A Novel Framework for the Analysis and Design of Heterogeneous Federated Learning

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, *Member, IEEE*, and H. Vincent Poor, *Fellow, IEEE*

***Abstract*—In federated learning, heterogeneity in the clients' local datasets and computation speeds results in large variations in the number of local updates performed by each client in each communication round. Naive weighted aggregation of such models causes objective inconsistency, that is, the global model converges to a stationary point of a mismatched objective function which can be arbitrarily different from the true objective. This paper provides a general framework to analyze the convergence of federated optimization algorithms with heterogeneous local training progress at clients. The analyses are conducted for both smooth non-convex and strongly convex settings, and can also be extended to partial client participation case. Additionally, it subsumes previously proposed methods such as FedAvg and FedProx, and provides the first principled understanding of the solution bias and the convergence slowdown due to objective inconsistency. Using insights from this analysis, we propose FedNova, a normalized averaging method that eliminates objective inconsistency while preserving fast error convergence.**

***Index Terms*—Federated learning, distributed optimization.**

I. Introduction

FEDERATED learning [2]–[4] is an emerging sub-area of distributed optimization where both data collection and model training is pushed to a large number of edge clients that have limited communication and computation capabilities. Unlike traditional distributed optimization [5], [6] where consensus (either through a central server or peer-to-peer communication) is performed after every local gradient computation, in federated learning, the subset of clients selected in each communication round perform multiple local updates before these models are aggregated in order to update a global model.

Heterogeneity in the Number of Local Updates in Federated Learning: The clients participating in federated learning are typically highly heterogeneous, both in the size of their local datasets as well as their computation speeds. The original paper on federated learning [2] proposed that each client performs E *epochs* (traversals of their local dataset) of local-update stochastic gradient descent (SGD) with a mini-batch size B. Thus, if a client has n_i local data samples, the number of local SGD iterations is $\tau_i = \lfloor En_i/B \rfloor$, which can vary widely across clients. The heterogeneity in the number of local SGD iterations is exacerbated by relative variations in the clients' computing speeds. When clients are required to upload their local updates after a given wall-clock time interval to mitigate the straggler effects, faster clients will perform more local updates than slower clients. The number of local updates made by a client can also vary across communication rounds due to unpredictable straggling or slowdown caused by background processes, outages, memory limitations etc.

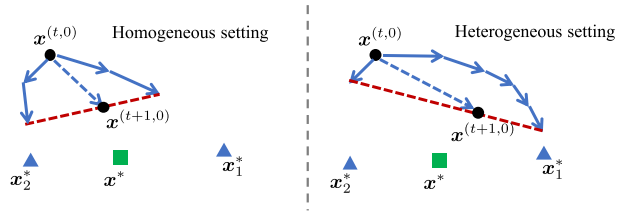


Fig. 1. Model updates in the parameter space. Green squares and blue triangles denote the minima of global and local objectives, respectively.

Heterogeneity in Local Updates Causes Objective Inconsistency: Most recent works that analyze the convergence of federated optimization algorithms [7]–[19] assume that number of local updates is the same across all clients (that is, $\tau_i = \tau$ for all clients i). These works show that, when the learning rate is properly tuned, periodic consensus between the locally trained client models attains a stationary point of the global objective function $F(\boldsymbol{x}) = \sum_{i=1}^{m} n_i F_i(\boldsymbol{x})/n$, which is a sum of local objectives weighted by the dataset size n_i. However, none of these prior works provides insight into the convergence of local-update or federated optimization algorithms in the practical setting when the number of local updates τ_i varies across clients $1, \ldots, m$. In fact, as we show in Section III, *standard averaging of client models after heterogeneous local updates results in convergence to a stationary point – not of the original objective function $F(\boldsymbol{x})$, but of an inconsistent objective $\widetilde{F}(\boldsymbol{x})$, which can be arbitrarily different from $F(\boldsymbol{x})$ depending upon the relative values of τ_i and the similarity among local objectives.* We refer to this problem as objective inconsistency. To gain intuition into this phenomenon, observe in Fig. 1 that if client 1 performs more local updates, then the updated global model $\boldsymbol{x}^{(t+1,0)}$ strays towards the local minimum $\boldsymbol{x}_1^*$, away from the true global minimum $\boldsymbol{x}^*$.

Manuscript received December 29, 2020; revised April 22, 2021 and July 8, 2021; accepted August 3, 2021. Date of publication August 24, 2021; date of current version September 24, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ketan Rajawat. This work was supported in part by NSF under Grants CCF-1850029 and CCF-2045694, in part by the 2018 IBM Faculty Research Award, and in part by the Qualcomm Innovation fellowship. *(Corresponding author: Jianyu Wang.)*

Jianyu Wang and Gauri Joshi are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: jianyuw1@andrew.cmu.edu; gaurij@andrew.cmu.edu).

Hao Liang is with the Department of Electrical and Computer Engineering, Rice University, Houston TX 77005 USA (e-mail: hl106@rice.edu).

Qinghua Liu and H. Vincent Poor are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: qinghual@princeton.edu; poor@princeton.edu).

Digital Object Identifier 10.1109/TSP.2021.3106104

The Need for a General Analysis Framework: A naive approach to overcome heterogeneity is to fix a target number of local updates τ that each client must finish within a communication round and keep fast nodes idle while the slow clients finish their updates. This method will ensure objective consistency (that is, the surrogate objective $\widetilde{F}(\boldsymbol{x})$ equals to the true objective $F(\boldsymbol{x})$). Nonetheless, waiting for the slowest one can significantly increase the total training time [20]. More sophisticated approaches such as FEDPROX [21], VRLSGD [16] and SCAFFOLD [15], designed to handle non-IID local datasets, can be used to reduce (not eliminate) objective inconsistency to some extent, but these methods either result in slower convergence or require additional communication and memory. So far, there is no rigorous understanding of the objective inconsistency and the speed of convergence for this challenging setting of federated learning with heterogeneous local updates.

Other Sources of Heterogeneous local Progress: We note that the root cause of the objective inconsistency is the imbalanced local training progress at clients. When clients use different local learning rates or different local solvers, there will also be a similar effects to taking different local steps. Therefore, in this paper, instead of just focusing on different local steps, we study a more general problem: how *heterogeneous local progress* influences the convergence of federated learning algorithms?

Main Contributions: The main contributions of this paper are listed below.

- We propose a general theoretical framework that subsumes a suite of federated optimization algorithms (such as FEDAVG and FEDPROX) and helps to analyze the effects of heterogeneous local training progress on their error convergence. The framework allows heterogeneous number of local updates, non-IID local datasets as well as different local solvers such as GD, SGD, SGD with proximal gradients, gradient tracking, adaptive learning rates, momentum, etc.
- Based on the general framework, we are able to find out the analytical expression of the surrogate objective function $\widetilde{F}(\boldsymbol{x})$ and show that previous federated optimization algorithms converge to the stationary points of $\widetilde{F}(\boldsymbol{x})$ rather than $F(\boldsymbol{x})$. There is an objective inconsistency problem.
- In order to eliminate the inconsistency problem, we propose FEDNOVA, a method that correctly normalizes local model updates when averaging. The main idea of FEDNOVA is that instead of averaging the cumulative local model changes, the aggregator averages the normalized local gradients according to the local training progress. FEDNOVA ensures objective consistency while preserving fast error convergence and outperforms existing methods as shown in Section VII. By enabling aggregation of models with heterogeneous local progress, FEDNOVA gives the bonus benefit of overcoming the problem of stragglers, or unpredictably slow nodes by allowing fast clients to perform more local updates than slow clients within each communication round.

To the best of our knowledge, this work provides the first fundamental understanding of the bias in the solution (caused by objective inconsistency) and how the convergence rate is influenced by heterogeneity in clients' local progress.

II. SYSTEM MODEL AND PRIOR WORK

The Federated Heterogeneous Optimization Setting: In federated learning, a total of m clients aim to jointly solve the following optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d}\left[F(\boldsymbol{x}) := \sum_{i=1}^{m} p_i F_i(\boldsymbol{x})\right] \tag{1}$$

where $p_i = n_i/n$ denotes the relative sample size, and $F_i(\boldsymbol{x}) = \frac{1}{n_i}\sum_{\xi\in\mathcal{D}_i} f_i(\boldsymbol{x};\xi)$ is the local objective function at the i-th client. Here, f_i is the loss function (possibly non-convex) defined by the learning model and ξ represents a data sample from local dataset $\mathcal{D}_i$. In the t-th communication round, each client independently runs τ_i iterations of local solver (*e.g.*, SGD) starting from the current global model $\boldsymbol{x}^{(t,0)}$ to optimize its own local objective.

In our theoretical framework, we treat τ_i as an arbitrary scalar which can also vary across rounds. In practice, if clients run for the same local epochs E, then $\tau_i = \lfloor En_i/B \rfloor$, where B is the mini-batch size. Alternately, if each communication round has a fixed length in terms of wall-clock time, then τ_i represents the local iterations completed by client i within the time window and may change across clients (depending on their computation speeds and availability) and across communication rounds.

The FedAvg Baseline Algorithm: Federated Averaging (FEDAVG) [2] is the first and most common algorithm used to aggregate these locally trained models at the central server at the end of each communication round. The shared global model $\boldsymbol{x}^{(t,0)}$ at round t is updated as follows:

$$\boldsymbol{x}^{(t+1,0)} - \boldsymbol{x}^{(t,0)} = \sum_{i=1}^{m} p_i \Delta_i^{(t)} \tag{2}$$

$$= -\sum_{i=1}^{m} p_i \cdot \eta \sum_{k=0}^{\tau_i-1} g_i(\boldsymbol{x}_i^{(t,k)}|\xi_i^{(t,k)}) \tag{3}$$

where $\boldsymbol{x}_i^{(t,k)}$ denotes client i's model after the k-th local update in the t-th communication round, $\boldsymbol{x}_i^{(t,0)} = \boldsymbol{x}^{(t,0)}$ is client i's initial model at the t-th round, and $\Delta_i^{(t)} = \boldsymbol{x}_i^{(t,\tau_i)} - \boldsymbol{x}_i^{(t,0)}$ denotes the cumulative local progress made by client i. Also, η is the client learning rate and $g_i(\boldsymbol{x}_i^{(t,k)}|\xi_i^{(t,k)})$ represents the stochastic gradient over a mini-batch $\xi_i^{(t,k)} \subset \mathcal{D}_i$ of B samples. For the ease of writing, we will use $g_i(\boldsymbol{x}_i^{(t,k)})$ to represent the stochastic gradients in the following texts. When the number of clients m is large, then the central server may only randomly select a subset of clients to perform computation at each round.

Convergence Analysis of FedAvg: The papers [7]–[9] first analyze FEDAVG by assuming the local objectives are identical and show that FEDAVG is guaranteed to converge to a stationary point of $F(\boldsymbol{x})$. This analysis was further expanded to the non-IID data partition and client sampling cases by [10]–[13]. However, in all these works, they assume that the number of local steps and the client optimizer are the same across all clients. Besides, asynchronous federated optimization algorithms proposed in [8], [22] take a different approach of allowing clients make updates to stale versions of the global model, and their analyses are limited to IID local datasets and convex local functions.

FedProx: Improving FedAvg by Adding a Proximal Term: To alleviate inconsistency due to non-IID data and heterogeneous

local updates, [21] proposes adding a proximal term $\frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}^{(t,0)}\|^2$ to each local objective, where $\mu \geq 0$ is a tunable parameter. This proximal term pulls each local model backward closer to the global model $\boldsymbol{x}^{(t,0)}$. Although [21] empirically shows that FEDPROX improves FEDAVG, its convergence analysis is limited by assumptions that are stronger than previous FEDAVG analysis and only works for sufficiently large μ. Since FEDPROX is a special case of our general framework, our convergence analysis provides sharp insights into the effect of μ. We show that a larger μ mitigates (but does not eliminate) objective inconsistency, albeit at an expense of slower convergence. Our proposed FEDNOVA method can improve FEDPROX by guaranteeing consistency without slowing down convergence.

Improving FedAvg via Momentum and Cross-Client Variance Reduction: The performance of FEDAVG has been improved in recent literature by applying momentum on the server side [17], [23], [24], or using cross-client variance reduction such as VRLSGD and SCAFFOLD [15], [16]. Again, these works do not consider heterogeneous local progress. Our proposed normalized averaging method FEDNOVA is orthogonal to and can be easily combined with these acceleration or variance-reduction techniques. Moreover, FEDNOVA is also compatible with and complementary to gradient compression/quantization [25]–[31] and fair aggregation techniques [32], [33].

Connections With Classic Distributed Optimization Literature: While this paper studies the bias induced by imbalanced local training progress at clients, there are other kinds of bias in SGD convergence discussed in distributed optimization literature. For example, stochastic gradients are biased if the mini-batch is not chosen uniformly at random [34]; consensus optimization algorithms need to use the push-sum protocol to eliminate the bias associated with the underlying directed network [35], [36]. The bias introduced in our paper is orthogonal to these previous works, as they are caused by different mechanisms. All the above mentioned bias can appear simultaneously in certain algorithms. Moreover, the objective inconsistency problem is not limited to federated learning algorithms. Classic distributed or decentralized optimization algorithms can also have the inconsistency problem when they allow workers/clients to use different learning rates or perform heterogeneous local updates before synchronization.

III. A CASE STUDY TO DEMONSTRATE THE OBJECTIVE INCONSISTENCY PROBLEM

In this section, we use a simple quadratic model to illustrate the convergence problem. Suppose that the local objective functions are $F_i(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{e}_i\|^2$, where $\boldsymbol{e}_i \in \mathbb{R}^d$ is an arbitrary vector and it is the minimum of the local objective. Consider that the global objective function is defined as

$$F(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} F_i(\boldsymbol{x}) = \frac{1}{2m}\sum_{i=1}^{m}\|\boldsymbol{x} - \boldsymbol{e}_i\|^2 \tag{4}$$

which is minimized by $\boldsymbol{x}^* = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{e}_i$. Below, we show that the convergence point of FEDAVG can be arbitrarily away from $\boldsymbol{x}^*$.

Lemma 1 (Objective Inconsistency in FedAvg): For the objective function in (4), if client i performs τ_i local steps per round, then FEDAVG (with sufficiently small learning rate η, deterministic gradients and full client participation) will converge to

$$\tilde{\boldsymbol{x}}^*_{\text{FEDAVG}} = \lim_{T\to\infty}\boldsymbol{x}^{(T,0)} = \frac{\sum_{i=1}^{m}\tau_i\boldsymbol{e}_i}{\sum_{i=1}^{m}\tau_i}. \tag{5}$$

The proof (of a more general version of Lemma 1) is deferred to Section VIII-A. While FEDAVG aims at optimizing $F(\boldsymbol{x})$, it actually converges to the optimum of a surrogate objective $\widetilde{F}(\boldsymbol{x})$. As illustrated in Fig. 2, there can be an arbitrarily large gap between $\tilde{\boldsymbol{x}}^*_{\text{FEDAVG}}$ and $\boldsymbol{x}^*$ depending on the relative values of τ_i and $F_i(\boldsymbol{x})$. This non-vanishing gap also occurs when the local steps τ_i are IID random variables across clients and communication rounds (see the right panel in Fig. 2).

