FRIENDS TO HELP: SAVING FEDERATED LEARNING FROM CLIENT DROPOUT

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) is an outstanding distributed machine learning framework due to its benefits on data privacy and communication efficiency. Since full client participation in many cases is infeasible due to constrained resources, partial participation FL algorithms have been investigated that *proactively* select/sample a subset of clients, aiming to achieve learning performance close to the full participation case. This paper studies a *passive* partial client participation scenario that is much less well understood, where partial participation is a result of external events, namely client dropout, rather than a decision of the FL algorithm. We cast FL with client dropout as a special case of a larger class of FL problems where clients can submit substitute (possibly inaccurate) local model updates. Based on our convergence analysis, we develop a new algorithm FL-FDMS that discovers friends of clients (i.e., clients whose data distributions are similar) on-the-fly and uses friends' local updates as substitutes for the dropout clients, thereby reducing the substitution error and improving the convergence performance. A complexity reduction mechanism is also incorporated into FL-FDMS, making it both theoretically sound and practically useful. Experiments on MNIST and CIFAR-10 confirmed the superior performance of FL-FDMS in handling client dropout in FL.

1 INTRODUCTION

Federated learning (FL) is a distributed machine learning paradigm where a set of clients with decentralized data work collaboratively to learn a model under the coordination of a centralized server. Depending on whether or not all clients participate in every learning round, FL is classified as either *full participation* or *partial participation*. While full participation is the ideal FL mode that achieves the best convergence performance, a lot of effort has been devoted to developing partial participation strategies via client selection/sampling Karimireddy et al. (2019); Li et al. (2019a); Yang et al. (2020); Ribero & Vikalo (2020); Chen et al. (2020a); Cho et al. (2020); Lai et al. (2021); Wu & Wang (2022); Balakrishnan et al. (2021) due to the attractive benefit of reduced resource (i.e. communication and computation) consumption. Existing works show that some of these partial participation strategies Li et al. (2019a); Yang et al. (2020) can indeed achieve performance close to full participation. Although the details differ, the principal idea of these strategies is the careful selection of *appropriate* clients to participate in each FL round. For example, in many cases Karimireddy et al. (2019); Li et al. (2019a); Yang et al. (2020), clients are sampled uniformly at random so that the participating clients form an "unbiased" representation of the whole client population in terms of the data distribution. In others Ribero & Vikalo (2020); Chen et al. (2020a); Cho et al. (2020); Lai et al. (2021); Wu & Wang (2022); Balakrishnan et al. (2021), "important" clients are selected more often to lead FL towards the correct loss descending direction.

This paper studies partial participation FL, but from an angle in stark contrast with existing works. In our considered problem, partial participation is a result of an arbitrary client dropout process, which the FL algorithm has absolutely no control over. For example, a client may not be able to participate (in other words, drop out) in a FL round due to, e.g., dead/low battery or loss of the communication signal. This means that the subset of clients participating in a FL round may not be "representative" or "important" in any sense. Note that the client dropout process by no means must be stochastic. Does FL still work with arbitrary client dropout? How does client dropout affect the

FL performance? How to mitigate the negative impact of client dropout on FL convergence? These are the central questions that this paper strives to answer.

We shall note that client dropout can occur simultaneously with client selection/sampling and hence partial participation can be a mixed result of both. As will become clear, our algorithm can be readily applied to this scenario and our theoretical results can also be extended provided that the client selection/sampling strategy used in conjunction has its own theoretical performance guarantee. However, since these results will depend on the specific client selection/sampling strategy adopted, and in order to better elucidate our main idea, this paper will not consider client selection/sampling.

Main Contributions. (1) Our study starts with a general convergence analysis for a class of FL problems where clients can submit substitute (inaccurate) local model updates for their true local model updates in each round. It turns out that this class of FL problems include FL with client dropout as a special case. The main insight derived from this analysis is that the FL convergence performance depends on the total substitution error (i.e., the difference between the true update and its substitute) over the entire course of learning. As such, reducing the substitution error is the key to improving the FL performance with client dropout. (2) We then introduce the notion of "friendship" among clients, where clients are friends if their data distributions and hence their local model updates are similar enough. Therefore, an intuitive idea for mitigating the impact of client dropout is to use the local model update of a friend client (which does not drop out) as a substitute of that of the dropout client when computing the next global model, since doing so incurs a small substitution error. (3) Although the idea sounds promising, realizing it is highly non-trivial because the friendship is *unknown*. Thus, our algorithm must discover the friendship over the FL rounds and use the discovered friends for the local model update substitution purpose. We prove that our algorithm asymptotically achieves the same FL convergence bound when assuming that the friendship is fully revealed. Furthermore, because the vanilla friends discovery mechanism can involve a large number of model update comparison computations, we develop an improved algorithm that significantly reduces the computational complexity. With this improvement, our algorithm is both theoretically sound and practically useful. (4) Experiments on MNIST and CIFAR-10 confirmed that our algorithm can significantly improve the FL performance in the presence of client dropout.

Related Work. The FedAvg algorithm is an early FL algorithm proposed in konevcny et al. (2016), which sparked many follow-up works on FL. Early works focus on FL under the assumptions of i.i.d. datasets and full client participation Stich (2018); Yu et al. (2019b); Wang & Joshi (2018), and most of the theoretical works show a linear speedup for convergence for a sufficiently large number of learning rounds. For non-i.i.d. datasets, the performance of FedAvg and its variants were demonstrated empirically Karimireddy et al. (2019); Jeong et al. (2018); Zhao et al. (2018); Li et al. (2020); Sattler et al. (2019), and convergence bounds were proven for the full participation case in Karimireddy et al. (2019); Stich et al. (2018); Yu et al. (2019a); Reddi et al. (2020). With partial client participation, the convergence of FedAvg was proven in Li et al. (2019a) for strongly convex functions, and the convergence of a generalized FedAvg with two-sided learning rates was analyzed in Karimireddy et al. (2019); Yang et al. (2020).

