A General Theory for Federated Optimization with Asynchronous and Heterogeneous Clients Updates

Yann Fraboni, Richard Vidal, Laetitia Kameni, Marco Lorenzi

A General Theory for Federated Optimization with Asynchronous and Heterogeneous Clients Updates

Yann Fraboni
Université Côte d'Azur, Inria Sophia Antipolis, Epione Research Group, France
Accenture Labs, Sophia Antipolis, France

Richard Vidal
Accenture Labs, Sophia Antipolis, France

Laetitia Kameni
Accenture Labs, Sophia Antipolis, France

Marco Lorenzi
Université Côte d'Azur, Inria Sophia Antipolis, Epione Research Group, France

Abstract

We propose a novel framework to study asynchronous federated learning optimization with delays in gradient updates. Our theoretical framework extends the standard FEDAVG aggregation scheme by introducing stochastic aggregation weights to represent the variability of the clients update time, due for example to heterogeneous hardware capabilities. Our formalism applies to the general federated setting where clients have heterogeneous datasets and perform at least one step of stochastic gradient descent (SGD). We demonstrate convergence for such a scheme and provide sufficient conditions for the related minimum to be the optimum of the federated problem. We show that our general framework applies to existing optimization schemes including centralized learning, FEDAVG, asynchronous FEDAVG, and FEDBUFF. The theory here provided allows drawing meaningful guidelines for designing a federated learning experiment in heterogeneous conditions. In particular, we develop in this work FEDFIX, a novel extension of FEDAVG enabling efficient asynchronous federated training while preserving the convergence stability of synchronous aggregation. We empirically demonstrate our theory on a series of experiments showing that asynchronous FEDAVG leads to fast convergence at the expense of stability, and we finally demonstrate the improvements of FEDFIX over synchronous and asynchronous FEDAVG.

1 Introduction

Federated learning (FL) is a training paradigm enabling different clients to jointly learn a global model without sharing their respective data. Federated learning is a generalization of distributed learning (DL), which was first introduced to optimize a given model in star-shaped networks composed of a server communicating with computing machines (Bertsekas and Tsitsiklis, 1989; Nedić et al., 2001; Zinkevich et al., 2009). In DL, the server owns the dataset and distributes it across machines. At every optimization round, the machines return the estimated gradients, and the server aggregates them to perform an SGD step. DL was later extended to account for SGD, and FL extends DL to enable optimization without sharing data between clients. Typical federated training schemes are based on the averaging of clients model parameters optimized locally by each client, such as in FEDAVG (McMahan et al., 2017), where at every optimization round clients perform a fixed amount of stochastic gradient descent (SGD) steps initialized with the current global model parameters, and subsequently return the optimized parameters to the server. The server computes the new global model as the average of the clients updates weighted by their respective data ratio.

A key methodological difference between the optimization problem solved in FL and the one of DL lies in the assumption of potentially non independent and identically distributed (iid) data instances (Kairouz et al., 2019; Yang et al., 2019). Proving convergence in the non-iid setup is more challenging, and in some settings, FEDAVG has been shown to converge to a sub-optimum, e.g. when each client performs a different amount of local work (Wang et al., 2020a), or when clients are not sampled in expectation according to their importance (Cho et al., 2020).

A major drawback of FEDAVG concerns the time needed to complete an optimization round, as the server must wait for all the clients to perform their local work to *synchronize* their update and create a new global model. As a consequence, due to the potential heterogeneity of the hardware across clients, the time for an optimization round is conditioned to the one of the slowest update, while the fastest clients stay idle once they have sent their updates to the server. To address these limitations, asynchronous FL has been proposed to take full advantage of the clients computation capabilities (Xu et al., 2021; Nguyen et al., 2018; Koloskova et al., 2019; De Sa et al., 2015). In the asynchronous setting, whenever the server receives a client's contribution, it creates a new global model and sends it back to the client. In this way, clients are never idle and always perform local work on a different version of the global model. While asynchronous FL has been investigated in the iid case (Stich and Karimireddy, 2020), a unified theoretical and practical investigation in the non-iid scenario is currently missing.