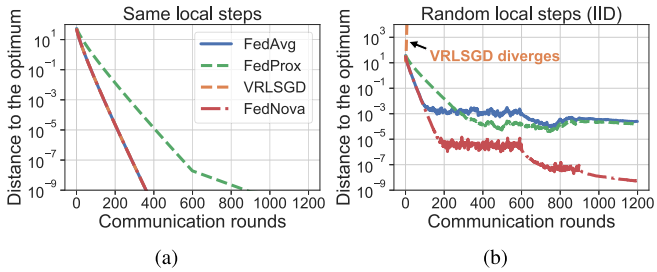


Fig. 2. Simulations comparing the FEDAVG, FEDPROX ($\mu = 1$), VRLSGD and our proposed FEDNOVA algorithms for 30 clients with the quadratic objectives defined in (4), where $\boldsymbol{e}_i \sim \mathcal{N}(0, 0.01\boldsymbol{I}), i \in [1, 30]$. Clients perform GD with $\eta = 0.05$, which is decayed by a factor of 5 at rounds 600 and 900. **(a)**: Clients perform the same number of local steps $\tau_i = 30$ – FEDNOVA is equivalent to FEDAVG in this case; **(b)**: local steps are IID, and time-varying Gaussians with mean 30, *i.e.*, $\tau_i(t) \in [1, 96]$. FEDNOVA significantly outperforms others in the heterogeneous τ_i setting.

Convergence Problem in Other Federated Algorithms: We can generalize Lemma 1 to the case of FEDPROX to demonstrate its convergence gap. From the simulations shown in Fig. 2, observe that FEDPROX can slightly improve on the optimality gap of FEDAVG, but it converges slower. Besides, previous cross-client variance reduction methods such as variance-reduced local SGD (VRLSGD) [16] and SCAFFOLD [15] are only designed for homogeneous local steps case. In the considered heterogeneous setting, if we replace the same local steps τ in VRLSGD by different τ_i's, then we observe that it has drastically different convergence under different settings and even diverge when clients perform random local steps (see the right panel in Fig. 2). These observations emphasize the critical need for a deeper understanding of objective inconsistency and new federated heterogeneous optimization algorithms.

IV. NEW THEORETICAL FRAMEWORK FOR HETEROGENEOUS FEDERATED OPTIMIZATION

We now present a general theoretical framework that subsumes a suite of federated optimization algorithms and helps analyze the effect of objective inconsistency on their error convergence. Although the results are presented for the full client participation setting, it is fairly easy to extend them to the case where a subset of clients are randomly sampled in each round. More discussions on client sampling case will be presented in Section V-B.

A. A Generalized Update Rule for Heterogeneous Federated Optimization

Recall from (3) that the update rule of federated optimization algorithms can be written as $\boldsymbol{x}^{(t+1,0)} - \boldsymbol{x}^{(t,0)} = \sum_{i=1}^m p_i \Delta_i^{(t)}$, where $\Delta_i^{(t)} := \boldsymbol{x}^{(t,\tau_i)} - \boldsymbol{x}^{(t,0)}$ denote the local parameter changes of client i at round t and $p_i = n_i/n$, the fraction of data at client i. We re-write this update rule in a more general form as follows:

$$\boldsymbol{x}^{(t+1,0)} - \boldsymbol{x}^{(t,0)} = -\tau_{\text{eff}} \sum_{i=1}^m w_i \cdot \eta \boldsymbol{d}_i^{(t)} \tag{6}$$

which optimizes $\widetilde{F}(\boldsymbol{x}) = \sum_{i=1}^m w_i F_i(\boldsymbol{x})$. The three key elements τ_{eff}, w_i and $\boldsymbol{d}_i^{(t)}$ of this update rule take different forms for different algorithms. Below, we provide detailed descriptions of these key elements.

1) *Locally averaged gradient:* $\boldsymbol{d}_i^{(t)}$: Without loss of generality, we can rewrite the accumulated local changes as $\Delta_i^{(t)} = -\eta \boldsymbol{G}_i^{(t)} \boldsymbol{a}_i$, where $\boldsymbol{G}_i^{(t)} = [g_i(\boldsymbol{x}_i^{(t,0)}), g_i(\boldsymbol{x}_i^{(t,1)}), \ldots, g_i(\boldsymbol{x}_i^{(t,\tau_i-1)})] \in \mathbb{R}^{d\times\tau_i}$ stacks all stochastic gradients in the t-th round, and $\boldsymbol{a}_i \in \mathbb{R}^{\tau_i}$ is a non-negative vector and defines how stochastic gradients are locally accumulated. Then, by normalizing the gradient weights $\boldsymbol{a}_i$, the locally averaged gradient is defined as $\boldsymbol{d}_i^{(t)} = \boldsymbol{G}_i^{(t)} \boldsymbol{a}_i / \|\boldsymbol{a}_i\|_1$. The normalizing factor $\|\boldsymbol{a}_i\|_1$ in the denominator is the ℓ_1 norm of the vector $\boldsymbol{a}_i$. By setting different $\boldsymbol{a}_i$, (6) works for most common client optimizers such as SGD with proximal updates, local momentum, and variable learning rate, and more generally, any solver whose accumulated gradient $\Delta_i^{(t)} = -\eta \boldsymbol{G}_i^{(t)} \boldsymbol{a}_i$, a linear combination of local gradients.
   Specifically, if the client optimizer is vanilla SGD (*i.e.*, the case of FEDAVG), then $\boldsymbol{a}_i = [1, 1, \ldots, 1] \in \mathbb{R}^{\tau_i}$ and $\|\boldsymbol{a}_i\|_1 = \tau_i$. As a result, the normalized gradient is just a simple average of all stochastic gradients within current round: $\boldsymbol{d}_i^{(t)} = \boldsymbol{G}_i^{(t)} \boldsymbol{a}_i / \tau_i = \sum_{k=0}^{\tau_i-1} g_i(\boldsymbol{x}_i^{(t,k)})/\tau_i$. Later in this section, we will present more specific examples on how to set $\boldsymbol{a}_i$ in other algorithms.
2) *Aggregation weights:* w_i: Each client's locally averaged gradient $\boldsymbol{d}_i$ is multiplied with weight w_i when computing the aggregated gradient $\sum_{i=1}^m w_i \boldsymbol{d}_i$. By definition, these weights satisfy $\sum_{i=1}^m w_i = 1$. Observe that these weights determine the surrogate objective $\widetilde{F}(\boldsymbol{x}) = \sum_{i=1}^m w_i F_i(\boldsymbol{x})$, which is optimized by the general algorithm in (6) instead of the true global objective $F(\boldsymbol{x}) = \sum_{i=1}^m p_i F_i(\boldsymbol{x})$ – we will prove this formally in Theorem 1.
3) *Effective number of steps:* τ_{eff}: Since client i makes τ_i local updates, the average number of local SGD steps per communication round is $\bar{\tau} = \sum_{i=1}^m \tau_i / m$. However, the server can scale up or scale down the effect of the aggregated updates by setting the parameter τ_{eff} larger or smaller than $\bar{\tau}$ (analogous to choosing a global learning rate [17], [24]). We refer to the ratio $\bar{\tau}/\tau_{\text{eff}}$ as the slowdown, and it features prominently in the convergence analysis presented in Section V.

Remark 1 (General Local Update Rule): It is worth noting that the length and exact value of accumulation vector $\boldsymbol{a}_i$ are determined by the number of local steps. We use $\boldsymbol{a}_i(k)$ to denote the accumulation vector after performing k local steps on client i. Unless otherwise stated, we set $\boldsymbol{a}_i = \boldsymbol{a}_i(\tau_i)$. With these notation, we can write down the local update rule as $\boldsymbol{x}_i^{(t,k)} = \boldsymbol{x}_i^{(t,0)} - \eta \sum_{s=0}^{k-1} a_{i,s}(k) g_i(\boldsymbol{x}_i^{(t,s)})$ for any $k \geq 0$, where $a_{i,s}(k)$ is the s-th element in vector $\boldsymbol{a}_i(k) \in \mathbb{R}^k$.

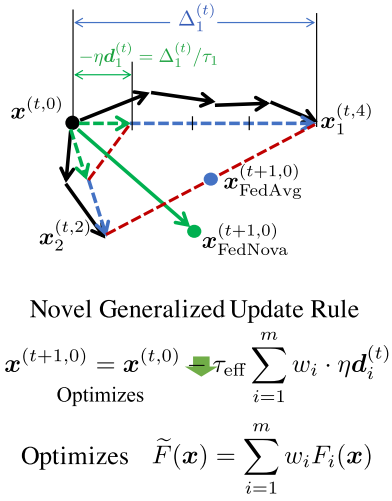


Fig. 3. Comparison between the novel framework and FEDAVG in the model parameter space. Solid black arrows denote local updates at clients. Green and blue dots denote the global updates made by the novel generalized update rule and FEDAVG respectively. While w_i controls the direction of the solid green arrow, effective steps τ_{eff} determines how far the global model moves along with this direction. FEDAVG implicitly assigns too higher weights for clients with more local steps, resulting in a biased global direction.

In Fig. 3, we further illustrate how the above key elements influence the algorithm and compare the novel generalized update rule and FEDAVG in the model parameter space. The general rule (6) enables us to freely choose τ_{eff} and w_i for a given local solver $\boldsymbol{a}_i$, which helps design fast and consistent algorithms such as FEDNOVA, the normalized averaging method proposed in Section VI. To implement this generalized update rule, each client can send the normalized update $-\eta \boldsymbol{d}_i^{(t)}$ to the central server, which is just a re-scaled version of $\Delta_i^{(t)}$, the accumulated local parameter update sent by clients in the vanilla update rule (3). The server does not need to know the specific form of local accumulation vector $\boldsymbol{a}_i$.

B. Previous Algorithms as Special Cases

Any previous algorithm whose accumulated local changes $\Delta_i^{(t)} = -\eta \boldsymbol{G}_i^{(t)} \boldsymbol{a}_i$, a linear combination of local gradients, is subsumed by the above formulation, as shown below:

$$\begin{aligned}
\boldsymbol{x}^{(t+1,0)} - \boldsymbol{x}^{(t,0)} &= \sum_{i=1}^m p_i \Delta_i^{(t)} \\
&= -\sum_{i=1}^m p_i \|\boldsymbol{a}_i\|_1 \cdot \frac{\eta \boldsymbol{G}_i^{(t)} \boldsymbol{a}_i}{\|\boldsymbol{a}_i\|_1} \\
&= -\underbrace{\left(\sum_{i=1}^m p_i \|\boldsymbol{a}_i\|_1\right)}_{\tau_{\text{eff}}\text{: effective local steps}} \sum_{i=1}^m \eta \underbrace{\left(\frac{p_i \|\boldsymbol{a}_i\|_1}{\sum_{i=1}^m p_i \|\boldsymbol{a}_i\|_1}\right)}_{w_i\text{: weight}} \underbrace{\left(\frac{\boldsymbol{G}_i^{(t)} \boldsymbol{a}_i}{\|\boldsymbol{a}_i\|_1}\right)}_{\boldsymbol{d}_i}.
\end{aligned} \tag{7}$$

Unlike the more general form (6), in (7), which subsumes the following previous methods, τ_{eff} and w_i are implicitly fixed by the choice of the local solver (*i.e.*, the choice of $\boldsymbol{a}_i$).

1) Vanilla SGD as Local Solver (FedAvg): In FEDAVG, the local solver is SGD such that $\boldsymbol{a}_i = [1, 1, \ldots, 1] \in \mathbb{R}^{\tau_i}$ and $\|\boldsymbol{a}_i\|_1 = \tau_i$. As a consequence, the normalized gradient $\boldsymbol{d}_i$ is a simple average over τ_i iterations, $\tau_{\text{eff}} = \sum_{i=1}^m p_i \tau_i$, and $w_i = p_i \tau_i / \sum_{i=1}^m p_i \tau_i$. That is, the normalized gradients with more local steps will be implicitly assigned higher weights.

2) Proximal SGD as Local Solver (FedProx): In FEDPROX, local SGD steps are corrected by a proximal term. It can be shown that $\boldsymbol{a}_i = [(1-\alpha)^{\tau_i - 1}, (1-\alpha)^{\tau_i - 2}, \ldots, (1-\alpha), 1] \in \mathbb{R}^{\tau_i}$, where $\alpha = \eta\mu$ and μ is a tunable parameter. In this case, we have $\|\boldsymbol{a}_i\|_1 = [1 - (1-\alpha)^{\tau_i}]/\alpha$ and hence,

$$\tau_{\text{eff}} = \frac{1}{\alpha} \sum_{i=1}^m p_i [1 - (1-\alpha)^{\tau_i}], w_i = \frac{p_i [1 - (1-\alpha)^{\tau_i}]}{\sum_{i=1}^m p_i [1 - (1-\alpha)^{\tau_i}]}. \tag{8}$$

When $\alpha = 0$, FEDPROX is equivalent to FEDAVG. As $\alpha = \eta\mu$ increases, the w_i in FEDPROX is more similar to p_i, thus making the surrogate objective $\widetilde{F}(\boldsymbol{x})$ more consistent. However, a larger α corresponds to smaller τ_{eff}, which slows down convergence, as we discuss more in Section V.

3) SGD With Decayed Learning Rate as Local Solver: Suppose the clients' local learning rates are exponentially decayed, then we have $\boldsymbol{a}_i = [1, \gamma_i, \ldots, \gamma_i^{\tau_i - 1}]$ where $\gamma_i \geq 0$ can vary across clients. As a result, we have $\|\boldsymbol{a}_i\|_1 = (1 - \gamma_i^{\tau_i})/(1 - \gamma_i)$ and $w_i \propto p_i (1 - \gamma_i^{\tau_i})/(1 - \gamma_i)$. Comparing with the case of FEDPROX (8), changing the values of γ_i has a similar effect as changing $(1 - \alpha)$.

4) Momentum SGD as Local Solver: If we use momentum SGD where the local momentum buffers of active clients are reset to zero at the beginning of each round [17] due to the stateless nature of FL [3], then we have $\boldsymbol{a}_i = [1 - \rho^{\tau_i}, 1 - \rho^{\tau_i - 1}, \ldots, 1 - \rho]/(1 - \rho)$, where ρ is the momentum factor, and $\|\boldsymbol{a}_i\|_1 = [\tau_i - \rho(1 - \rho^{\tau_i})/(1 - \rho)]/(1 - \rho)$.

More generally, the new formulation (7) suggests that $w_i \neq p_i$ whenever clients have different $\|\boldsymbol{a}_i\|_1$, which may be caused by imbalanced local updates (*i.e.*, $\boldsymbol{a}_i$'s have different dimensions), or various local learning rate/momentum schedules (*i.e.*, $\boldsymbol{a}_i$'s have different scales).

V. CONVERGENCE ANALYSIS

A. Main Results: Analysis for Smooth Non-Convex Functions

In Theorem 1 and Theorem 2 below we provide a convergence analysis for the general update rule (6) and quantify the solution bias due to objective inconsistency. The analysis relies on Assumptions 1 and 2 used in the standard analysis of SGD [37] and Assumption 3 commonly used in the federated optimization literature [3], [11], [15], [21], [24], [38], [39] to capture the dissimilarities of local objectives.

Assumption 1 (Smoothness): Each local objective function is Lipschitz smooth, that is, $\|\nabla F_i(\boldsymbol{x}) - \nabla F_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \forall i \in \{1, 2, \ldots, m\}$.

