ACCELERATING ADAPTIVE FEDERATED OPTIMIZATION WITH LOCAL GOSSIP COMMUNICATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, adaptive federated optimization methods, such as FedAdam and FedAMSGrad, have gained increasing attention for their fast convergence and stable performance, especially in training models with heavy-tail stochastic gradient distributions. However, these adaptive federated methods suffer from the *dilemma of local steps*, i.e., the convergence rate gets worse as the number of local steps increases in partial participation settings, making it challenging to further improve the efficiency of adaptive federated optimization. In this paper, we propose a novel method to accelerate adaptive federated optimization with local gossip communications when data is heterogeneous. Particularly, we aim to lower the impact of data dissimilarity by gathering clients into disjoint clusters inside which they are connected with local client-to-client links and are able to conduct local gossip communications. We show that our proposed algorithm achieves a faster convergence rate as the local steps increase thus solving the *dilemma of local steps*. Specifically, our solution improves the convergence rate from $\mathcal{O}(\sqrt{\tau}/\sqrt{TM})$ in FedAMSGrad to $\mathcal{O}(1/\sqrt{T\tau M})$ in partial participation scenarios for nonconvex stochastic setting. Extensive experiments and ablation studies demonstrate the effectiveness and broad applicability of our proposed method.

1 INTRODUCTION

Federated Learning (Konečnỳ et al., 2016; McMahan et al., 2017) has become a crucial large-scale machine learning paradigm where multiple clients jointly train a machine learning model coordinated by a central server. Unlike traditional centralized training, where data is stored in a single central server, in federated learning, training data are stored on each client and only the local trained models are iteratively exchanged and synchronized to the central server. FedAvg (McMahan et al., 2017) (also known as Local SGD (Stich, 2018)) has become one of the most popular federated optimization methods, where each client locally performs multiple steps of SGD updates then aggregates together for the global model update. Aside from the advantage of data privacy protection, the design of multiple local update steps also intends to reduce the communication between the server and clients. Compared with distributed learning (McMahan et al., 2017; Stich, 2018) where each local update step is followed by server aggregation, federated learning can further reduce the communication rounds. Recently, as the booming interests in training large-scale models such as BERT (Devlin et al., 2018), GPT-3 (Brown et al., 2020) and ViT(Dosovitskiy et al., 2021), adaptive federated optimization methods such as FedAdam (Reddi et al., 2020), FedAGM (Tong et al., 2020) and FedAMS (Wang et al., 2022b) has also been proposed and attracted a lot of attention. Specifically, adaptive federated optimization retains the multiple steps of SGD update on local clients but changes the global update of FedAvg from one-step SGD to one-step adaptive gradient methods update. By introducing adaptivity into federated learning, it achieves fast convergence, especially for heavy-tail stochastic gradient noise distributions.

While various adaptive federated optimization algorithms have been proposed, there still exist several key bottlenecks in applying adaptive federated optimization in practice, such as (1) *large client-to-server communication overhead* due to the limited bandwidth and repetitive transmission between the server and clients; (2) *intense sensitivity on data heterogeneity* since nonidentical data distribution on different clients introduce extra variance between clients and slow down the training process of federated learning. What's even worse, these two objectives may conflict with each other: while

increasing the number of local training steps and using partial participation strategies can certainly save the communication costs between the server and clients, it has been shown that the variance overhead term grows as the number of local steps increases in partial participation settings, which leads to worse convergence rate in adaptive federated optimization (Reddi et al., 2020; Wang et al., 2022b). Such worse convergence result is largely due to data heterogeneity, as in the i.i.d setting, the increasing of local steps can indeed lead to a better convergence rate. In this work, we refer this problem as the *dilemma of local steps*. Similar issues have also been shown in FedAvg that a larger number of local SGD steps may cause over-fitting on local clients, also known as client-drift, which slows down the convergence or leads to an unstable result (Karimireddy et al., 2020b). This motivates us to study the following question:

*Can we resolve the **dilemma of local steps** for adaptive federated optimizations? i.e., achieving a faster convergence rate as the number of local steps increases under the non-i.i.d. setting?*

Note that previous studies have shown that traditional variance reduction techniques (Johnson & Zhang, 2013; Fang et al., 2018) can help reduce the client-drift and improve the convergence rate in FedAvg by additionally computing and communicating a control variate or a full-batch gradient (Karimireddy et al., 2020a;b). However, it still remains an open problem how to apply such variance reduction techniques to adaptive federated optimization as it requires precise characterization of each local SGD iteration, which is incompatible with adaptive federated optimization, whose current analysis can only give the characterization of cumulative gradient estimators between two communication rounds. Therefore, we take a different route here to solve the *dilemma of local steps* in adaptive federated optimization: since the core idea of variance reduction is to lower the impact of data dissimilarity between clients, we could obtain a similar effect by enabling the local client-to-client communications similar to gossip averaging in decentralized training (Boyd et al., 2006; Lian et al., 2017; Li et al., 2019b) for reducing the dissimilarity variance between clients. Specifically, in this paper, we propose a novel hybrid adaptive federated optimization method, HA-Fed, which benefits from both adaptive federated optimization (Reddi et al., 2020; Tong et al., 2020; Wang et al., 2022b) and techniques in decentralized training (Lian et al., 2017; Koloskova et al., 2020; Li et al., 2019b). HA-Fed is structured by partitioning a global network into disjoint network clusters, where clients in the same cluster are connected via locally gossip communication links. These locally communications are fast and frequent, which incurs neglectable extra communication overhead compared with client-to-server communication links.