Most existing works on partial participation study strategies that *proactively* select or sample a subset of clients to participate in FL. The convergence of simple strategies such as random selection has been studied in Karimireddy et al. (2019); Li et al. (2019a); Yang et al. (2020). This analysis is relatively easy because of the "unbiased" nature of random selection. As a more sophisticated strategy, Cho et al. (2020) selects clients with higher local loss and proves an increased convergence speed. However, the vast majority of client selection strategies Ribero & Vikalo (2020); Lai et al. (2021); Wu & Wang (2022) only empirically shows the performance improvement. Very few works studied biased client selection in FL. In Cho et al. (2022), the authors presented a convergence rate bound under biased client selection, which contains a constant term due to the bias. In our paper, partial participation is *not* a decision of the FL algorithm, and our focus is on how to mitigate the negative impact of client dropout.

Client dropout is related to the "straggler" issue in FL, which is caused by the delayed local model uploading by some clients. Existing solutions to the straggler issue can be categorized into the following three types: doing nothing but waiting McMahan et al. (2017), allowing clients to upload their local models asynchronously to the server Wu et al. (2020); Li et al. (2019b); Xie et al. (2019); Chen et al. (2020b), and using the stored last updates of the inactive clients to join the model aggregation Yan et al. (2020); Gu et al. (2021). A straggler-resilient FL is proposed in Reisizadeh et al.

(2020) that incorporates statistical characteristics of the clients' data to adaptively select clients. Some other recent works Ruan et al. (2021); Yang et al. (2022) studied FL with flexible client participation where client dropout is part of the considered scenario and provided general convergence analysis. However, unlike our paper, these works Ruan et al. (2021); Yang et al. (2022) do not propose mitigation schemes for client dropout.

Our proposed algorithm exploits client similarity for model substitution, which is related to clustered FL Ghosh et al. (2020); Ruan & Joe-Wong (2022); Dennis et al. (2021) which clusters clients with similar data distributions into the same group. However, our algorithm have a very different motivation and objective. Moreover, our algorithm only requires computing pair-wise similarity, which can be computed easily on the server, whereas clustered FL needs to identify a set of clients belong to the same cluster, which incurs much higher computation and communication overhead.

2 FEDERATED LEARNING WITH CLIENT DROPOUT

We consider a server and a set of K clients, indexed by $\mathcal{K} = \{1, ..., K\}$, who work together to train a machine learning model by solving a distributed optimization problem:

$$\min_{w \in \mathbb{R}^d} \left\{ f(w) := \frac{1}{K} \sum_{k=1}^{K} F^k(w) := \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{\xi^k \sim \mathcal{D}^k}[F^k(w; \xi^k)] \right\} \tag{1}$$

where $F^k : \mathbb{R}^d \to \mathbb{R}$ denotes the objective function, $\xi^k \sim \mathcal{D}^k$ represents the sample/s drawn from distribution $\mathcal{D}^k$ at the k-th client and $w \in \mathbb{R}^d$ is the model parameter to learn. In a non-i.i.d. data setting, which is the focus of this paper, the distributions $\mathcal{D}^k$ are different across the clients.

We consider a typical FL algorithm konevcny et al. (2016) working in the client dropout setting. In each round t, only a subset $\mathcal{S}_t \subseteq \mathcal{K}$ of clients participate due to external reasons uncontrollable by the FL algorithm. We call the clients that are cannot participate or complete the task *dropout* (or *inactive*) clients. Then, FL executes the following four steps among the *non-dropout* (or *active*) clients in round t:

1. **Global model download**. Each client $k \in \mathcal{S}_t$ downloads the global model w_t from the server.

2. **Local model update**. Each client $k \in \mathcal{S}_t$ uses w_t as the initial model to train a new local model w_{t+1}^k, typically by using mini-batch stochastic gradient descent (SGD) as $w_{t,0}^k = w_t, w_{t+1}^k = w_{t,E}^k, w_{t,\tau+1}^k = w_{t,\tau}^k - \eta_L g_{t,\tau}^k, \forall \tau = 1, ..., E$ where $\xi_{t,\tau}^k$ is a mini-batch of data samples independently sampled uniformly at random from the local dataset of client k, $g_{t,\tau}^k = \nabla F^k(w_{t,\tau}^k; \xi_{t,\tau}^k)$ is the mini-batch stochastic gradient, η_L is the client local learning rate and E is the number of iterations for local training.

3. **Local model upload**. Clients upload their local model updates to the server. Instead of uploading the local model w_{t+1}^k itself, client k can simply upload the *local model update* Δ_t^k, which is defined as the accumulative model parameter difference as $\Delta_t^k = -\sum_{\tau=0}^{E-1} g_{t,\tau}^k$.

4. **Global model update**. The server updates the global model by using the aggregated local model updates of the clients in $\mathcal{S}_t$ as $w_{t+1} = w_t + \eta \eta_L \Delta_t$, where $\Delta_t := \frac{1}{S_t} \sum_{k \in \mathcal{S}_t} \Delta_t^k$ and η is the global learning rate and $S_t \triangleq |\mathcal{S}_t|$ denotes the number of the non-dropout clients.