Assumption 2 (Unbiased Gradient and Bounded Variance): The stochastic gradient at each client is an unbiased estimator of the local gradient: $\mathbb{E}_\xi[g_i(\boldsymbol{x}|\xi)] = \nabla F_i(\boldsymbol{x})$ where ξ represents a randomly sampled mini-batch from the local dataset $\mathcal{D}_i$, and has bounded variance $\mathbb{E}_\xi[\|g_i(\boldsymbol{x}|\xi) - \nabla F_i(\boldsymbol{x})\|^2] \leq \sigma^2, \forall \boldsymbol{x} \in \mathbb{R}^d, \forall i \in \{1, 2, \ldots, m\}, \sigma^2 \geq 0$.

Assumption 3 (Bounded Dissimilarity): For any sets of weights $\{w_i \geq 0\}_{i=1}^m, \sum_{i=1}^m w_i = 1$, there exist constants $\beta^2 \geq 1, \kappa^2 \geq 0$ such that $\sum_{i=1}^m w_i \|\nabla F_i(\boldsymbol{x})\|^2 \leq \beta^2 \|\sum_{i=1}^m w_i \nabla F_i(\boldsymbol{x})\|^2 + \kappa^2$. If local functions are identical to each other, then we have $\beta^2 = 1, \kappa^2 = 0$.

Assumption 4 (Accumulation Vector): All elements in the accumulation vector $\boldsymbol{a}_i(k)$, in which $k \in [1, \tau_i], \forall i$, are upper bounded by Λ. Also, $\|\boldsymbol{a}_i(k)\|_p \leq \|\boldsymbol{a}_i(k+1)\|_p$ for $p = \{1, 2\}$.

One can easily validate that Assumption 4 holds for many common local sovlers, such as vanilla SGD, proximal SGD and momentum SGD. In all these special cases, we have $\Lambda = 1$. Under the above assumptions, our main theorem is stated as follows.

Theorem 1 (Convergence to the Surrogate Objective $\widetilde{F}(\boldsymbol{x})$'s Stationary Point): Under Assumptions 1 to 4, any federated optimization algorithm that follows the update rule (6), will converge to a stationary point of a surrogate objective $\widetilde{F}(\boldsymbol{x}) = \sum_{i=1}^m w_i F_i(\boldsymbol{x})$. More specifically, if the total communication rounds T is pre-determined and the learning rate η is small enough $\eta = \sqrt{m/\overline{\tau} T}$ where $\overline{\tau} = \frac{1}{m}\sum_{i=1}^m \tau_i$, then the optimization error $\min_{t \in [T]} \mathbb{E}\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\|^2$ will be bounded by:

$$\underbrace{\mathcal{O}\left(\frac{\overline{\tau}/\tau_{\text{eff}}}{\sqrt{m\overline{\tau}T}}\right) + \mathcal{O}\left(\frac{A\sigma^2}{\sqrt{m\overline{\tau}T}}\right) + \mathcal{O}\left(\frac{mB\sigma^2}{\overline{\tau}T}\right) + \mathcal{O}\left(\frac{mC\kappa^2}{\overline{\tau}T}\right)}_{\text{denoted by } \epsilon_{\text{opt}} \text{ in } (13)} \tag{9}$$

where $\mathcal{O}$ swallows all constants (including L), and quantities A, B, C are defined as follows:

$$A = m\tau_{\text{eff}} \sum_{i=1}^m \frac{w_i^2 \|\boldsymbol{a}_i\|_2^2}{\|\boldsymbol{a}_i\|_1^2}, \tag{10}$$

$$B = \Lambda \sum_{i=1}^m w_i (\tau_i - 1) \|\boldsymbol{a}_i\|_2^2 / \|\boldsymbol{a}_i\|_1, \tag{11}$$

$$C = \Lambda^2 \max_i \{\tau_i(\tau_i - 1)\}. \tag{12}$$

In Section VIII-B, we also provide another version of this theorem that explicitly contains the local learning rate η. Moreover, since the surrogate objective $\widetilde{F}(\boldsymbol{x})$ and the original objective $F(\boldsymbol{x})$ are just different linear combinations of the local functions, once the algorithm converges to a stationary point of $\widetilde{F}(\boldsymbol{x})$, one can also obtain some guarantees in terms of $F(\boldsymbol{x})$, as given by Theorem 2 below.

Theorem 2 (Convergence in Terms of the True Objective $F(\boldsymbol{x})$): Under the same conditions as Theorem 1, the minimal gradient norm of the true global objective function $\min_{t \in [T]} \mathbb{E}\|\nabla F(\boldsymbol{x}^{(t,0)})\|^2$ will be bounded by:

$$\underbrace{2\left[\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}(\beta^2 - 1) + 1\right]\epsilon_{\text{opt}}}_{\text{vanishing error term}} + \underbrace{2\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}\kappa^2}_{\text{non-vanishing error due to obj. inconsistency}} \tag{13}$$

where ϵ_{opt} denotes the vanishing optimization error given by (9) and $\chi^2_{\boldsymbol{p}\|\boldsymbol{w}} = \sum_{i=1}^m (p_i - w_i)^2 / w_i$ represents the chi-square divergence between vectors $\boldsymbol{p} = [p_1, \ldots, p_m]$ and $\boldsymbol{w} = [w_1, \ldots, w_m]$.

Proof: Please refer to Appendix A-A. ■

Discussion: Theorems 1 and 2 describe the convergence behavior of a broad class of federated heterogeneous optimization algorithms. Observe that when $\boldsymbol{p} = \boldsymbol{w}$ we have that $\chi^2 = 0$.

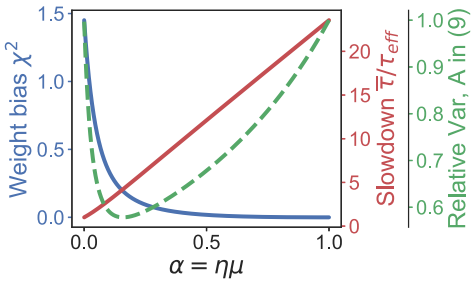


Fig. 4. Illustration on how the parameter $\alpha = \eta\mu$ influences the convergence of FEDPROX. We set $m = 30, p_i = 1/m, \tau_i \sim \mathcal{N}(20, 20)$. '*Weight bias*' denotes the chi-square distance between $\boldsymbol{p}$ and $\boldsymbol{w}$. '*Slowdown*' and '*Relative Variance*' quantify how the first and the second terms in (9) change.

Also, when all local functions are identical to each other, we have $\beta^2 = 1, \kappa^2 = 0$. Only in these two special cases, is there no objective inconsistency. For most other algorithms subsumed by the general update rule in (6), both w_i and τ_{eff} are influenced by the choice of $\boldsymbol{a}_i$. When clients have different local progress (*i.e.*, different $\boldsymbol{a}_i$ vectors), previous algorithms will end up with a non-zero error floor $\chi^2\kappa^2$, which does not vanish to 0 even with sufficiently small learning rate. In Appendix A-B, we further construct a lower bound and show that $\lim_{T\to\infty} \min_{t\in[T]} \|\nabla F(\boldsymbol{x}^{(t,0)})\|^2 = \Omega(\chi^2\kappa^2)$, suggesting (13) is tight.

Novel Insights Into the Convergence of FedProx and the Effect of μ: Recall that in FEDPROX $\boldsymbol{a}_i = [(1-\alpha)^{\tau_i - 1}, \ldots, (1-\alpha), 1]$, where $\alpha = \eta\mu$. Accordingly, substituting the effective steps and aggregated weight, given by (8), into (9) and (13), we get the convergence guarantee for FEDPROX. Again, it has objective inconsistency because $w_i \neq p_i$. As we increase α, the weights w_i come closer to p_i and thus, the non-vanishing error $\chi^2\kappa^2$ in (13) decreases (see blue curve in Fig. 4). However increasing α worsens the slowdown $\overline{\tau}/\tau_{\text{eff}}$, which appears in the first error term in (9) (see the red curve in Fig. 4). In the extreme case when $\alpha = 1$, although FEDPROX achieves objective consistency, it has a significantly slower convergence because $\tau_{\text{eff}} = 1$ and the first term in (9) is $\overline{\tau}$ times larger than that with FEDAVG (eq. to $\alpha = 0$).

Theorem 1 also reveals that, in FEDPROX, there should exist a best value of α that balances all terms in (9). It can be shown that $\alpha = \mathcal{O}(m^{\frac{1}{2}}/\overline{\tau}^{\frac{1}{2}}T^{\frac{1}{6}})$ optimizes the error bound (9) of FEDPROX and yields a convergence rate of $\mathcal{O}(1/\sqrt{m\overline{\tau}T} + 1/T^{\frac{2}{3}})$ on the surrogate objective. This can serve as a guideline on setting α in practice.

Linear Speedup Analysis: Another implication of Theorem 1 is that when the communication rounds T is sufficiently large, then the convergence of the surrogate objective will be dominated by the first two terms in (9), which is $1/\sqrt{m\overline{\tau}T}$. This suggests that the algorithm only uses T/γ total rounds when using γ times more clients (*i.e.*, achieving linear speedup) to reach the same error level.

B. Extension: Analysis for Partial Client Participation

In this subsection, we extend the general analysis to the case where only a random subset of clients participate into training at each round. Following previous works [11], [12], [15], [21], we assume the sampling scheme guarantees that the update rule (7) holds in expectation. This can be achieved by sampling with replacement from $\{1, 2, \ldots, m\}$ with probabilities $\{p_i\}$, and averaging local updates from selected clients with equal weights. Specifically, we have

$$\boldsymbol{x}^{(t+1,0)} - \boldsymbol{x}^{(t,0)} = \frac{1}{q}\sum_{j=1}^{q} \Delta_{l_j}^{(t)} \tag{14}$$

where q is the number of selected clients per round, and l_j is a random index sampled from $\{1, 2, \ldots, m\}$ satisfying $\mathbb{P}(l_j = i) = p_i$. Recall that $p_i = n_i/n$ is the relative sample size at client i. One can directly validate that

$$\mathbb{E}_{\mathcal{S}}\left[\frac{1}{q}\sum_{j=1}^{q} \Delta_{l_j}^{(t)}\right] = \sum_{i=1}^{m} p_i \Delta_i^{(t)} \tag{15}$$

where $\mathbb{E}_{\mathcal{S}}$ represents the expectation over random indices $\mathcal{S} = \{l_1, \ldots, l_q\}$ at the current round. When the client sampling scheme satisfies (15), we can obtain the following theorem.

Theorem 3: Under the same condition as in Theorem 1, suppose at each round, the server randomly selects q clients with replacement to perform local computation. Any federated optimization algorithms satisfying (14) and 15 converge to a stationary point of a surrogate objective $\widetilde{F}(\boldsymbol{x}) = \sum_{i=1}^{m} w_i F_i(\boldsymbol{x})$. If we set $\eta = \sqrt{q/\widetilde{\tau}T}$ where $\widetilde{\tau} = \mathbb{E}_{\mathcal{S}}[\sum_{j=1}^{q} \tau_{l_j}/q] = \sum_{i=1}^{m} p_i\tau_i$ is the average local update across clients, then the expected gradient norm $\min_{t\in[T]} \mathbb{E}\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\|^2$ is bounded as follows:

$$\mathcal{O}\left(\frac{\widetilde{\tau}/\tau_{\text{eff}}}{\sqrt{q\widetilde{\tau}T}}\right) + \mathcal{O}\left(\frac{A'\sigma^2}{\sqrt{q\widetilde{\tau}T}}\right) + \mathcal{O}\left(\frac{q(B\sigma^2 + C\kappa^2)}{\widetilde{\tau}T}\right) \tag{16}$$

where $A' = \tau_{\text{eff}} \sum_{i=1}^{m} \frac{w_i^2 \|\boldsymbol{a}_i\|_2^2}{p_i \|\boldsymbol{a}_i\|_1^2}$, B, C are defined in (11) and (12) and $\mathcal{O}$ combines all other constants (including L).

Proof: Please refer to Appendix A-C. ■

Discussion: Comparing with the full client participation case, Theorem 3 has a similar form as Theorem 1. When the number of communication rounds T is sufficiently large, the convergence rate will be dominated by the first two terms, which is $\mathcal{O}(1/\sqrt{q\widetilde{\tau}T})$. This suggests that in the case of client sampling, the algorithm can still achieve linear speedup in terms of the number of sampled clients.

C. Extension: Analysis for Strongly Convex Functions

Another benefit of using our general theoretical analysis is that it can be easily extended to the strongly-convex case as a corollary. In particular, when the global objective is strongly convex, it satisfies the Polyak-Łojasiewicz (PL) condition, stated as follows:

$$\|\nabla F(\boldsymbol{x})\|^2 \geq 2c[F(\boldsymbol{x}) - F_{\text{inf}}] \tag{17}$$

where c is a positive constant. Under the PL condition, the convergence rate of federated optimization algorithms can be further improved. In particular, we have the following theorem.

Theorem 4 (Convergence under PL Condition): When each local objective function is strongly convex with constant c, any federated optimization algorithm that follows the update rule (6) will converge to the minimum of a surrogate objective $\widetilde{F}(\boldsymbol{x}) = \sum_{i=1}^{m} w_i F_i(\boldsymbol{x})$. Specifically, if the client learning rate is set as $\eta^{(t)} = 6/[c\tau_{\text{eff}}(t+\gamma)]$, where $\gamma = L/(c\nu), \nu > 0$, then the optimization error $\widetilde{F}(\boldsymbol{x}^{(T,0)}) - \widetilde{F}_{\text{inf}}$ will converge to 0 at the

following rate:

$$\mathcal{O}\left(\frac{L}{sc^2}\frac{\sigma^2 A}{mT\overline{\tau}}\right)+\mathcal{O}\left(\frac{L^2}{s^2c^3}\frac{\sigma^2 B+\kappa^2 C}{T^2\overline{\tau}^2}\right). \tag{18}$$

where $s=\tau_{\text{eff}}/\overline{\tau}$ and A,B,C are constants as defined in (10) to (12). Moreover, in order to achieve the above rate, τ_{eff} should be upper bounded by the following quantity:

$$\tau_{\text{eff}}^2 \le \frac{648\nu^2(\sigma^2 B+\kappa^2 C)}{cL^2[\widetilde{F}(\boldsymbol{x}^{(0,0)})-\widetilde{F}_{\text{inf}}]-36\nu L^2\sigma^2} \tag{19}$$

where $\nu>0$ is a constant.

Proof: Please refer to Appendix A-D. ■

Discussion: Theorem 4 shows that under the PL condition, the convergence rate of federated optimization algorithms is dominated by $\mathcal{O}(1/(mT\overline{\tau}))$, which is the same as synchronous mini-batch SGD [37]. Similar to the non-convex results (Theorem 1), there is an additional error caused by performing local updates. The additional error decays to 0 at a faster rate $\mathcal{O}(1/T^2\overline{\tau}^2)$ and has limited influence on the error bound.

Moreover, observe that the error bound (18) monotonically decreases when τ_{eff} becomes larger. However, τ_{eff} cannot be arbitrarily large. Given a set of local steps at clients $\{\tau_i\}$, τ_{eff} has an upper bound as given in (19). Again, Theorem 4 not only applies to FEDAVG but also works for other federated optimization algorithms using proximal SGD, or momentum SGD, as the local solver.