Our contributions can be summarized as follows:

1. We propose a new hybrid adaptive federated optimization method, HA-Fed, which benefits from the frequently local gossip communications to resolve the *dilemma of local steps* in adaptive federated optimization methods. i.e., achieves faster convergence rate as the local steps increases.
2. We show the theoretical convergence improvements for our proposed HA-Fed in the stochastic nonconvex optimization settings. Specifically, we prove that HA-Fed achieves a faster convergence rate than FedAMSGrad [1] on the non-dominant term in full participation scenarios. Moreover, we show that in the more practical partial participation setting, HA-Fed improves the convergence rate (dominant term) from $\mathcal{O}(\sqrt{\tau}/\sqrt{TM})$ to $\mathcal{O}(1/\sqrt{T\tau M})$ w.r.t. global communication rounds T, local update steps τ and the number of participation clients M.
3. Extensive experiments are conducted on several benchmarks dataset and show that our proposed HA-Fed effectively saves the client-to-server communication overhead while achieving faster convergence with heterogeneous data. Extensive ablation studies also show the broad applicability of our proposed method.

Notation: We consider column vectors throughout this paper except special explanations. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, denote $\sqrt{\mathbf{x}}, \mathbf{x}^2, \mathbf{x}/\mathbf{y}$ as the element-wise square root, square, and division of the vectors. For vector $\mathbf{x}$ and matrix A, $\|\cdot\|$ abbreviates the ℓ_2 norm of the vector and Frobenius norm of the matrix, i.e., $\|\mathbf{x}\| = \|\mathbf{x}\|_2$ and $\|A\| = \|A\|_F$, and $\|A\|_2$ denotes the spectral norm of matrix A. We

[1] The convergence rate of FedAMSGrad is obtained from the convergence analysis for FedAMS (Wang et al., 2022b), where FedAMSGrad gets a similar convergence to FedAMS. FedAMSGrad is also included in (Tong et al., 2020).

denote $\mathbf{1}$ as vector with all elements equal to 1 with appropriate dimension, and $\mathbf{I}$ as the identity matrix with appropriate dimension.

2 Related Work

Federated learning: Federated learning (Konečnỳ et al., 2016) has attracted growing interest recently due to the demand for training models locally at edge devices and the requirements of privacy protection. Federated optimization methods such as SGD-based optimization algorithm, FedAvg (McMahan et al., 2017), also known as Local SGD (Stich, 2018), have been widely used in federated learning. Aside from FedAvg, since adaptive gradient methods such as Adam (Kingma & Ba, 2014) and its variant AMSGrad (Reddi et al., 2018) overcame the sensitivity to parameters and slow to convergence issue of SGD, adaptive federated optimizations such as FedAdam(Reddi et al., 2020), FedAGM (Tong et al., 2020) and FedAMS (Wang et al., 2022b) studied the corresponding adaptive optimization algorithms in federated learning. Moreover, several works (Hsu et al., 2019; Ghosh et al., 2019; Karimireddy et al., 2020b; Li et al., 2019a; Yang et al., 2021) addressed and focused on the data heterogeneity issues of federated learning, where Karimireddy et al. (2020b) proposed a federated learning variance reduction method that overcomes the data heterogeneity, but it requires extra communication costs for variance reduction operations. Guo et al. (2021) considered heterogeneous communications for modern communication networks that improve communication efficiency. Hierarchical federated learning algorithms (Liu et al., 2020; Abad et al., 2020; Castiglia et al., 2020) are developed by aggregating client models to edge servers first before synchronizing them to the central server.

Decentralized learning and other frameworks: Decentralized learning is a large-scale machine learning paradigm without a central server. It has been firstly studied from gossip averaging techniques (Tsitsiklis, 1984; Boyd et al., 2006). Decentralized (gossip) SGD algorithms (Lian et al., 2017; Li et al., 2019b; Boyd et al., 2006; Tang et al., 2018) are then proposed that consider client-to-client communications after each step of SGD update on the client. Lu & De Sa (2021) proved a tight lower bound for decentralized training under the nonconvex setting. Teng et al. (2019) proposes a leader-distributed SGD algorithm that pulls workers to the currently best-performing model among all models, which also utilizes inexpensive gossip communication. Moreover, recent studies generalized various distributed SGD algorithms under unified frameworks (Wang & Joshi, 2021; Koloskova et al., 2020), where Wang & Joshi (2021) included reducing communication costs and decentralized training in i.i.d. settings, and Koloskova et al. (2020) studied a general network topology-changing gossip SGD methods that summarize several algorithms in distributed and federated learning.

Communication-efficient federated learning: In terms of reducing the communication overhead in federated learning, one of the common approaches is to save the communication bits when synchronizing, such as the compressed and quantized FedAvg-based methods (Reisizadeh et al., 2020; Jin et al., 2020; Jhunjhunwala et al., 2021; Chen et al., 2021a). Note that the bit compression strategy is orthogonal to our hybrid adaptive federated learning framework and can potentially be combined to further reduce communication overheads.

3 Preliminaries on Adaptive Federated Optimization

Firstly, let's begin with the general federated learning problem under nonconvex stochastic optimization settings. Suppose we have N local clients, and our goal is to minimize the following objective:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}), \tag{3.1}$$

where $\mathbf{x}$ denotes the model parameters, $f_i(\mathbf{x}) = \mathbb{E}_{\xi\sim\mathcal{D}_i} f_i(\mathbf{x},\xi_i)$ is the local nonconvex loss function corresponding to client i, and $\mathcal{D}_i$ is the local data distribution associated with client i. FedAvg (McMahan et al., 2017) is a popular optimization algorithm to solve Eq. 3.1, with the sequential implementation of local SGD updates and global averaging.