For the main result of this paper, we consider the most general case of the client dropout process by imposing only an upper limit on the dropout ratio. That is, there exists a constant $\alpha \in [0, 1)$ such that $(K - S_t)/K \le \alpha$. If all clients drop out in a round, then essentially the round is skipped. Later in this paper, we will impose additional conditions on the dropout process to facilitate our theoretical analysis. Also note that if $\mathcal{S}_t$ were a choice of the FL algorithm, then the problem would become FL with client selection/sampling. We stress again that in our problem, $\mathcal{S}_t$ is not a choice.

3 CONVERGENCE ANALYSIS

Consider a FL round t where the set $\mathcal{S}_t$ of clients are active while the remaining set $\mathcal{K} \backslash \mathcal{S}_t$ of clients dropped out. Thus, one can only use the local model updates Δ_t^k of the active clients in $\mathcal{S}_t$ to

perform global model update since the inactive clients upload nothing to the server. However, rather than completely ignoring the inactive clients, we write the aggregate model update Δ_t in a different way to include all clients in the equation:

$$\Delta_t := \frac{1}{S_t} \sum_{k \in \mathcal{S}_t} \Delta_t^k = \frac{1}{K} \left(\sum_{k \in \mathcal{S}_t} \Delta_t^k + \sum_{k \in \mathcal{K} \backslash \mathcal{S}_t} \tilde{\Delta}_t^k \right) \tag{2}$$

where in the second equality we simply take $\tilde{\Delta}_t^k = \frac{1}{S_t} \sum_{k \in \mathcal{S}_t} \Delta_t^k$. In other words, although the inactive clients did not participate in the round t's learning, it is equivalent to the case where an inactive client $k \in \mathcal{K} \backslash \mathcal{S}_t$ uses $\tilde{\Delta}_t^k = \frac{1}{S_t} \sum_{k \in \mathcal{S}_t} \Delta_t^k$ as a substitute of its true local update Δ_t^k (which it may not even calculate due to dropout). Apparently, because $\tilde{\Delta}_t^k \neq \Delta_t^k$ in general, such a substitute leads to an biased error in the global update and hence affects the FL convergence performance.

Leveraging the above observation, we consider a larger class of FL problems that include client dropout as a special case. Specifically, imagine that an inactive client k, instead of contributing nothing, uses a substitute $\tilde{\Delta}_t^k$ for Δ_t^k when submitting its local model update. Apparently, $\tilde{\Delta}_t^k = \frac{1}{S_t} \sum_{k \in \mathcal{S}_t} \Delta_t^k$ is a specific choice of the substitute. We will still use the notation Δ_t as the aggregate model update with local update substitution and the readers should not be confused. Our convergence analysis will utilize the following standard assumptions about the FL problem.

Assumption 1 (Lipschitz Smoothness). *The local objective functions satisfy the Lipschitz smoothness property, i.e.,$\exists L > 0$, such that $\|\nabla F^k(x) - \nabla F^k(y)\| \leq L\|x - y\|$, $\forall x, y \in \mathbb{R}^d$ and $\forall k \in \mathcal{K}$.*

Assumption 2 (Unbiased Local Gradient Estimator). *The mini-batch based local gradient estimator is unbiased, i.e. $\mathbb{E}_{\xi^k \sim \mathcal{D}^k}[\nabla F^k(x; \xi^k)] = \nabla F^k(x)$, $\forall k \in \mathcal{K}$.*

Assumption 3 (Bounded Local and Global Variance). *There exist constants $\rho_L > 0$ and $\rho_G > 0$ such that the variance of each local gradient estimator is bounded, i.e., $\mathbb{E}_{\xi^k \sim \mathcal{D}^k} \left[\|\nabla F^k(x; \xi^k) - \nabla F^k(x)\|^2 \right] \leq \rho_L^2, \forall x, \forall k \in \mathcal{K}$. And the global variability of the local gradient is bounded by $\|\nabla F^k(x) - \nabla f(x)\|^2 \leq \rho_G^2, \forall x, \forall k \in \mathcal{K}$.*

The following result on the upper bound for the τ-step SGD under full participation will be used.

Lemma 1 (Lemma 4 in Reddi et al. (2020)). *For any step-size satisfying $\eta_L \leq \frac{1}{8LE}$, we have: $\forall \tau = 0, ..., E - 1$*

$$\frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[\|w_{t,\tau}^k - w_t\|^2] \leq 5E\eta_L^2(\rho_L^2 + 6E\rho_G^2) + 30E^2\eta_L^2\|\nabla f(w_t)\|^2 \tag{3}$$

Several additional notations will be handy. Let $\bar{\Delta}_t := \frac{1}{K} \sum_{k \in \mathcal{K}} \Delta_t^k$ be the average local model update assuming that all clients are active in round t and submitted their true local updates. Thus, $e_t := \Delta_t - \bar{\Delta}_t$ represents aggregate global update error due to client dropout and local update substitution in round t. Furthermore, let $e_t^k := \tilde{\Delta}_t^k - \Delta_t^k$, $\forall k \in \mathcal{K} \backslash \mathcal{S}_t$ be the individual substitution error for an individual inactive client k in round t.