VI. FEDNOVA: PROPOSED FEDERATED NORMALIZED AVERAGING ALGORITHM

Theorems 1 and 2 suggest an extremely simple solution to overcome the problem of objective inconsistency. When we set $w_i=p_i$ in (6), then the second non-vanishing term $\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}\kappa^2$ in (13) will just become zero. This simple intuition yields the following new algorithm:

$$\text{FEDNOVA}: \quad \boldsymbol{x}^{(t+1,0)}-\boldsymbol{x}^{(t,0)}=-\tau_{\text{eff}}^{(t)}\sum_{i=1}^{m}p_i\cdot\eta\boldsymbol{d}_i^{(t)} \tag{20}$$

where $\boldsymbol{d}_i^{(t)}=\frac{\boldsymbol{G}_i^{(t)}\boldsymbol{a}_i^{(t)}}{\|\boldsymbol{a}_i^{(t)}\|_1}$. The proposed algorithm is named *federated normalized averaging* (FEDNOVA), because the normalized stochastic gradients $\boldsymbol{d}_i$ are averaged/aggregated instead of the local changes $\Delta_i=-\eta\boldsymbol{G}_i\boldsymbol{a}_i$. When the local solver is vanilla SGD, then $\boldsymbol{a}_i=[1,1,\ldots,1]\in\mathbb{R}^{\tau_i}$ and $\boldsymbol{d}_i^{(t)}$ is a simple average over current round's gradients. In order to be consistent with FEDAVG whose update rule is (7), one can simply set $\tau_{\text{eff}}^{(t)}=\sum_{i=1}^m p_i\tau_i^{(t)}$. Then, in this case, the update rule of FEDNOVA is equivalent to $\boldsymbol{x}^{(t+1,0)}-\boldsymbol{x}^{(t,0)}=(\sum_{i=1}^m p_i\tau_i^{(t)})\sum_{i=1}^m p_i\Delta_i^{(t)}/\tau_i^{(t)}$. Comparing to previous algorithm $\boldsymbol{x}^{(t+1,0)}-\boldsymbol{x}^{(t,0)}=\sum_{i=1}^m p_i\Delta_i^{(t)}$, each accumulative local change Δ_i in FEDNOVA is re-scaled by $(\sum_{i=1}^m p_i\tau_i^{(t)})/\tau_i^{(t)}$. This simple tweak in the aggregation weights eliminates inconsistency in the solution and gives better convergence than previous methods.

Flexibility in Choosing Hyper-parameters and Local Solvers: Besides vanilla SGD, the new formulation of FEDNOVA naturally allows clients to choose various local solvers (*i.e.*, client-side optimizer). As discussed in Section IV-A, the local solver can be any optimizers as long as the local model changes can be written as a linear combination of gradients.[1] Examples include SGD with decayed local learning rate, SGD with proximal updates, SGD with local momentum, etc. Furthermore, the value of τ_{eff} is not necessarily to be controlled by the local solver as previous algorithms. For example, when using SGD with proximal updates, one can simply set $\tau_{\text{eff}}=\sum_{i=1}^m p_i\tau_i$ instead of its default value $\sum_{i=1}^m p_i[1-(1-\alpha)^{\tau_i}]/\alpha$. This can help alleviate the slowdown problem discussed in Section V.

Combination With Acceleration Techniques: If clients have additional communication bandwidth, they can use cross-client variance reduction techniques to further accelerate the training [15], [16]. In this case, each local gradient step at the t-round will be corrected by $\sum_{i=1}^m p_i\boldsymbol{d}_i^{(t-1)}-\boldsymbol{d}_i^{(t-1)}$. That is, the local gradient at the k-th local step becomes $g_i(\boldsymbol{x}^{(t,k)})+\sum_{i=1}^m p_i\boldsymbol{d}_i^{(t-1)}-\boldsymbol{d}_i^{(t-1)}$. Besides, on the server side, one can also implement server momentum or adaptive server optimizers [17], [23], [24], in which the aggregated normalized gradient $-\tau_{\text{eff}}\sum_{i=1}^m\eta p_i\boldsymbol{d}_i$ is used to update the server momentum buffer instead of directly updating the server model.

Convergence Analysis: In FEDNOVA, the local solvers at clients do not necessarily need to be the same or fixed across rounds. In the following theorem, we obtain strong convergence guarantee for FEDNOVA, even with *arbitrarily time-varying* local updates and client optimizers.

Theorem 5 (Convergence of: FEDNOVA to a Consistent Solution): Suppose that each client performs arbitrary number of local updates $\tau_i(t)$ using arbitrary gradient accumulation method $\boldsymbol{a}_i(t), t\in[T]$ per round. Under Assumptions 1 to 3, and local learning rate as $\eta=\sqrt{m/(\hat{\tau}T)}$, where $\hat{\tau}=\sum_{t=0}^{T-1}\overline{\tau}(t)/T$ denotes the average local steps over all rounds at clients, then FEDNOVA converges to a stationary point of $F(\boldsymbol{x})$ in a rate of $\mathcal{O}(1/\sqrt{m\hat{\tau}T})$. The detailed bound is the same as the right hand side of (9), except that $\overline{\tau}, A, B, C$ are replaced by their average values over all rounds.

The proof of Theorem 5 can be found in Appendix A-E. Using the same technique as Theorem 3, one can further generalize Theorem 5 to incorporate client sampling schemes.

VII. EXPERIMENTAL RESULTS

Experimental Setup: We evaluate all algorithms on two setups with non-IID data partitioning: (1) *Logistic Regression on a Synthetic Federated Dataset:* The dataset `Synthetic(1,1)` is originally constructed in [21]. The local dataset sizes $n_i, i\in[1,30]$ follows a power law. (2) *DNN trained on a Non-IID partitioned CIFAR-10 dataset:* We train a VGG-11 [40] network on the CIFAR-10 dataset [41], which is partitioned across 16 clients using a Dirichlet distribution $\text{Dir}_{16}(0.1)$, as done in [42]. The original CIFAR-10 test set (without partitioning) is used to evaluate the generalization performance of the trained global model. The local learning rate η is decayed by a constant factor after finishing 50% and 75% of the communication rounds. The initial value of η is tuned separately for FEDAVG with different local solvers. When using the same solver, FEDNOVA uses the same η as FEDAVG to guarantee a fair comparison. On CIFAR-10, we run each experiment with 3 random seeds and report the average and standard deviation. Our code is available at here: https://github.com/JYWa/FedNova.

[1] Adaptive optimization methods (such as Adam, AdaGrad) do not meet this criteria.

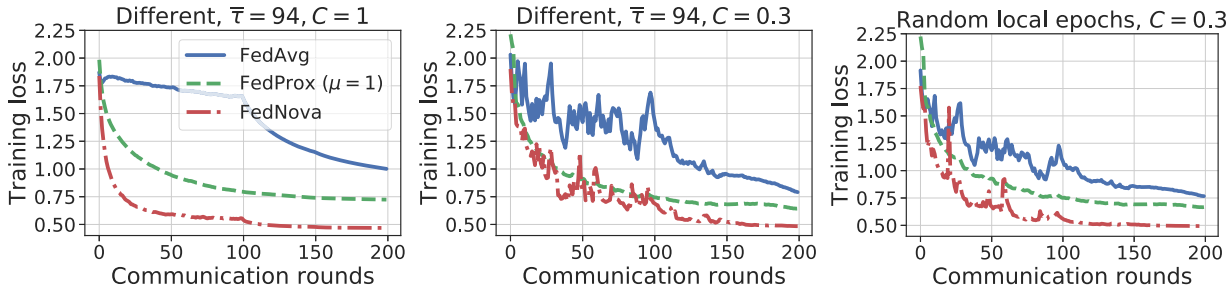


Fig. 5. Results on the synthetic dataset constructed in [21] under three different settings. ***Left***: All clients perform $E_i = 5$ local epochs; ***Middle***: Only $C = 0.3$ fraction of clients are randomly selected per round to perform $E_i = 5$ local epochs; ***Right***: Only $C = 0.3$ fraction of clients are randomly selected per round to perform random and time-varying local epochs $E_i(t) \sim \mathcal{U}(1, 5)$.

TABLE I
RESULTS COMPARING FEDAVG AND FEDNOVA WITH VARIOUS CLIENT OPTIMIZERS (*I.E.*, LOCAL SOLVERS) TRAINED ON NON-IID CIFAR-10 DATASET. FEDPROX AND SCAFFOLD CORRESPOND TO FEDAVG WITH PROXIMAL SGD UPDATES AND CROSS-CLIENT VARIANCE-REDUCTION (VR), RESPECTIVELY

Local Epochs	Client Opt.	Test Accuracy %	
		FEDAVG	FEDNOVA
$E_i = 2$ $(16 \le \tau_i \le 408)$	Vanilla	60.68±1.05	**66.31**±0.86
	Momentum	65.26±2.42	**73.32**±0.29
	Proximal [21]	60.44±1.21	**69.92**±0.34
$E_i^{(t)} \sim \mathcal{U}(2, 5)$ $(16 \le \tau_i^{(t)} \le 1020)$	Vanilla	64.22±1.06	**73.22**±0.32
	Momentum	70.44±2.99	**77.07**±0.12
	Proximal [21]	63.74±1.44	**73.41**±0.45
	VR [15]	74.72±0.34	**74.72**±0.19
	Momen.+VR	Not Defined	**79.19**±0.17

Synthetic Dataset Simulations: In Fig. 5, we observe that by simply changing w_i to p_i, FEDNOVA not only converges significantly faster than FEDAVG but also achieves consistently the best performance under three different settings. Note that the only difference between FEDNOVA and FEDAVG is the aggregated weights when averaging the normalized gradients.

Non-IID CIFAR-10 Experiments: In Table I we compare the performance of FEDNOVA and FEDAVG on non-IID CIFAR-10 with various client optimizers run for 100 communication rounds. When the client optimizer is SGD or SGD with momentum, simply changing the weights yields a 6–9% improvement on the test accuracy; When the client optimizer is proximal SGD, FEDAVG is equivalent to FEDPROX. By setting $\tau_{\text{eff}} = \sum_{i=1}^m p_i \tau_i$ and correcting the weights $w_i = p_i$ while keeping $\boldsymbol{a}_i$ same as FEDPROX, `FedNova-Prox` achieves about 10% higher test accuracy than FEDPROX. It turns out that FEDNOVA consistently converges faster than FEDAVG. When using variance-reduction methods such as SCAFFOLD (that requires doubled communication), FEDNOVA-based method preserves the same test accuracy. Furthermore, combining local momentum and variance-reduction can be easily achieved in FEDNOVA. It yields the highest test accuracy among all other local solvers. This kind of combination is non-trivial and has not appeared yet in the literature.

Effectiveness of Local Momentum: From Table I, it is worth noting that using momentum SGD as the local solver is an effective way to improve the performance. It generally achieves 3–7% higher test accuracy than vanilla SGD. This local momentum scheme can be further combined with server momentum [17], [23], [24]. When $E_i(t) \sim \mathcal{U}(2, 5)$, the hybrid momentum scheme achieves test accuracy 81.15 ± 0.38% As a reference, using server momentum alone achieves 77.49 ± 0.25%.

VIII. DEFERRED PROOFS OF MAIN THEOREMS

A. Proof of Lemma 1: Quadratic Case Analysis

Here, we provide a general proof for arbitrary quadratic functions. Since FEDAVG can be treated as a special case of FEDPROX, we analyze the convergence of FEDPROX instead. Consider a setting where each local objective function is strongly convex and defined as follows:

$$F_i(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{H}_i \boldsymbol{x} - \boldsymbol{e}_i^\top \boldsymbol{x} + \frac{1}{2}\boldsymbol{e}_i^\top \boldsymbol{H}_i^{-1} \boldsymbol{e}_i \ge 0 \tag{21}$$

where $\boldsymbol{H}_i \in \mathbb{R}^{d\times d}$ is an invertible matrix and $\boldsymbol{e}_i \in \mathbb{R}^d$ is an arbitrary vector. It is easy to show that the optimum of the i-th local function is $\boldsymbol{x}_i^* = \boldsymbol{H}_i^{-1}\boldsymbol{e}_i$. Without loss of generality, we assume the global objective function to be a weighted average across all local functions, that is

$$F(\boldsymbol{x}) = \sum_{i=1}^m p_i F_i(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \overline{\boldsymbol{H}} \boldsymbol{x} - \overline{\boldsymbol{e}}^\top \boldsymbol{x} + \frac{1}{2}\sum_{i=1}^m p_i \boldsymbol{e}_i^\top \boldsymbol{H}_i^{-1} \boldsymbol{e}_i$$

where $\overline{\boldsymbol{H}} = \sum_{i=1}^m p_i \boldsymbol{H}_i$ and $\overline{\boldsymbol{e}} = \sum_{i=1}^m p_i \boldsymbol{e}_i$. As a result, the global minimum is $\boldsymbol{x}^* = \overline{\boldsymbol{H}}^{-1}\overline{\boldsymbol{e}}$. Now, let us study whether previous federated optimization algorithms can converge to this global minimum.

The local update rule of FEDPROX for the i-th device can be written as follows:

$$\begin{aligned}\boldsymbol{x}_i^{(t,k+1)} &= \boldsymbol{x}_i^{(t,k)} - \eta\left[\boldsymbol{H}_i \boldsymbol{x}_i^{(t,k)} - \boldsymbol{e}_i + \mu(\boldsymbol{x}_i^{(t,k)} - \boldsymbol{x}^{(t,0)})\right] \\ &= (\boldsymbol{I} - \eta\mu\boldsymbol{I} - \eta\boldsymbol{H}_i)\boldsymbol{x}_i^{(t,k)} + \eta\boldsymbol{e}_i + \eta\mu\boldsymbol{x}^{(t,0)} \end{aligned} \tag{22}$$

where $\boldsymbol{x}_i^{(t,k)}$ denotes the local model parameters at the k-th local iteration after t communication rounds, η denotes the local learning rate and μ is a tunable hyper-parameter in FEDPROX. When $\mu = 0$, the algorithm will reduce to FEDAVG. We omit the device index in $\boldsymbol{x}^{(t,0)}$, since it is synchronized and the same across all devices.

After minor rearranging of (22), we obtain

$$\boldsymbol{x}_i^{(t,k+1)} - \boldsymbol{c}_i^{(t)} = (\boldsymbol{I} - \eta\mu\boldsymbol{I} - \eta\boldsymbol{H}_i)\left(\boldsymbol{x}_i^{(t,k)} - \boldsymbol{c}_i^{(t)}\right). \tag{23}$$

where $\boldsymbol{c}_i^{(t)} = (\boldsymbol{H}_i + \mu\boldsymbol{I})^{-1}(\boldsymbol{e}_i + \mu\boldsymbol{x}^{(t,0)})$. Then, after performing τ_i steps of local updates, the local model becomes

$$\boldsymbol{x}_i^{(t,\tau_i)} = (\boldsymbol{I} - \eta\mu\boldsymbol{I} - \eta\boldsymbol{H}_i)^{\tau_i}\left(\boldsymbol{x}^{(t,0)} - \boldsymbol{c}_i^{(t)}\right) + \boldsymbol{c}_i^{(t)}. \quad (24)$$

Subtracting $\boldsymbol{x}^{(t,0)}$ on both sides and rearranging, it follows that

$$\boldsymbol{x}_i^{(t,\tau_i)} - \boldsymbol{x}^{(t,0)} = \boldsymbol{K}_i\left(\boldsymbol{e}_i - \boldsymbol{H}_i\boldsymbol{x}^{(t,0)}\right) \quad (25)$$

where $\boldsymbol{K}_i = [\boldsymbol{I} - (\boldsymbol{I} - \eta\mu\boldsymbol{I} - \eta\boldsymbol{H}_i)^{\tau_i}](\boldsymbol{H}_i + \mu\boldsymbol{I})^{-1}$.

