to improve the aggregated model of participating clients in each round. We explore this idea in the following two directions, depending on the underlying FL algorithm:

- *SL as correction*: the server updates the global model in the same fashion as clients and follows an incremental gradient method [2]. This applies to FedAvg and SCAFFOLD.
- *SL as regularization*: the server provides regularization steps, e.g., by following the ADMM approach [3]. This applies to FedDyn and FedDC for instance.

In the following, we focus on FedAvg with incremental SL, namely FSL shown in Algorithm 1 below, for our analysis. For the other three aforementioned algorithms augmented with SL, we provide basic ideas and preliminary implementations in Appendix B; we leave their analyses for future work and, instead, report experimental results here.

C. FSL Algorithm

Lines 1–7 of Algorithm 1 are the same as the FedAvg algorithm [25], where in each global round t , each selected client i (1) receives the current global model x_t from the server, (2) performs K steps of the Stochastic Gradient Descent (SGD) algorithm using its local data \mathcal{D}_i (LOCALSGD) with learning rate η_t , and (3) returns to the server its latest update. The server then combines its current model x_t with the updates from the clients using some weight $\eta_g > 0$ (lines 7); note that FedAvg uses $\eta_g=1$. The server then uses the resulting updated model to learn locally by performing K_0 steps of LOCALSGD with learning rate η_0 (line 8). Clearly, FSL has the same computation and communication costs at the clients as the FedAvg algorithm. Our convergence analysis of FSL in Section III will shed light on the selection of its parameters.

Before presenting a formal analysis and experimental results, let us provide some insights into FSL. First, FSL is similar to the incremental (stochastic) gradient method, which has been shown to be much faster than the non-incremental version when the model is far from a (locally) optimal point [2]. Second, if the distributions of \mathcal{D}_0 and \mathcal{D} are close, server's loss function f_0 will be similar to the overall loss function F in (1). Thus, when the current model is far from an optimal point, the gradient ∇f_0 will track the global gradient ∇F , even when individual clients' gradients ∇f_i do not follow ∇F closely. Therefore, when the updated model obtained by

Algorithm 1: FSL: FL with Server Learning

Server: initial global model x_0 , learning rates η_g, η_0 , no. steps K_0 , weight γ
Clients: learning rate η_l , no. steps K

- 1 **for** $t = 0, \dots, T - 1$ **do**
- 2 sample a subset \mathcal{S} of clients
- 3 broadcast x_t to clients in \mathcal{S}
- 4 **forall** clients $i \in \mathcal{S}$ **do**
- 5 $x_{t,K}^{(i)} \leftarrow \text{LOCALSGD}(f_i, x_t, \eta_l, K)$
- 6 upload $x_{t,K}^{(i)} \rightarrow \text{Server}$
- 7 $\bar{x}_t \leftarrow x_t + \eta_g \left(\frac{\sum_{i \in \mathcal{S}} x_{t,K}^{(i)}}{|\mathcal{S}|} - x_t \right)$
- 8 $x_{t+1} \leftarrow \text{LOCALSGD}(\gamma f_0, \bar{x}_t, \eta_0, K_0)$

LOCALSGD(f, x, η, K):

- 9 $y_0 = x$
- 10 **for** $k = 0, \dots, K - 1$ **do**
- 11 $g(y_k) \leftarrow$ unbiased estimate of $\nabla f(y_k)$
- 12 $y_{k+1} \leftarrow y_k - \eta g(y_k)$
- 13 **return:** y_K

aggregating clients' updated models does not make (much) progress, ∇f_0 will help improve the updated model. In fact, significant improvements can still be achieved even when the distributions of \mathcal{D}_0 and \mathcal{D} are not very similar as long as their difference is small in relation to the non-IIDness of clients' data. We will elaborate on these points in Section III below.

Finally, we conclude this section with the following remark. As one can see in the FSL algorithm above (as well as SL combined with other algorithms shown in Appendix B), the weight γ is the most important parameter of our proposed approach, reflecting the contribution of server learning to the baseline FL approach. As a result, we will discuss in subsequent sections on how to select/tune this parameter, analytically and numerically in Sections III and IV (as well as Appendix C) respectively.

III. CONVERGENCE RESULTS

In this section, we study the convergence of Algorithm 1. Specifically, we will prove that, under suitable conditions on step sizes, FSL converges to a neighborhood of a stationary point of the following modified loss function

$$\tilde{F} = \frac{1}{1+\gamma} F + \frac{\gamma}{1+\gamma} f_0$$

which is simply the normalized version of that in (5), where the weight $\gamma > 0$ is chosen by the server. The value of γ should depend on the quality of server's dataset \mathcal{D}_0 : when the distribution of \mathcal{D}_0 is close to that of \mathcal{D} , a larger value would offer greater benefits. *But, our analysis presented below does not assume that their distributions are close.* Also, our experimental results in Section IV below demonstrate that FSL can deliver significant benefits even when the two distributions differ considerably. First, we state several assumptions under which our analysis of Algorithm 1 is carried out.

Assumption 1: The server and client's local loss functions $\{f_i\}_{i=0}^N$ are L -smooth on \mathbb{R}^d .

This assumption is standard in the literature and often holds in practice. It also implies that both F and \tilde{F} are L -smooth.

Our next assumption is used to bound the gradient dissimilarity caused by clients' non-IID data; see, e.g., [26].

Assumption 2: There exists a finite constant G such that $\frac{1}{N} \sum_{i \in [N]} \|\nabla f_i(x) - \nabla F(x)\|^2 \leq G^2$ for all $x \in \mathbb{R}^d$.

Here, G bounds the average disparity in the gradients of clients' loss functions and the empirical loss caused by non-IID samples at the clients; the IID case corresponds to $G \rightarrow 0$. Similarly, when the distributions of \mathcal{D}_0 and \mathcal{D} are different, there is a discrepancy between ∇f_0 and ∇F . We use the following assumption to characterize the quality of server data.

Assumption 3: There exists a finite constant $\bar{\xi}$ such that $\|\nabla f_0(x) - \nabla F(x)\|^2 \leq \bar{\xi}^2$ for all $x \in \mathbb{R}^d$.

This assumption does *not* imply that the server data distribution is similar to that of the clients' aggregate data (although this would be an ideal situation). In other words, $\bar{\xi}^2$ is not necessarily small, and *our analysis presented below examines how this bound affects the performance of FSL.*

Note that the uniform bounds in Assumptions 2 and 3 are used to simplify presentation; our analysis only requires the bounds to hold for the generated sequence $\{x_t\}_{t \geq 0}$. This holds, e.g., when $\{x_t\}$ is bounded. Although those bounds are usually unknown, they quantify the extent of non-IIDness in clients' and server's data and facilitate our analysis.

Finally, we assume that the clients and the server can obtain unbiased estimates of their local gradients for updating their local models. This is also standard in stochastic optimization.

Assumption 4: All clients $i \in [N]$ and the server ($i = 0$) have access to unbiased estimates g_i of ∇f_i with variance bounded by σ_i^2 . For simplicity, assume that $\sigma_i = \sigma, \forall i \in [N]$.

Here, σ bounds the variance of noisy estimates for the clients and the server. Note that it is not uncommon in practice that the server has enough computing capability to obtain gradient estimates with small variance. E.g., when n_0 is not too large, the server may utilize all samples to compute the exact gradient for each update, in which case we have $\sigma_0 = 0$.

Let us now briefly describe the idea used to prove the convergence of FSL. For the special case when $N = 1, K = 1$, and $\sigma_i = 0$, FSL reduces to the incremental gradient method. For a general case, we can relate the sequence $\{x_t\}$ generated by FSL to that of a centralized incremental SGD applied to the global loss function \tilde{F} , where the difference between the two sequences is caused by client sampling and local learning steps. By choosing step sizes sufficiently small in connection with the bounds in Assumptions 1–4, we can then bound such differences and establish the convergence of FSL.

Our first result below demonstrates the progress in each round of FSL. Here, we use $\mathbb{E}_t[\cdot]$ to denote the conditional expectation over the randomness at round t and define

$$\rho_s = \frac{N-S}{N-1}, \quad \Psi = \frac{\gamma^2 \sigma_0^2}{K_0} + \frac{\sigma^2}{KS} + \frac{\rho_s G^2}{S}.$$

Theorem 1: Suppose that Assumptions 1–4 hold and

$$K\eta_l\eta_g = K_0\eta_0 \leq \frac{1}{4L} \min\{\eta_g, 1/\gamma, 8/9\}. \quad (6)$$

Then, the following holds for any $t \geq 0$:

$$\begin{aligned} \mathbb{E}_t[\tilde{F}(x_{t+1})] &\leq \tilde{F}(x_t) - K_0\eta_0 h \|\nabla \tilde{F}(x_t)\|^2 \\ &\quad + 5K_0^2\eta_0^2 L\Psi + 8K_0^3\eta_0^3 L^2 \left(\frac{\gamma\kappa}{1+\gamma} \bar{\xi}^2 + \Phi \right), \end{aligned} \quad (7)$$

where $h = \gamma + \frac{1}{2} - K_0\eta_0 L \frac{1+\gamma}{2} (3\gamma + 3 + 16\kappa K_0\eta_0 L)$, $\kappa = \max\{4\gamma^3, 2\eta_g^{-2} + 3\gamma^2\}$, and $\Phi = \gamma^2\Psi + \frac{2\gamma^2}{S}(\frac{\sigma^2}{K} + \rho_s G^2) + \eta_g^{-2}(2G^2 + \frac{\sigma^2}{K})$.

We have the following remarks. First, condition (6) means that the server and the clients use the same effective step size per round, which is sufficiently small in the order of $\mathcal{O}(1/L(1+\gamma))$. Second, by choosing a sufficiently small $K_0\eta_0$, we have $h \geq 1/2$; in fact, it can be shown that if

$$K_0\eta_0 \leq \frac{1}{8L(\gamma+1)} \min\left\{1, \frac{(\gamma+1)^2}{2\kappa}\right\}, \quad (8)$$

then $h \geq \frac{3\gamma+1}{4}$. Thus, when the current model is far from a stationary point and $\|\nabla\tilde{F}(x_t)\|^2$ is large, it is desirable to use large γ . But, if γ is too large, the last two terms in (7) will likely dominate and prevent the algorithm from making significant improvements, potentially causing it to diverge. Although this suggests that one could use a diminishing γ , we consider a fixed γ in the following analysis for simplicity. We can quantify the overall progress of the algorithm as follows.

Theorem 2: Suppose Assumptions 1–4 and condition (6) hold. Let $\mathcal{E}_T = \min_{t \leq T-1} \mathbb{E}\|\nabla F(x_t)\|^2$. Then,

$$h\mathcal{E}_T \leq \frac{\tilde{D}_0}{TK_0\eta_0} + 5K_0\eta_0 L\Psi + 8K_0^2\eta_0^2 L^2 \left(\frac{\gamma\kappa}{1+\gamma}\bar{\xi}^2 + \Phi\right)$$

for any $T > 0$, where $\tilde{D}_0 = \tilde{F}(x_0) - \tilde{F}^*$.

Here, $\Phi = \frac{\gamma^4\sigma_0^2}{K_0} + \phi$ with $\phi = \frac{3\gamma^2}{S}(\rho_s G^2 + \frac{\sigma^2}{K}) + \frac{1}{\eta_g^2}(2G^2 + \frac{\sigma^2}{K})$.

Thus $\rho_s \approx 1$ and $\phi = \Theta((\frac{\gamma^2}{S} + \frac{1}{\eta_g^2})(G^2 + \frac{\sigma^2}{K}))$ when $S \ll N$. As both ϕ and κ decrease in η_g , in principle we can select large η_g to reduce the upper bound in Theorem 2. Here, since we are interested in scenarios where $\gamma = \mathcal{O}(1)$ and $\rho_s \approx 1$, η_g need not be too large either. Based on these observations, let us consider $\eta_g = \Theta(\sqrt{S})$, which gives $\phi = \Theta(\frac{\gamma^2+1}{S}(G^2 + \frac{\sigma^2}{K}))$ and thus $\Phi = \mathcal{O}((\gamma^2 + 1)\Psi)$. Under these conditions and by dividing both sides of the inequality in Theorem 2 by $h = \Omega(\gamma + 1)$ (cf. (8)), we obtain the following result.

Corollary 1: If $\eta_g = \Theta(\sqrt{S})$, $K_0 = \Theta(K)$, $K_0\eta_0 = \Theta(\frac{\sqrt{KS}}{\sqrt{LT}(\gamma+1)})$, and condition (8) hold, then

$$\mathcal{E}_T = \mathcal{O}\left(\frac{\sqrt{L}}{\sqrt{KST}}(\tilde{D}_0 + \frac{M^2}{(\gamma+1)^2}) + \frac{L}{T}\left(\frac{M^2}{1+\gamma} + \frac{\gamma\kappa KS\bar{\xi}^2}{(1+\gamma)^4}\right)\right) \quad (9)$$

with $M^2 = \gamma^2\sigma_0^2 S + \sigma^2 + \rho_s K G^2$.

Let us make the following remarks. First, the sublinear rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$ is to be expected for FL with a nonconvex loss function and is also similar to that of the usual SGD method.

Second, the FedAvg [25] is a special case of FSL with $\gamma = 0$, i.e., without SL. In this case, $M^2 = \sigma^2 + \rho_s K G^2$ and thus $\mathcal{E}_T = \mathcal{O}(\frac{\sqrt{L}}{\sqrt{KST}}(\tilde{D}_0 + M^2) + \frac{L}{T}M^2)$ is large when clients' data is highly nonhomogeneous and G^2 is large. By increasing γ , we can alleviate the adverse effect of non-IID data, as the dependence on G^2 scales as $\mathcal{O}(\frac{G^2}{\sqrt{T}(1+\gamma)^2} + \frac{G^2}{T(1+\gamma)})$, assuming that the last term in (9) is not dominant. This happens when $\bar{\xi}^2$ is small compared to G^2 and γ is not too large, especially in cases of our interest where $\sigma_0 \ll \sigma$ and $\bar{\xi}^2 \ll G^2$. For instance, this is true when the server samples are taken from \mathcal{D} via uniform sampling without replacement as done in [37].²

²This case is only of theoretical interest, where $\bar{\xi}$ tends to decrease with the size of \mathcal{D}_0 according to $\mathbb{E}_{\mathcal{D}_0}\|\nabla f_0(x) - \nabla F(x)\|^2 = (\frac{n}{n_0} - 1)\frac{\sigma_0^2(x)}{n-1}$, where $\sigma_0^2(x) = \frac{1}{n} \sum_{s \in \mathcal{D}} \|\nabla_x \ell(x, s) - \nabla F(x)\|^2$ is the population variance.

Similarly, in the applications such as AVs, the manufacturers can likely ensure that samples collected for the server by test vehicles are more diverse and representative than those of a typical ‘‘single’’ client because the collection process is under their control, suggesting $\bar{\xi}^2 \ll G^2$ in such cases. We will empirically illustrate this scenario in Appendix C-B1 below.

Third, note that $\tilde{D}_0 = \frac{D_0 + \gamma(f_0(x_0) - f_0(\tilde{x}^*))}{1+\gamma}$, where $D_0 = F(x_0) - F(\tilde{x}^*)$ and \tilde{x}^* is any global minimizer of \tilde{F} . If x_0 is chosen far from \tilde{x}^* or a stationary point and the distributions of \mathcal{D}_0 and \mathcal{D} are similar, it is likely that \tilde{D}_0 is large and $\tilde{D}_0 \approx D_0$. On the other hand, if the server pre-trains its model using its own data so as to minimize f_0 , then \tilde{D}_0 can be improved. In fact, when the server dataset is small, overfitting can happen and thus $f_0(x_0) \approx 0$ and $\tilde{D}_0 \leq \frac{D_0}{1+\gamma}$. This shows that both pre-training and increasing γ can help.