Adaptive federated optimization is then proposed to incorporate adaptivity in federated optimization methods by replacing the global averaging in FedAvg with one-step adaptive gradient optimization. For example, FedAMSGrad is designed with multi-steps of local SGD updates and followed by

one step of global AMSGrad (Reddi et al., 2018) update. Specifically, at global round t, the server broadcasts the model $\mathbf{x}_t$ to selected clients. Each client i conducts τ steps of local SGD updates with local learning rate η_l and obtains the local model $\mathbf{x}_{t,\tau}^i$. The model difference $\Delta_t^i = \mathbf{x}_{t,\tau}^i - \mathbf{x}_t$ for each client is aggregated to the server and averaged to Δ_t. The server updates the global model $\mathbf{x}_{t+1}$ by taking Δ_t as a pseudo gradient for calculating momentum $\mathbf{m}_t$ and variance $\mathbf{v}_t$ for AMSGrad optimizer, and performs one step AMSGrad update with global learning rate η, i.e.,

$$\begin{aligned} &\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\Delta_t, \mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\Delta_t^2, \\ &\widehat{\mathbf{v}}_t = \max\{\widehat{\mathbf{v}}_{t-1}, \mathbf{v}_t\}, \mathbf{x}_{t+1} = \mathbf{x}_t + \eta \frac{\mathbf{m}_t}{\sqrt{\widehat{\mathbf{v}}_t + \epsilon}}, \end{aligned} \tag{3.2}$$

the server obtains model $\mathbf{x}_{t+1}$ after one global round. Besides FedAdam and FedAMSGrad, there are several adaptive federated optimization methods with slightly changes in update formulas, e.g., FedAdagrad and FedYogi (Reddi et al., 2020), FedAGM (Tong et al., 2020) and FedAMS (Wang et al., 2022b).

The convergence of FedAMSGrad is affected by several factors such as the number of local steps τ, global rounds T, and the number of participating clients M. In full participation settings, where M is equal to the total number of clients N, FedAMSGrad enjoys a convergence rate of $\mathcal{O}(1/\sqrt{T\tau N})$. This suggests that even for heterogeneous data, a larger number of local steps τ can help save the client-to-server communication rounds and lead to faster convergence. However, previous study shows that under more practical partial participation settings, FedAMSGrad only achieves a convergence rate of $\mathcal{O}(\sqrt{\tau}/\sqrt{TM})$ with heterogeneous data. This suggests that while larger τ can reduce communication frequency, it scarifies the convergence rate and requires more communication rounds to converge. We refer to this problem as the *dilemma of local steps*.

The *dilemma of local steps* arises in partial participation settings since the heterogeneous data induces a large variance term in the final convergence result, which is proportional to the number of local steps τ and thus leads to a worse convergence rate. For full participation settings, it is fortunate that this variance overhead only appears on the non-dominant term, thus it does not slow down the overall convergence. While for partial participation settings, the larger τ amplifies the over-fitting issue on local clients as fewer clients participate in each round of global training and becomes a dominant term in the convergence result. Although variance reduction techniques (Johnson & Zhang, 2013; Fang et al., 2018) can help reduce the client-drift (or the *dilemma of local steps*) in the local iterations of FedAvg (Karimireddy et al., 2020b;a), the success of applying variance reduction techniques to FedAvg rely on the precise characteristic of each local SGD iteration. However, as shown in Eq. 3.2, the global adaptive optimizer updates via the cumulative model difference Δ_t between two communication rounds, which makes how to apply iterative variance reduction bounds to adaptive federated optimization an open problem. In the following, we will present our attempt to resolve the *dilemma of local steps* by a new hybrid adaptive federated optimization method.

4 Proposed Method

In this paper, we propose a hybrid adaptive federated optimization method (HA-Fed) where the clients are partitioned into disjoint clusters inside which they can communicate by fast client-to-client links, and clusters communicate with the central server with client-to-server communication links. Specifically, assuming we have one central server and K disjoint clusters, each of which contains n local clients and there are connected by client-to-client links (denoted by the adjacency matrix W_k). Let's denote the total number of clients as $N = Kn$. Our goal is to solve the following optimization problem:

$$\min_{x\in\mathbb{R}^d} f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} \bar{f}_k(\mathbf{x}), \tag{4.1}$$

where $f_i(\mathbf{x}) = \mathbb{E}_{\xi\sim\mathcal{D}_i} f_i(\mathbf{x}, \xi_i)$ is the nonconvex loss function for the i-th client, and $\bar{f}_k(\mathbf{x}) := \frac{1}{n}\sum_{i\in\mathcal{V}_k} f_i(\mathbf{x})$ is the average loss on cluster k. We consider $\mathcal{V}_k$ as the set of local clients in the cluster k, and clients in cluster k are linked by a connected graph $\mathcal{G}_k$[2].

[2]The connected graph implies there is a path from any client to any other client in the graph.

In order to accelerate FedAMSGrad under heterogeneous data settings, our HA-Fed starts from FedAMSGrad and introduces intra-cluster gossip communications. Gossip communication is designed for clients in a network to communicate with their neighbors without a central server, and it has been a popular approach in decentralized learning (Lian et al., 2017; Koloskova et al., 2020; Chen et al., 2021b). Our proposed HA-Fed adds frequent client-to-client gossip communication inside each cluster to leverage the over-fitting issue within the cluster. These gossip communications rely on inexpensive local client-to-client communications without incurring extra client-to-server communication rounds, but at the same time, prevent over-fitting on local clients since the model on each client sufficiently communicates with their neighbors.

Algorithm 1 summarizes the proposed HA-Fed in full participation scenarios. The major difference between HA-Fed and FedAMSGrad lies in the local update step within each cluster (Line 9 in Algorithm 1): at the s-th step of intra-cluster training for cluster k, after client i finishes their local update and obtains $\mathbf{x}^i_{t,s+\frac{1}{2}}$ by one step SGD, we conduct one gossip averaging step within the cluster, i.e., let each client communicate with its neighbors $\mathcal{N}^i_k$ and aggregate the nearby local models with a weighted matrix W_k. The rest part of the algorithm is similar to FedAMSGrad.

In order to further reduce client-to-server communication rounds, we also adopt partial participation setting for HA-Fed[3]. Generally, in partial participation settings, the server samples a subset of m clients in each cluster before each round starts and only broadcasts the current model to these m selected clients and the selected clients will broadcast the received model to other clients within the same cluster with client-to-client links. For global model updates, all selected clients send the model difference Δ^i_t to the central server, and the server aggregates them to Δ_t. The rest of the partial participation update is the same as the full participation scenarios.