Theorem 1. *Let constant local and global learning rates η_L and η be chosen as such that $\eta_L \leq \frac{1}{8EL}$ and $\eta\eta_L \leq \frac{1}{4EL}$. Under Assumption 1-3, the sequence of model w_t generated by using model update substitution with a substitution error sequence $e_0, ..., e_{T-1}$ satisfies*

$$\min_{t=0,...,T-1} \mathbb{E}\|\nabla f(w_t)\|^2 \leq \frac{f_0 - f_*}{c\eta\eta_L ET} + \Phi + \Psi(e_0, ..., e_{T-1}) \tag{4}$$

where $\Phi = \frac{1}{c} \left[5\eta_L^2 EL^2(\rho_L^2 + 6E\rho_G^2) + \frac{\eta\eta_L L}{K}\sigma_L^2 \right]$, c is a constant, $f_0 \triangleq f(w_0)$, $f_* \triangleq f(w_*)$, w_* is the optimal model and

$$\Psi(e_0, ..., e_{T-1}) = \frac{1 + 3\eta\eta_L LE}{cE^2 T} \sum_{t=0}^{T-1} \mathbb{E}_t[\|e_t\|^2] \tag{5}$$

where the expectation is over the local dataset samples among the clients.

The above convergence bound contains three parts: a vanishing term $\frac{f_0 - f_*}{c\eta\eta_L ET}$ as T increases, a constant term Φ whose size depends on the problem instance parameters and is independent of T, and a third term that depends on the sequence of substitution errors $e_0, ..., e_{T-1}$. Thus, by using constant learning rates η and η_L and assuming that $\|e_t\|^2$ is bounded, then the convergence bound is $O(1/T) + C$, where C is a constant. The key insight derived by Theorem 1 is that the FL convergence bound depends on the cumulative substitution error $\sum_{t=0}^{T-1} \mathbb{E}_t[\|e_t\|^2]$. When there is no dropout client, namely all clients participated in every round and submitted their true local model updates, the cumulative substitution error is 0 and hence, $\Psi(e_0, ..., e_{T-1}) = 0$. Thus, the convergence bound is simply $\frac{f_0 - f_*}{c\eta\eta_L ET} + \Phi$, which degenerates to the same bound established in Yang et al. (2020) for the normal full participation case. Next, we derive a more specific bound on $\Psi(e_0, ..., e_{T-1})$ for the naive dropout case where an inactive client uploads nothing to the server or, equivalently, uses $\frac{1}{S_t} \sum_{k \in \mathcal{S}_t} \Delta_t^k$ as a substitute. The following additional assumption is needed.

Assumption 4. *For any two clients i and j, the local model update difference is bounded as follows:*

$$\mathbb{E}[\|\Delta^i(w) - \Delta^j(w)\|^2] \leq \sigma_{i,j}^2, \forall w \tag{6}$$

where the expectation is over the local dataset samples.

Assumption 4 provides a pairwise characterization of clients' dataset heterogeneity in terms of the local model updates. When two clients i, j have the same data distribution, and assuming that the min-batch SGD utilizes the entire local dataset (i.e., the local gradient estimator is accurate), then it is obvious $\sigma_{i,j}^2 = 0$. We let $\sigma_P^2 \triangleq \max_{i,j} \sigma_{i,j}^2$ be the maximum pairwise difference.

With Assumption 4, the round-t substitution error can then be bounded as $\mathbb{E}[\|e_t\|^2] \leq \alpha^2 \sigma_P^2$ (See details in Appendix A.4). Plugging this bound into $\Psi(e_0, ..., e_{T-t})$, we have

$$\Psi(e_0, ..., e_{T-1}) \leq \frac{\alpha^2 \sigma_P^2 (1 + 3\eta\eta_L LE)}{cE^2} \triangleq \bar{\Psi} \tag{7}$$

Note that $\bar{\Psi}$ is a constant independent of T. This implies that, with constant learning rates η_L and η, $\min_t \mathbb{E}\|\nabla f(w_t)\|^2$ converges to some value at most $\Phi + \bar{\Psi}$ as $T \to \infty$.

Convergence Rate: By using a local learning rate $\eta_L \sim \mathcal{O}(\frac{1}{\sqrt{T}})$ and a global learning rate $\eta \sim \mathcal{O}(1)$, the convergence rate can be improved to $\mathcal{O}(\frac{1}{E\sqrt{T}}) + \mathcal{O}(\frac{E^2}{T}) + \mathcal{O}(\frac{1}{K\sqrt{T}}) + \mathcal{O}(\frac{\alpha^2 \sigma_P^2}{\sqrt{T}}) + \mathcal{O}(\frac{\alpha^2 \sigma_P^2}{E})$. Similar to the biased client selection in Cho et al. (2022), even with decaying learning rates, the convergence bound contains a non-vanishing term (i.e., the last term) in the general case. However, in our case, the non-vanishing term is due to replacing the model of the dropout clients with an arbitrary model. Nevertheless, with a decreasing and vanishing dropout rate α, the last term also vanishes and the $\min_t \mathbb{E}\|\nabla f(w_t)\|^2$ converges to 0.

4 FRIEND MODEL SUBSTITUTION

In this section, we develop a new algorithm to reduce or even eliminate the non-vanishing term in the convergence bound of FL with client dropout. Our key idea is to find a better substitute $\tilde{\Delta}_t^k$ for Δ_t^k when client k drops out in round t in order to reduce $\Psi(e_0, ..., e_{T-1})$. This is possible by noticing that $\sigma_{i,j}^2$ are different across client pairs and the local model updates are more similar when the clients' data distributions are more similar. Thus, when a client i drops out, one can use the local model update Δ_t^j as a replacement of Δ_t^i if j shares a similar data distribution with i, or in our terminology, j is a friend of i. We make "friendship" formal in the following definition.

Definition 1 (Friendship). *Let $\sigma_F^2 < \sigma_P^2$ be some constant. We say that clients i and j are friends if $\sigma_{i,j}^2 \leq \sigma_F^2$. Further, denote $\mathcal{B}_k$ as the set of friends of client k and $B_k = |\mathcal{B}_k|$ as the size of $\mathcal{B}_k$.*

Assumption 5 (Friend Presence). *In any round t, for any inactive client $i \in \mathcal{K} \backslash \mathcal{S}_t$, there exists at least one active client j that is client i's friend.*

Assumption 5 states that using local model update of a friend to replace that of an inactive client is feasible. Nevertheless, this assumption can be relaxed so that friends may not be present in every round when a client drops out. We will cover the relaxed case in the supplementary material.