Fourth, the first term of the bound in (9) often dominates and scales as $\mathcal{O}(\frac{\tilde{D}_0 + \gamma^2\sigma_0^2 S + \sigma^2}{\sqrt{KST}} + \frac{\rho_s \sqrt{K}G^2}{\sqrt{ST}})$. This implies that while increasing K helps reduce the effect of stochastic noises and initialization, it increases client and server drifts and consequently amplifies the effect of non-IIDness via the terms $\frac{\sqrt{K}G^2}{\sqrt{TS}}$ and $\frac{K\rho_s G^2}{T} + \frac{K\bar{\xi}^2}{T}$. Similarly, increasing S will reduce the dominant term, which scales as $\mathcal{O}(\frac{1}{\sqrt{S}})$, at the cost of slightly increasing the smaller term $\mathcal{O}(\frac{S\bar{\xi}^2}{T})$.

Finally, let us remark on the optimality of the original loss. Since $\|\nabla F(x_t)\|^2 \leq (1 + \gamma)\|\nabla\tilde{F}(x_t)\|^2 + \frac{\gamma}{1+\gamma}\bar{\xi}^2$, it follows that $\min_{t \leq T-1} \mathbb{E}\|\nabla F(x_t)\|^2 \leq (1 + \gamma)\mathcal{E}_T + \frac{\gamma}{1+\gamma}\bar{\xi}^2$. Here, \mathcal{E}_T can be bounded using Corollary 1, while the second term affects the neighborhood to which the model converges. Thus, in principle, one should select γ judiciously to trade off between these two terms. However, we show numerically in the next section that this can be done fairly easily.

IV. EXPERIMENTAL RESULTS

We now illustrate the benefits of SL through experiments using EMNIST [6], CIFAR-10 and CIFAR-100 [18]. Specifically, we will demonstrate that (i) SL is more beneficial than data sharing, and (ii) SL can improve FL algorithms, including FedAvg, SCAFFOLD, FedDyn and FedDC. We employed the usual SGD method for local learning in each algorithm.

Data and Model: For both CIFAR-10 and CIFAR-100, we use 50k samples for training and 10k samples for testing. For EMNIST, we use a balanced dataset with 45 label classes, 108k samples for training and 18k for testing. Similar to [17], [25], we use simple neural networks with 2 convolutional layers and 2-3 dense layers, and cross-entropy loss for training; see Appendix C-A for further details. These models are by no means the state-of-the-art, but are enough for our purpose of illustrations and comparisons.

Implementation and Evaluation For simplicity, we partition training data roughly evenly among N clients. Each client chosen by the server at each round trains its local model for E_c epochs on its local data with batch size B . In FSL, the server also updates its model for E_0 epochs in each round using batch size B_0 , where E_0 and B_0 are selected such that the number of local steps at the server is the same as that of active clients. We also use $\eta_g = \sqrt{S}$ unless stated otherwise.

We consider the following two scenarios (which mimic the settings in [37] and [17], respectively): (1) N is small, and server data is IID and of small size compared to client’s data, and (2) N is large, server data is non-IID, and client data size is relatively small. Numbers reported below are the averages of 3 runs and a rolling window of size 10.

A. Comparison of SL and DS

Following [37], we partition training data evenly among N clients so that each client i has $n_i = \frac{n}{N}$ samples of C label classes with $\frac{n_i}{C}$ samples per label class, selected uniformly at random without replacement from training data. We vary C to study the effect of client data heterogeneity – smaller C means more non-IID and $C = 1$ is a pathological case.

Consider $(N, n_i, n_0) = (10, 5000, 500)$ for CIFAR-10 and $(45, 2400, 225)$ for EMNIST. Here, \mathcal{D}_0 has roughly $\frac{n_0}{C}$ samples per label class, sampled without replacement from \mathcal{D} . We study the role of different parameters in FSL and compare it against regular FedAvg (FL) and DS. We also tested FSL with non-incremental SL (Section II), but put its results in Supplementary Material for reference as it underperforms FSL. Here, we run all algorithms for 1k rounds with $B = B_0 = 100$ and $E_c = 1$. Both DS and FSL use pretraining where the server trains its local model with learning rate of 0.01 for 500 epochs over its data \mathcal{D}_0 . We varied $\gamma \in \{\frac{Nn_0}{n}, 0.5, 1, 1.5, 2\}$; note that when $\gamma = \frac{Nn_0}{n}$, FSL has the same global objective as DS.

Effects of Client Data Distributions: Fig. 1 shows the test accuracy as C varies. We have the following observations.

First, all algorithms achieve similar final accuracies in the IID case ($C = 45$ for EMNIST and $C = 10$ for CIFAR-10) as expected. When client data become more non-IID as C decreases, FL suffers significantly in both accuracy and convergence time, which is expected and reported in the literature. Second, DS greatly improves over FL, but has a similar convergence property: slower learning with wide oscillations. This is to be expected as DS is essentially FL where each client has an additional small set of shared data. Third, in all cases, FSL provides the highest accuracy and fastest convergence with considerable acceleration at the beginning and much smaller oscillations in accuracy, thanks to only a small dataset at the server (which is about 0.21% of training data for EMNIST and 1% for CIFAR-10). Fourth, FSL performs fairly consistently for a range of γ values, suggesting that fine tuning might be unnecessary in this case. Finally, although we use a pretrained model for FSL and DS but not FL, we show in Supplementary Material that similar observations can be obtained when FSL, DS, and FL all use the same initial model. In fact, FSL provides more significant acceleration, even in the IID cases where DS offers little to no benefits over FL.

Benefits of SL: Fig. 2 shows the final accuracy and rise time (i.e., number of rounds to reach 90% final accuracy) of FSL when varying the weight $\gamma > 0$, learning rate η_l , and server data size n_0 . For comparison, we show the numbers of DS at $\gamma=0$.

Role of γ : First, in general, increasing γ from 0 improves the accuracy and convergence time significantly compared to DS. The improvement is more pronounced when comparing to FL. Second, such improvements remain significant over a

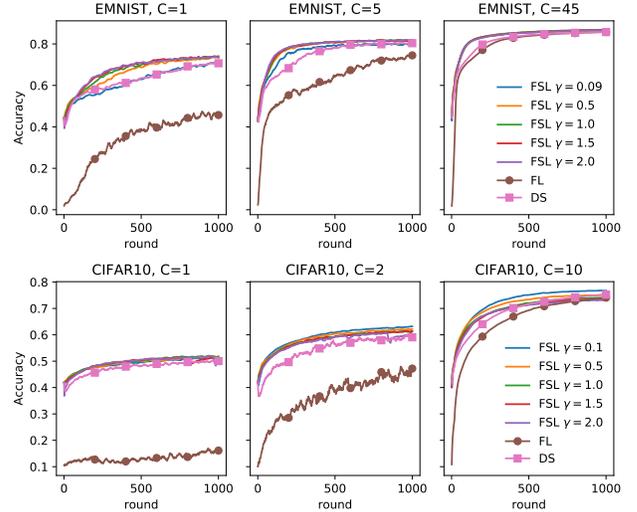


Fig. 1. Test accuracy for varying C and γ , where $(n_0, S, \eta_l) = (225, 5, 0.01)$ for EMNIST, and $(500, 4, 0.01)$ for CIFAR-10. The accuracy of FSL and DS is higher at the beginning (at round 0) because a pretrained model obtained using server data is used as an initial model for FSL and DS (while a random initial model is adopted for FL as no server data is available).

wide range of γ values. For example, γ over $[0.5, 1.5]$ provides similar performance for all considered local learning rates η_l , server data sizes n_0 , and for both datasets. For CIFAR-10, it appears that a smaller γ provides better results, while a large value may slightly degrade the performance; the opposite holds true for EMNIST (except when $C = 1$ and η_l is large, increasing $\gamma > 1$ actually decreases the accuracy). This can be attributed to the fact that the client data are more non-IID and server samples are more dissimilar in CIFAR-10 than in EMNIST; see the cases $C = 2$ and $C = 5$ in Fig. 1.

Server data size: First, with a small (good quality) dataset, the server can already have a pretrained model much better than random initialization. Second, increasing the server data size helps improve FSL further. Here, the accuracy improvement is greater for CIFAR-10 than EMNIST. The rise time improvement is significant when η_l is small and diminishes for larger η_l . Note that increasing the local learning rate η_l also increases the server’s effective learning rate $\gamma\eta_0 = \gamma\eta_l\sqrt{S}$.

B. Benefits of SL with FL Algorithms

We now illustrate the benefits of SL when it is adopted as a *complementary* approach with other FL methods. Even though our analysis in the previous section was carried out only for FedAvg, we also consider SCAFFOLD, FedDyn, and FedDC as baselines. These algorithms augmented with SL are described in Appendix B, which we denote as FedAvg+ (FSL), SCAFFOLD+, FedDyn+, and FedDC+.

We report their performance in a large scale setting with low client participation rates using 3 datasets: $(N, S) = (450, 5)$ for EMNIST, $(1000, 10)$ for CIFAR-10, and $(500, 10)$ for CIFAR-100. Following previous works [9], [10] we use Dirichlet distributions for allocating data among clients to create non-IID settings. Specifically, we set the Dirichlet parameter at 0.1 and 0.3 and refer to these settings as Dir0.1 and Dir0.3. For all algorithms, we use a learning rate of 0.1 for clients (and server). Other hyperparameters are given in Appendix C-A.

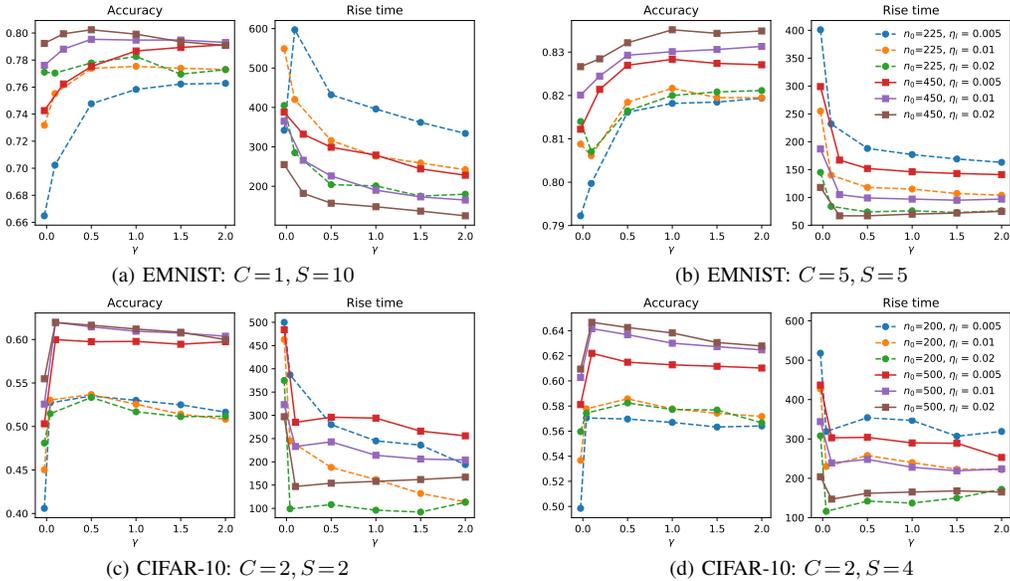


Fig. 2. Test accuracy and rise time (in global rounds) when varying the weight γ , learning rate η_l , and server data size n_0 . Values at $\gamma > 0$ represent FSL and those at $\gamma = 0$ represent DS.

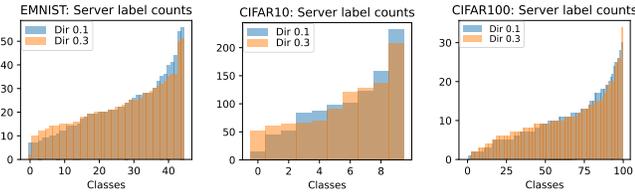


Fig. 3. Sorted label counts of the server in one set of experiments when getting data from 20 clients, each with 50 examples.

For SL, the server obtains data from a subset of $c = 20$ clients,³ and each client provides $s = 50$ samples selected uniformly at random without replacement. This produces non-IID server data in general. But, since the server data is collected from 20 clients, its distribution tends to be closer to that of the aggregate client data than individual clients' data. Fig. 3 shows some examples of label counts of server data in our experiments, which is highly class-imbalanced. Empirical evidence given in Appendix C-B1 also suggests that the bounds in Assumptions 2–3 are likely to satisfy $\xi^2 \ll G^2$. We also showed the benefits of SL when the server makes use of *synthetic* data. Due to a space constraint, these results are omitted here and can be found in Supplementary Material.

Table I shows the final accuracy and the number of rounds needed to reach a target accuracy for all the baseline algorithms and their SL-augmented versions for a fixed choice of γ and without pretraining.⁴ Generally, SL is able to speed up these algorithms and improve their final accuracy despite the fact that server data is rather small and highly imbalanced in both Dir0.1 and Dir0.3 cases. Although there are a few cases in

³These clients can be, for example, test vehicles in our AV example; here they are sampled without replacement once prior to training for simplicity.

⁴Although the original version of SCAFFOLD and FedDC require $2\times$ communication overhead per round compared to FedAvg and FedDyn, we presented another version that requires only $1.5\times$ overhead, where server broadcasts the global control variate but clients do not need to send their local control variable back to server. For simplicity, we reported the number of communication rounds instead of actual total communication overheads.

TABLE I
FINAL ACCURACY AND NUMBER OF ROUNDS TO REACH TARGET ACCURACY T_x . IN FEDAVG+ AND SCAFFOLD+, WE SET $\gamma = 1$ FOR EMNIST AND $\gamma = 0.5$ FOR CIFAR-10/100. IN FEDDYN+ AND FEDDC+, WE USE $\gamma = 0.1$ FOR EMNIST AND $\gamma = 0.05$ FOR CIFAR-10/100.