Algorithm 1 HA-Fed:full participation

Input: initial point $\mathbf{x}_1$, global step size η, local step size η_l, $\beta_1, \beta_2, \epsilon$, weighting matrix W_k for all clusters $k \in [K]$

1: $\mathbf{m}_0 \leftarrow 0$, $\mathbf{v}_0 \leftarrow 0$
2: **for** $t = 1$ to T **do**
3: **for** each cluster $k \in [K]$ in parallel **do**
4: **for** each client $i \in \mathcal{V}_k$ in parallel **do**
5: Receive model from the server: $\mathbf{x}^i_{t,0} = \mathbf{x}_t$
6: **for** $s = 0, ..., \tau - 1$ **do**
7: Compute local stochastic gradient: $\mathbf{g}^i_{t,s} = \nabla F_i(\mathbf{x}^i_{t,s}; \xi^i_{t,s})$
8: Local update: $\mathbf{x}^i_{t,s+\frac{1}{2}} = \mathbf{x}^i_{t,s} - \eta_l \mathbf{g}^i_{t,s}$
9: Gossip communication: $\mathbf{x}^i_{t,s+1} = \sum_{j \in \mathcal{N}^i_k} (W_k)_{i,j} \mathbf{x}^j_{t,s+\frac{1}{2}}$
10: **end for**
11: Get the model difference: $\Delta^i_t = \mathbf{x}^i_{t,\tau} - \mathbf{x}_t$
12: **end for**
13: **end for**
14: Server gets model difference: $\Delta_t = \frac{1}{K} \sum_{k \in [K]} \frac{1}{n} \sum_{i \in \mathcal{V}_k} \Delta^i_t$
15: Update: $\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\Delta_t$
16: Update: $\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2)\Delta^2_t$
17: $\widehat{\mathbf{v}}_t = \max(\widehat{\mathbf{v}}_{t-1}, \mathbf{v}_t)$ and $\widehat{\mathbf{V}}_t = \mathrm{diag}(\widehat{\mathbf{v}}_t + \epsilon)$
18: Server updates $\mathbf{x}_{t+1} = \mathbf{x}_t + \eta \frac{\mathbf{m}_t}{\sqrt{\widehat{\mathbf{v}}_t + \epsilon}}$
19: **end for**

In a nutshell, HA-Fed takes advantage of decentralized training to resolve the *dilemma of local steps* in adaptive federated optimization while preserving the benefit of adaptive optimizations: The server aggregation rule and update schemes follow standard adaptive federated optimization, which enjoys nice convergence properties, especially for heavy-tail stochastic gradient noise distributions. Meanwhile, the local gossip communications alleviate the impact of data dissimilarity between clients on the final convergence rate. Of course, this design requires all clients within each cluster to stay active and perform gossip communications. Yet we also want to emphasize that HA-Fed can also be

[3]Due to the space limit, see details in Algorithm 2 in the Appendix.

compatible with scenarios where not all clients are active at each iteration by simply adapting the frequency of local gossip communications. We refer interested readers to Appendix F.3 for more details.

5 CONVERGENCE ANALYSIS

In this section, we provide the theoretical convergence analysis of the proposed HA-Fed method. Before starting with the main theoretical results, let us first state the following assumptions:

Assumption 5.1 (Smoothness)**.** Each loss function on the i-th client $f_i(\mathbf{x})$ is L-smooth, i.e., $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $\big|f_i(\mathbf{x}) - f_i(\mathbf{y}) - \langle \nabla f_i(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle\big| \leq \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2$.

Assumption 5.2 (Bounded Gradient)**.** Each loss function on the i-th client $f_i(\mathbf{x})$ has G-bounded stochastic gradient on ℓ_2, i.e., for all ξ, we have $\|\nabla f_i(\mathbf{x}, \xi)\| \leq G$.

Assumption 5.3 (Bounded Stochastic Variance)**.** Each stochastic gradient on the i-th client has a bounded local variance, i.e., for all $\mathbf{x}, i \in [m]$,we have $\mathbb{E}\big[\|\nabla f_i(\mathbf{x}, \xi) - \nabla f_i(\mathbf{x})\|^2\big] \leq \sigma^2$.

Assumption 5.1 also implies the L-gradient Lipschitz condition, i.e., $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$, it is a standard assumption in nonconvex optimization problems (Kingma & Ba, 2014; Reddi et al., 2018; Li et al., 2019a; Yang et al., 2021). Assumption 5.2 is usually adopted in studying adaptive gradient methods (Kingma & Ba, 2014; Reddi et al., 2018; Zhou et al., 2018; Chen et al., 2020). Assumption 5.3 is frequently stated in studying distributed and federated learning optimization problems (Reddi et al., 2020; Yang et al., 2021; Chen et al., 2021b; Wang et al., 2022a).

Assumption 5.4 (Bounded Inter-Client Variances)**.** The variance between local client's objective function and the objective function on the corresponding cluster is bounded, i.e., for all $\mathbf{x}, k \in [K]$, we have $\frac{1}{n}\sum_{i \in \mathcal{V}_k} \|\nabla f_i(\mathbf{x}) - \nabla \bar{f}_k(\mathbf{x})\|^2 \leq \sigma_k^2$. The objective function on each cluster and the global function has a bounded variance: for $\alpha \geq 1$ and $\sigma_g \geq 0$, there is $\frac{1}{K}\sum_{k \in [K]} \|\nabla \bar{f}_k(\mathbf{x})\|^2 \leq \alpha^2 \|\nabla f(\mathbf{x})\|^2 + \sigma_g^2$.