Note that although such "friendship" exists among the clients, they are **hidden** from the FL algorithm. Thus, it is not straightforward to substitute the model update of a dropout client with that of its friends. Nevertheless, it is still useful to understand what can be achieved assuming that the friendship information is fully revealed. The convergence bound in this *non-realizable* case will serve as an optimal baseline for our algorithm to be developed.

With friend model substitution, the accumulate substitution error can be bounded as $\mathbb{E}[\|e_t\|^2] \leq \alpha^2 \sigma_F^2$ (see Appendix A.4). Substituting this bound into $\Psi(e_0, ..., e_{T-1})$, we have

$$\Psi(e_0, ..., e_{T-1}) \leq \frac{\alpha^2 \sigma_F^2 (1 + 3\eta\eta_L LE)}{cE^2} \triangleq \Psi^* \tag{8}$$

Note that Ψ^* is still a constant independent of T but it is much smaller than $\bar{\Psi}$, since typically $\sigma_F^2 \ll \sigma_P^2$. Therefore, the convergence bound can be significantly improved if the algorithm utilizes the friendship information.

Convergence Rate: With a local learning rate $\eta_L \sim \mathcal{O}(\frac{1}{\sqrt{T}})$ and a global learning rate $\eta \sim \mathcal{O}(1)$, the convergence rate can be improved to $\mathcal{O}(\frac{1}{E\sqrt{T}}) + \mathcal{O}(\frac{E^2}{T}) + \mathcal{O}(\frac{1}{K\sqrt{T}}) + \mathcal{O}(\frac{\alpha^2\sigma_F^2}{\sqrt{T}}) + \mathcal{O}(\frac{\alpha^2\sigma_F^2}{E})$. The above convergence bound still has a non-vanishing constant term (i.e., the last term) in the general case. In some special cases where σ_F^2 dynamically decreases over round, this term may also vanish as $T \to \infty$. For example, consider that clients are grouped into clusters and clients from the same cluster have the identical data distribution. Borrowing the idea from Shi et al. (2022) on the increasing mini-batch size, the local model update difference of two clients, i.e., $\sigma_{i,j}^2$ from the same cluster also decreases with an increasing batch size B. Assuming $\mathbb{E}[\|\Delta^i(w) - \Delta^j(w)\|^2] \leq \sigma_F^2/B, \forall w$, and by using the mini-batch size sequence $B_t = B_0 t$ where B_0 is the initial mini-batch size, then the last term becomes $\mathcal{O}(\frac{\alpha^2\sigma_F^2 \ln T}{TE})$, which goes to 0 as $T \to \infty$. Therefore, the constant term in the convergence rate is eliminated even with constant dropout rate α.

The above analysis shows that the FL convergence can be substaintially improved if the algorithm can utilize the friendship information. Next, we develop a learning-assisted FL algorithm, called FL with Friend Discovery and Model Substitution (FL-FDMS), that discovers the friends of clients and use the model update of the discovered friends for substitution. We prove that FL-FDMS achieves asymptotically the optimal error bound Ψ^*.

4.1 ALGORITHM (FL-FDMS)

In each FL round t, in addition to the regular steps in the FL algorithm described in Section 3, FL-FDMS performs different actions depending on whether or not a client drops out. For active clients, FL-FDMS calculates pairwise similarity scores to learn the similarity between clients. For inactive clients, FL-FDMS uses the historical similarity scores to find active friend clients and use their local model updates as substitutes. We describe these two cases in more detail below.

Active Clients. For any pair of active clients i and j in $\mathcal{S}_t$. The server calculates a similarity score $r_t^{i,j} = r(\Delta_t^i, \Delta_t^j)$ based on their uploaded local model updates Δ_t^i and Δ_t^j. Many functions can be used to calculate the score. For example, $r(\Delta_t^i, \Delta_t^j)$ can simply be the negative model difference, i.e., $-\|\Delta_t^i - \Delta_t^j\|$, or the normalized cosine similarity, i.e.,

$$r(\Delta_t^i, \Delta_t^j) = \frac{1}{2}\left(\frac{\langle\Delta_t^i, \Delta_t^j\rangle}{\|\Delta_t^i\|\|\Delta_t^j\|} + 1\right) \tag{9}$$

Because the normalized cosine similarity takes value from a bounded and normalized range $[0, 1]$, which is more amenable for mathematical analysis, we will use this function in this paper. Clearly, a higher similarity score implies that the two clients are more similar in terms of their data distribution.

However, a single similarity score calculated in one particular round does not provide accurate similarity information because of the randomness in the initial model in that round and the randomness in mini-batch SGD for local model computation. Thus, the server maintains and updates an average similarity score $R_t^{i,j}$ for clients i and j based on all similarity scores calculated so far as follows,

$$R_t^{i,j} = \begin{cases} \frac{N_{t-1}^{i,j}}{N_{t-1}^{i,j}+1}R_{t-1}^{i,j} + \frac{1}{N_{t-1}^{i,j}+1}r_t^{i,j}, & \text{if } i,j \in \mathcal{S}_t \\ R_{t-1}^{i,j}, & \text{otherwise} \end{cases} \tag{10}$$

where $N_t^{i,j}$ is the number of rounds where both clients i and j did not drop out up to round t.