EMNIST	Dirichlet 0.1			Dirichlet 0.3		
	$T_{.80}$	$T_{.82}$	Acc@400	$T_{.80}$	$T_{.82}$	Acc@400
FedAvg	134	238	83.11 \pm 0.69	81	139	84.41 \pm 0.67
FedAvg+	49	91	84.80 \pm 0.24	35	65	85.51 \pm 0.37
SCAFFOLD	187	248	82.86 \pm 0.98	107	162	84.43 \pm 0.87
SCAFFOLD+	38	65	85.92 \pm 0.24	34	56	86.20 \pm 0.17
FedDyn	97	168	84.08 \pm 0.63	70	101	85.40 \pm 0.48
FedDyn+	53	88	85.39 \pm 0.32	43	70	85.89 \pm 0.30
FedDC	85	119	85.34 \pm 0.39	57	81	86.36 \pm 0.18
FedDC+	47	74	86.00 \pm 0.17	42	62	86.50 \pm 0.19
CIFAR-10	Dirichlet 0.1			Dirichlet 0.3		
	$T_{.65}$	$T_{.70}$	Acc@1.5k	$T_{.65}$	$T_{.70}$	Acc@1.5k
FedAvg	608	1612	68.70 \pm 1.42	354	751	73.09 \pm 0.77
FedAvg+	247	631	73.26 \pm 0.68	133	327	76.45 \pm 0.35
SCAFFOLD	508	1289	67.29 \pm 3.14	295	538	74.58 \pm 2.46
SCAFFOLD+	187	395	73.89 \pm 1.57	120	243	77.86 \pm 0.64
FedDyn	393	738	70.79 \pm 1.55	262	513	74.15 \pm 1.13
FedDyn+	365	702	72.24 \pm 1.27	250	465	75.41 \pm 0.92
FedDC	364	749	70.53 \pm 1.12	233	460	74.36 \pm 0.96
FedDC+	332	677	71.81 \pm 1.31	223	408	75.61 \pm 0.70
CIFAR-100	Dirichlet 0.1			Dirichlet 0.3		
	$T_{.35}$	$T_{.40}$	Acc@2k	$T_{.35}$	$T_{.40}$	Acc@2k
FedAvg	–	–	33.40 \pm 0.58	–	–	34.03 \pm 0.26
FedAvg+	–	–	34.53 \pm 0.57	1463	–	36.26 \pm 0.36
SCAFFOLD	950	1695	40.34 \pm 0.95	992	1753	40.00 \pm 0.76
SCAFFOLD+	679	1432	41.00 \pm 0.58	670	1338	41.70 \pm 0.65
FedDyn	787	1252	43.14 \pm 0.38	758	1147	44.60 \pm 0.45
FedDyn+	753	1229	43.59 \pm 0.41	731	1109	44.61 \pm 0.21
FedDC	806	1525	40.55 \pm 0.61	863	1593	40.47 \pm 0.43
FedDC+	694	963	45.74 \pm 0.56	727	999	45.70 \pm 0.55

Table I where SL does not provide a significant improvement, e.g., FedDyn+ in CIFAR-100Dir0.3, SL does not degrade the performance of any baseline algorithm.

Fig. 4 below shows the test accuracy convergence for different values of γ in the Dir0.1 setting; see also Figs. 6 and 7 in Appendix C-B for the Dir0.3 case and the sensitivity of final

accuracy for a wider range of γ . First, as expected, the benefits of SL vary from one algorithm to another, and also depend on the dataset. Second, it appears that FedAvg and SCAFFOLD generally benefit more from SL than FedDyn and FedDC do. For example, among the plots for CIFAR-10Dir0.1 in Fig. 4, FedAvg+ with $\gamma = 0.1$ actually achieves highest Acc@2k at 74.47 ± 0.72 and requires the least communication overhead with $T_{0.65} = 206$ and $T_{0.70} = 443$, noting that SCAFFOLD+ and FedDC+ incur at least $1.5\times$ more per-round overhead than FedAvg+ and FedDC+; see also footnote 4. Finally, by relying on SL as correction, FedAvg+ and SCAFFOLD+ allow to use a larger range of γ compared to that of FedDyn+ and FedDC+ which use SL as regularization instead.

V. CONCLUSIONS

We considered auxiliary SL as a means of mitigating the performance degradation of FL on non-IID data. Our approach augments FL with SL using a small dataset, and thus is complementary in that it can be utilized in conjunction with other existing approaches. Our analysis and experiments revealed that SL can offer significant improvements in terms of accuracy and convergence time over conventional FL algorithms even when the server dataset is relatively small. As expected, the improvements depend not only on server data size but also on the divergence between its distribution and that of the aggregate training data. We also studied the benefits of SL with other techniques proposed to cope with non-IID data. Our experiments suggest that SL can still provide significant benefits in many cases, e.g., SCAFFOLD+ for EMNIST and CIFAR-10, while closing the performance gap among these techniques. We are currently exploring the relationship between the performance improvements and the server data size/the distributional divergence. Our future work will include more rigorous analyses of SL when combined with FL algorithms other than FedAvg. A better understanding of SL as regularization is also worth investigating to improve further the performance of FedDyn+ and FedDC+.

APPENDIX A PROOFS

Our proofs will use the following technical lemmas.

Lemma 1: If $\{z_1, z_2, \dots, z_m\}$ are zero-mean independent random variables, then $\mathbb{E}[\|\sum_{i=1}^m z_i\|^2] = \mathbb{E}[\sum_{i=1}^m \|z_i\|^2]$.

Lemma 2: (Cauchy-Schwarz) The following holds:

1. $\|v_i + v_j\|^2 \leq (1+a)\|v_i\|^2 + (1+\frac{1}{a})\|v_j\|^2$ for any $a > 0$ and $v_i, v_j \in \mathbb{R}^d$; (CS1)
2. $\|\sum_{i=1}^m v_i\|^2 \leq m \sum_{i=1}^m \|v_i\|^2$ for any $m \in \{1, 2, 3, \dots\}$ and $\{v_1, \dots, v_m\} \subset \mathbb{R}^d$. (CS2)

A. Proof of Theorem 1

To prove the theorem, we first use the smoothness of $\tilde{F} = \frac{1}{1+\gamma}(F + \gamma f_0)$ to find an upper bound on the difference $\mathbb{E}_t[\tilde{F}(x_{t+1})] - \tilde{F}(x_t)$ consisting of two terms in (14). We then bound each term separately by considering the learning steps of clients and the server with the help of (CS1) and (CS2) in Lemma 2.

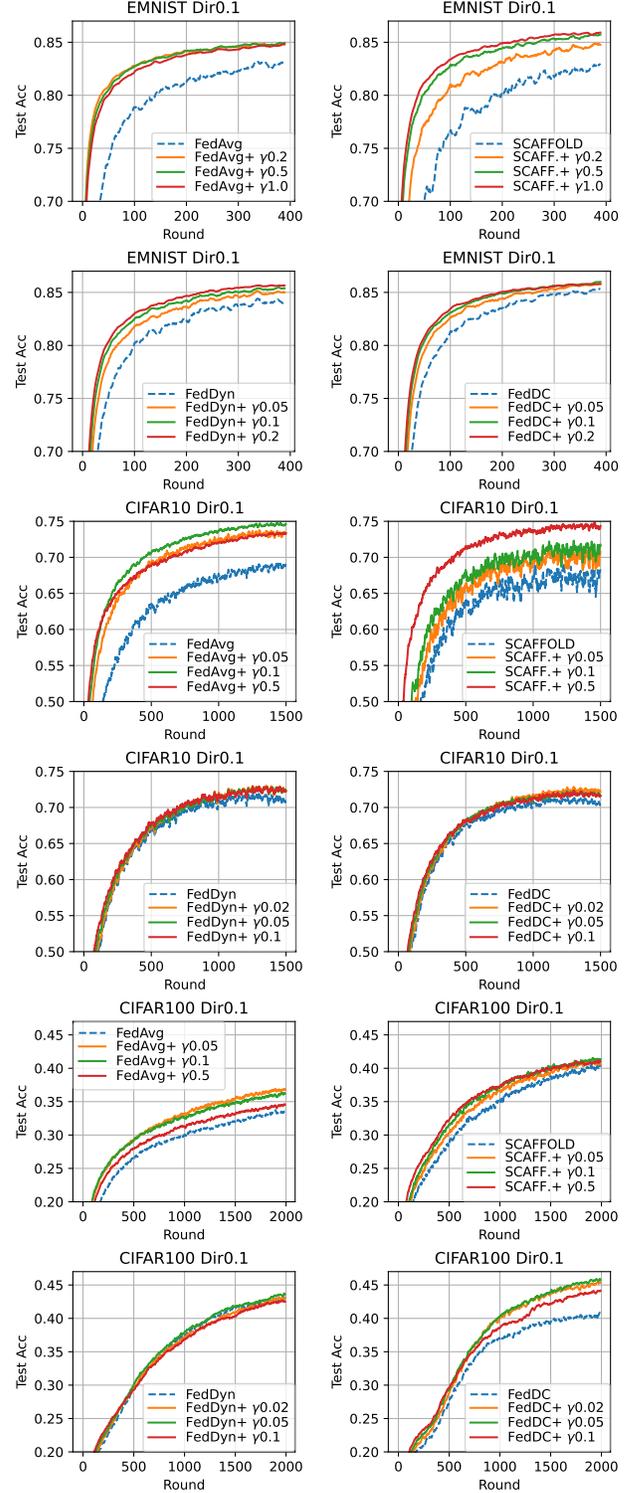


Fig. 4. Test accuracy when using Dirichlet-0.1 data distribution for N clients. Here, $N = 450$ for EMNIST, 1000 for CIFAR-10, and 500 for CIFAR-100. For server learning, Server gets data from 20 clients, each with 50 samples.

Let us rewrite our algorithm as follows. For any $t \geq 1$,

$$\begin{aligned}
 x_{t,k}^{(i)} &= x_{t,k-1}^{(i)} - \eta g_{t,k-1}^{(i)}, & \text{with } x_{t,0}^{(i)} &= x_t, \forall k \in [K], i \in \mathcal{S}_t \\
 \bar{x}_t &= x_t + \frac{\eta g}{S} \sum_{i \in \mathcal{S}_t} (x_{t,K}^{(i)} - x_t) & (10) \\
 w_{t,k} &= w_{t,k-1} - \gamma \eta_0 g_{t,k-1}^{(0)}, & \text{with } w_{t,0} &= \bar{x}_t, \forall k \in [K_0] \\
 x_{t+1} &= w_{t,K_0}
 \end{aligned}$$

where \mathcal{S}_t is the random set of clients chosen at round t with $S = |\mathcal{S}_t|$, $g_{t,k-1}^{(i)}$ is an unbiased estimate of $\nabla f_i(x_{t,k-1}^{(i)})$ for $i \in \mathcal{S}_t$, $g_{t,k-1}^{(0)}$ is an unbiased estimate of $\nabla f_0(w_{t,k-1})$, and the step sizes satisfy $K_0\eta_0 = K\eta_g\eta_l$; see (6). Note that

$$\begin{aligned} x_{t+1} - x_t &= x_{t+1} - \bar{x}_t + \bar{x}_t - x_t \\ &= -K_0\eta_0 \left(\frac{\sum_{k=1}^{K_0} \gamma g_{t,k-1}^{(0)}}{K_0} + \frac{\sum_{k=1}^K \sum_{i \in \mathcal{S}} g_{t,k-1}^{(i)}}{KS} \right). \end{aligned} \quad (11)$$

Let us also define

$$E_t^{(c)} = \mathbb{E}_t \left[\frac{1}{KN} \sum_{i \in [N], k \in [K]} \|x_t - x_{t,k-1}^{(i)}\|^2 \right] \quad \text{and} \quad (12)$$

$$E_t^{(0)} = \mathbb{E}_t \left[\frac{1}{K_0} \sum_{k \in [K_0]} \|x_t - w_{t,k-1}\|^2 \right], \quad (13)$$

where $E_t^{(c)}$ is known as the drift caused by the clients' local updates, while $E_t^{(0)}$ is the correction/drift due to SL in our algorithm. The following results are simply an application of the Lipschitz conditions of ∇f_i for $i = 0, 1, \dots, N$.

Lemma 3: We have the following relations:

$$\mathbb{E}_t \sum_{k \in [K_0]} \|\nabla f_0(x_t) - \nabla f_0(w_{t,k-1})\|^2 \leq K_0 L^2 E_t^{(0)}$$

$$\mathbb{E}_t \sum_{i \in [N], k \in [K]} \|\nabla f_i(x_t) - \nabla f_i(x_{t,k-1}^{(i)})\|^2 \leq NKL^2 E_t^{(c)}$$

First, the descent lemma for L -smooth function \tilde{F} implies

$$\begin{aligned} &\mathbb{E}_t [\tilde{F}(x_{t+1})] - \tilde{F}(x_t) \\ &\leq \underbrace{\langle \nabla \tilde{F}(x_t), \mathbb{E}_t [x_{t+1}] - x_t \rangle}_{=: T_1} + L \underbrace{\frac{1}{2} \mathbb{E}_t [\|x_{t+1} - x_t\|^2]}_{=: T_2}. \end{aligned} \quad (14)$$

We now bound T_1 and T_2 on the right-hand side (RHS) of (14).

• **Bound for T_1 :** From (6), (11), and $(1+\gamma)\tilde{F} = F + \gamma f_0$,

$$\begin{aligned} T_1 &= \left\langle \nabla \tilde{F}(x_t), \left((1+\gamma)K_0\eta_0 \nabla \tilde{F}(x_t) - \eta_0 \mathbb{E}_t \left[\sum_{k=1}^{K_0} \gamma g_{t,k-1}^{(0)} \right] \right. \right. \\ &\quad \left. \left. - \eta_g \frac{\eta_l}{S} \mathbb{E}_t \left[\sum_{i \in \mathcal{S}, k \in [K]} g_{t,k-1}^{(i)} \right] - (1+\gamma)K_0\eta_0 \nabla \tilde{F}(x_t) \right) \right\rangle \\ &= \left\langle \nabla \tilde{F}(x_t), \left(\eta_0 \gamma \mathbb{E}_t \sum_{k \in [K_0]} (\nabla f_0(x_t) - g_{t,k-1}^{(0)}) \right. \right. \\ &\quad \left. \left. + \eta_g \eta_l \mathbb{E}_t \sum_{k \in [K]} \left(\nabla F(x_t) - \frac{1}{S} \sum_{i \in \mathcal{S}} g_{t,k-1}^{(i)} \right) \right) \right\rangle \quad (15) \\ &\quad - (1+\gamma)K_0\eta_0 \|\nabla \tilde{F}(x_t)\|^2. \end{aligned}$$

Let us define the inner product in (15), namely the first-term on the RHS, to be T_3 . Note that by taking expectation over \mathcal{S} and using Assumption 4, we obtain

$$\mathbb{E}_t \frac{1}{S} \sum_{i \in \mathcal{S}} g_{t,k-1}^{(i)} = \frac{1}{N} \mathbb{E}_t \sum_{i \in [N]} \nabla f_i(x_{t,k-1}^{(i)}) \quad (16)$$

$$\mathbb{E}_t \sum_{k \in [K_0]} g_{t,k-1}^{(0)} = \mathbb{E}_t \sum_{k \in [K_0]} \nabla f_0(w_{t,k-1}). \quad (17)$$

Using (6), (16), (17), $F = \frac{1}{N} \sum_{i \in [N]} f_i$ and then the inequality $2\langle v_1, v_2 \rangle \leq \|v_1\|^2 + \|v_2\|^2$, T_3 is bounded as follows:

$$\begin{aligned} \frac{2T_3}{K_0\eta_0} &= \mathbb{E}_t \left[2 \left\langle \nabla \tilde{F}(x_t), \frac{\sum_{i,k} (\nabla f_i(x_t) - \nabla f_i(x_{t,k-1}^{(i)}))}{KN} \right. \right. \\ &\quad \left. \left. + \frac{\sum_{k \in [K_0]} \gamma (\nabla f_0(x_t) - \nabla f_0(w_{t,k-1}))}{K_0} \right\rangle \right] \\ &\leq \|\nabla \tilde{F}(x_t)\|^2 + \mathbb{E}_t \left[\left\| \frac{\sum_{i,k} (\nabla f_i(x_t) - \nabla f_i(x_{t,k-1}^{(i)}))}{KN} \right. \right. \\ &\quad \left. \left. + \frac{\sum_k \gamma (\nabla f_0(x_t) - \nabla f_0(w_{t,k-1}))}{K_0} \right\|^2 \right] \quad (18) \end{aligned}$$