Assumption 5.4 represents the data heterogeneity in a cluster and between clusters. The similar data heterogeneity assumption, which considers the variance between local clients, is common in federated learning (Reddi et al., 2020; Yang et al., 2021) and decentralized learning (Lian et al., 2017; Li et al., 2019b; Koloskova et al., 2020).

Assumption 5.5 (Gossip Weighting Matrix)**.** The local clients in cluster k are connected in the graph $\mathcal{G}_k$, and the corresponding weighting matrix W_k is a doubly stochastic matrix with the fact: $W_k \in [0, 1]^{n \times n}$, $W_k \mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top W_k = \mathbf{1}^\top$ and $\text{null}(\mathbf{I} - W_k) = \text{span}(\mathbf{1})$. We further assume the spectral gap ρ_k: there exists $\rho_k \in [0, 1)$ such that $\|W_k - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\|_2 \leq \rho_k$.

Assumption 5.5 is usually assumed for decentralized learning framework (Koloskova et al., 2020; Chen et al., 2021b; Guo et al., 2021). Specifically, $\rho_k = 0$ means the matrix W_k with all elements $\frac{1}{n}$, corresponding to a fully connected graph $\mathcal{G}_k$ and $\rho_k \to 1$ means the matrix W_k tends to be elements with either 0 or 1, corresponding to a graph that is nearly disconnected. Several works (Lian et al., 2017; Li et al., 2019b) alternatively assume the spectral gap ρ of a weighting matrix W as the second largest eigenvalue of a doubly stochastic matrix W, i.e., $\rho = |\lambda_2(W)|$, and this spectral gap holds the same role for revealing the connectivity of the graph.

5.1 CONVERGENCE ANALYSIS FOR HA-FED: FULL PARTICIPATION

We first study the convergence behaviour of HA-Fed under full participation scenarios.

Theorem 5.6 (HA-Fed full participation)**.** Under Assumptions 5.1-5.5, if the local learning rate satisfies $\eta_l \leq \min\big\{\frac{\sqrt[4]{\epsilon}}{\alpha\sqrt{CC_0\tau(\tau+\rho_{\max}^2 D_{\tau,\rho})}}, \frac{\epsilon}{2\tau C_0 C_{\beta,\eta}}\big\}$, then the iterates of Algorithm 1 satisfy

$$\min_{t \in [T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq 8(\beta_2 \eta_l^2 \tau^2 G^2 + \epsilon)^{\frac{1}{2}} \left\{ \frac{f_0 - f_*}{\eta \eta_l \tau T} + \frac{\Psi}{T} + \Phi_1 + \Phi_2 \right\}, \tag{5.1}$$

where $\Psi = \frac{C_\beta G^2 d}{\sqrt{\epsilon}} + \frac{2C_\beta^2 \eta \eta_l \tau L G^2 d}{\epsilon}$, $\Phi_1 = \frac{CL^2\eta_l^2}{4\sqrt{\epsilon}}\Big[\tau^2\sigma_g^2 + \tau\rho_{\max}^2 D_{\tau,\rho}\bar{\sigma}_L^2 + \tau\sigma^2\big(\frac{1}{n} + \rho_{\max}^2\big)\Big]$, $\Phi_2 = C_{\beta,\eta}\frac{\eta_l}{2\epsilon N}\sigma^2$, where $C_\beta = \frac{\beta_1}{1-\beta_1}$, $C_{\beta,\eta} = \big((C_\beta^2 + 3)\eta L + 2\sqrt{1-\beta_2}G\big)$, C is a constant

irrelevant to parameters, $\rho_{\max} = \max_{k\in[K]} \rho_k$ is the maximum spectral gap of all K clusters, $D_{\tau,\rho} = \min\big\{\frac{1}{1-\rho_{\max}}, \tau\big\}$ describes the density and connectivity of clusters, and $\bar{\sigma}_L^2 = \frac{1}{K}\sum_{k=1}^K \sigma_k^2$ is the average dissimilarity between local clients in the same cluster.

Remark 5.7. The convergence rate Eq. 5.1 is composed of four terms. The first and second terms are related to T and vanish as T increases. The third term Φ_1 represents the variance overhead introduced by both stochastic and inter-client variances. The last term Φ_2 represents the stochastic variance from all N clients. Note that only Φ_1 is related to the cluster connectivity $\rho_{\max}$ while the other three terms are identical to the corresponding term in the convergence rate of N-clients FedAMSGrad. Specifically, the dependency of Φ_1 for HA-Fed is $\Phi_1 = \mathcal{O}\big(\eta_l^2\tau^2\sigma_g^2 + \eta_l^2\rho_{\max}^2\tau^2\bar{\sigma}_L^2 + \eta_l^2\big(\frac{1}{n} + \rho_{\max}^2\big)\tau\sigma^2\big)$, while the corresponding term $\widetilde{\Phi}_1$ for FedAMSGrad is $\mathcal{O}(\eta_l^2\tau^2\widetilde{\sigma}_g^2 + \eta_l^2\tau\sigma^2)$. When $\rho_{\max} = 0$, Φ_1 in HA-Fed becomes $\mathcal{O}\big(\eta_l^2\tau^2\sigma_g^2 + \eta_l^2\tau\frac{\sigma^2}{n}\big)$, which is better than that of FedAMSGrad. And when $\rho_{\max} \to 1$, Φ_1 in HA-Fed becomes $\mathcal{O}\big(\eta_l^2\tau^2(\sigma_g^2 + \bar{\sigma}_L^2) + \eta_l^2\tau\sigma^2\big)$, which matches the results in FedAMSGrad[4]. In terms of the overall convergence rate, since Φ_1 in HA-Fed has the same order of dependency w.r.t. τ and η_l as in FedAMSGrad, suppose we pick the learning rates $\eta = \Theta(\sqrt{\tau N})$ and $\eta_l = \Theta(1/\sqrt{T\tau^2})$ and when T is sufficient large, i.e., $T > \tau N$, HA-Fed achieves the same convergence rate of $O(1/\sqrt{T\tau N})$ as FedAMSGrad (Wang et al., 2022b) and also same as other general federated nonconvex optimization methods such as FedAvg (Yu et al., 2019; Yang et al., 2021) and FedAdam (Reddi et al., 2020).