In FEDPROX and FEDAVG, the server averages all local models according to the sample size

$$\boldsymbol{x}^{(t+1,0)} - \boldsymbol{x}^{(t,0)} = \sum_{i=1}^{m} p_i\boldsymbol{K}_i\left(\boldsymbol{e}_i - \boldsymbol{H}_i\boldsymbol{x}^{(t,0)}\right). \quad (26)$$

Accordingly, we get the following update rule for the central model:

$$\boldsymbol{x}^{(t+1,0)} = \left[\boldsymbol{I} - \sum_{i=1}^{m} p_i\boldsymbol{K}_i\boldsymbol{H}_i\right]\boldsymbol{x}^{(t,0)} + \sum_{i=1}^{m} p_i\boldsymbol{K}_i\boldsymbol{e}_i. \quad (27)$$

This is equivalent to

$$\boldsymbol{x}^{(t+1,0)} - \widetilde{\boldsymbol{x}} = \left[\boldsymbol{I} - \sum_{i=1}^{m} p_i\boldsymbol{K}_i\boldsymbol{H}_i\right]\left[\boldsymbol{x}^{(t,0)} - \widetilde{\boldsymbol{x}}\right]. \quad (28)$$

where

$$\widetilde{\boldsymbol{x}} = \left(\sum_{i=1}^{m} p_i\boldsymbol{K}_i\boldsymbol{H}_i\right)^{-1}\left(\sum_{i=1}^{m} p_i\boldsymbol{K}_i\boldsymbol{e}_i\right). \quad (29)$$

After T communication rounds, one obtains

$$\boldsymbol{x}^{(T,0)} = \left[\boldsymbol{I} - \sum_{i=1}^{m} p_i\boldsymbol{K}_i\boldsymbol{H}_i\right]^{T}\left[\boldsymbol{x}^{(t,0)} - \widetilde{\boldsymbol{x}}\right] + \widetilde{\boldsymbol{x}}. \quad (30)$$

Accordingly, when $\|\boldsymbol{I} - \sum_{i=1}^{m} p_i\boldsymbol{K}_i\boldsymbol{H}_i\|_2 < 1$, the iterates will converge to

$$\lim_{T\to\infty}\boldsymbol{x}^{(T,0)} = \widetilde{\boldsymbol{x}} = \left(\sum_{i=1}^{m} p_i\boldsymbol{K}_i\boldsymbol{H}_i\right)^{-1}\left(\sum_{i=1}^{m} p_i\boldsymbol{K}_i\boldsymbol{e}_i\right). \quad (31)$$

Now let us focus on a concrete example where $p_1 = p_2 = \cdots = p_m = 1/m$, $\boldsymbol{H}_1 = \boldsymbol{H}_2 = \cdots = \boldsymbol{H}_m = \boldsymbol{I}$ and $\mu = 0$. In this case, $\boldsymbol{K}_i = 1 - (1-\eta)^{\tau_i}$. As a result, we have

$$\lim_{T\to\infty}\boldsymbol{x}^{(T,0)} = \frac{\sum_{i=1}^{m}\left[1 - (1-\eta)^{\tau_i}\right]\boldsymbol{e}_i}{\sum_{i=1}^{m}\left[1 - (1-\eta)^{\tau_i}\right]}. \quad (32)$$

Furthermore, when the learning rate is sufficiently small (*e.g.*, can be achieved by gradually decaying the learning rate), according to L'Hôpital's Rule, we obtain

$$\lim_{\eta\to 0}\lim_{T\to\infty}\boldsymbol{x}^{(T,0)} = \frac{\sum_{i=1}^{m}\tau_i\boldsymbol{e}_i}{\sum_{i=1}^{m}\tau_i}. \quad (33)$$

Here, we complete the proof of Lemma 1.

B. Proof of Theorem 1: Convergence of Surrogate Objective

For the ease of writing, let us define the following auxiliary variables:

$$\boldsymbol{h}_i^{(t)} = \frac{1}{a_i}\sum_{k=0}^{\tau_i-1} a_{i,k}\nabla F_i(\boldsymbol{x}_i^{(t,k)}) \quad (34)$$

where $a_{i,k} \geq 0$ is an arbitrary scalar, $\boldsymbol{a}_i = [a_{i,0}, \ldots, a_{i,\tau_i-1}]^\top$, and $a_i = \|\boldsymbol{a}_i\|_1$. One can show that $\mathbb{E}[\boldsymbol{d}_i^{(t)} - \boldsymbol{h}_i^{(t)}] = 0$. In addition, since clients are independent of each other, we have $\mathbb{E}\langle\boldsymbol{d}_i^{(t)} - \boldsymbol{h}_i^{(t)}, \boldsymbol{d}_j^{(t)} - \boldsymbol{h}_j^{(t)}\rangle = 0, \forall i \neq j$.

According to the update rule and Lipschitz-smooth assumption, we have

$$\mathbb{E}[\widetilde{F}(\boldsymbol{x}^{(t+1,0)})] - \widetilde{F}(\boldsymbol{x}^{(t,0)}) \leq \frac{\tau_{\text{eff}}^2\eta^2 L}{2}\underbrace{\mathbb{E}\left[\left\|\sum_{i=1}^{m} w_i\boldsymbol{d}_i^{(t)}\right\|^2\right]}_{T_1} - \tau_{\text{eff}}\eta\underbrace{\mathbb{E}\left[\left\langle\nabla\widetilde{F}(\boldsymbol{x}^{(t,0)}), \sum_{i=1}^{m} w_i\boldsymbol{d}_i^{(t)}\right\rangle\right]}_{T_2} \quad (35)$$

where the expectation is taken over mini-batches $\xi_i^{(t,k)}, \forall i \in \{1,2,\ldots,m\}, k \in \{0,1,\ldots,\tau_i-1\}$. Before diving into the detailed bounds for T_1 and T_2, we first introduce a useful lemma.

Lemma 2: Suppose $\{A_k\}_{k=1}^{T}$ is a sequence of random matrices and $\mathbb{E}[A_k|A_{k-1}, A_{k-2}, \ldots, A_1] = \boldsymbol{0}, \forall k$. Then,

$$\mathbb{E}\left[\left\|\sum_{k=1}^{T} A_k\right\|_{\text{F}}^2\right] = \sum_{k=1}^{T}\mathbb{E}\left[\|A_k\|_{\text{F}}^2\right]. \quad (36)$$

Proof:

$$\begin{aligned}\mathbb{E}\left[\left\|\sum_{k=1}^{T} A_k\right\|_{\text{F}}^2\right] &= \sum_{k=1}^{T}\mathbb{E}\left[\|A_k\|_{\text{F}}^2\right] + \sum_{i=1}^{T}\sum_{j=1,j\neq i}^{T}\times\mathbb{E}\left[\text{Tr}\{A_i^\top A_j\}\right]\\ &= \sum_{k=1}^{T}\mathbb{E}\left[\|A_k\|_{\text{F}}^2\right] + \sum_{i=1}^{T}\sum_{j=1,j\neq i}^{T}\times\text{Tr}\{\mathbb{E}\left[A_i^\top A_j\right]\}\end{aligned}$$

Assume $i < j$. Then, using the law of total expectation,

$$\mathbb{E}\left[A_i^\top A_j\right] = \mathbb{E}\left[A_i^\top\mathbb{E}\left[A_j|A_i,\ldots,A_1\right]\right] = \boldsymbol{0}. \quad (37)$$

■

1) Bounding the First Term in (35): For the First term on the RHS of (35), we have

$$T_1 \leq 2\mathbb{E}\left[\left\|\sum_{i=1}^{m} w_i\left(\boldsymbol{d}_i^{(t)} - \boldsymbol{h}_i^{(t)}\right)\right\|^2\right] + 2\mathbb{E}\left[\left\|\sum_{i=1}^{m} w_i\boldsymbol{h}_i^{(t)}\right\|^2\right] \quad (38)$$

$$= 2\sum_{i=1}^{m} w_i^2\mathbb{E}\left[\left\|\boldsymbol{d}_i^{(t)} - \boldsymbol{h}_i^{(t)}\right\|^2\right] + 2\mathbb{E}\left[\left\|\sum_{i=1}^{m} w_i\boldsymbol{h}_i^{(t)}\right\|^2\right] \quad (39)$$

where (38) follows from the fact: $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ and (39) uses the special property of $\boldsymbol{d}_i^{(t)}, \boldsymbol{h}_i^{(t)}$, that is, $\mathbb{E}\langle\boldsymbol{d}_i^{(t)} - \boldsymbol{h}_i^{(t)}, \boldsymbol{d}_j^{(t)} - \boldsymbol{h}_j^{(t)}\rangle = 0, \forall i \neq j$. Then, let us expand the expression of $\boldsymbol{d}_i^{(t)}$ and $\boldsymbol{h}_i^{(t)}$, to obtain that

$$T_1 \leq \sum_{i=1}^{m}\frac{2w_i^2}{a_i^2}\sum_{k=0}^{\tau_i-1}[a_{i,k}]^2\mathbb{E}\left[\left\|g_i(\boldsymbol{x}_i^{(t,k)}) - \nabla F_i(\boldsymbol{x}_i^{(t,k)})\right\|^2\right]$$

$$+ 2\mathbb{E}\left[\left\|\sum_{i=1}^{m} w_i \boldsymbol{h}_i^{(t)}\right\|^2\right] \tag{40}$$

$$\leq 2\sigma^2 \sum_{i=1}^{m} \frac{w_i^2 \left\|\boldsymbol{a}_i\right\|^2}{\left\|\boldsymbol{a}_i\right\|_1^2} + 2\mathbb{E}\left[\left\|\sum_{i=1}^{m} w_i \boldsymbol{h}_i^{(t)}\right\|^2\right] \tag{41}$$

where (40) is derived using Lemma 2, and (41) follows Assumption 2.

2) Bounding the Second Term in (35): For the second term on the right hand side (RHS) in (35), we have

$$T_2 = \mathbb{E}\left[\left\langle \nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}), \sum_{i=1}^{m} w_i \left(\boldsymbol{d}_i^{(t)} - \boldsymbol{h}_i^{(t)}\right)\right\rangle\right] + \mathbb{E}\left[\left\langle \nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}), \sum_{i=1}^{m} w_i \boldsymbol{h}_i^{(t)}\right\rangle\right] \tag{42}$$

$$= \mathbb{E}\left[\left\langle \nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}), \sum_{i=1}^{m} w_i \boldsymbol{h}_i^{(t)}\right\rangle\right] \tag{43}$$

$$= \frac{1}{2}\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t)})\right\|^2 + \frac{1}{2}\mathbb{E}\left[\left\|\sum_{i=1}^{m} w_i \boldsymbol{h}_i^{(t)}\right\|^2\right] - \frac{1}{2}\mathbb{E}\left[\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}) - \sum_{i=1}^{m} w_i \boldsymbol{h}_i^{(t)}\right\|^2\right] \tag{44}$$

where the last equation uses the fact: $2\langle a, b\rangle = \|a\|^2 + \|b\|^2 - \|a-b\|^2$.

3) Intermediate Result: Substituting (44) and (41) back into (35) and assuming $\tau_{\text{eff}}\eta L \leq 1/2$, we have

$$\frac{\mathbb{E}\left[\widetilde{F}(\boldsymbol{x}^{(t+1,0)})\right] - \widetilde{F}(\boldsymbol{x}^{(t,0)})}{\eta\tau_{\text{eff}}} \leq -\frac{1}{2}\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2 + \tau_{\text{eff}}\eta L\sigma^2 \sum_{i=1}^{m} \frac{w_i^2 \left\|\boldsymbol{a}_i\right\|_2^2}{\left\|\boldsymbol{a}_i\right\|_1^2} + \frac{1}{2}\mathbb{E}\left[\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}) - \sum_{i=1}^{m} w_i \boldsymbol{h}_i^{(t)}\right\|^2\right] \tag{45}$$

$$\leq -\frac{1}{2}\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2 + \tau_{\text{eff}}\eta L\sigma^2 \sum_{i=1}^{m} \frac{w_i^2 \left\|\boldsymbol{a}_i\right\|_2^2}{\left\|\boldsymbol{a}_i\right\|_1^2} + \frac{1}{2}\sum_{i=1}^{m} w_i \mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)}) - \boldsymbol{h}_i^{(t)}\right\|^2\right] \tag{46}$$

where the last inequality uses the fact $\widetilde{F}(\boldsymbol{x}) = \sum_{i=1}^{m} w_i F_i(\boldsymbol{x})$ and Jensen's Inequality: $\|\sum_{i=1}^{m} w_i z_i\|^2 \leq \sum_{i=1}^{m} w_i \|z_i\|^2$. In order to bound the last term in (46), we can use the following lemma.

Lemma 3: The difference between the locally averaged gradient and the server gradient $\nabla F_i(\boldsymbol{x}^{(t,0)})$ can be bounded as follows:

$$\sum_{i=1}^{m} w_i \mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)}) - \boldsymbol{h}_i^{(t)}\right\|^2\right] \leq \frac{1}{2}\mathbb{E}\left[\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2\right] + 3\eta^2 L^2 \sigma^2 B + 6\eta^2 L^2 \kappa^2 C \tag{47}$$

where $B = \Lambda \sum_{i=1}^{m} w_i(\tau_i - 1)\|\boldsymbol{a}_i\|_2^2 / \|\boldsymbol{a}_i\|_1, C = \Lambda^2 \max_i \tau_i (\tau_i - 1)$, and Λ denotes the upper bound of all elements in any accumulation vector $\boldsymbol{a}_i$. That is, $\Lambda = \max_{i,s,k} a_{i,s}(k)$.