By applying (CS2) to (18), we obtain

$$\begin{aligned} \frac{T_3}{K_0\eta_0} &\leq \frac{\|\nabla \tilde{F}(x_t)\|^2}{2} + \mathbb{E}_t \frac{\sum_{i,k} \|\nabla f_i(x_t) - \nabla f_i(x_{t,k-1}^{(i)})\|^2}{KN} \\ &\quad + \mathbb{E}_t \frac{\sum_k \gamma^2 \|\nabla f_0(x_t) - \nabla f_0(w_{t,k-1})\|^2}{K_0} \\ &\leq \frac{1}{2} \|\nabla \tilde{F}(x_t)\|^2 + L^2 E_t^{(c)} + L^2 \gamma^2 E_t^{(0)}, \end{aligned}$$

where the second inequality follows from Lemma 3. Using this bound in (15) yields

$$T_1 \leq K_0\eta_0 (L^2 (E_t^{(c)} + \gamma^2 E_t^{(0)}) - (0.5 + \gamma) \|\nabla \tilde{F}(x_t)\|^2). \quad (19)$$

• **Bound for T_2 :** From (11) and $(1+\gamma)\tilde{F} = F + \gamma f_0$,

$$\begin{aligned} \frac{2T_2}{3K_0^2\eta_0^2} &= \frac{1}{3} \mathbb{E}_t \left\| \frac{\sum_{k \in [K_0]} \gamma g_{t,k-1}^{(0)}}{K_0} + \frac{\sum_{k \in [K], i \in \mathcal{S}} g_{t,k-1}^{(i)}}{KS} \right\|^2 \\ &= \frac{1}{3} \mathbb{E}_t \left[\left\| \frac{\sum_{k \in [K_0]} \gamma g_{t,k-1}^{(0)}}{K_0} - \gamma \nabla f_0(x_t) \right. \right. \\ &\quad \left. \left. + \frac{\sum_{k \in [K], i \in \mathcal{S}} g_{t,k-1}^{(i)}}{KS} - \nabla F(x_t) + (1+\gamma) \nabla \tilde{F}(x_t) \right\|^2 \right] \end{aligned}$$

Applying (CS2) to the RHS, we get

$$\begin{aligned} \frac{2T_2}{3K_0^2\eta_0^2} &\leq \gamma^2 \mathbb{E}_t \underbrace{\left\| \left(\frac{1}{K_0} \sum_{k \in [K_0]} g_{t,k-1}^{(0)} \right) - \nabla f_0(x_t) \right\|^2}_{=: T_4} \\ &\quad + \mathbb{E}_t \underbrace{\left\| \left(\frac{1}{KS} \sum_{k \in [K], i \in \mathcal{S}} g_{t,k-1}^{(i)} \right) - \nabla F(x_t) \right\|^2}_{=: T_5} \quad (20) \\ &\quad + (1+\gamma)^2 \|\nabla \tilde{F}(x_t)\|^2. \end{aligned}$$

Below, we first drive a bound for T_4 and then that of T_5 .

We first rewrite T_4 .

$$\begin{aligned} \frac{T_4}{2} &= \frac{1}{2K_0^2} \mathbb{E}_t \left[\left\| \sum_{k \in [K_0]} (g_{t,k-1}^{(0)} - f_0(w_{t,k-1})) \right. \right. \\ &\quad \left. \left. + \sum_{k \in [K_0]} (f_0(w_{t,k-1}) - \nabla f_0(x_t)) \right\|^2 \right] \end{aligned}$$

By applying (CS2) to the RHS, we obtain

$$\begin{aligned} \frac{T_4}{2} &\leq K_0^{-2} \mathbb{E}_t \left\| \underbrace{\sum_{k \in [K_0]} (g_{t,k-1}^{(0)} - \nabla f_0(w_{t,k-1}))}_{=: T_{4a}} \right\|^2 \\ &\quad + K_0^{-2} \mathbb{E}_t \left\| \underbrace{\sum_{k \in [K_0]} (f_0(w_{t,k-1}) - \nabla f_0(x_t))}_{=: T_{4b}} \right\|^2. \end{aligned}$$

From Lemma 1 and Assumption 4,

$$T_{4a} = K_0^{-2} \mathbb{E}_t \sum_{k \in [K_0]} \|g_{t,k-1}^{(0)} - \nabla f_0(w_{t,k-1})\|^2 \leq \frac{\sigma_0^2}{K_0}.$$

Applying (CS2) to T_{4b} and then using the first inequality in Lemma 3 yields

$$T_{4b} \leq \mathbb{E}_t \frac{\sum_{k=1}^{K_0} \|\nabla f_0(w_{t,k-1}) - \nabla f_0(x_t)\|^2}{K_0} \leq L^2 E_t^{(0)}.$$

Using these bounds on T_{4a} and T_{4b} , we get the following bound on T_4 .

$$T_4 \leq 2T_{4a} + 2T_{4b} \leq \frac{2\sigma_0^2}{K_0} + 2L^2 E_t^{(0)} \quad (21)$$

Next, we proceed to bound T_5 .

$$\begin{aligned} K^2 T_5 &= \mathbb{E}_t \left\| \left(\sum_{i \in \mathcal{S}, k \in [K]} \frac{1}{S} g_{t,k-1}^{(i)} \right) - K \nabla F(x_t) \right\|^2 \\ &= \mathbb{E}_t \left\| \frac{1}{S} \sum_{i \in \mathcal{S}, k \in [K]} (g_{t,k-1}^{(i)} - \nabla F(x_t)) \right\|^2 \\ &= \mathbb{E}_t \left[\left\| \frac{1}{S} \sum_{i \in \mathcal{S}, k \in [K]} \left(g_{t,k-1}^{(i)} - \nabla f_i(x_{t,k-1}^{(i)}) + \nabla f_i(x_{t,k-1}^{(i)}) \right. \right. \right. \\ &\quad \left. \left. \left. - \nabla f_i(x_t) + \nabla f_i(x_t) - \nabla F(x_t) \right) \right\|^2 \right]. \end{aligned}$$

Rearranging terms and applying (CS2) yields

$$\begin{aligned} \frac{K^2 T_5}{3} &\leq \mathbb{E}_t \left\| \underbrace{\frac{1}{S} \sum_{i \in \mathcal{S}, k \in [K]} (g_{t,k-1}^{(i)} - \nabla f_i(x_{t,k-1}^{(i)}))}_{=: T_{5a}} \right\|^2 \\ &\quad + \mathbb{E}_t \left\| \underbrace{\frac{1}{S} \sum_{i \in \mathcal{S}, k \in [K]} (\nabla f_i(x_{t,k-1}^{(i)}) - \nabla f_i(x_t))}_{=: T_{5b}} \right\|^2 \\ &\quad + \mathbb{E}_t \left\| \underbrace{\frac{1}{S} \sum_{i \in \mathcal{S}, k \in [K]} (\nabla f_i(x_t) - \nabla F(x_t))}_{=: T_{5c}} \right\|^2. \end{aligned}$$

Each term on the RHS can be bounded as follows. First,

$$T_{5a} = \frac{1}{S^2} \mathbb{E}_t \sum_{i \in \mathcal{S}, k \in [K]} \|g_{t,k-1}^{(i)} - \nabla f_i(x_{t,k-1}^{(i)})\|^2 \leq \frac{K}{S} \sigma^2.$$

Here, the equality follows from Lemma 1, and the inequality is a consequence of Assumption 4. Second, first using (CS2)

and then the L -smoothness and $E_t^{(c)}$ defined in (12), we get

$$\begin{aligned} T_{5b} &\stackrel{(CS2)}{\leq} \frac{K}{S} \mathbb{E}_t \sum_{i \in \mathcal{S}, k \in [K]} \|\nabla f_i(x_{t,k-1}^{(i)}) - \nabla f_i(x_t)\|^2 \\ &\stackrel{(L\text{-smooth.})}{\leq} \frac{KL^2}{S} \mathbb{E}_t \sum_{i \in \mathcal{S}, k \in [K]} \|x_{t,k-1}^{(i)} - x_t\|^2 \\ &= K^2 L^2 \frac{1}{KN} \mathbb{E}_t \sum_{i \in [N], k \in [K]} \|x_{t,k-1}^{(i)} - x_t\|^2 \\ &= K^2 L^2 E_t^{(c)}. \end{aligned}$$

Third, we bound T_{5c} using the formula for the variance of sampling without replacement and Assumption 2.

$$\begin{aligned} T_{5c} &= K^2 \mathbb{E}_t \left\| \frac{\sum_{i \in \mathcal{S}} \nabla f_i(x_t)}{S} - \nabla F(x_t) \right\|^2 \\ &= \frac{K^2}{S} \left(1 - \frac{S}{N} \right) \frac{\sum_{i=1}^N \|\nabla f_i(x_t) - \nabla F(x_t)\|^2}{N-1} \\ &\leq \frac{K^2}{S} \left(1 - \frac{S}{N} \right) \frac{NG^2}{N-1} \stackrel{(\text{Assump. 2})}{\leq} K^2 \tau_s G^2, \end{aligned}$$

where $\tau_s = \frac{(N-S)}{S(N-1)} = \frac{\rho_s}{S}$. Using these three bounds,

$$T_5 \leq \frac{3}{K^2} (T_{5a} + T_{5b} + T_{5c}) \leq \frac{3\sigma^2}{KS} + 3\tau_s G^2 + 3L^2 E_t^{(c)} \quad (22)$$

Using the bounds in (21) and (22) for T_4 and T_5 , respectively, in (20), we obtain the following bound for T_2 .

$$\begin{aligned} T_2 &\leq 1.5 K_0^2 \eta_0^2 \left(\frac{2\gamma^2 \sigma_0^2}{K_0} + 2L^2 \gamma^2 E_t^{(0)} + (1+\gamma)^2 \|\nabla \tilde{F}(x_t)\|^2 \right. \\ &\quad \left. + 3 \left(\frac{\sigma^2}{KS} + L^2 E_t^{(c)} + \tau_s G^2 \right) \right) \quad (23) \end{aligned}$$

Use the bounds in (19) and (23) for T_1 and T_2 , respectively, in (14).

$$\begin{aligned} &\mathbb{E}_t [\tilde{F}(x_{t+1})] - \tilde{F}(x_t) \\ &\leq -K_0 \eta_0 (0.5 + \gamma - 1.5(1+\gamma)^2 K_0 \eta_0 L) \|\nabla \tilde{F}(x_t)\|^2 \\ &\quad + K_0 \eta_0 L^2 \gamma^2 (1 + 3K_0 \eta_0 L) E_t^{(0)} \\ &\quad + K_0 \eta_0 L^2 (1 + 4.5K_0 \eta_0 L) E_t^{(c)} + K_0^2 \eta_0^2 L G_2 \quad (24) \end{aligned}$$

with $G_2 := \frac{3\gamma^2 \sigma_0^2}{K_0} + \frac{9\sigma^2}{2KS} + \frac{9\tau_s G^2}{2}$. Note that the condition in (6) implies $1 + 3K_0 \eta_0 L < 1 + 4.5K_0 \eta_0 L \leq 2$. Consequently,

$$\begin{aligned} &\mathbb{E}_t [\tilde{F}(x_{t+1})] - \tilde{F}(x_t) \\ &\leq -K_0 \eta_0 (0.5 + \gamma - 1.5(1+\gamma)^2 K_0 \eta_0 L) \|\nabla \tilde{F}(x_t)\|^2 \\ &\quad + 2K_0 \eta_0 L^2 \underbrace{(\gamma^2 E_t^{(0)} + E_t^{(c)})}_{=: T_6} + K_0^2 \eta_0^2 L G_2. \quad (25) \end{aligned}$$

To bound T_6 , we make use of the following results for bounding the drift terms above. Their proofs are given in Appendices A-B and A-C below.

Lemma 4: If $K\eta_l \leq \frac{1}{4L}$, then

$$E_t^{(c)} \leq 4K^2 \eta_l^2 \left(\|\nabla F(x_t)\|^2 + \frac{\sigma^2}{2K} + G^2 \right). \quad (26)$$

Lemma 5: Let $G_3 = \frac{\sigma^2}{KS} + \tau_s G^2 + \frac{\gamma^2 \sigma_0^2}{3K_0}$. If $K_0 \eta_0 \gamma \leq \frac{1}{4L}$, then

$$E_t^{(0)} \leq 12K_0^2 \eta_0^2 (L^2 E_t^{(c)} + \|\nabla F(x_t)\|^2 + \frac{4}{3} \gamma^2 \|\nabla f_0(x_t)\|^2 + G_3). \quad (27)$$

Using the results above, we can bound T_6 in (25) as follows.

$$\begin{aligned} T_6 &= \gamma^2 E_t^{(0)} + E_t^{(c)} \\ &\leq \underbrace{(1 + 12K_0^2 \eta_0^2 L_0^2) E_t^{(c)}}_{=: T_7} \\ &\quad + 12K_0^2 \eta_0^2 \gamma^2 \left(\|\nabla F(x_t)\|^2 + \frac{4}{3} \gamma^2 \|\nabla f_0(x_t)\|^2 + G_3 \right), \end{aligned}$$

where $L_0 = \gamma L$. Under condition (6), we have $12K_0^2 \eta_0^2 L_0^2 \leq 1$. Using the bound on $E_t^{(c)}$ in (26),

$$T_7 \leq 2E_t^{(c)} \leq 8K^2 \eta_l^2 \left(\|\nabla F(x_t)\|^2 + \frac{\sigma^2}{2K} + G^2 \right).$$

Using the equality $K_0 \eta_0 = K \eta_l \eta_g$ in the above bound on T_7 ,

$$\begin{aligned} T_6 &\leq 8K_0^2 \eta_0^2 \eta_g^{-2} \left(\|\nabla F(x_t)\|^2 + \frac{\sigma^2}{2K} + G^2 \right) \\ &\quad + 12K_0^2 \eta_0^2 \gamma^2 \left(\|\nabla F(x_t)\|^2 + \frac{4}{3} \gamma^2 \|\nabla f_0(x_t)\|^2 + G_3 \right) \end{aligned}$$

By rearranging the terms on the RHS,

$$\begin{aligned} T_6 &\leq K_0^2 \eta_0^2 \|\nabla F(x_t)\|^2 (8\eta_g^{-2} + 12\gamma^2) \\ &\quad + K_0^2 \eta_0^2 \gamma \|\nabla f_0(x_t)\|^2 (16\gamma^3) \\ &\quad + 4K_0^2 \eta_0^2 \underbrace{(3\gamma^2 G_3 + \eta_g^{-2} (2G^2 + \sigma^2 K^{-1}))}_{=: G_4} \\ &\leq 4\kappa K_0^2 \eta_0^2 (\|\nabla F(x_t)\|^2 + \gamma \|\nabla f_0(x_t)\|^2) + 4K_0^2 \eta_0^2 G_4, \end{aligned}$$

where $\kappa = \max\{4\gamma^3, 2\eta_g^{-2} + 3\gamma^2\}$. Since $\tilde{F} = \frac{1}{(1+\gamma)}(F + \gamma f_0)$, we have $\|\nabla F(x_t)\|^2 + \gamma \|\nabla f_0(x_t)\|^2 = (1 + \gamma) \|\nabla \tilde{F}(x_t)\|^2 + \frac{\gamma}{1+\gamma} \xi^2(x_t)$ with $\xi^2(x_t) = \|\nabla F(x_t) - \nabla f_0(x_t)\|^2$. Therefore,

$$T_6 \leq 4\kappa K_0^2 \eta_0^2 [(1+\gamma) \|\nabla \tilde{F}(x_t)\|^2 + \frac{\gamma}{1+\gamma} \xi^2(x_t)] + 4K_0^2 \eta_0^2 G_4.$$

Applying this bound on T_6 to (25), we get

$$\begin{aligned} &\mathbb{E}_t [\tilde{F}(x_{t+1})] - \tilde{F}(x_t) \\ &\leq -K_0 \eta_0 (0.5 + \gamma - 1.5(1 + \gamma)^2 K_0 \eta_0 L) \|\nabla \tilde{F}(x_t)\|^2 \\ &\quad + K_0^2 \eta_0^2 L G_2 + 2K_0 \eta_0 L^2 T_6 \\ &\leq -\frac{K_0 \eta_0}{2} \left(2\gamma + 1 - K_0 \eta_0 L (1 + \gamma) (3(\gamma + 1) \right. \\ &\quad \left. + 16\kappa K_0 \eta_0 L) \right) \|\nabla \tilde{F}(x_t)\|^2 \\ &\quad + K_0^2 \eta_0^2 L G_2 + 8K_0^3 \eta_0^3 L^2 \left(\frac{\gamma \kappa}{1+\gamma} \xi^2(x_t) + G_4 \right). \quad (28) \end{aligned}$$

The proof is completed by using $G_3 \leq \frac{2}{9} G_2 \leq \Psi$ and $G_4 \leq \Phi$.