Inactive Clients. For any inactive client k, the server looks up $R_t^{k,i}$ between k and every active client $i \in \mathcal{S}_t$, finds the one with the highest similarity score, denoted by $\phi_t(k) = \arg\max_{i \in \mathcal{S}_t} R_t^{k,i}$, and uses the local model update $\Delta_t^{\phi_t(k)}$ as a substitute for Δ_t^k when computing the global update.

4.2 CONVERGENCE ANALYSIS

In this subsection, we analyze the convergence of FL-FDMS. Since Theorem 1 has already proven the FL convergence bound depending on a general error sequence, we will focus only on bounding $\Psi(e_0, e_1, ..., e_{T-1})$ in our algorithm. The following additional assumptions on the dropout process are needed.

Assumption 6 (Sufficiently Many Common Rounds). *There exists a constant $\beta \in (0, 1]$ so that for any pair of clients i and j, $N_t^{i,j}$ satisfies $N_t^{i,j} \geq \beta t, \forall t$.*

We first establish a bound on the probability that $\phi_t(k)$ selected by our algorithm is not a friend of client k. Let $\mathbb{E}_t[r_t^{i,j}] = \mu^{i,j}$ be the expected similarity score between clients i and j. Thus $\delta_k = \min_{i \in \mathcal{B}_k} \mu^{k,i} - \max_{i \notin \mathcal{B}_k} \mu^{k,i}$ denotes the minimum similarity score gap between a friend client and a non-friend client. We further let $\delta_{\min} = \min_k \delta_k$ for all $k \in \mathcal{K}$.

Lemma 2. *The probability that our algorithm selects a non-friend client for an inactive client in round t is upper bounded by $2K \exp\left(\frac{-\beta \delta_{min}^2 t}{2}\right)$.*

Now, we are ready to bound $\mathbb{E}\|e_t\|^2$ by using FL-FDMS. Firstly, we can show the following bound on $\mathbb{E}\|e_t\|^2$ (see Appendix A.4):

$$\mathbb{E}\|e_t\|^2 \leq \alpha^2 \left(\sigma_F^2 + 2K \exp\left(\frac{-\beta \delta_{min}^2 t}{2}\right)(\sigma_P^2 - \sigma_F^2) \right) \tag{11}$$

Plugging this bound into Ψ (see Appendix A.5), we have

$$\Psi(e_0, ..., e_{T-1}) \leq \Psi^* + 2K\bar{\Psi} \frac{1 - \exp\left(\frac{-\beta \delta_{\min}^2 T}{2}\right)}{T\left(1 - \exp\left(\frac{-\beta \delta_{\min}^2}{2}\right)\right)} \tag{12}$$

Note that as $T \to \infty$, $\Psi(e_0, ..., e_{T-1}) \to \Psi^*$. Therefore, FL-FDMS asymptotically approaches the convergence bound obtained in the fully revealed friendship information case.

4.3 REDUCING SIMILARITY COMPUTATION COMPLEXITY

FL-FDMS requires pairwise similarity score computation for all active clients in every round, causing a big computation burden when $\mathcal{S}_t$ is large. We now design a simple mechanism inspired by Hoeffding races Maron & Moore (1993) to reduce the computational complexity and incorporate it in FL-FDMS. To this end, we introduce a notion called candidate (friend) set $\mathcal{C}_t^k$ for each client k, which keeps the potential friend list of client k. Initially, $\mathcal{C}_t^k$ is the entire client set $\mathcal{K}$, but over time, $\mathcal{C}_t^k$ shrinks by eliminating non-friend clients with high probability. Thus, the server only needs to compute the similarity score of clients in the candidate friend set of a client k and pick a client from this set for the substitution purpose. The updating rule for $\mathcal{C}_t^k$ is as follows. In every round t, for each client k: (1) Find the client $\phi_t(k)$ with the highest similarity score, i.e., $\phi_t(k) = \arg\max_{i \in \mathcal{C}_k^t} R_t^{k,i}$; (2) Compute the similarity score gap $g_{k,i} = R_t^{k,\phi_t(k)} - R_t^{k,i}$ for any other client $i \neq \phi_t(k), i \in \mathcal{C}_t^k$; (3) Eliminate i from $\mathcal{C}_t^k$ if $g_{k,j} \geq \Theta_t$, namely $\mathcal{C}_{t+1}^k \leftarrow \mathcal{C}_t^k - \{i : g_{k,i} \geq \Theta_t\}$, where Θ_t is a threshold parameter decreasing over t.

In Theorem 2, we design a specific threshold sequence Θ_t and prove that our convergence bound in the previous section holds with high probability.

Theorem 2. *Let $\delta_f = \max_k\{\max_{i \in \mathcal{B}_k} \mu^{k,i} - \min_{i \in \mathcal{B}_k} \mu^{k,i}\}$ be the maximum similarity score gap among friends of any client, and $B_{\max} = \max_k B_k$ be the maximum number of friends that any client can have. For any $p \in (0, 1)$, by setting $\Theta_t = \sqrt{\frac{2 \ln(2K^2 T B_{\max}) - 2 \ln p}{\beta t}} + \delta_f$, FL-FDMS with complexity reduction yields, with probability at least $1 - p$, the same bound on Ψ as in equation 69.*

Note that in order to establish the convergence bound, we used several loose bounds in the proof of Theorem 2, thereby resulting an unnecessarily large threshold sequence Θ_t. In practice, the threshold sequence Θ_t can be chosen much smaller than what is given in Theorem 2, and hence non-friend clients can be eliminated more quickly with high confidence.