Proof: Due to space limitations, we delegate the proof to Appendix A-F. ■

4) Final Results: Substituting (47) back into (46), we have

$$\frac{\mathbb{E}\left[\widetilde{F}(\boldsymbol{x}^{(t+1,0)})\right] - \widetilde{F}(\boldsymbol{x}^{(t,0)})}{\eta\tau_{\text{eff}}} \leq -\frac{1}{4}\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2 + \tau_{\text{eff}}\eta L\sigma^2 \sum_{i=1}^{m} \frac{w_i^2 \left\|\boldsymbol{a}_i\right\|_2^2}{\left\|\boldsymbol{a}_i\right\|_1^2} + \frac{3}{2}\eta^2 L^2 \sigma^2 B + 3\eta^2 L^2 \kappa^2 C \tag{48}$$

Taking the average across all rounds, we get

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2\right] \leq \frac{4\left[\widetilde{F}(\boldsymbol{x}^{(0,0)}) - \widetilde{F}_{\text{inf}}\right]}{\eta\tau_{\text{eff}}T} + \frac{4\eta L\sigma^2 A}{m} + 6\eta^2 L^2 \sigma^2 B + 12\eta^2 L^2 \kappa^2 C.$$

where $A = m\tau_{\text{eff}} \sum_{i=1}^{m} \frac{w_i^2 \|\boldsymbol{a}_i\|_2^2}{\|\boldsymbol{a}_i\|_1^2}$. Since $\min \|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\|^2 \leq \frac{1}{T}\sum_{t=0}^{T-1} \|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\|^2$, we have

$$\min_{t\in[T]} \mathbb{E}\left[\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2\right] \leq \frac{4\left[\widetilde{F}(\boldsymbol{x}^{(0,0)}) - \widetilde{F}_{\text{inf}}\right]}{\eta\tau_{\text{eff}}T} + \frac{4\eta L\sigma^2 A}{m} + 6\eta^2 L^2 \sigma^2 B + 12\eta^2 L^2 \kappa^2 C. \tag{49}$$

5) Constraint on the Local Learning Rate: Here, let us summarize the constraints on the local learning rate:

$$\eta L \leq \frac{1}{2\tau_{\text{eff}}}, \tag{50}$$

$$4\eta^2 L^2 \max_i \{\|\boldsymbol{a}_i\|_1 (\|\boldsymbol{a}_i\|_1 - a_{i,-1})\} \leq \frac{1}{2\beta^2 + 1}. \tag{51}$$

For the second constraint, we can further tighten it as follows:

$$4\eta^2 L^2 \max_i \|\boldsymbol{a}_i\|_1^2 \leq \frac{1}{2\beta^2 + 1} \tag{52}$$

That is,

$$\eta L \leq \frac{1}{2}\min\left\{\frac{1}{\max_i \|\boldsymbol{a}_i\|_1 \sqrt{2\beta^2 + 1}}, \frac{1}{\tau_{\text{eff}}}\right\}. \tag{53}$$

6) Further Optimizing the Bound: By setting $\eta = \sqrt{\frac{m}{\overline{\tau}T}}$ where $\overline{\tau} = \frac{1}{m}\sum_{i=1}^{m} \tau_i$, $\min_{t\in[T]} \mathbb{E}\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\|^2$ will be upper bounded by

$$\mathcal{O}\left(\frac{\overline{\tau}/\tau_{\text{eff}}}{\sqrt{m\overline{\tau}T}}\right) + \mathcal{O}\left(\frac{A\sigma^2}{\sqrt{m\overline{\tau}T}}\right) + \mathcal{O}\left(\frac{mB\sigma^2}{\overline{\tau}T}\right) + \mathcal{O}\left(\frac{mC\kappa^2}{\overline{\tau}T}\right).$$

Here, we complete the proof of Theorem 1.

IX. Concluding Remarks

In federated learning, the participating clients (*e.g.*, IoT sensors, mobile devices) are typically highly heterogeneous, both in the size of their local datasets and in their computation speeds. Clients can also join and leave the training at any time according to their availabilities. Therefore, it is common that clients perform different amounts of works within one round of local computation. However, previous analyses of federated optimization algorithms have been limited to the homogeneous case where all clients have the same local steps, hyper-parameters, and client optimizers. In this paper, we have developed a novel theoretical framework to analyze the challenging heterogeneous setting. We have shown that original FedAvg algorithm will converge to a stationary point of a mismatched objective function which can be arbitrarily different from the true objective. To the best of our knowledge, we have thus provided the first fundamental understanding of how the convergence rate and bias in the final solution of federated optimization algorithms are influenced by heterogeneity in clients' local progress. The new framework naturally allows clients to have different local steps and local solvers, such as GD, SGD, SGD with momentum, proximal updates, etc. Inspired by the theoretical analysis, we have proposed FedNova, which can automatically adjust the aggregated weights and effective local steps according to the local progress. We have validated the effectiveness of FedNova both theoretically and empirically. On a non-IID version of the CIFAR-10 dataset, FedNova generally achieves 6–9% higher test accuracy than FedAvg. Future directions include extending the theoretical framework to adaptive optimization methods or gossip-based training methods.

Future Directions: There are many open directions to extend this work. For example, the main theorems are based on Assumption 3. However, this assumption on dissimilarity among local objectives can be removed when using cross-client variance-reduction techniques [15]. Besides, as illustrated by Theorem 2, the bias term is caused by improper weighting scheme as well as the differences between local objectives. While our proposed algorithm FedNova corrects the weighting scheme, we believe algorithms that reduce the differences among clients' local updates can also mitigate the objective inconsistency problem. Furthermore, our current algorithmic framework requires the local model changes to be a linear combination of gradients and cannot work for adaptive optimization methods. Given the popularity of Adam [43] and AdaGrad [44] on language-related training tasks, the adaptive variants of FedNova could be a promising direction.

Appendix
Proofs of Other Theorems

A. Proof of Theorem 2: Including Bias in the Error Bound

Lemma 4: For any model parameter $\boldsymbol{x}$, the difference between the gradients of $F(\boldsymbol{x})$ and $\widetilde{F}(\boldsymbol{x})$ can be bounded as follows:

$$\left\|\nabla F(\boldsymbol{x}) - \nabla \widetilde{F}(\boldsymbol{x})\right\|^2 \leq \chi^2_{\boldsymbol{p}\|\boldsymbol{w}} \left[(\beta^2 - 1)\left\|\nabla \widetilde{F}(\boldsymbol{x})\right\|^2 + \kappa^2\right]$$

where $\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}$ denotes the chi-square distance between $\boldsymbol{p}$ and $\boldsymbol{w}$, *i.e.*, $\chi^2_{\boldsymbol{p}\|\boldsymbol{w}} = \sum_{i=1}^m (p_i - w_i)^2 / w_i$.

Proof: According to the definition of $F(x)$ and $\widetilde{F}(\boldsymbol{x})$, we have

$$\nabla F(x) - \nabla \widetilde{F}(\boldsymbol{x}) = \sum_{i=1}^m (p_i - w_i)\nabla F_i(\boldsymbol{x}) \tag{54}$$

$$= \sum_{i=1}^m \frac{p_i - w_i}{\sqrt{w_i}} \cdot \sqrt{w_i}\left(\nabla F_i(\boldsymbol{x}) - \nabla \widetilde{F}(\boldsymbol{x})\right). \tag{55}$$

Applying Cauchy-Schwarz inequality, it follows that

$$\left\|\nabla F(x) - \nabla \widetilde{F}(\boldsymbol{x})\right\|^2 \leq \left[\sum_{i=1}^m \frac{(p_i - w_i)^2}{w_i}\right]\left[\sum_{i=1}^m w_i \left\|\nabla F_i(x) - \nabla \widetilde{F}(\boldsymbol{x})\right\|^2\right] \tag{56}$$

$$\leq \chi^2_{\boldsymbol{p}\|\boldsymbol{w}}\left[(\beta^2 - 1)\left\|\nabla \widetilde{F}(\boldsymbol{x})\right\|^2 + \kappa^2\right]. \tag{57}$$

where the last inequality uses Assumption 3. ■

Note that

$$\|\nabla F(\boldsymbol{x})\|^2 \leq 2\left\|\nabla F(\boldsymbol{x}) - \nabla \widetilde{F}(\boldsymbol{x})\right\|^2 + 2\left\|\nabla \widetilde{F}(\boldsymbol{x})\right\|^2 \tag{58}$$

$$\leq 2\left[\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}(\beta^2 - 1) + 1\right]\left\|\nabla \widetilde{F}(\boldsymbol{x})\right\|^2 + 2\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}\kappa^2. \tag{59}$$

As a result, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla F(\boldsymbol{x}^{(t,0)})\right\|^2 \leq 2\left[\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}(\beta^2 - 1) + 1\right]\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2 + 2\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}\kappa^2$$

$$\leq 2\left[\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}(\beta^2 - 1) + 1\right]\epsilon_{\text{opt}} + 2\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}\kappa^2 \tag{60}$$

where ϵ_{opt} denotes the optimization error.

B. Constructing a Lower Bound

In this subsection, we are going to construct a lower bound of $\mathbb{E}\|\nabla F(\boldsymbol{x}^{(t,0)})\|^2$, showing that (13) is tight and the non-vanishing error term in Theorem 2 is not an artifact of our analysis.

Lemma 5: One can manually construct a strongly convex objective function such that FedAvg with heterogeneous local updates cannot converge to its global optimum. In particular, the gradient norm of the objective function does not vanish as learning rate approaches to zero. We have the following lower bound:

$$\lim_{T\to\infty}\mathbb{E}\left\|\nabla F(\boldsymbol{x}^{(T,0)})\right\|^2 = \Omega(\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}\kappa^2) \tag{61}$$

where $\chi^2_{\boldsymbol{p}\|\boldsymbol{w}}$ denotes the chi-square divergence between weight vectors and κ^2 quantifies the dissimilarities among local objective functions and is defined in Assumption 3.

Proof: Suppose that there are only two clients with local objectives $F_1(x) = \frac{1}{2}(x-a)^2$ and $F_2(x) = \frac{1}{2}(x+a)^2$. The global objective is defined as $F(x) = \frac{1}{2}F_1(x) + \frac{1}{2}F_2(x)$. For

any set of weights $w_1, w_2, w_1 + w_2 = 1$, we define the surrogate objective function as $\widetilde{F}(\boldsymbol{x}) = w_1 F_1(\boldsymbol{x}) + w_2 F_2(\boldsymbol{x})$. As a consequence, we have

$$\sum_{i=1}^{m} w_i \left\| \nabla F_i(x) - \nabla \widetilde{F}(x) \right\|^2 = 2 w_1 w_2 a^2 \tag{62}$$

Comparing with Assumption 3, we can define $\kappa^2 = 2 w_1 w_2 a^2$ and $\beta^2 = 1$ in this case. Furthermore, according to the derivations in Section VIII-A, the iterate of FEDAVG can be written as follows:

$$\lim_{T\to\infty} x^{(T,0)} = \frac{\tau_1 a - \tau_2 a}{\tau_1 + \tau_2}. \tag{63}$$

In this case, $w_1 = \tau_1/(\tau_1+\tau_2), w_2 = \tau_2/(\tau_1+\tau_2)$. As a results, we have

$$\begin{aligned}
&\lim_{T\to\infty} \left\| \nabla F(x^{(T,0)}) \right\|^2 \\
&= \lim_{T\to\infty} \left[\frac{1}{2}(x^{(T,0)} - a) + \frac{1}{2}(x^{(T,0)} + a) \right]^2 \\
&= \left(\frac{\tau_1 - \tau_2}{\tau_1 + \tau_2} \right)^2 a^2 \\
&= \frac{(\tau_2 - \tau_1)^2}{2\tau_1\tau_2} \kappa^2 = \Omega(\chi^2_{\boldsymbol{p}\|\boldsymbol{w}} \kappa^2).
\end{aligned} \tag{64}$$

where $\chi^2_{\boldsymbol{p}\|\boldsymbol{w}} = \sum_{i=1}^m (p_i - w_i)^2/w_i = (w_1 - 1/2)^2/w_1 + (w_2 - 1/2)^2/w_2 = (\tau_2 - \tau_1)^2/(2\tau_1\tau_2)$. ■

C. Proof of Theorem 3

The main part of the proof is nearly the same as the proof of Theorem 1, except for a few initial steps. According to the Lipschitz-smooth assumption, it follows that

$$\begin{aligned}
&\mathbb{E}\left[\widetilde{F}(\boldsymbol{x}^{(t+1,0)})\right] - \widetilde{F}(\boldsymbol{x}^{(t,0)}) \\
&\le \underbrace{\mathbb{E}\left[\left\langle \nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}), \sum_{j=1}^{q} \frac{\Delta_{l_j}^{(t)}}{q} \right\rangle \right]}_{T_3} + \frac{L}{2} \underbrace{\mathbb{E}\left[\left\| \sum_{j=1}^{q} \frac{\Delta_{l_j}^{(t)}}{q} \right\|^2 \right]}_{T_4}
\end{aligned} \tag{65}$$

where the expectation is taken over randomly selected indices $\{l_j\}$ as well as mini-batches $\xi_i^{(t,k)}, \forall i \in \{1,2,\ldots,m\}, k \in \{0,1,\ldots,\tau_i - 1\}$.

For the first term in (65), we can first take the expectation over indices and obtain

$$T_3 = \mathbb{E}\left[\left\langle \nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}), \sum_{i=1}^{m} p_i \Delta_i^{(t)} \right\rangle \right] \tag{66}$$

$$= -\tau_{\text{eff}} \eta \mathbb{E}\left[\left\langle \nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}), \sum_{i=1}^{m} w_i \boldsymbol{d}_i^{(t)} \right\rangle \right] \tag{67}$$

where $\tau_{\text{eff}} = \sum_{i=1}^m p_i \|\boldsymbol{a}_i\|_1, w_i = p_i\|\boldsymbol{a}_i\|_1 / \sum_{i=1}^m (p_i \|\boldsymbol{a}_i\|_1)$. This term is exactly the same as the first term in (35). We can directly reuse previous results in the proof of Theorem 1. Comparing with (44), we have

$$\begin{aligned}
T_3 \le &-\frac{\tau_{\text{eff}}\eta}{2} \left\| \nabla \widetilde{F}(\boldsymbol{x}^{(t)}) \right\|^2 - \frac{\tau_{\text{eff}}\eta}{2} \mathbb{E}\left[\left\| \sum_{i=1}^{m} w_i \boldsymbol{h}_i^{(t)} \right\|^2 \right] \\
&+ \frac{\tau_{\text{eff}}\eta}{2} \sum_{i=1}^{m} w_i \mathbb{E}\left[\left\| \nabla F_i(\boldsymbol{x}^{(t,0)}) - \boldsymbol{h}_i^{(t)} \right\|^2 \right].
\end{aligned} \tag{68}$$

For the second term in (65), we have

$$T_4 = \eta^2 \tau_{\text{eff}}^2 \mathbb{E}\left[\left\| \frac{1}{q} \sum_{j=1}^{q} \frac{w_{l_j}}{p_{l_j}} \boldsymbol{d}_{l_j}^{(t)} \right\|^2 \right] \tag{69}$$

$$\begin{aligned}
&= \eta^2 \tau_{\text{eff}}^2 \mathbb{E}\left[\left\| \frac{1}{q} \sum_{j=1}^{q} \frac{w_{l_j}}{p_{l_j}} \boldsymbol{h}_{l_j}^{(t)} \right\|^2 \right] \\
&\quad + \eta^2 \tau_{\text{eff}}^2 \mathbb{E}\left[\left\| \frac{1}{q} \sum_{j=1}^{q} \frac{w_{l_j}}{p_{l_j}} (\boldsymbol{d}_{l_j}^{(t)} - \boldsymbol{h}_{l_j}^{(t)}) \right\|^2 \right]
\end{aligned} \tag{70}$$

$$\le \eta^2 \tau_{\text{eff}}^2 \underbrace{\mathbb{E}\left[\left\| \frac{1}{q} \sum_{j=1}^{q} \frac{w_{l_j}}{p_{l_j}} \boldsymbol{h}_{l_j}^{(t)} \right\|^2 \right]}_{T_5} + \frac{1}{q} \sum_{i=1}^{m} \frac{w_i^2 \sigma^2}{p_i} \frac{\|\boldsymbol{a}_i\|_2^2}{\|\boldsymbol{a}_i\|_1^2} \tag{71}$$

where the last inequality follows Assumption 2. Additionally, for the term T_5, we can bound it as follows:

$$\begin{aligned}
T_5 \le &\, 3\mathbb{E}\left\| \frac{1}{q} \sum_{j=1}^{q} \frac{w_{l_j}}{p_{l_j}} (\boldsymbol{h}_{l_j}^{(t)} - \nabla F_{l_j}(\boldsymbol{x}^{(t,0)})) \right\|^2 \\
&+ 3\mathbb{E}\left\| \frac{1}{q} \sum_{j=1}^{q} \frac{w_{l_j}}{p_{l_j}} \nabla F_{l_j}(\boldsymbol{x}^{(t,0)}) - \nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}) \right\|^2 \\
&+ 3 \left\| \nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}) \right\|^2 \\
\le &\, 3r \sum_{i=1}^{m} w_i \left\| \boldsymbol{h}_i^{(t)} - \nabla F_i(\boldsymbol{x}^{(t,0)}) \right\|^2 \\
&+ 3\left(1 + \frac{r\beta^2}{q}\right) \left\| \nabla \widetilde{F}(\boldsymbol{x}^{(t,0)}) \right\|^2 + 3r\kappa^2
\end{aligned} \tag{72}$$

where r is defined as $\max_i w_i/p_i$. The derivation of (72) is based on the fact that $\sum_{i=1}^m p_i \|w_i \nabla F_i / p_i\|^2 \le r \sum_{i=1}^m w_i \|\nabla F_i\|^2$ and Assumption 3. Substituting T_3, T_4, T_5 into (65) and applying Lemma 3, one can complete the proof.