B. Proof of Lemma 4

The proof follows the same line of arguments as in the proof of Lemma 8 in [17]; we provide it here for completeness and for later reference in the proof of Lemma 5. For notational

simplicity, we drop the index t in this proof, including conditional expectation \mathbb{E}_t . Clearly, the result holds for $K = 1$ (by Assumptions 2 and 4). Thus, we consider only $K \geq 2$ below.

$$\begin{aligned} \mathbb{E} \|x_k^{(i)} - x\|^2 &= \mathbb{E} \|x_{k-1}^{(i)} - x - \eta_l g_{k-1}^{(i)}\|^2 \\ &\leq \mathbb{E} \|x_{k-1}^{(i)} - x - \eta_l \nabla f_i(x_{k-1}^{(i)})\|^2 + \eta_l^2 \sigma^2 \\ &\stackrel{\text{(CS1)}}{\leq} \frac{K}{K-1} \mathbb{E} \|x_{k-1}^{(i)} - x\|^2 + K \eta_l^2 \|\nabla f_i(x_{k-1}^{(i)})\|^2 + \eta_l^2 \sigma^2 \\ &\stackrel{\text{(CS2)}}{\leq} \frac{K}{K-1} \mathbb{E} \|x_{k-1}^{(i)} - x\|^2 + 2K \eta_l^2 \|\nabla f_i(x_{k-1}^{(i)}) - \nabla f_i(x)\|^2 \\ &\quad + 2K \eta_l^2 \|\nabla f_i(x)\|^2 + \eta_l^2 \sigma^2 \\ &\leq \left(\frac{K}{K-1} + 2K \eta_l^2 L^2 \right) \mathbb{E} \|x_{k-1}^{(i)} - x\|^2 \\ &\quad + 2K \eta_l^2 \|\nabla f_i(x)\|^2 + \eta_l^2 \sigma^2 \\ &\leq (1+a) \mathbb{E} \|x_{k-1}^{(i)} - x\|^2 + \eta_l^2 (2K \|\nabla f_i(x)\|^2 + \sigma^2), \end{aligned}$$

where $a := \frac{1.125}{K-1}$, the first inequality is a consequence of Assumption 4, the fourth inequality follows from the L -smoothness in Assumption 1, and the last inequality holds because $4K \eta_l L \leq 1$ (eq. (6) in Theorem 1) and $2K \eta_l^2 L^2 = \frac{(4K \eta_l L)^2}{8K} < \frac{1}{8(K-1)}$ for any K . Unrolling the relation above

$$\begin{aligned} \mathbb{E} \|x_k^{(i)} - x\|^2 &\leq \eta_l^2 (2K \|\nabla f_i(x)\|^2 + \sigma^2) \sum_{k=0}^{K-1} (1+a)^k \\ &\leq 1.85K \eta_l^2 (2K \|\nabla f_i(x)\|^2 + \sigma^2), \quad (29) \end{aligned}$$

where the last inequality holds since $a = \frac{1.125}{K-1}$ and $\sum_{k=0}^{K-1} \frac{(1+a)^k}{K} = \frac{(1+a)^K - 1}{aK} < \frac{e^{1.125} - 1}{1.125} < 1.85$ for any $K \geq 2$. Here, the first bound on $\frac{(1+a)^K - 1}{aK}$ follows from the observation that it is increasing in K and converges to the bound $\frac{e^{1.125} - 1}{1.125}$ in the limit (as $K \rightarrow \infty$). Thus, averaging

the above relation over k and i yields $\mathbb{E} \frac{\sum_{i,k} \|x_k^{(i)} - x\|^2}{3.7KN} \leq K^2 \eta_l^2 \left(\frac{\sum_i \|\nabla f_i(x)\|^2}{N} + \frac{\sigma^2}{2K} \right) \leq K^2 \eta_l^2 (\|\nabla F(x)\|^2 + G^2 + \frac{\sigma^2}{2K})$.

C. Proof of Lemma 5

Note that $w_{t,0} = \bar{x}_t$ and $E_t^{(0)} = \mathbb{E}_t \sum_{k \in [K_0]} \frac{\|x_t - w_{t,k-1}\|^2}{K_0}$.

$$\begin{aligned} E_t^{(0)} &\stackrel{\text{(CS2)}}{\leq} \mathbb{E}_t \frac{2}{K_0} \sum_{k \in [K_0]} \left(\|x_t - \bar{x}_t\|^2 + \|\bar{x}_t - w_{t,k-1}\|^2 \right) \\ &= 2\mathbb{E}_t \|x_t - \bar{x}_t\|^2 + \frac{2}{K_0} \mathbb{E}_t \sum_{k \in [K_0]} \|w_{t,k-1} - w_{t,0}\|^2. \end{aligned}$$

Following the same line of arguments to obtain (29) as in the proof of Lemma 4, we have

$$\sum_{k=1}^{K_0} \frac{\mathbb{E}_t \|w_{t,k-1} - w_{t,0}\|^2}{K_0} \leq 4K_0^2 \eta_0^2 \gamma^2 (\mathbb{E}_t \|\nabla f_0(\bar{x}_t)\|^2 + \frac{\sigma_0^2}{2K_0})$$

Note that, from (CS2) and L -smoothness in Assumption 1,

$$\begin{aligned} \|\nabla f_0(\bar{x}_t)\|^2 &= \|\nabla f_0(x_t) + \nabla f_0(\bar{x}_t) - \nabla f_0(x_t)\|^2 \\ &\leq 2\|\nabla f_0(x_t)\|^2 + 2\|\nabla f_0(\bar{x}_t) - \nabla f_0(x_t)\|^2 \\ &\leq 2\|\nabla f_0(x_t)\|^2 + 2L^2 \|\bar{x}_t - x_t\|^2 \end{aligned}$$

Therefore, since $4K_0\eta_0\gamma L \leq 1$ (see (6)), we have

$$\begin{aligned} E_t^{(0)} &\leq (2 + 4^2 K_0^2 \eta_0^2 L^2 \gamma^2) \mathbb{E}_t \|x_t - \bar{x}_t\|^2 \\ &\quad + 4^2 K_0^2 \eta_0^2 \gamma^2 \|\nabla f_0(x_t)\|^2 + 4K_0 \eta_0^2 \gamma^2 \sigma_0^2 \\ &\leq 3\mathbb{E}_t \left[\|x_t - \bar{x}_t\|^2 \right] + 4K_0 \eta_0^2 \gamma^2 (4K_0 \|\nabla f_0(x_t)\|^2 + \sigma_0^2). \end{aligned}$$

Next, let us consider the term $\mathbb{E}_t \|\bar{x}_t - x_t\|^2$. Recall $\bar{x}_t - x_t = \frac{\eta_g}{S} (\sum_{i \in \mathcal{S}} \sum_{k \in [K]} \eta_i g_{i,k-1}^{(t)})$. Therefore,

$$\begin{aligned} &\frac{1}{K^2 \eta_g^2 \eta_l^2} \mathbb{E}_t \left[\|\bar{x}_t - x_t\|^2 \right] \\ &= \mathbb{E}_t \left\| \left(\frac{1}{KS} \sum_{i \in \mathcal{S}, k \in [K]} g_{i,k-1}^{(i)} \right) - \nabla F(x_t) + \nabla F(x_t) \right\|^2 \\ &\stackrel{(CS1)}{\leq} 4 \|\nabla F(x_t)\|^2 \\ &\quad + \frac{4}{3} \mathbb{E}_t \left[\underbrace{\left\| \left(\frac{1}{KS} \sum_{i \in \mathcal{S}, k \in [K]} g_{i,k-1}^{(i)} \right) - \nabla F(x_t) \right\|^2}_{=T_5 \text{ in (20)}} \right] \end{aligned}$$

$$\stackrel{(\text{eq. (22)})}{\leq} 4((KS)^{-1} \sigma^2 + \tau_s G^2 + L^2 E_t^{(c)} + \|\nabla F(x_t)\|^2).$$

Therefore, $\mathbb{E}_t \left[\|\bar{x}_t - x_t\|^2 \right] \leq 4K_0^2 \eta_0^2 \left(\frac{\sigma^2}{KS} + \tau_s G^2 + L^2 E_t^{(c)} + \|\nabla F(x_t)\|^2 \right)$. Thus, with $G_3 = \frac{\sigma^2}{KS} + \tau_s G^2 + \frac{\gamma^2 \sigma_0^2}{3K_0}$, we have $E_t^{(0)} \leq 12K_0^2 \eta_0^2 (L^2 E_t^{(c)} + \|\nabla F(x_t)\|^2 + \frac{4\gamma^2}{3} \|\nabla f_0(x_t)\|^2 + G_3)$.

D. Proof of Theorem 2

From (7) we obtain $h\mathbb{E} \|\nabla \tilde{F}(x_t)\|^2 \leq \frac{1}{K_0 \eta_0} (\mathbb{E}[\tilde{F}(x_t) - \tilde{F}^*] - \mathbb{E}[\tilde{F}(x_{t+1}) - \tilde{F}^*]) + 5K_0 \eta_0 L \Psi + 8K_0^2 \eta_0^2 L^2 \left(\frac{\gamma \kappa}{1+\gamma} \bar{\xi}^2 + \Phi \right)$. Summing this relation over $t = 0, \dots, T-1$ and then simplifying terms yields the desired result.

APPENDIX B

COMBINING SL WITH OTHER FL ALGORITHMS

We describe an extension of SCAFFOLD, FedDyn, and FedDC with SL. For these algorithms, we first reformulate them using slightly different notation from the original papers. In the case of SCAFFOLD and FedDC, the uplink communication overhead per round is also reduced by half. Then, the incremental SL module is added and highlighted in blue. Our approach is to rely on the idea of the underlying FL algorithm; specifically, we use SL as correction for SCAFFOLD and as regularization for FedDyn and FedDC. We emphasize that our implementations below are not necessarily the best or the only way to incorporate incremental SL; for example, we also attempted to use SL as correction in FedDyn and FedDC but found that to be ineffective.

1) *SCAFFOLD+SL*: An extension of SCAFFOLD is shown in Algorithm 2, where the server maintains an additional local control variate c_0 , which is updated using the global control variate c and its local data. Here, the server updates its local model and control variate in the same fashion as the clients.

2) *FedDyn+SL*: The extension of FedDyn is shown in Algorithm 3. The idea of FedDyn+SL is to follow the ADMM approach [3] using the augmented Lagrangian: $\mathcal{L} = \gamma f_0(x_0) + \frac{1}{N} \sum_{i=1}^N f_i(x_i) - \langle h_i, x_i - x_0 \rangle + \frac{\alpha}{2} \|x_i - x_0\|^2$ with h_i being dual variables, and use γf_0 as a regularizer. Thus, the goal

Algorithm 2: SCAFFOLD+SL

Server: global model x , local and global states $c_0 = c = 0$, learning rates η_0, η_g , no. steps K_0 , weight γ
Clients: local state $c_i = 0$, learning rate η_i , no. steps K

```

1 for  $t = 0, \dots, T-1$  do
2   sample clients  $\mathcal{S} \subset [N]$ 
3   broadcast  $(x, c) \rightarrow$  all  $i \in \mathcal{S}$ 
4   forall clients  $i \in \mathcal{S}$  do
5      $x_i \leftarrow$  LOCALOPT( $f_i, x, \eta_i, c - c_i, K$ )
6      $\Delta x_i \leftarrow x_i - x$  and  $c_i \leftarrow c_i - c - \frac{1}{K\eta_i} \Delta x_i$ 
7     upload  $\Delta x_i \rightarrow$  Server
8    $\Delta x \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \Delta x_i$ 
9    $\Delta c \leftarrow \frac{1}{N} \sum_{i \in \mathcal{S}} (c + \frac{\Delta x_i}{K\eta_i})$ 
10   $x \leftarrow x + \eta_g \Delta x$  and  $c \leftarrow c + \Delta c$ 
11   $x_0 \leftarrow$  LOCALOPT( $f_0, x, \eta_0, c - c_0, K_0$ )
12   $\Delta x_0 \leftarrow x_0 - x$  and  $\Delta c_0 \leftarrow -(c + \frac{1}{K_0 \eta_0} \Delta x_0)$ 
13   $x \leftarrow x + \gamma \Delta x_0$  and  $c \leftarrow c + \gamma \Delta c_0$  and  $c_0 \leftarrow c_0 + \Delta c_0$ 
14  $y_0 \leftarrow x$ 
15 for  $k = 0, \dots, K-1$  do
16    $g(y_k) \leftarrow$  unbiased estimate of  $\nabla f(y_k)$ 
17    $y_{k+1} \leftarrow y_k - \eta(g(y_k) + d_c)$ 
18 return  $y_K$ 

```

LOCALOPT(f, x, η, d_c, K):

```

14  $y_0 \leftarrow x$ 
15 for  $k = 0, \dots, K-1$  do
16    $g(y_k) \leftarrow$  unbiased estimate of  $\nabla f(y_k)$ 
17    $y_{k+1} \leftarrow y_k - \eta(g(y_k) + d_c)$ 
18 return  $y_K$ 

```

of the local learning LOCALOPT in each round is to solve the following two optimization problems in an alternating manner. **Clients:** $x_i \leftarrow \arg \min_y f_i(y) - \alpha \langle h_i, y \rangle + \frac{\alpha}{2} \|y - x_0\|^2$

Server: $x_0 \leftarrow \arg \min_y \gamma f_0(y) + \sum_{i=1}^N \frac{\alpha}{N} \langle h_i, y \rangle + \frac{\alpha}{2N} \|y - x_i\|^2$

Here client i updates its local dual variable h_i but does not send it back to the server. Line 9 in the algorithm allows the server to keep track of $h = \frac{1}{N} \sum_{i=1}^N h_i$ in each round. In addition, due to partial participation, the server approximates the quadratic term $\sum_{i \in [N]} \frac{1}{N} \|y - x_i\|^2$ by $\sum_{i \in \mathcal{S}} \frac{1}{S} \|y - x_i\|^2$. Thus, we recover the FedDyn algorithm by setting $\gamma = 0$. For simplicity and fair comparison, we adopted the usual SGD for LOCALOPT in Algorithm 3.