This work introduces a novel theoretical framework for asynchronous FL based on the generalization of the aggregation scheme of FEDAVG, where asynchronicity is modeled as a stochastic process affecting clients' contribution at a given federated aggregation step. More specifically, our framework is based on a stochastic formulation of FL, where clients are given stochastic aggregation weights dependent on their effectiveness in returning an update. Based on this formulation, we provide sufficient conditions for asynchronous FL to converge, and we subsequently give sufficient conditions for convergence to the FL optimum of the associated synchronous FL problem. Our conditions depend on the clients computation time (which can be eventually estimated by the server), and are independent from the clients data heterogeneity, which is usually unknown to the server.

With asynchronous FL, the server only waits for one client contribution to create the new global. As a result, optimization rounds are potentially faster even though the new global improves only for the participating client at the detriment of the other ones. This aspect may affect the stability of asynchronous FEDAVG as compared to synchronous FEDAVG and, as we demonstrate in this work, even diverge in some cases. To tackle this issue, we propose FEDFIX, a robust asynchronous FL scheme, where new global models are created with all the clients contributions received after a fixed amount of time. We prove the convergence of FEDFIX and verify experimentally that it outperforms standard asynchronous FEDAVG in the considered experimental scenarios.

The paper is structured as follows. In Section 2, we introduce our aggregation scheme and the closed-form of its aggregation weights in function of the clients computation capabilities and the considered FL optimization routine. Based on our aggregation scheme, in Section 3, we provide convergence guarantees, and we give sufficient conditions for the learning procedure to converge to the optimum of the FL optimization problem. In Section 4, we apply our theoretical framework to synchronous and asynchronous FEDAVG, and show that our work extends current state-of-the-art approaches to asynchronous optimization in FL. Finally, in Section 5, we demonstrate experimentally our theoretical results.

2 Background

We define here the formalism required by the theory that will be introduced in the following sections. We first introduce in Section 2.1 the FL optimization problem, and we adapt it in section 2.2 to account for delays in client contributions. We then generalize in Section 2.3 the FEDAVG aggregation scheme to account for contributions delays. In Section 2.4, we introduce the notion of virtual global models as a direct generalization of gradient descent, and introduce in Section 2.5 the final asynchronous FL optimization problem. Finally, we introduce in Section 2.6 a formalization of the concept of data heterogeneity across clients.

2.1 Federated Optimization Problem

We have M participants owning n_i data points $\{\boldsymbol{z}_{k,i}\}_{k=1}^{n_i}$ independently sampled from a fixed unknown distribution over a sample space $\{\mathcal{Z}_i\}_{i=1}^{M}$. We have $\boldsymbol{z}_{k,i} = (\boldsymbol{x}_{k,i}, \boldsymbol{y}_{k,i})$ for supervised learning, where $\boldsymbol{x}_{k,i}$ is the input of the statistical model, and $\boldsymbol{y}_{k,i}$ its desired target, while we denote $\boldsymbol{z}_{k,i} = \boldsymbol{x}_{k,i}$ for unsupervised learning. Each client optimizes the model's parameters $\boldsymbol{\theta}$ based on the estimated local loss $l(\boldsymbol{\theta}, \boldsymbol{z}_{k,i})$. The aim of FL is solving a distributed optimization problem associated with the averaged loss across clients

$$\mathcal{L}(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{z\sim\hat{\mathcal{Z}}}\left[l(\boldsymbol{\theta}, \boldsymbol{z})\right] = \frac{1}{\sum_{i=1}^{M} n_i} \sum_{i=1}^{M}\sum_{k=1}^{n_i} l(\boldsymbol{\theta}, \boldsymbol{z}_{k,i}), \tag{1}$$

where the expectation is taken with respect to the sample distribution $\hat{\mathcal{Z}}$ across the M participating clients. We consider a general form of the federated loss of equation (1) where clients local losses are weighted by an associated parameter p_i such that $\sum_{i=1}^{n} p_i = 1$, i.e.

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{M} p_i \mathcal{L}_i(\boldsymbol{\theta}) \text{ s.t. } \mathcal{L}_i(\boldsymbol{\theta}) = \frac{1}{n_i}\sum_{k=1}^{n_i} l(\boldsymbol{\theta}, \boldsymbol{z}_{k,i}). \tag{2}$$

The weight p_i can be interpreted as the importance given by the server to client i in the federated optimization problem. While any combination of $\{p_i\}$ is possible, we note that in typical FL formulations, either (a) every client has equal importance, i.e. $p_i = 1/M$, or (b) every data point is equally important, i.e. $p_i = n_i / \sum_{i=1}^{M} n_i$.