D. Proof of Theorem 4

Since each local objective is c-strongly convex, their weighted summation $\widetilde{F}(\boldsymbol{x}) = \sum_{i=1}^m w_i F_i(\boldsymbol{x})$ is also c-strongly convex and satisfies the PL condition. Substituting the PL condition into (48), we have

$$\begin{aligned}
&\frac{\mathbb{E}[\widetilde{F}(\boldsymbol{x}^{(t+1,0)})] - \widetilde{F}(\boldsymbol{x}^{(t,0)})}{\eta\tau_{\text{eff}}} \\
&\le -\frac{c[\widetilde{F}(\boldsymbol{x}^{(t,0)}) - \widetilde{F}_{\text{inf}}]}{2} + \frac{\eta L \sigma^2 A}{m} \\
&\quad + \frac{3}{2}\eta^2 L^2 \sigma^2 B + 3\eta^2 L^2 \kappa^2 C.
\end{aligned} \tag{73}$$

After minor rearranging, we obtain

$$\mathbb{E}[\widetilde{F}(\boldsymbol{x}^{(t+1,0)})] - \widetilde{F}_{\text{inf}} \leq \left(1 - \frac{\eta\tau_{\text{eff}}c}{2}\right)[\widetilde{F}(\boldsymbol{x}^{(t,0)}) - \widetilde{F}_{\text{inf}}] + \frac{\eta^2\tau_{\text{eff}}L\sigma^2 A}{m} + \frac{3}{2}\eta^3\tau_{\text{eff}}L^2\sigma^2 B + 3\eta^3\tau_{\text{eff}}L^2\kappa^2 C. \quad (74)$$

For ease of writing, we define $\widetilde{\eta}^{(t)} = \eta^{(t)}\overline{\tau}$, $s = \tau_{\text{eff}}/\overline{\tau}$, $D = sL\sigma^2 A/(\overline{\tau}m)$ and $E = 3sL^2\sigma^2 B/(2\overline{\tau}^2) + 3sL^2\kappa^2 C/\overline{\tau}^2$. Let us first prove by induction that, for any $t \geq 0$, $\mathbb{E}[\widetilde{F}(\boldsymbol{x}^{(t,0)})] - \widetilde{F}_{\text{inf}} \leq \widetilde{\eta}^{(t)}\beta D + [\widetilde{\eta}^{(t)}]^2\beta E$. We assume this holds for $t > 0$. According to (73), we have

$$\begin{aligned}
&\mathbb{E}[\widetilde{F}(\boldsymbol{x}^{(t+1,0)})] - \widetilde{F}_{\text{inf}} \\
&\leq \left(1 - \frac{\widetilde{\eta}^{(t)}sc}{2}\right)\left(\widetilde{\eta}^{(t)}\beta D + [\widetilde{\eta}^{(t)}]^2\beta E\right) + [\widetilde{\eta}^{(t)}]^2 D + [\widetilde{\eta}^{(t)}]^3 E \\
&= \left[\beta\left(1 - \frac{\widetilde{\eta}^{(t)}sc}{2}\right) + \widetilde{\eta}^{(t)}\right]\left[\widetilde{\eta}^{(t)}D + [\widetilde{\eta}^{(t)}]^2 E\right]. \quad (75)
\end{aligned}$$

Let us set $\beta = \frac{6}{sc}$ and $\widetilde{\eta}^{(t)} = \frac{6}{sc(t+\gamma)}$. After some manipulations, one can show that

$$\left[\beta\left(1 - \frac{\widetilde{\eta}^{(t)}sc}{2}\right) + \widetilde{\eta}^{(t)}\right]\widetilde{\eta}^{(t)} \leq \beta\widetilde{\eta}^{(t+1)}, \quad (76)$$

$$\left[\beta\left(1 - \frac{\widetilde{\eta}^{(t)}sc}{2}\right) + \widetilde{\eta}^{(t)}\right][\widetilde{\eta}^{(t)}]^2 \leq \frac{216}{s^3c^3}\frac{1}{(t+1+\gamma)^2} = \beta[\widetilde{\eta}^{(t+1)}]^2 \quad (77)$$

Substituting (76) and (77) into (75), we have

$$\mathbb{E}[\widetilde{F}(\boldsymbol{x}^{(t+1,0)})] - \widetilde{F}_{\text{inf}} \leq \beta\widetilde{\eta}^{(t+1)}D + \beta[\widetilde{\eta}^{(t+1)}]^2 E. \quad (78)$$

When $t = 0$, all the hyper-parameters should satisfy

$$\widetilde{F}(\boldsymbol{x}^{(0,0)}) - \widetilde{F}_{\text{inf}} \leq \frac{36}{c^2\gamma}\frac{L\sigma^2 A}{\tau_{\text{eff}}m} + \frac{216}{c^3\gamma^2}\left(\frac{3L^2\sigma^2 B}{2\tau_{\text{eff}}^2} + \frac{3L^2\kappa^2 C}{\tau_{\text{eff}}^2}\right). \quad (79)$$

Substituting the definition of A into (79),

$$\begin{aligned}
\widetilde{F}(\boldsymbol{x}^{(0,0)}) - \widetilde{F}_{\text{inf}} &\leq \frac{36L\sigma^2}{c^2\gamma}\sum_{i=1}^{m}\frac{w_i^2\|\boldsymbol{a}_i\|_2^2}{\|\boldsymbol{a}_i\|_1^2} + \frac{648L^2(\sigma^2 B + \kappa^2 C)}{c^3\gamma^2\tau_{\text{eff}}^2} \\
&\leq \frac{36L\sigma^2}{c^2\gamma} + \frac{648L^2(\sigma^2 B + \kappa^2 C)}{c^3\gamma^2\tau_{\text{eff}}^2}. \quad (80)
\end{aligned}$$

After minor rearranging, we get the constraint on τ_{eff} as follows:

$$\tau_{\text{eff}}^2 \leq \frac{648L^2(\sigma^2 B + \kappa^2 C)}{c^3\gamma^2[\widetilde{F}(\boldsymbol{x}^{(0,0)}) - \widetilde{F}_{\text{inf}}] - 36c\gamma L\sigma^2} \quad (81)$$

When we set $\gamma = L/(\nu c)$, it follows that

$$\tau_{\text{eff}}^2 \leq \frac{648\nu^2(\sigma^2 B + \kappa^2 C)}{cL^2[\widetilde{F}(\boldsymbol{x}^{(0,0)}) - \widetilde{F}_{\text{inf}}] - 36\nu L^2\sigma^2}. \quad (82)$$

After a total of T communication rounds,

$$\begin{aligned}
&\widetilde{F}(\boldsymbol{x}^{(T,0)}) - \widetilde{F}_{\text{inf}} \\
&\leq \frac{36}{sc^2}\frac{L\sigma^2 A}{(T+\gamma)\overline{\tau}m} + \frac{216}{s^2c^3(T+\gamma)^2\overline{\tau}^2}\left(\frac{3L^2\sigma^2 B}{2} + 3L^2\kappa^2 C\right) \\
&= \mathcal{O}\left(\frac{L}{sc^2}\frac{\sigma^2 A}{mT\overline{\tau}}\right) + \mathcal{O}\left(\frac{L^2}{s^2c^3}\frac{\sigma^2 B + \kappa^2 C}{T^2\overline{\tau}^2}\right). \quad (83)
\end{aligned}$$

E. Proof of Theorem 5

In the case of FEDNOVA, the aggregated weights w_i equals to p_i. Therefore, the surrogate objective $\widetilde{F}(\boldsymbol{x}) = \sum_{i=1}^m w_i F_i(\boldsymbol{x})$ is the same as the original objective function $F(\boldsymbol{x}) = \sum_{i=1}^m p_i F_i(\boldsymbol{x})$. We can directly reuse the intermediate results in the proof of Theorem 1. According to (48), for the t-th round, we have

$$\begin{aligned}
&\frac{\mathbb{E}[F(\boldsymbol{x}^{(t+1,0)})] - F(\boldsymbol{x}^{(t,0)})}{\eta\tau_{\text{eff}}} \\
&\leq -\frac{1}{4}\left\|\nabla F(\boldsymbol{x}^{(t,0)})\right\|^2 + \frac{\eta L\sigma^2 A^{(t)}}{m} + \frac{3}{2}\eta^2 L^2\sigma^2 B^{(t)} + 3\eta^2 L^2\kappa^2 C^{(t)} \quad (84)
\end{aligned}$$

where quantities $A^{(t)}, B^{(t)}, C^{(t)}$ have the same definitions as (10) to (12), except replacing $\boldsymbol{a}_i$ with $\boldsymbol{a}_i^{(t)}$. Then, taking the total expectation and averaging over all rounds, it follows that

$$\begin{aligned}
&\frac{\mathbb{E}[F(\boldsymbol{x}^{(T,0)})] - F(\boldsymbol{x}^{(0,0)})}{\eta\tau_{\text{eff}}T} \\
&\leq -\frac{1}{4T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla F(\boldsymbol{x}^{(t,0)})\right\|^2 + \frac{\eta L\sigma^2\widetilde{A}}{m} + \frac{3}{2}\eta^2 L^2\sigma^2\widetilde{B} + 3\eta^2 L^2\kappa^2\widetilde{C} \quad (85)
\end{aligned}$$

where $\widetilde{A} = \sum_{t=0}^{T-1} A^{(t)}/T$, $\widetilde{B} = \sum_{t=0}^{T-1} B^{(t)}/T$, and $\widetilde{C} = \sum_{t=0}^{T-1} C^{(t)}/T$. Finally, repeating the same procedure in the proof of Theorem 1, we complete the proof.

F. Proof of Lemma 3

Recall the definition of $\boldsymbol{h}_i^{(t)}$, one can derive that

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)}) - \boldsymbol{h}_i^{(t)}\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)}) - \frac{1}{a_i}\sum_{k=0}^{\tau_i-1} a_{i,k}\nabla F_i(\boldsymbol{x}_i^{(t,k)})\right\|^2\right] \quad (86) \\
&= \mathbb{E}\left[\left\|\frac{1}{a_i}\sum_{k=0}^{\tau_i-1} a_{i,k}\left(\nabla F_i(\boldsymbol{x}^{(t,0)}) - \nabla F_i(\boldsymbol{x}_i^{(t,k)})\right)\right\|^2\right] \quad (87) \\
&\leq \frac{1}{a_i}\sum_{k=0}^{\tau_i-1}\left\{a_{i,k}\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)}) - \nabla F_i(\boldsymbol{x}_i^{(t,k)})\right\|^2\right]\right\} \quad (88) \\
&\leq \frac{L^2}{a_i}\sum_{k=0}^{\tau_i-1}\left\{a_{i,k}\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)} - \boldsymbol{x}_i^{(t,k)}\right\|^2\right]\right\} \quad (89)
\end{aligned}$$

$$\leq \frac{L^2\Lambda}{a_i}\sum_{k=0}^{\tau_i-1}\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)}-\boldsymbol{x}_i^{(t,k)}\right\|^2\right] \tag{90}$$

where Λ denotes the upper bound of all elements in the accumulation vector, (88) uses Jensen's Inequality again: $\|\sum_{i=1}^m w_i z_i\|^2 \leq \sum_{i=1}^m w_i\|z_i\|^2$, and (89) follows Assumption 1. Now, we turn to bounding the difference between the server model $\boldsymbol{x}^{(t,0)}$ and the local model $\boldsymbol{x}_i^{(t,k)}$. Plugging into the local update rule and using the fact $\|a+b\|^2 \leq 2\|a\|^2+2\|b\|^2$,

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)}-\boldsymbol{x}_i^{(t,k)}\right\|^2\right]\\ &\leq 2\eta^2\mathbb{E}\left[\left\|\sum_{s=0}^{k-1}a_{i,s}(k)\left(g_i(\boldsymbol{x}_i^{(t,s)})-\nabla F_i(\boldsymbol{x}_i^{(t,s)})\right)\right\|^2\right] &(91)\\ &\quad+2\eta^2\mathbb{E}\left[\left\|\sum_{s=0}^{k-1}a_{i,s}(k)\nabla F_i(\boldsymbol{x}_i^{(t,s)})\right\|^2\right] &(92)\end{aligned}$$

Applying Lemma 2 to the first term,

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)}-\boldsymbol{x}_i^{(t,k)}\right\|^2\right]\\ &\leq 2\eta^2\sigma^2\sum_{s=0}^{k-1}[a_{i,s}(k)]^2+2\eta^2\mathbb{E}\left[\left\|\sum_{s=0}^{k-1}a_{i,s}(k)\nabla F_i(\boldsymbol{x}_i^{(t,s)})\right\|^2\right] &(93)\\ &\leq 2\eta^2\sigma^2\sum_{s=0}^{k-1}[a_{i,s}(k)]^2\\ &\quad+2\eta^2\left[\sum_{s=0}^{k-1}a_{i,s}(k)\right]\sum_{s=0}^{k-1}a_{i,s}(k)\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}_i^{(t,s)})\right\|^2\right] &(94)\\ &\leq 2\eta^2\sigma^2\sum_{s=0}^{k-1}[a_{i,s}(k)]^2\\ &\quad+2\eta^2\Lambda\left[\sum_{s=0}^{k-1}a_{i,s}(k)\right]\sum_{s=0}^{\tau_i-1}\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}_i^{(t,s)})\right\|^2\right] &(95)\end{aligned}$$

where (94) follows from Jensen's Inequality, and (95) uses the fact $a_{i,s}(k)\leq\Lambda$. Note that

$$\begin{aligned}\sum_{k=0}^{\tau_{i-1}}\left[\sum_{s=0}^{k-1}[a_{i,s}(k)]^2\right] &= \sum_{k=0}^{\tau_{i-1}}\|\boldsymbol{a}_i(k)\|_2^2 &(96)\\ &= \sum_{k=1}^{\tau_{i-1}}\|\boldsymbol{a}_i(k)\|_2^2 \leq (\tau_{i-1})\|\boldsymbol{a}_i\|_2^2 &(97)\\ \sum_{k=0}^{\tau_i-1}\left[\sum_{s=0}^{k-1}[a_{i,s}(k)]\right] &= \sum_{k=0}^{\tau_i-1}\|\boldsymbol{a}_i(k)\|_1 &(98)\\ &= \sum_{k=1}^{\tau_i-1}\|\boldsymbol{a}_i(k)\|_1 \leq (\tau_i-1)\|\boldsymbol{a}_i\|_1 &(99)\end{aligned}$$

where the above inequalities uses the fact $\|\boldsymbol{a}_i(k)\|\leq\|\boldsymbol{a}_i(\tau_i)\|=\|\boldsymbol{a}_i\|$. As a result, we have

$$\begin{aligned}&\frac{1}{\|\boldsymbol{a}_i\|_1}\sum_{k=0}^{\tau_i-1}\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)}-\boldsymbol{x}_i^{(t,k)}\right\|^2\right]\\ &\leq 2\eta^2\sigma^2\frac{(\tau_i-1)\|\boldsymbol{a}_i\|_2^2}{\|\boldsymbol{a}_i\|_1}\\ &\quad+2\eta^2\Lambda(\tau_i-1)\|\boldsymbol{a}_i\|_1\frac{1}{\|\boldsymbol{a}_i\|_1}\sum_{k=0}^{\tau_i-1}\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}_i^{(t,k)})\right\|^2\right] &(100)\end{aligned}$$