3) *FedDC+SL*: Note that FedDC can be viewed as a combination of SCAFFOLD and FedDyn. In the extension of FedDC, shown in Algorithm 4, the local learning step LOCALOPT is the same as that in Algorithm 3.

APPENDIX C

SETUP & ADDITIONAL EXPERIMENTAL RESULTS

A. Neural Network Models and Training Parameters

Models: We use simple networks with 2 convolutional layers followed by 2-3 dense layers. All layers use ReLU as activation functions except for the last dense layer that uses softmax. The model structures are as follows:

EMNIST: Conv2d(1,32)–Conv2d(32,64)–MaxPool2d(2)–Dropout(.25)–Dense(128)–Dropout(.5)–Dense(45). Conv2d uses a kernel size of 3 with padding=1.

CIFAR-10: Conv2d(3,32)–MaxPool2d(2)–Conv2d(32,64)–MaxPool2d(2)–Dropout(.25)–Dense(128)–Dropout(.5)–Dense(10). Conv2d uses a kernel size of 3 with padding=1.

CIFAR-100: Conv2d(3,64)–MaxPool2d(2)–Conv2d(32,64)–MaxPool2d(2)–Dense(384)–Dense(192)–Dense(100). Conv2d uses a kernel size of 5 with padding=0.

Training parameters: We use the following:

Algorithm 3: FEDDYN+SL

Server: global model x , local state $h = 0$, learning rate η_0 , no. steps K_0 , weight γ , reg. coef. α
Clients: local state $h_i = 0$, learning rate η_l , no. steps K , reg. coef. α

- 1 **for** $t = 0, \dots, T - 1$ **do**
- 2 sample clients $\mathcal{S} \subset [N]$
- 3 broadcast $x \rightarrow$ all $i \in \mathcal{S}$
- 4 **forall** clients $i \in \mathcal{S}$ **do**
- 5 $x_i \leftarrow \text{LOCALOPT}(f_i, x, \eta_l, h_i - x, \alpha, K)$
- 6 $h_i \leftarrow h_i + (x_i - x)$
- 7 **upload** $x_i \rightarrow$ Server
- 8 $\bar{x} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} x_i$
- 9 $h \leftarrow h + \frac{|\mathcal{S}|}{N} (\bar{x} - x)$
- 10 $x \leftarrow \bar{x} + h$
- 11 $x \leftarrow \text{LOCALOPT}(\gamma f_0, x, \eta_0, -h - x, \alpha, K_0)$

$\text{LOCALOPT}(f, x, \eta, d_x, \alpha, K)$:

- 12 $y_0 \leftarrow x$
- 13 **for** $k = 0, \dots, K - 1$ **do**
- 14 $g(y_k) \leftarrow$ unbiased estimate of $\nabla f(y_k)$
- 15 $y_{k+1} \leftarrow y_k - \eta(g(y_k) + \alpha d_x + \alpha y_k)$
- 16 **return** y_K

Algorithm 4: FEDDC+SL

Server: global model x , local and global states $h = c = 0$, learning rate η_0 , no. steps K_0 , weight γ , reg. coef. α
Clients: local states $h_i = c_i = 0$, learning rate η_l , no. steps K , reg. coef. α

- 1 **for** $t = 0, \dots, T - 1$ **do**
- 2 sample clients $\mathcal{S} \subset [N]$
- 3 broadcast $(x, c) \rightarrow$ all $i \in \mathcal{S}$
- 4 **forall** clients $i \in \mathcal{S}$ **do**
- 5 $x_i \leftarrow \text{LOCALOPT}(f_i, x, \eta_l, \frac{c - c_i}{\alpha} + h_i - x, \alpha, K)$
- 6 $h_i \leftarrow h_i + x_i - x$
- 7 $c_i \leftarrow c_i - c - \frac{1}{K\eta} (x_i - x)$
- 8 **upload** $x_i \rightarrow$ Server
- 9 $\bar{x} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} x_i$
- 10 $h \leftarrow h + \frac{|\mathcal{S}|}{N} (\bar{x} - x)$
- 11 $c \leftarrow c - \frac{|\mathcal{S}|}{N} (c + \frac{1}{K\eta} (\bar{x} - x))$
- 12 $x \leftarrow \bar{x} + h$
- 13 $x \leftarrow \text{LOCALOPT}(\gamma f_0, x, \eta_0, -h - x, \alpha, K_0)$

EMNIST: $B = 50$, $E_c = 5$, $B_0 = 200$, $E_0 = 5$, $S = 5$, weight decay 10^{-4} , no learning rate decay.

CIFAR-10: $B = 25$, $E_c = 5$, $B_0 = 200$, $E_0 = 2$, $S = 10$, weight decay 10^{-3} , learning rate decay 0.998. For FedDyn+ and FedDC+, γ is reduced by a factor of 0.99 every global round. FedAvg+ and SCAFFOLD+ use global step size $\eta_g = 2$.

CIFAR-100: $B = 50$, $E_c = 5$, $B_0 = 200$, $E_0 = 2$, $S = 10$, weight decay 10^{-3} , no learning rate decay. FedAvg+ and SCAFFOLD+ use global step size $\eta_g = 2$.

B. Additional Experiments

1) *Quantifying the Non-IIDness of Client and Server Data:* Generally, it is difficult to obtain uniform bounds $\bar{\xi}$ and G in Assumptions 2 and 3. In Fig. 5, we instead show empirical bounds $\xi_t^2 := \|\nabla f_0(x_t) - \nabla F(x_t)\|^2$ and $G_t^2 := \frac{1}{N} \sum_{i \in [N]} \|\nabla f_i(x_t) - \nabla F(x_t)\|^2$ when using FedAvg+ with $\gamma = 1$ for EMNIST and $\gamma = 0.5$ for CIFAR-10/100, where

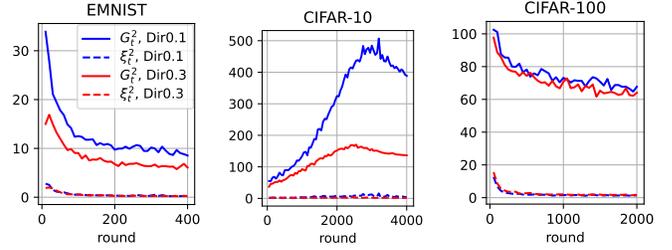


Fig. 5. Empirical bounds in Assumptions 2 and 3 in FedAvg+.

the server obtains data from a subset of $c=20$ clients, each with 50 samples. Since server data is highly class-imbalanced in both cases Dir0.1 and Dir0.3 (see Fig. 3), we use simple augmentation (random cropping and rotations) to have at least n_0/n_C samples for each present class, where n_C is the number of present classes. The plots clearly show $\xi_t^2 \ll G_t^2$.

2) *Accuracy with varying γ :* Fig. 6 shows the convergence of accuracy for different values of γ in the Dir0.3 setting. Fig. 7 shows the sensitivity of final accuracy for a wider range of γ in both settings Dir0.1 and Dir0.3. We note the following.

First, as expected, the performance of every algorithm improves as clients' data becomes more IID with a larger Dirichlet parameter. Here, FedAvg and SCAFFOLD can be improved significantly further by SL in both final accuracy and convergence time. In many cases, FedAvg+ and SCAFFOLD+ perform better than FedDyn and FedDC. The improvement in FedDyn+ and FedDC+ is not always significant.

Second, FedAvg+ and SCAFFOLD+ can adopt larger values and a wider range of γ , compared to those in FedDyn+ and FedDC+. We believe this difference stems from different uses of SL – as correction or regularization. In fact, FedDyn and FedDC without SL already use some form of regularization (by adding both linear and quadratic terms to the client loss function according to the augmented Lagrangian approach; see Appendix B above).

Finally, we note that FedAvg+ might underperform FedAvg if the weight γ is too small and the global step size η_g is too large. This is the case for EMNIST plots shown in Fig. 7. The main reason for this is that, for the case of non-IID data with low participation rate, the aggregated update from clients in each round will be far from an optimal direction and their drift is magnified further by a large global step size when $\eta_g > 1$ in FedAvg+ (FedAvg is a special case of FedAvg+ with $\eta_g = 1$ and $\gamma = 0$). Thus, when γ is too small, SL does not produce sufficient correction, leading to possible degradation compared to FedAvg.

REFERENCES

- [1] S. Augenstein, A. Hard, L. Ning, K. Singhal, S. Kale, K. Partridge, and R. Mathews. Mixed federated learning: Joint decentralized and centralized learning. *arXiv preprint arXiv:2205.13655*, 2022.
- [2] D. P. Bertsekas et al. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optim. Mach. Learn.*, 2010(1-38):3, 2011.
- [3] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.
- [4] C. Briggs, Z. Fan, and P. Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *Int. Jt. Conf. Neural Netw.*, pages 1–9. IEEE, 2020.

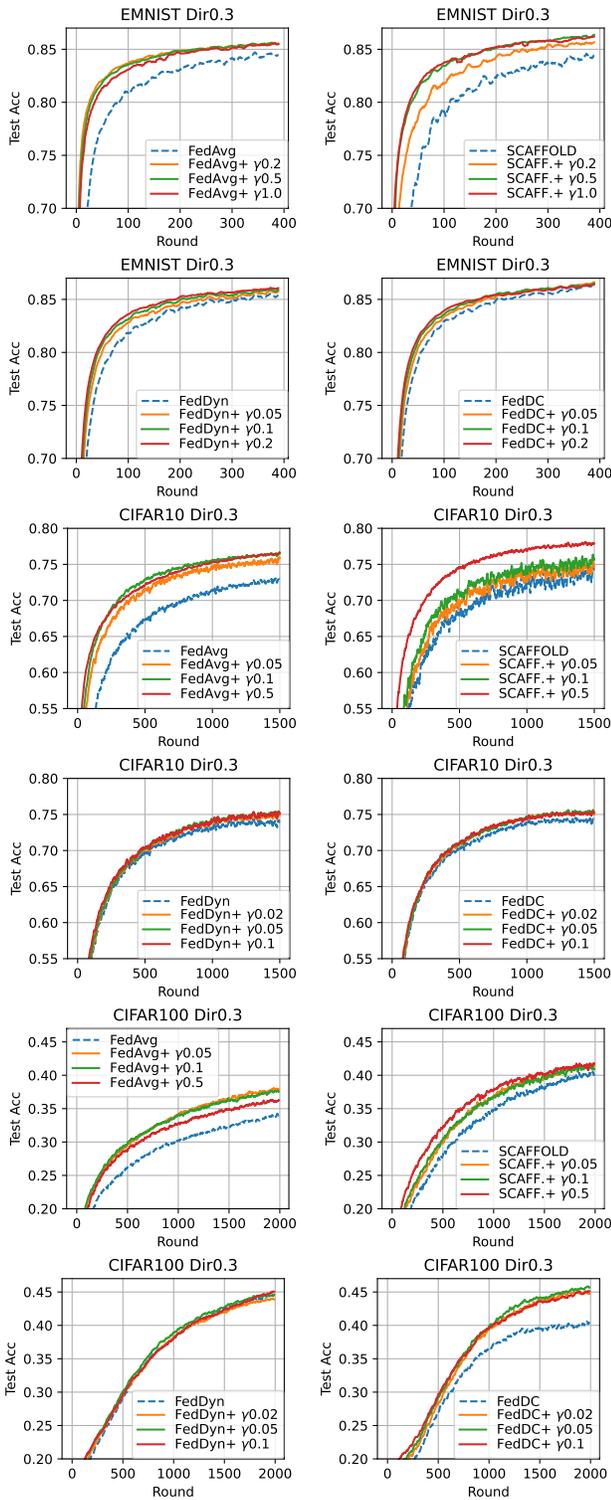


Fig. 6. Test accuracy when using Dirichlet-0.3 data distribution for N clients. Here, $N = 450$ for EMNIST, 1000 for CIFAR-10, and 500 for CIFAR-100. For SL, Server gets data from 20 clients, each with 50 samples.

- [5] M. Chiang and T. Zhang. Fog and IoT: An overview of research opportunities. *IEEE Internet Things J.*, 3(6):854–864, 2016.
- [6] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *Int. Jt. Conf. Neural Netw.*, pages 2921–2926, 2017.
- [7] Y. Deng, M. M. Kamani, and M. Mahdavi. Adaptive personalized federated learning. *arXiv:2003.13461*, 2020.
- [8] E. Diao, J. Ding, and V. Tarokh. SemiFL: Semi-supervised federated

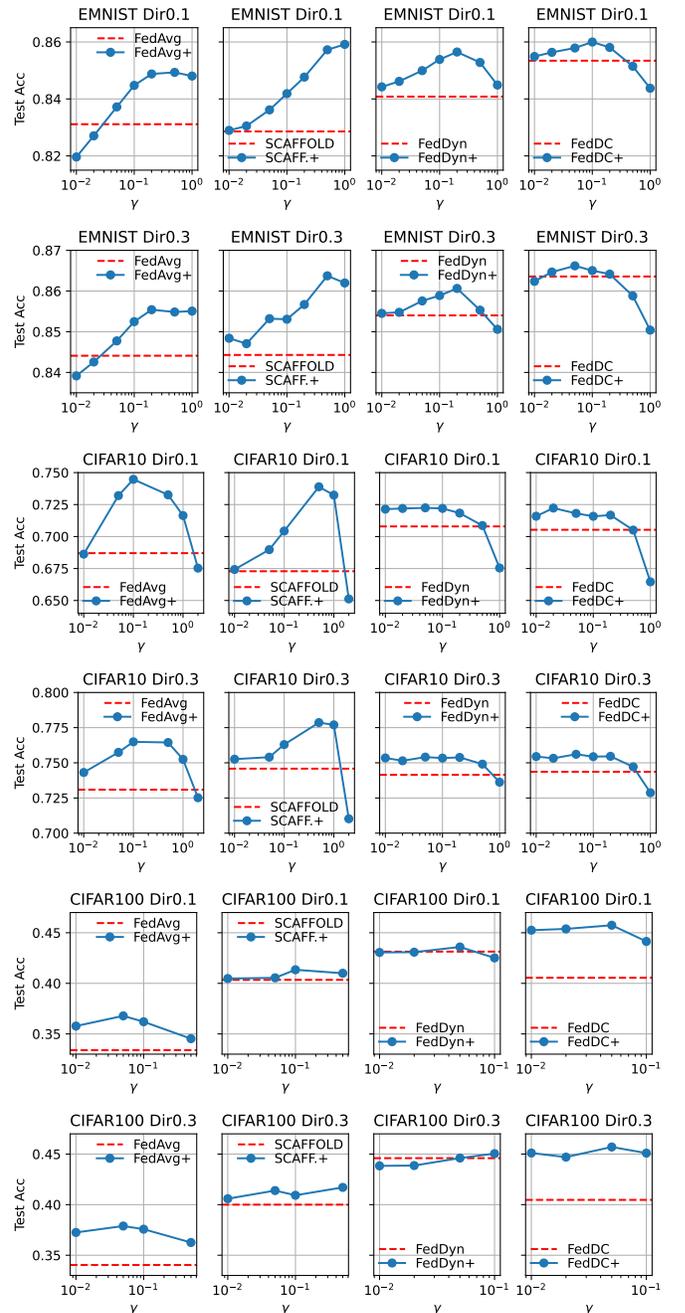


Fig. 7. Accuracy vs. γ after 400 rounds for EMNIST, 1.5k rounds for CIFAR-10, and 2k rounds for CIFAR-100.

learning for unlabeled clients with alternate training. *Adv. Neural Inf. Process. Syst.*, 35:17871–17884, 2022.