5.2 Convergence Analysis for HA-Fed: Partial Participation

In such settings, we assume that only selected clients participate in each round of global synchronization. We assume the sampling strategy is random sampling without replacement in each cluster. Generally, at the beginning of global iteration t, the server samples a subset $\mathcal{S}_t^k$ for cluster k that contains m clients, these $M = Km$ clients receive the model from the server and synchronize their model difference for the global update.

Theorem 5.8 (HA-Fed partial participation)**.** Under Assumptions 5.1-5.5, if the local learning rate satisfies $\eta_l \le \min\Big\{\frac{1}{4C_0C_{\beta,\eta}(\tau-1)}, \frac{\sqrt[4]{\epsilon}}{2\alpha L\sqrt{\widetilde{C}C_0\tau(\tau+\rho_{\max}^2 D_{\tau,\rho})}}, \frac{1}{128\alpha^2\widetilde{C}C_0C_{\beta,\eta}\rho_{\max}^2 D_{\tau,\rho}}\big(\frac{n-m}{m(n-1)} + \frac{1}{\tau^2}\big)^{-1}\Big\}$, then the iterates of Algorithm 1 in partial participation scenarios satisfy

$$\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \le 8(\beta_2\eta_l^2\tau^2G^2 + \epsilon)^{\frac{1}{2}}\left\{\frac{f_0 - f_*}{\eta\eta_l\tau T} + \frac{\Psi}{T} + \Phi_1 + \Phi_2 + \Phi_3 + \Phi_4\right\}, \tag{5.2}$$

where $\Psi = \frac{C_\beta G^2 d}{\sqrt{\epsilon}} + \frac{2C_\beta^2\eta\eta_l\tau LG^2 d}{\epsilon}$, $\Phi_1 = \frac{CL^2}{4\sqrt{\epsilon}}\eta_l^2\Big[\tau^2\sigma_g^2 + \tau\rho_{\max}^2 D_{\tau,\rho}\bar{\sigma}_L^2 + \tau\sigma^2\big(\frac{1}{n} + \rho_{\max}^2\big)\Big]$, $\Phi_2 = C_{\beta,\eta}\big[1 + \big(\frac{n-m}{m}\big)\rho_{\max}^2\big]\frac{\eta_l}{\epsilon N}\sigma^2$, $\Phi_3 = \widetilde{C}C_{\beta,\eta}\cdot\frac{n-m}{m(n-1)}\eta_l D_{\tau,\rho}\rho_{\max}^2\big[\sigma_g^2 + \bar{\sigma}_L^2 + \sigma^2 + D_{\tau,\rho}\frac{\sigma^2}{\tau^2 n}\big]$, $\Phi_4 = \widetilde{C}C_{\beta,\eta}\cdot\eta_l^3L^2D_{\tau,\rho}\rho_{\max}^2\big[\sigma_g^2 + \bar{\sigma}_L^2 + \sigma^2 + D_{\tau,\rho}\frac{\sigma^2}{\tau^2 n}\big]$, where C and $\widetilde{C}$ are constants irrelevant to parameters and $\rho_{\max}, D_{\tau,\rho}, \bar{\sigma}_L^2, C_{\beta,\eta}, C_\beta$ are same defined as Theorem 5.6.

Remark 5.9. When $\rho_{\max} = 0$, i.e., clients in each cluster are fully connected, in such case, there are $\Phi_1 = \mathcal{O}\big(\eta_l^2\tau^2\sigma_g^2 + \eta_l^2\tau\frac{\sigma^2}{n}\big)$, $\Phi_2 = \mathcal{O}\big(\frac{\eta_l\sigma^2}{N}\max\{\eta, 1\}\big)$ and $\Phi_3 = \Phi_4 = 0$ in Eq. 5.2, which matches the result of fully participated HA-Fed with $\rho_{\max} = 0$. It is worth noting that although partially participated HA-Fed aggregates M client models in each global round, since clients are fully connected inside the clusters, picking a part of the clients (inside each cluster) for global aggregation is the same as picking all the clients. Therefore, partially participated HA-Fed recovers to fully participated HA-Fed under such a setting.

Remark 5.10. When $\rho_{\max} \to 1$ and $K = 1$, i.e., all clients are tending to disconnected, HA-Fed will reduce to partial participated FedAMSGrad with M clients. Under such cases, we have $D_{\tau,\rho} = \min\big\{\frac{1}{1-\rho_{\max}}, \tau\big\} = \tau$. By choosing same learning rates $\eta = \Theta(\sqrt{\tau M})$ and $\eta_l = \Theta(1/\sqrt{T\tau^2})$ as in FedAMSGrad, $\Phi_3 = \mathcal{O}\big(\frac{\sqrt{\tau}}{\sqrt{TM}}\big)$ dominates the convergence rate of HA-Fed, which recovers the convergence of partially participated FedAMSGrad.

[4] $\widetilde{\sigma}_g^2$ is the global variance obtaining by a similar assumption on clients' loss function , i.e., the loss function on each client of FedAMSGard satisfies $\frac{1}{N}\sum_{i=1}^N \|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \le \widetilde{\sigma}_g^2$.

Remark 5.9 and 5.10 implies that when clients are sparsely connected, the convergence of partial participated HA-Fed still suffers from *dilemma of local steps* as in FedAMSGrad, while HA-Fed indeed resolves the dilemma when clients are densely connected. Therefore, it is crucial to investigate how cluster connectivity helps solve the *dilemma of local steps*. The following corollary gives a precise characterization on condition of $\rho_{\max}$ needed for solving the *dilemma of local steps*.