In addition, we can bound the second term using the following inequality:

$$\begin{aligned}&\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}_i^{(t,k)})\right\|^2\right]\\ &\leq 2\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}_i^{(t,k)})-\nabla F_i(\boldsymbol{x}^{(t,0)})\right\|^2\right]+2\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)})\right\|^2\right]\\ &\leq 2L^2\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)}-\boldsymbol{x}_i^{(t,k)}\right\|^2\right]+2\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)})\right\|^2\right]. &(101)\end{aligned}$$

Substituting (101) into (95), we get

$$\begin{aligned}&\frac{1}{\|\boldsymbol{a}_i\|_1}\sum_{k=0}^{\tau_i-1}\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)}-\boldsymbol{x}_i^{(t,k)}\right\|^2\right]\\ &\leq 2\eta^2\sigma^2\frac{(\tau_i-1)\|\boldsymbol{a}_i\|_2^2}{\|\boldsymbol{a}_i\|_1}\\ &\quad+4\eta^2L^2\Lambda(\tau_i-1)\|\boldsymbol{a}_i\|_1\frac{1}{\|\boldsymbol{a}_i\|_1}\sum_{k=0}^{\tau_i-1}\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)}-\boldsymbol{x}_i^{(t,k)}\right\|^2\right]\\ &\quad+4\eta^2\Lambda\tau_i(\tau_i-1)\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}_i^{(t,0)})\right\|^2\right] &(102)\end{aligned}$$

After minor rearranging, it follows that

$$\begin{aligned}&\frac{1}{\|\boldsymbol{a}_i\|_1}\sum_{k=0}^{\tau_i-1}\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)}-\boldsymbol{x}_i^{(t,k)}\right\|^2\right]\\ &\leq\frac{2\eta^2\sigma^2}{1-4\eta^2L^2\Lambda(\tau_i-1)\|\boldsymbol{a}_i\|_1}\frac{(\tau_i-1)\|\boldsymbol{a}_i\|_2^2}{\|\boldsymbol{a}_i\|_1}\\ &\quad+\frac{4\eta^2\Lambda\tau_i(\tau_i-1)}{1-4\eta^2L^2\Lambda(\tau_i-1)\|\boldsymbol{a}_i\|_1}\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)})\right\|^2\right]. &(103)\end{aligned}$$

Note that $\|\boldsymbol{a}_i\|_1\leq\Lambda\tau_i$, we have

$$\begin{aligned}&\frac{L^2\Lambda}{\|\boldsymbol{a}_i\|_1}\sum_{k=0}^{\tau_i-1}\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)}-\boldsymbol{x}_i^{(t,k)}\right\|^2\right]\\ &\leq\frac{2\eta^2L^2\Lambda\sigma^2}{1-4\eta^2L^2\Lambda^2\tau_i(\tau_i-1)}\frac{(\tau_i-1)\|\boldsymbol{a}_i\|_2^2}{\|\boldsymbol{a}_i\|_1}\\ &\quad+\frac{4\eta^2L^2\Lambda^2\tau_i(\tau_i-1)}{1-4\eta^2L^2\Lambda^2\tau_i(\tau_i-1)}\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)})\right\|^2\right] &(104)\end{aligned}$$

Define $D = 4\eta^2 L^2 \Lambda^2 \max_i \tau_i(\tau_i - 1) < 1$. We can simplify (104) as follows

$$\frac{L^2\Lambda}{a_i}\sum_{k=0}^{\tau_i-1}\mathbb{E}\left[\left\|\boldsymbol{x}^{(t,0)} - \boldsymbol{x}_i^{(t,k)}\right\|^2\right]$$
$$\leq \frac{2\eta^2 L^2 \Lambda \sigma^2}{1-D}\frac{(\tau_i-1)\left\|\boldsymbol{a}_i\right\|_2^2}{\left\|\boldsymbol{a}_i\right\|_1} + \frac{D}{1-D}\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)})\right\|^2\right]. \quad (105)$$

Then, taking the average across all workers and applying Assumption 3, one can obtain

$$\frac{1}{2}\sum_{i=1}^{m} w_i \mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)}) - \boldsymbol{h}_i^{(t)}\right\|^2\right]$$
$$\leq \frac{\Lambda\eta^2 L^2\sigma^2}{1-D}\sum_{i=1}^{m} w_i \frac{(\tau_i-1)\left\|\boldsymbol{a}_i\right\|_2^2}{\left\|\boldsymbol{a}_i\right\|_1} + \frac{D}{2(1-D)}\sum_{i=1}^{m} w_i \mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)})\right\|^2\right] \quad (106)$$
$$\leq \frac{\Lambda\eta^2 L^2\sigma^2}{1-D}\sum_{i=1}^{m} w_i \frac{(\tau_i-1)\left\|\boldsymbol{a}_i\right\|_2^2}{\left\|\boldsymbol{a}_i\right\|_1} + \frac{D\beta^2}{2(1-D)}\mathbb{E}\left[\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2\right] + \frac{D\kappa^2}{2(1-D)}. \quad (107)$$

If $D \leq \frac{1}{2\beta^2+1}$, then it follows that $\frac{1}{1-D} \leq 1 + \frac{1}{2\beta^2} \leq \frac{3}{2}$ and $\frac{D\beta^2}{1-D} \leq \frac{1}{2}$. These facts can help us further simplify (107). We have

$$\sum_{i=1}^{m} w_i \mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{x}^{(t,0)}) - \boldsymbol{h}_i^{(t)}\right\|^2\right]$$
$$\leq \frac{1}{2}\mathbb{E}\left[\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2\right] + \frac{3}{2}\eta^2 L^2\sigma^2 B + 3\eta^2 L^2\kappa^2 C \quad (108)$$
$$\leq \frac{1}{2}\mathbb{E}\left[\left\|\nabla \widetilde{F}(\boldsymbol{x}^{(t,0)})\right\|^2\right] + 3\eta^2 L^2\sigma^2 B + 6\eta^2 L^2\kappa^2 C \quad (109)$$

where $B = \Lambda\sum_{i=1}^{m} w_i(\tau_i - 1)\|\boldsymbol{a}_i\|_2^2/\|\boldsymbol{a}_i\|_1, C = \Lambda^2 \max_i \tau_i (\tau_i - 1)$.

APPENDIX B
EXPERIMENTAL SETTINGS

Platform: All experiments in this paper are conducted on a cluster of 16 machines, each of which is equipped with one NVIDIA TitanX GPU. The machines communicate (*i.e.*, transfer model parameters) with each other via Ethernet. We treat each machine as one client in the federated learning setting. The algorithms are implemented by `PyTorch`. We run each experiments for 3 times with different random seeds.

Hyper-parameter Choices: On non-IID CIFAR10 dataset, we fix the mini-batch size per client as 32. When clients use momentum SGD as the local solver, the momentum factor is 0.9; when clients use proximal SGD, the proximal parameter μ is selected from $\{0.0005, 0.001, 0.005, 0.01\}$. It turns out that when $E_i = 2$, $\mu = 0.005$ is the best and when $E_i(t) \sim \mathcal{U}(2, 5)$, $\mu = 0.001$ is the best. The client learning rate η is tuned from $\{0.005, 0.01, 0.02, 0.05, 0.08\}$ for FEDAVG with each local solver separately. When using the same local solver, FEDNOVA uses the same client learning rate as FEDAVG. Specifically, if the local solver is momentum SGD, then we set $\eta = 0.02$. In other cases, $\eta = 0.05$ consistently performs the best. On the synthetic dataset, the mini-batch size per client is 20 and the client learning rate is 0.02.

ACKNOWLEDGMENT

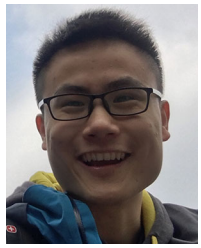
The authors thank Anit Kumar Sahu, Tian Li, Zachary Charles, Zachary Garrett, and Virginia Smith for helpful discussions.

REFERENCES

[1] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 7611–7623, 2020.
[2] H. B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
[3] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2019, doi: 10.1561/2200000083.
[4] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, Jul.–Sep. 2020.
[5] M. Li *et al.*, "Scaling distributed machine learning with the parameter server," in *Proc. 11th USENIX Symp. Operating Syst. Design Implementation*, vol. 14, 2014, pp. 583–598.
[6] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, May 2018.
[7] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms," 2018, *arXiv:1808.07576*.
[8] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. Int. Conf. Learn. Representations*, 2019.
[9] F. Zhou and G. Cong, "On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3219–3227.
[10] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5693–5700.
[11] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HJxNAnVtDS
[12] F. Haddadpour, M. M. Kamani, M. Mahdavi, and V. Cadambe, "Local SGD with periodic averaging: Tighter analysis and adaptive synchronization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 11080–11092.
[13] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local SGD on identical and heterogeneous data," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 4519–4529.
[14] J. Wang and G. Joshi, "Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD," in *Proc. Conf. Mach. Learn. Syst.*, 2019, pp. 212–229.
[15] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for on-device federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.
[16] X. Liang, S. Shen, J. Liu, Z. Pan, E. Chen, and Y. Cheng, "Variance reduced local SGD with lower communication complexity," 2019, *arXiv:1912.12844*.
[17] J. Wang, V. Tantia, N. Ballas, and M. Rabbat, "SlowMo: Improving communication-efficient distributed SGD with slow momentum," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkxJ8REYPH
[18] B. E. Woodworth, J. Wang, A. Smith, B. McMahan, and N. Srebro, "Graph oracle models, lower bounds, and gaps for parallel stochastic optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8496–8506.
[19] A. Dieuleveut and K. K. Patel, "Communication trade-offs for local-SGD with large step size," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13579–13590.
[20] S. Dutta, G. Joshi, S. Ghosh, P. Dube, and P. Nagpurkar, "Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD," in *Proc. 21st Int. Conf. Artif. Intell. Statist.*, 2018, pp. 803–812.

[21] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Conf. Mach. Learn. Syst.*, 2020, pp. 429–450.
[22] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," 2019, *arXiv:1903.03934*.
[23] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," 2019, *arXiv:1909.06335*.
[24] S. Reddi *et al.*, "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Representations*, 2021.
[25] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2021–2031.
[26] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14668–14679.
[27] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "Atomo: Communication-efficient learning via atomic sparsification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9850–9861.
[28] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-IID data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.
[29] Z. Li, D. Kovalev, X. Qian, and P. Richtárik, "Acceleration for compressed gradient descent in distributed and federated optimization," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 5895–5904.
[30] F. Wu, S. He, Y. Yang, H. Wang, Z. Qu, and S. Guo, "On the convergence of quantized parallel restarted SGD for serverless learning," 2020, *arXiv:2004.09125*.
[31] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 500–514, Dec. 2021.
[32] T. Li, M. Sanjabi, and V. Smith, "Fair resource allocation in federated learning," in *Proc. Int. Conf. Learn. Representations*, 2020.
[33] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 4615–4625.
[34] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," *Optim. Mach. Learn.*, vol. 2010, no. 1-38, pp. 85–119, 2011.
[35] D. Bajović, J. M. Moura, J. Xavier, and B. Sinopoli, "Distributed inference over directed networks: Performance limits and optimal design," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3308–3323, Jul. 2016.
[36] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM J. Optim.*, vol. 20, no. 3, pp. 1157–1170, 2010.
[37] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, 2018.
[38] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," 2019, *arXiv:1910.14425*.
[39] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, "MATCHA: Speeding up decentralized SGD via matching decomposition sampling," 2019, *arXiv:1905.09435*.
[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[41] A. Krizhevsky, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.
[42] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. Int. Conf. Learn. Representations*, 2020.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[44] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 2121–2159, 2011.

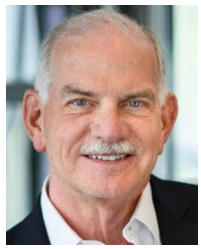
Jianyu Wang received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 2017. He is currently working toward the Ph.D. degree with Carnegie Mellon University, Pittsburgh, PA, USA, advised by Professor Gauri Joshi. In 2020 and 2021, he was a Research Intern with Google Research and in 2019, with Facebook AI Research. His research interests include federated learning, distributed optimization, and systems for large-scale machine learning. His research has been supported by Qualcomm Ph.D. fellowship (2019).



Qinghua Liu received the B.E. degree in electrical engineering and the B.S. degree in mathematics from Tsinghua University, Beijing, China, in 2018. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ, USA. His current research focuses on reinforcement learning.

Hao Liang received the bachelor's degree in opto-electrical information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2018 and the master's degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2020. He is currently working toward the Ph.D. degree with Rice University, Houston, TX, USA, majoring in electrical and computer engineering. His research interests include speech and image processing, and distributed optimization.

Gauri Joshi (Member, IEEE) received the B.Tech and M.Tech degrees in electrical engineering from the Indian Institute of Technology (IIT) Bombay, Mumbai, India, in 2010 and the Ph.D. degree in EECS from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2016. She is currently an Assistant Professor with ECE Department, Carnegie Mellon University, Pittsburgh, PA, USA. From 2016 to 2017, she was a Research Staff Member with IBM T. J. Watson Research Center. Her current research interests include federated learning, distributed optimization, and coding theory. Her awards and honors include the NSF CAREER Award in 2021, the ACM Sigmetrics Best Paper Award in 2020, and the Institute Gold Medal of IIT Bombay in 2010.



H. Vincent Poor (Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University, Princeton, NJ, USA, in 1977. From 1977 to 1990, he was on the Faculty with the University of Illinois at Urbana-Champaign, Champaign, IL, USA. Since 1990, he has been on the Faculty with Princeton, where he is currently the Michael Henry Strater University Professor. During 2006–2016, he was the Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at various other universities, including most recently with Berkeley and Cambridge. His research interests include information theory, machine learning and network science, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the forthcoming book *Machine Learning and Wireless Communications* (Cambridge University Press). He is a Member of the National Academy of Engineering and the National Academy of Sciences and is a Foreign Member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He was the recipient of the IEEE Alexander Graham Bell Medal in 2017.