5 EXPERIMENTS

Setup. We perform experiments on two standard public datasets, namely MNIST and CIFAR-10, which are widely used in FL experiments, in a clustered setting as well as a general setting, in the case of clustered setting, the clients are grouped into clusters with an equal number of clients, and in the case of general setting, we use a common way of generating non-i.i.d. FL dataset, the detail settings can be found in Appendix. We compare FL-FDMS with three variants of FedAvg since in this paper we describe FL-FDMS in the context of FedAvg.

1. **Full Participation (Full)**. This is the ideal case where all clients participate in FL without dropout. It is used as a performance upper-bound.

2. **Client Dropout (Dropout)**. In this case, the server simply ignores the dropout clients and performs global aggregation on the non-dropout clients.

3. **Staled Substitute (Stale)**. Another method to deal with dropout clients is to use their last uploaded local model updates for the current round's global aggregation. Such a method was also used to deal with the "straggler" issue in FL in some previous works Yan et al. (2020); Gu et al. (2021).
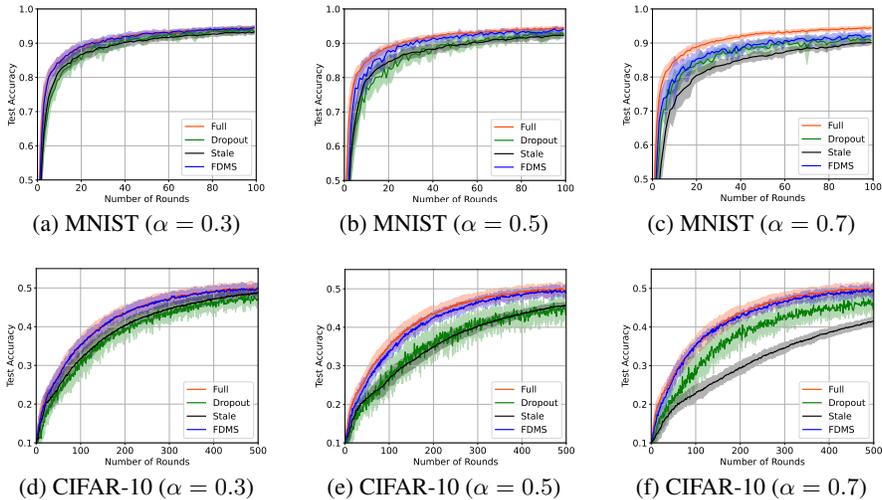

Figure 1: Performance comparison on the clustered setting with various α

Performance Comparison. We first compare the convergence performance in the clustered setting under different dropout ratio $\alpha \in \{0.3, 0.5, 0.7\}$. Fig. 1 plots the convergence curves on the MNIST dataset and the CIFAR-10 dataset, respectively. Several observations are made as follows. First, **FL-FDMS** outperforms **Dropout** and **Stale** in terms of test accuracy and convergence speed, and achieves performance close to **Full** in all cases. Second, **FL-FDMS** reduces the fluctuations caused by the client dropout on the convergence curve. Third, with a larger dropout ratio, the performance improvement of **FL-FDMS** is larger. Fourth, on more complex datasets (e.g., CIFAR-10), **FL-FDMS** achieves a even more significant performance improvement.

We note that **Stale** is the most sensitive to the dropout ratio α and its performance degrades significantly as α increases. This is because with a larger α, individual clients has few participating opportunities. As a result, the staled local models become too outdated to provide useful information for the current round's global model updating.

We also perform experiments in the more general non-iid case to illustrate the wide applicability of the proposed algorithm. Fig. 7 plots the convergence curves on CIFAR-10 under the general setting. The results confirm the superiority of **FL-FDMS**. However, we also note that the improvement is smaller than that in the clustered setting. This suggests a limitation of **FL-FDMS**, which works best when the "friendship" relationship among the clients is stronger.

(a) CIFAR-10 ($\alpha = 0.3$)   (b) CIFAR-10 ($\alpha = 0.5$)   (c) CIFAR-10 ($\alpha = 0.7$)
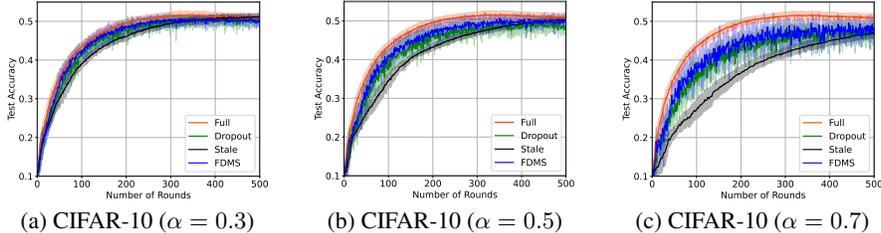
Figure 2: Performance comparison on the CIFAR-10 general setting with various α

Friend Discovery. **FL-FDMS** relies on successfully discovering the friends of dropout clients. In Fig. 3, we show the pairwise similarity scores in the final learning round. In our controlled clustered setting, 20 clients were grouped into 5 clusters, but this information was not known by the algorithm at the beginning. As the figure shows, the similarity scores obtained by **FL-FDMS** are larger for intra-cluster client pairs and smaller for inter-cluster client pairs, indicating that the clustering/friendship information can be successfully discovered. Moreover, our experiments show that the discovered friendship is more obvious for CIFAR-10 than for MNIST. This is likely due to the different dataset structures and the different CNN models adopted.
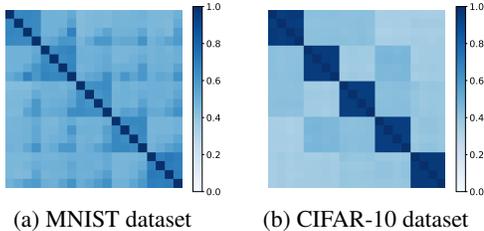