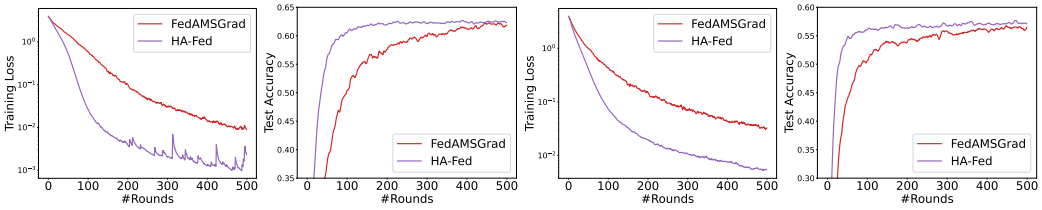
Corollary 5.11. Suppose all clusters satisfies $\rho_{\max} \leq \frac{1}{2\sqrt{n-m}}$ and $K < n$, then by choosing the global learning rate $\eta = \Theta(\sqrt{\tau M})$ and local learning rate $\eta_l = \Theta(\frac{1}{\sqrt{T\tau}})$, when T is sufficient large, i.e., $T > \tau M$, then the convergence rate for HA-Fed in partial participation settings satisfies $\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] = \mathcal{O}\big(\frac{1}{\sqrt{T\tau M}}\big)$.

Remark 5.12. Corollary 5.11 shows that HA-Fed successfully resolves the *dilemma of local steps*: larger number of local steps τ can now achieve a faster convergence rate if clusters satisfy certain constraints. Note that when $m = n$, i.e., in the full participation setting, this $\rho_{\max} \leq \frac{1}{2\sqrt{n-m}}$ condition imposes no actual constraint on $\rho_{\max}$. When m becomes smaller, the requirements for $\rho_{\max}$ also get stronger, i.e., the local cluster needs to be more densely connected. Also, for a given number of total clients N, the condition $K < n$ implies the number of clients in each cluster is larger than the number of clusters in the network, which ensures that each cluster has enough clients for local gossip communications and thus can reduce the variance and resolve the *dilemma of local steps* in the partial participation settings.

6 Experiments

In this section, we present the empirical evaluations for the HA-Fed algorithm. We mainly compare HA-Fed with the adaptive federated optimization counterpart, FedAMSGrad, and also conduct several ablation studies related to the algorithm framework and the intra-cluster topology.

Experimental Setup: We compare our proposed HA-Fed with FedAMSGrad, on CIFAR-10/CIFAR-100 (Krizhevsky et al., 2009) using (1) ResNet-18 (He et al., 2016) model, and (2) ConvMixer[5] model (Trockman & Kolter, 2022), and Fashion MNIST (Xiao et al., 2017) datasets using (1) ConvMixer model and (2) CNN model[6]. For HA-Fed, the global network topology is set up with 32 total clients, and they are equally divided into 4 clusters where each cluster contains 8 clients. We set the default partial participation ratio as $p = 0.25$, i.e., 2 clients participated per cluster per round. We adopt ring topology for all clusters by default with maximum spectral gap $\rho_{\max} = 0.805$. For FedAMSGrad, we set the number of clients and the partial participation ratio the same, i.e., 32 clients in total and 8 clients synchronize to the central server in each round. For both methods, we conduct $\tau = 48$ steps of local training with a batch size of 50. We search for the best training hyper-parameter for both models. Due to the space limit, we leave the CIFAR-10 and Fashion MNIST experiments as well as the other experimental details in Appendix F.



(a) ResNet-18 (b) ConvMixer-256-8

Figure 1: The learning curves for HA-Fed and FedAMSGrad in training CIFAR-100 data on (a) ResNet-18 model and (b) ConvMixer-256-8 model using ring topology for local communications.

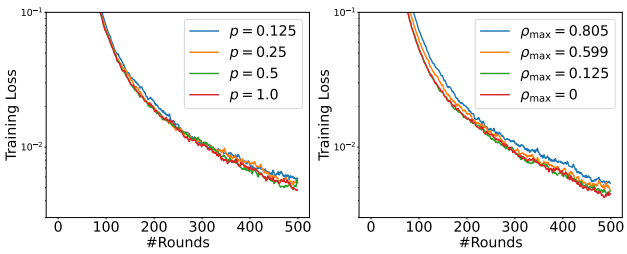
Figure 1 shows the convergence result of HA-Fed and FedAMSGrad on training CIFAR-100 with ResNet-18 and ConvMixer-256-8 model. We compare the training loss and test accuracy against global rounds for both models. For the ResNet-18 model, HA-Fed achieves faster convergence than

[5]ConvMixer shares similar ideas to vision transformer (Dosovitskiy et al., 2021) to use patch embeddings to preserve locality and similarly, and it is trained via adaptive gradient methods by default.

[6]See details for the CNN model in Appendix F.3.

FedAMSGrad in reducing training loss, and HA-Fed grows rapidly to obtain an overall higher test accuracy. For the ConvMixer-256-8 model, HA-Fed again shows its faster convergence speed on training loss; in the meantime, HA-Fed still holds a higher test accuracy compared to FedAMSGrad under the same settings.

Now we study how the participation ratio p and network connectivity $\rho_{\max}$ would affect the convergence of our proposed HA-Fed algorithm. Figure 2(a) illustrates the ablation study on the participation ratio p. Specifically, we test various values of p from $p = \{0.125, 0.25, 0.5, 1.0\}$. From Figure 2(a), we observe that a larger participation ratio p slightly improves the convergence on training loss. This is consistent with our theoretical convergence rate that increasing the number of participating clients improves the convergence rate, but the improvement is slight compared to a large number of global round T and local steps τ. Figure 2(b) then shows ablation study on clusters' maximum spectral gap $\rho_{\max}$. Specifically, we compare various of $\rho_{\max}$ from $\rho_{\max} = \{0, 0.125, 0.599, 0.805\}$ calculated by different network typologies. From Figure 2(b), we can observe that smaller $\rho_{\max}$ contributes to a faster convergence on training loss, which is shown as the red and green lines achieve faster convergence on training loss than the orange and blue lines. This result matches the theoretical result that $\rho_{\max}$ holds the non-dominant term in the convergence of HA-Fed even for partial participation scenarios. This suggests that without a dense network topology, HA-Fed can still take the benefit of gossip communication to achieve the expected convergence result.