- [9] A. E. Durmus, Z. Yue, M. Ramon, M. Matthew, W. Paul, and S. Venkatesh. Federated learning based on dynamic regularization. In *Int. Conf. Learn. Rep. (ICLR)*, pages 1–36, 2021.
- [10] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu. FedDC: Federated learning with non-iid data via local drift decoupling and correction. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 10112–10121, 2022.
- [11] F. Haddadpour and M. Mahdavi. On the convergence of local descent methods in federated learning. *arXiv:1910.14425*, 2019.
- [12] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang. Personalized cross-silo federated learning on non-iid data. In *Proc. AAAI Conf. Artif. Intell.*, volume 35, pages 7865–7873, 2021.
- [13] H. Jamali-Rad, M. Abdizadeh, and A. Singh. Federated learning with taskonomy for non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.*,

- 34(11):8719–8730, 2022.
- [14] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang. Federated semi-supervised learning with inter-client consistency & disjoint learning. *arXiv preprint arXiv:2006.12097*, 2020.
- [15] Y. Jiang, J. Konečný, K. Rush, and S. Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv:1909.12488*, 2019.
- [16] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1–2):1–210, 2021.
- [17] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, pages 5132–5143. PMLR, 2020.
- [18] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- [19] V. Kulkarni, M. Kulkarni, and A. Pant. Survey of personalization techniques for federated learning. In *Proc. 4th World Conf. Smart Trends Syst. Sec. Sustain. (WorldS4)*, pages 794–797, 2020.
- [20] D. Li and J. Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv:1910.03581*, 2019.
- [21] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. *Mach. Learn. Syst.*, 2:429–450, 2020.
- [22] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv:2102.07623*, 2021.
- [23] V. S. Mai, R. J. La, T. Zhang, Y. Huang, and A. Battou. Federated Learning with Server Learning for Non-IID Data. In *Proc. 57th Annu. Conf. Inf. Sci. Syst. (CISS)*, pages 1–6. IEEE, 2023.
- [24] G. Malinovskiy, D. Kovalev, E. Gasanov, L. Condat, and P. Richtarik. From local sgd to local fixed-point methods for federated learning. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 6692–6701. PMLR, 2020.
- [25] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artif. Intell. Stat. (AISTATS)*, pages 1273–1282. PMLR, 2017.
- [26] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.
- [27] N. Shoham, T. Avidor, A. Keren, N. Israel, D. Benditkis, L. Mor-Yosef, and I. Zeitak. Overcoming forgetting in federated learning on non-iid data. *arXiv:1910.07796*, 2019.
- [28] J. Song, M.-H. Oh, and H.-S. Kim. Personalized federated learning with server-side information. *IEEE Access*, 10:120245–120255, 2022.
- [29] H. Wang, Z. Kaplan, D. Niu, and B. Li. Optimizing federated learning on non-iid data with reinforcement learning. In *Proc. IEEE Conf. Compu. Commun. (INFOCOM)*, pages 1698–1707. IEEE, 2020.
- [30] M. Xie, G. Long, T. Shen, T. Zhou, X. Wang, J. Jiang, and C. Zhang. Multi-center federated learning. *arXiv:2108.08647*, 2021.
- [31] K. Yang, S. Chen, and C. Shen. On the convergence of hybrid server-clients collaborative training. *IEEE J. Sel. Areas Commun.*, 41(3):802–819, 2022.
- [32] Y. Yeganeh, A. Farshad, N. Navab, and S. Albarqouni. Inverse distance aggregation for federated learning with non-iid data. In *Proc. Domain Adapt. Represent. Transf. Distrib. Collab. Learn.*, pages 150–159. Springer, 2020.
- [33] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani. Hybrid-FL for wireless networks: Cooperative learning mechanism using non-IID data. In *IEEE Int. Conf. Commun. (ICC)*, pages 1–7. IEEE, 2020.
- [34] L. Zhang, Y. Luo, Y. Bai, B. Du, and L.-Y. Duan. Federated learning for non-iid data via unified feature learning and optimization objective alignment. In *Proc. IEEE/CVF Int. Conf. Compu. Vis. (ICCV)*, pages 4420–4428, October 2021.
- [35] T. Zhang. Toward automated vehicle teleoperation: Vision, opportunities, and challenges. *IEEE Internet Things J.*, 7(12):11347–11354, 2020.
- [36] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang. Client selection for federated learning with non-iid data in mobile edge computing. *IEEE Access*, 9:24462–24474, 2021.
- [37] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. Federated learning with non-IID data. *arXiv:1806.00582*, 2018.

Supplementary Material: A Study of Enhancing Federated Learning on Non-IID Data with Server Learning

In this supplemental material, we provide additional experimental results for the two cases of server data:

- IID data: Servers gets data directly from training data; see Section I below.
- non-IID data: Server gets data either from non-IID clients or from a different source of data; see Section II below.

We will use two datasets EMNIST and CIFAR-10 and partition training data among clients following the scheme in Section IV-A of the paper. In particular, the training data is split evenly among N clients so that each client i has $n_i = \frac{n}{N}$ samples of C label classes with $\frac{n_i}{C}$ samples per label class, selected uniformly at random without replacement from training data. We vary C to study the effect of client data heterogeneity – smaller C means more non-IID and $C = 1$ is a pathological case.

I. SL USING IID DATA

In this section, we will compare FSL with DS and FL (FedAvg) without pre-training for SL in subsection I-A, and compare FSL and DS with the non-incremental version of FSL, denoted by FSP-p in subsection I-B.

A. SL without using Pretrained Model

Figure 1 shows the performance of FSL, FL and DS when they start from a randomly initialized model instead of a pretrained one as in Figure 1 of the paper. Clearly, in this case, the acceleration provided by SL is much more significant, even in the IID cases, in which DS offers little to no benefits as one would expect.

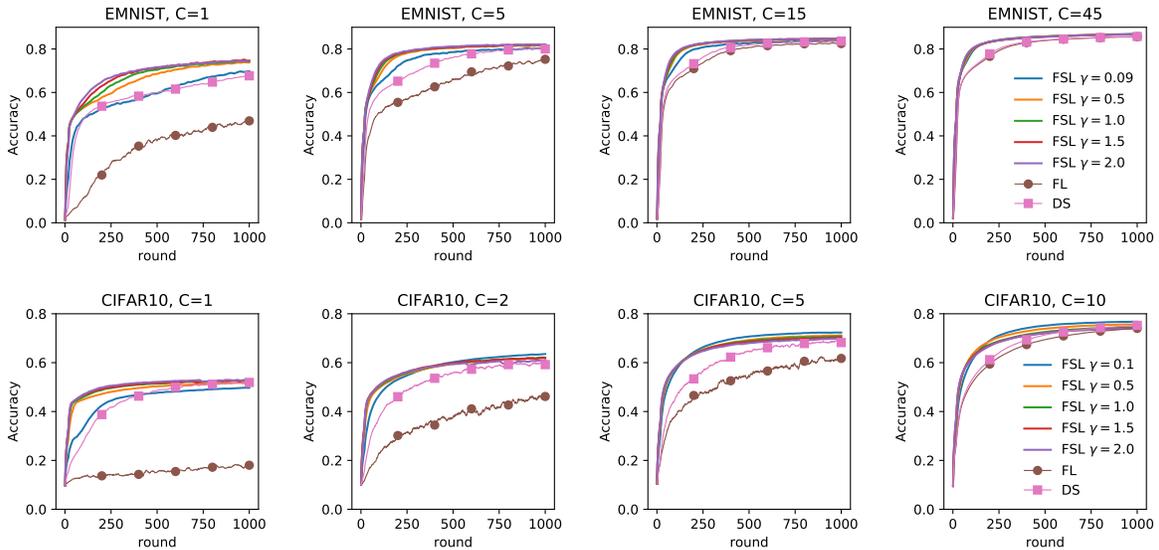


Fig. 1. Test accuracy of FSL, FL and DS when not using server pretrained model. Here, $n_0 = 225$, $S = 5$ and $\eta_l = 0.01$ in EMNIST experiments and $n_0 = 500$, $S = 4$, $\eta_l = 0.01$ for CIFAR-10.

B. Comparison with Non-incremental SL

Figure 2 compares the performance of FSL, DS and the non-incremental version of SL, denoted by FSL-p, when varying C and S . Clearly, FSL-p is slightly worse than DS while FSL significantly outperforms in all cases. A similar conclusion can be drawn as we vary γ as shown in Figure 3.

II. SL WITH NON-IID DATA

We now consider two different sources of data for the server in experiments with $(N, n_i) = (1000, 50)$ for CIFAR-10 and $(450, 240)$ for EMNIST.

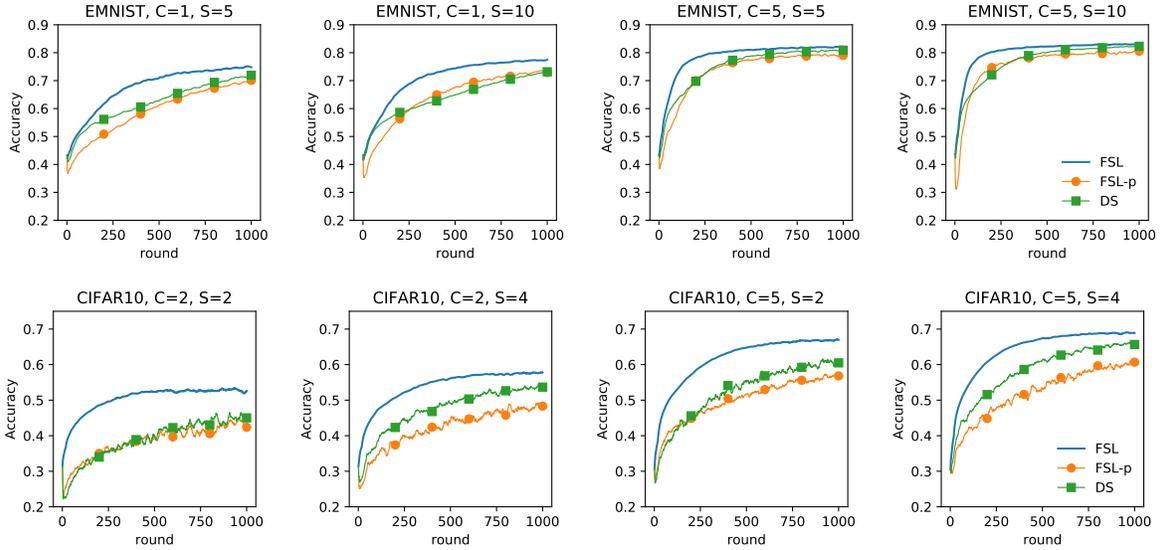


Fig. 2. Test accuracy of FSL, FSL-p and DS, where $n_0 = 225$, $\eta_l = 0.01$, $\gamma = 1$ for EMNIST, and $n_0 = 200$, $\eta_l = 0.01$, and $\gamma = 1$ for CIFAR-10.

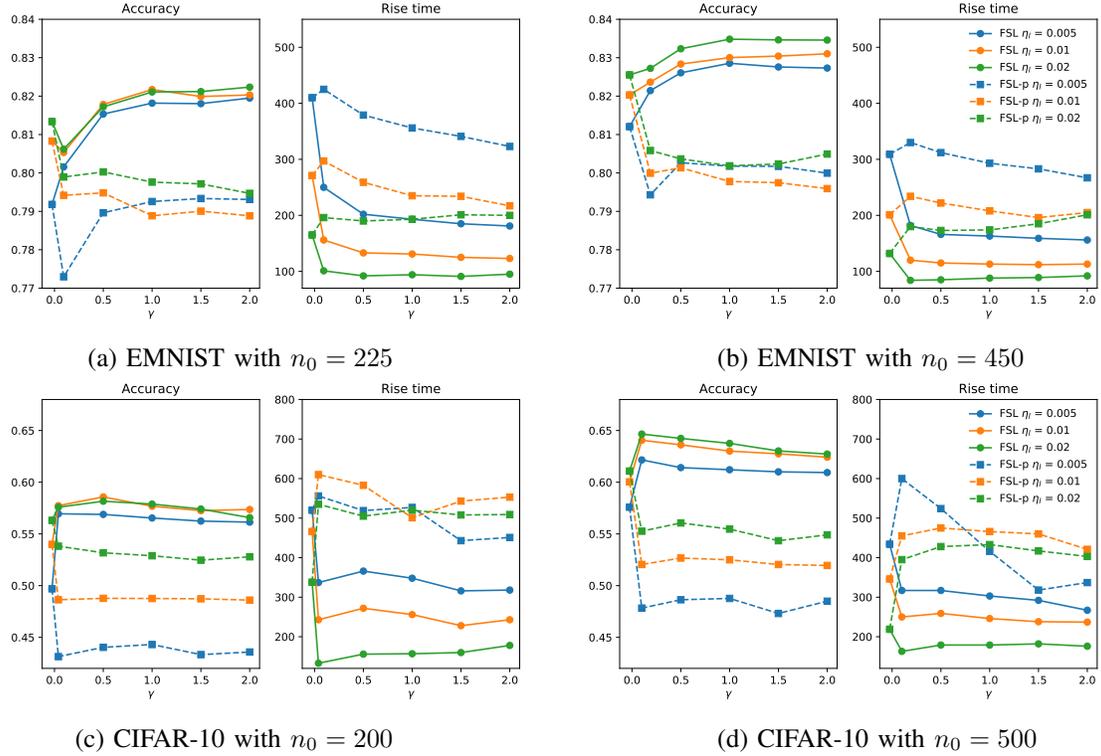


Fig. 3. Comparison between FSL, FSL-p, and DS (shown at $\gamma = 0$). Here, $(C, S) = (5, 5)$ for EMNIST and $(2, 4)$ for CIFAR-10.

a) Data from a few clients: The server obtains data only from a subset of c clients,¹ each contributing s samples (selected uniformly at random without replacement). Note that the server data with $n_0 = c \times s$ samples is highly imbalanced and non-IID (likely missing one or more label classes when $C = 1$).

b) Data from other source(s): For EMNIST, we provide the server $n_0 = 675$ synthetic examples by generating for each label class 15 images of the corresponding letter or number using a cursive font with 5 rotation angles $\{-20, -10, 0, 10, 20\}$

¹These clients can be, for example, test vehicles in our AV example; here they are sampled without replacement once prior to training for simplicity.

and 3 sizes;² see Fig. 4 for a comparison of this synthetic data and EMNIST. For CIFAR-10, we collect $n_0 = 504$ images from the dataset STL-10 with 9 similar label classes as in CIFAR-10, each with 56 examples;³ see Fig. 5 for an illustration of this data, and note that the class `frog` is absent in STL-10. We refer to our algorithm in this case as `FSLsyn`. Our goal with `FSLsyn` is to examine the benefits of server learning when it is performed on data with a significantly different distribution than that of clients’ data.