(a) MNIST dataset   (b) CIFAR-10 dataset

Figure 3: Pairwise similarity scores in the final learning round.
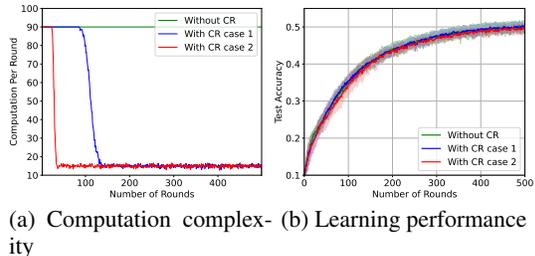

(a) Computation complexity   (b) Learning performance

Figure 4: Complexity reduction on the CIFAR-10 clustered setting ($\alpha = 0.5$).

Computation Complexity. In this set of experiments, we investigate the computational complexity of **FL-FDMS** with and without the complexity reduction mechanism (CR) on CIFAR-10 in the clustered setting. For the complexity reduction mechanism, we use two different threshold sequences: Θ_t in Theorem 2 (Case 1), and $0.5\Theta_t$ (Case 2). In Fig.4, we can see that the complexity reduction mechanism can significantly reduce the computation complexity while still achieving a similar learning performance. We also notice that a smaller threshold sequence accelerates the client elimination process while keeping a similar learning performance.

6   CONCLUSION

This paper investigated the impact of client dropout on the convergence of FL. Our analysis treats client dropout as a special case of local update substitution and characterizes the convergence bound in terms of the total substitution error. This inspired us to develop FL-FDMS, which discovers friend clients on-the-fly and uses friends' updates to reduce the substitution errors, thereby mitigating the negative impact of client dropout. Extensive experiment results show that discovering the client "friendship" is possible and friend clients can be a useful resort for addressing client dropout problems.

REFERENCES

Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. Diverse client selection for federated learning via submodular maximization. In *International Conference on Learning Representations*, 2021.

Wenlin Chen, Samuel Horvath, and Peter Richtarik. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020a.

Yujing Chen, Yue Ning, Martin Slawski, and Huzefa Rangwala. Asynchronous online federated learning for edge devices with non-iid data. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 15–24. IEEE, 2020b.

Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020.

Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 10351–10375. PMLR, 2022.

Don Kurian Dennis, Tian Li, and Virginia Smith. Heterogeneity for the win: One-shot federated clustering. In *International Conference on Machine Learning*, pp. 2611–2620. PMLR, 2021.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

Xinran Gu, Kaixuan Huang, Jingzhao Zhang, and Longbo Huang. Fast federated learning in the presence of arbitrary device unavailability. *Advances in Neural Information Processing Systems*, 34:12052–12064, 2021.

Ashwin R Jadhav. Federated-learning (pytorch), 2020. URL https://github.com/AshwinRJ/Federated-Learning-PyTorch.

Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. 2019.

Jakub konevcny, 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.

Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. Oort: Efficient federated learning via guided participant selection. In *15th Symposium on Operating Systems Design and Implementation*, pp. 19–35, 2021.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*, 2021.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

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.

Yanan Li, Shusen Yang, Xuebin Ren, and Cong Zhao. Asynchronous federated learning with differential privacy for edge intelligence. *arXiv preprint arXiv:1912.07902*, 2019b.

Oded Maron and Andrew Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. *Advances in neural information processing systems*, 6, 1993.

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.

Amirhossein Reisizadeh, Isidoros Tziotis, Hamed Hassani, Aryan Mokhtari, and Ramtin Pedarsani. Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity. *arXiv preprint arXiv:2012.14453*, 2020.

Monica Ribero and Haris Vikalo. Communication-efficient federated learning via optimal client sampling. *arXiv preprint arXiv:2007.15197*, 2020.

Yichen Ruan and Carlee Joe-Wong. Fedsoft: Soft clustered federated learning with proximal local updating. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8124–8131, 2022.

Yichen Ruan, Xiaoxi Zhang, Shu-Che Liang, and Carlee Joe-Wong. Towards flexible device participation in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3403–3411. PMLR, 2021.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

Dian Shi, Liang Li, Maoqiang Wu, Minglei Shu, Rong Yu, Miao Pan, and Zhu Han. To talk or to work: Dynamic batch sizes assisted time efficient federated learning over future mobile edge devices. *IEEE Transactions on Wireless Communications*, 2022.

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

Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. *Advances in Neural Information Processing Systems*, 31, 2018.

Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

Hongda Wu and Ping Wang. Node selection toward faster convergence for federated learning on non-iid data. *IEEE Transactions on Network Science and Engineering*, 2022.

Wentai Wu, Ligang He, Weiwei Lin, Rui Mao, Carsten Maple, and Stephen Jarvis. Safa: a semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Transactions on Computers*, 70(5):655–668, 2020.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019.

Yikai Yan, Chaoyue Niu, Yucheng Ding, Zhenzhe Zheng, Fan Wu, Guihai Chen, Shaojie Tang, and Zhihua Wu. Distributed non-convex optimization with sublinear speedup under intermittent client availability. *arXiv preprint arXiv:2002.07399*, 2020.

Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. In *International Conference on Learning Representations*, 2020.

Haibo Yang, Xin Zhang, Prashant Khanduri, and Jia Liu. Anarchic federated learning. In *International Conference on Machine Learning*, pp. 25331–25363. PMLR, 2022.

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, 2019a.

Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5693–5700, 2019b.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.