(a) Ablation on the ratio p (b) Ablation on $\rho_{\max}$

Figure 2: The learning curves with (a) different participating ratio p and (b) different maximum spectral gap $\rho_{\max}$ of clusters in training CIFAR-100 data on ConvMixer-256-8 model.

We further study how the number of local update steps τ would affect the convergence of our proposed HA-Fed algorithm. Figure 3 shows the ablation study about the number of local steps τ, we compare different τ from $\tau = \{24, 48, 96\}$. We observe that a larger number of local steps τ indeed helps accelerate convergence on training loss, as the green line ($\tau = 96$) in the left plot keeps the smallest training loss. From the right plot in Figure 3, larger τ generally achieves better generalization performance with higher test accuracy. This result backup our theory and show that HA-Fed achieves a faster convergence as the number of local steps increases, and HA-Fed indeed resolves the *dilemma of local steps*.

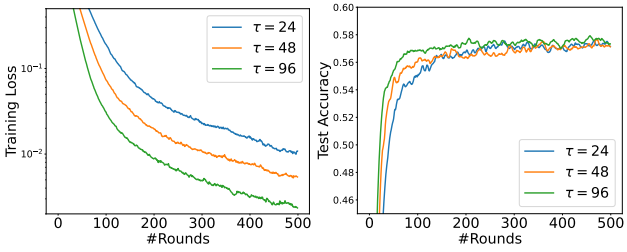


Figure 3: The learning curves with different numbers of local steps τ in training CIFAR-100 on ConvMixer-256-8 model.

7 Conclusions

In this paper, we propose a novel hybrid adaptive federated optimization algorithm, HA-Fed, that overcomes the *dilemma of local steps* and achieves a faster convergence rate as the local training step increases. HA-Fed mitigates the impact of data heterogeneity by adding inexpensive client-to-client communications hence resolving the *dilemma of local steps* without extra client-to-server communications. We present a completed theoretical convergence analysis for the proposed HA-Fed. We prove that HA-Fed achieves a faster convergence rate than the previous adaptive federated optimization method for both full and partial participation scenarios with heterogeneous data under nonconvex stochastic settings. Experiments on several benchmarks and ablation studies verify our theory.

REFERENCES

Mehdi Salehi Heydar Abad, Emre Ozfatura, Deniz Gunduz, and Ozgur Ercetin. Hierarchical federated learning across heterogeneous cellular networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8866–8870. IEEE, 2020.

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Timothy Castiglia, Anirban Das, and Stacy Patterson. Multi-level local sgd: Distributed sgd for heterogeneous hierarchical networks. In *International Conference on Learning Representations*, 2020.

Jinghui Chen, Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

Mingzhe Chen, Nir Shlezinger, H Vincent Poor, Yonina C Eldar, and Shuguang Cui. Communication-efficient federated learning. *Proceedings of the National Academy of Sciences*, 118(17), 2021a.

Yiming Chen, Kun Yuan, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. Accelerating gossip sgd with periodic global averaging. In *International Conference on Machine Learning*, pp. 1791–1802. PMLR, 2021b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

Yuanxiong Guo, Ying Sun, Rui Hu, and Yanmin Gong. Hybrid local sgd for federated learning with heterogeneous communications. In *International Conference on Learning Representations*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Divyansh Jhunjhunwala, Advait Gadhikar, Gauri Joshi, and Yonina C Eldar. Adaptive quantization of model updates for communication-efficient federated learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3110–3114. IEEE, 2021.

Richeng Jin, Yufan Huang, Xiaofan He, Huaiyu Dai, and Tianfu Wu. Stochastic-sign sgd for federated learning with theoretical guarantees. *arXiv preprint arXiv:2002.10940*, 2020.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020a.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pp. 5381–5393. PMLR, 2020.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019a.

Xiang Li, Wenhao Yang, Shusen Wang, and Zhihua Zhang. Communication-efficient local decentralized sgd methods. *arXiv preprint arXiv:1910.09126*, 2019b.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017.

Lumin Liu, Jun Zhang, SH Song, and Khaled B Letaief. Client-edge-cloud hierarchical federated learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6. IEEE, 2020.

Yucheng Lu and Christopher De Sa. Optimal complexity in decentralized training. In *International Conference on Machine Learning*, pp. 7111–7123. PMLR, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2021–2031. PMLR, 2020.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pp. 4848–4856. PMLR, 2018.

Yunfei Teng, Wenbo Gao, Francois Chalus, Anna E Choromanska, Donald Goldfarb, and Adrian Weller. Leader stochastic gradient descent for distributed training of deep learning models. *Advances in Neural Information Processing Systems*, 32, 2019.

Qianqian Tong, Guannan Liang, and Jinbo Bi. Effective federated adaptive gradient methods with non-iid decentralized data. *arXiv preprint arXiv:2009.06557*, 2020.

Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *Journal of Machine Learning Research*, 22(213):1–50, 2021. URL http://jmlr.org/papers/v22/20-147.html.

Yujia Wang, Lu Lin, and Jinghui Chen. Communication-compressed adaptive gradient method for distributed nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 6292–6320. PMLR, 2022a.

Yujia Wang, Lu Lin, and Jinghui Chen. Communication-efficient adaptive federated learning. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 22802–22838. PMLR, 2022b.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. *arXiv preprint arXiv:2101.11203*, 2021.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pp. 7184–7193. PMLR, 2019.

Dongruo Zhou, Jinghui Chen, Yuan Cao, Yiqi Tang, Ziyan Yang, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.