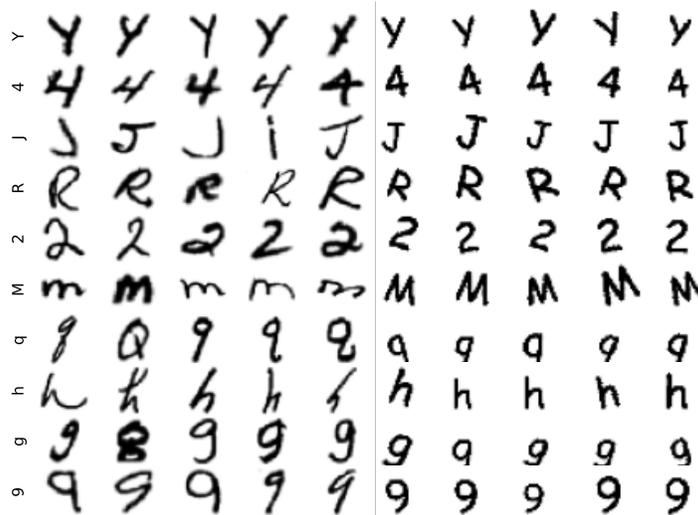


Fig. 4. Left: EMNIST training examples. Right: Server’s synthetic examples.



Fig. 5. Left: CIFAR-10 training examples. Right: Server’s STL-10 examples.

We compare `FSL` and `FSLsyn` (without using a pretrained model) against `SCAFFOLD` and `FedDyn` when $S = \lceil N/100 \rceil$ and $(c, s, n_0) = (10, 50, 500)$ for CIFAR-10 and $(9, 50, 450)$ for EMNIST. We use $\eta_g = \sqrt{S}$ for `FSL`, `FSLsyn`, and `SCAFFOLD`. Fig. 6 shows the test accuracy after $T = 1,000$ rounds with varying learning rate η_l and non-IIDness C . Here, we fix the weight $\gamma=1$ in `FSL` and `FSLsyn` and regularization parameter $\alpha = 0.01$ in `FedDyn`; better performance can be obtained by tuning these parameters as we will show later.

First, Fig. 6 shows that, compared to `SCAFFOLD` and `FedDyn`, our algorithms `FSL` and `FSLsyn` have comparable overall accuracy for EMNIST and much better for CIFAR-10, especially in very non-IID cases, even without tuning γ . The heatmap also suggests that it is fairly easy to select learning rates for `FSL` and `FSLsyn`. The results further indicate that using server learning with synthetic or other ‘good’ sources of data can provide significant benefits. In fact, `FSLsyn` has comparable performance to `FSL` for EMNIST and slightly worse performance for CIFAR-10 (but still better than `FedDyn` and `SCAFFOLD` in this case). Additional experimental results reported in Figures 7 and 8 below also show that `FSL` and `FSLsyn` have faster rise times in most cases. Note that our algorithm can be improved further by having more (and better) data for server learning and using a pretraining step.

Finally, Table 1 shows that both the quantity and the quality of server’s data \mathcal{D}_0 affect the performance of `FSL`. These results are obtained with CIFAR-10 when clients’ data is highly non-IID with $C = 2$, and we pick the learning rates according to the highest accuracy given in Fig. 6. We also fine-tune the regularization parameter of `FedDyn` with $\alpha \in \{0.01, 0.05, 0.1, 0.5\}$

²To generate synthetic data, we first plot each character or number in a 2 inch \times 2 inch figure using font sizes $\{100, 110, 120\}$ in points with each point equal to 1/72 inch, and then resize it to a 28 pixel \times 28 pixel figure.

³STL-10 images were acquired from labeled examples on ImageNet; data available at: <https://cs.stanford.edu/~acoates/stl10/>

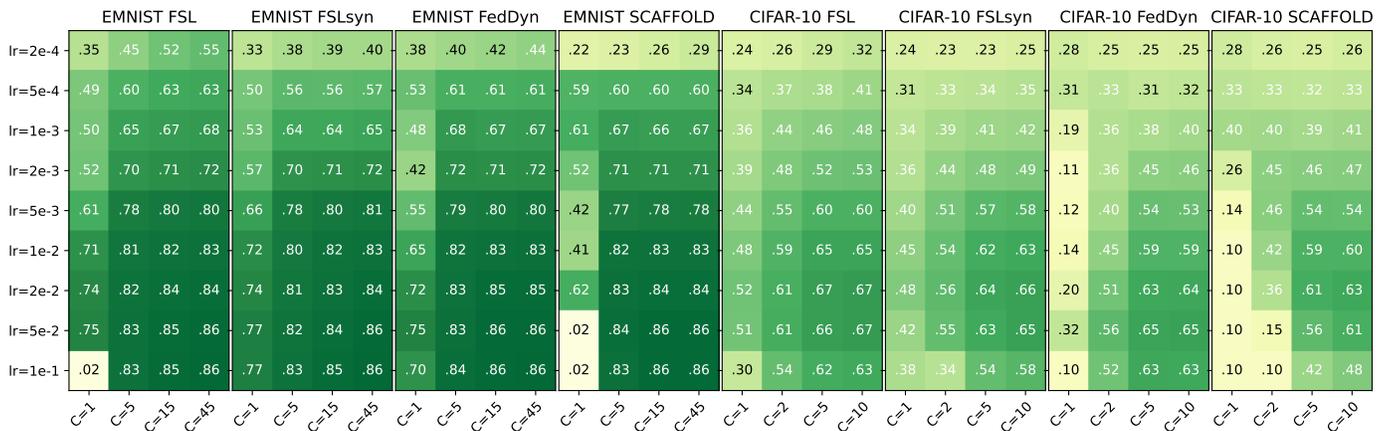


Fig. 6. Heat maps of test accuracy when varying η_l and C . Here, $B = 50$, $S = 5$ for EMNIST, and $B = 10$, $S = 10$ for CIFAR-10; B is chosen following [2] so that 1 epoch of clients corresponds to 5 local steps.

TABLE I
ACCURACY AFTER 1K ROUNDS AND NUMBER OF GLOBAL ROUNDS NEEDED TO REACH 0.5 ACCURACY $T_{0.5}$ IN CIFAR-10 WITH $C = 2$.

	FedDyn $\eta_l = 0.05$	FLSyn ($\eta_l = 0.02$) $n_0 = 504$ $n_0 = 720$		FSL ($\eta_l = 0.02$) $n_0 = 250$ $n_0 = 500$	
$T_{0.5}$	502	339	333	238	203
Acc	0.5779	0.5763	0.5835	0.5845	0.6144

following [1] and the server weight $\gamma \in \{0.6, 0.8, 1.0, 1.2\}$ in FSL and FLSyn – we report the best numbers and skip SCAFFOLD as it underperforms FedDyn. Moreover, we vary the server data size $n_0 \in \{250, 500\}$ for FSL, and $n_0 \in \{504, 720\}$ for FLSyn. Both the rise time and the accuracy improve as n_0 increases, with FSL featuring a more significant improvement since the server’s data are more similar to the clients’ data compared to synthetic data (see Fig. 5). In addition, both of our algorithms require a significantly smaller number of global rounds to reach 0.5 accuracy, showcasing the benefit of server learning. It is also interesting to note that FSL with $n_0 = 250$ is still slightly better than FLSyn with $n_0 = 720$, confirming that the synthetic data are likely taken from a different distribution.

REFERENCES

- [1] A. E. Durmus, Z. Yue, M. Ramon, M. Matthew, W. Paul, and S. Venkatesh. Federated learning based on dynamic regularization. In *ICLR*, 2021.
- [2] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *37th ICML*, pages 5132–5143. PMLR, 2020.

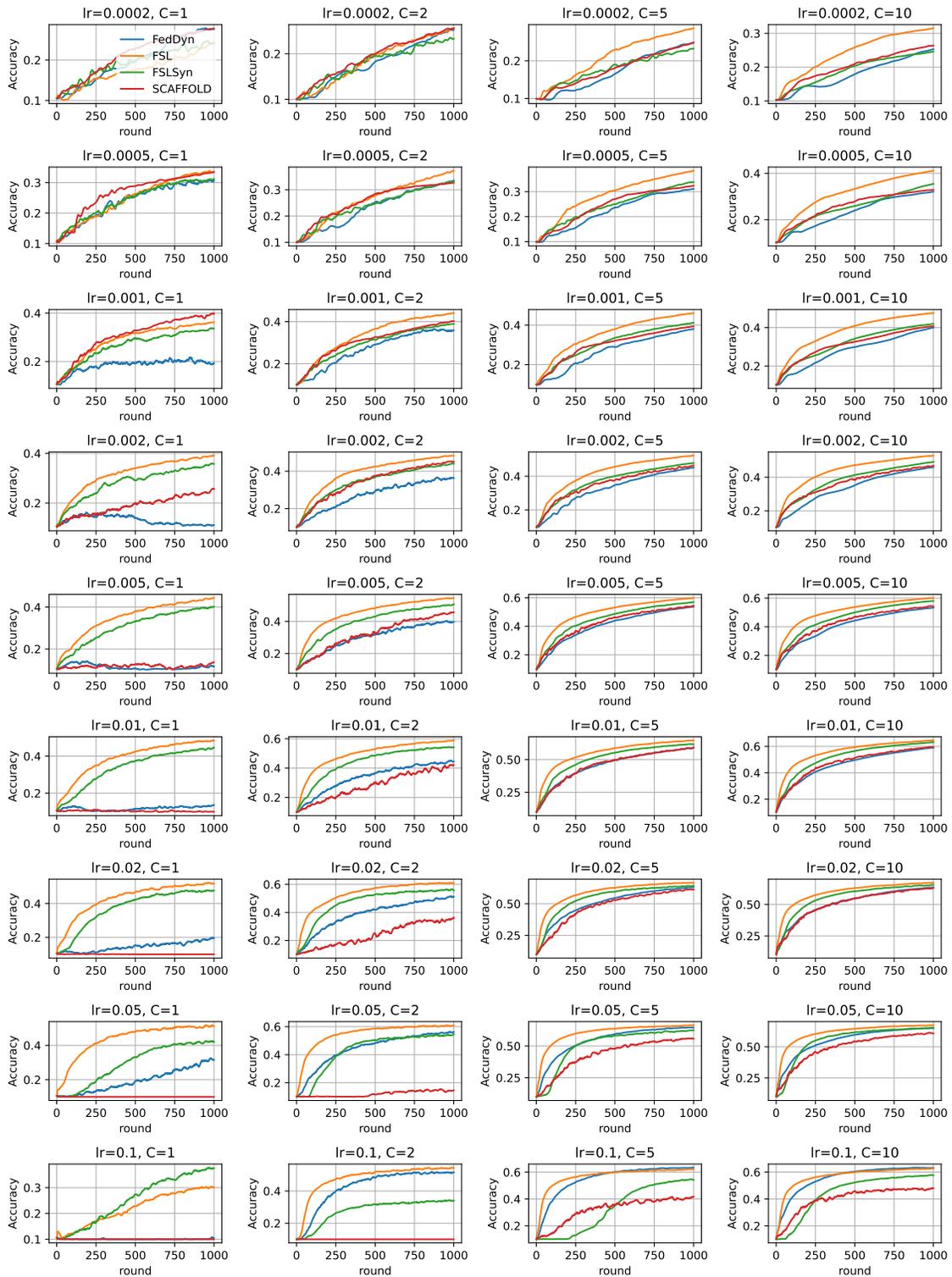


Fig. 7. Test accuracy of FSL, FLSyn, FedDyn, and SCAFFOLD in CIFAR-10 experiments when varying local learning rate $lr = \eta_i$ and number of label classes C each client has.

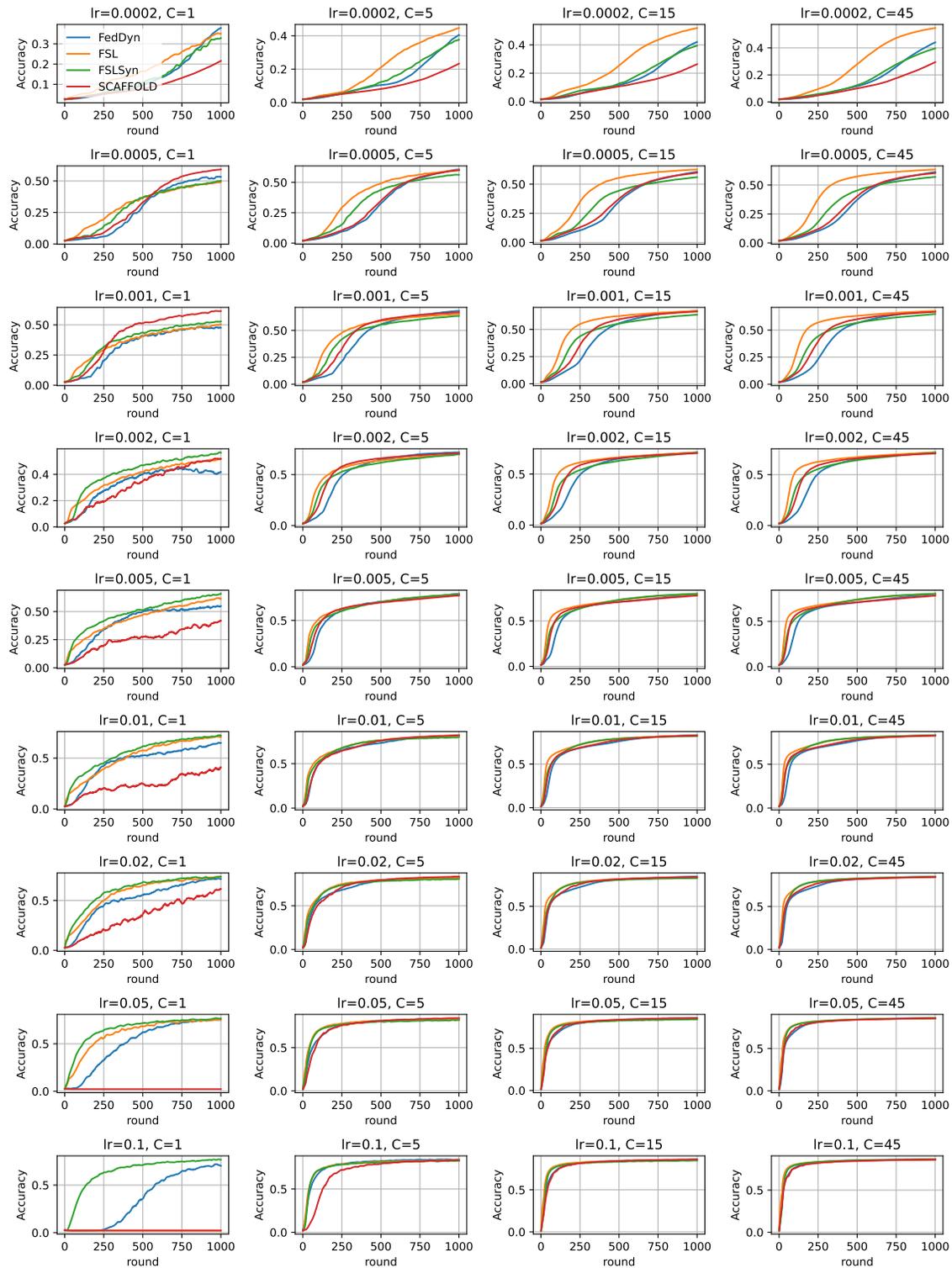


Fig. 8. Test accuracy of FSL, FLSyn, FedDyn, and SCAFFOLD in EMNIST experiments when varying local learning rate $lr = \eta_i$ and number of label classes C each client has.