2.2 Asynchronicity in Clients Updates

An optimization round starts at time t^n with global model $\boldsymbol{\theta}^n$, finishes at time t^{n+1} with the new global model $\boldsymbol{\theta}^{n+1}$, and takes $\Delta t^n = t^{n+1} - t^n$ time to complete. No assumptions are made on Δt^n, which can be a random variable, and we set for convenience $t^0 = 0$. In this section, we introduce the random variables needed to develop in Section 2.3 the server aggregation scheme connecting two consecutive global models $\boldsymbol{\theta}^n$ and $\boldsymbol{\theta}^{n+1}$.

We define the random variable T_i representing the update time needed for client i to perform its local work and send it to the server for aggregation. T_i depends on the client computation and communication hardware, and is assumed to be independent from the current optimization round n. If the server sets the FL round time to $\Delta t^n = \max_i T_i$, the aggregation is performed by waiting for the contribution of every client, and we retrieve the standard client-server communication scheme of synchronous FEDAVG.

With asynchronous FEDAVG, we need to relate T_i to the server aggregation time Δt^n. We introduce $\rho_i(n)$ the index of the most recent global model received by client i at optimization round n and, by construction, we have $0 \leq \rho_i(n) \leq n$. We define by

$$T_i^n \coloneqq T_i - (t^n - t^{\rho_i(n)})$$

the remaining time at optimization round n needed by client i to complete its local work.

Comparing T_i^n with Δt^n indicates whether a client is participating to the optimization round or not, through the stochastic event $\mathbb{I}(T_i^n \leq \Delta t^n)$. When $\mathbb{I}(T_i^n \leq \Delta t^n) = 1$, the local work of client i is used to create the new global model $\boldsymbol{\theta}^{n+1}$, while client i does not contribute when $\mathbb{I}(T_i^n \leq \Delta t^n) = 0$. With synchronous FEDAVG, we retrieve $\mathbb{I}(T_i^n \leq \Delta t^n) = \mathbb{I}(T_i \leq \max_i T_i) = 1$ for every client.

Figure 1 illustrates the notations described in this section in a FL process with $M = 2$ clients.

2.3 Server Aggregation Scheme

We consider $\boldsymbol{\Delta}_i(n)$ the contribution of client i received by the server at optimization round n. In the rest of this work, we consider that clients perform K steps of SGD on the model they receive from the server. By calling their trained model $\boldsymbol{\theta}_i^{(n,k)}$ after k SGD, we can rewrite clients contribution for FEDAVG as $\boldsymbol{\Delta}_i(n) \coloneqq \boldsymbol{\theta}_i^{(n,K)} - \boldsymbol{\theta}^n$, and the FEDAVG aggregation scheme as

$$\boldsymbol{\theta}^{n+1} \coloneqq \boldsymbol{\theta}^n + \sum_{i=1}^{M} p_i \boldsymbol{\Delta}_i(n). \tag{3}$$

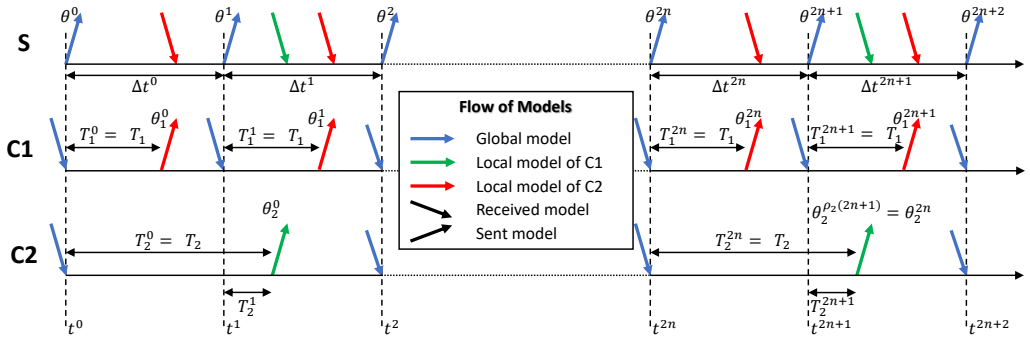

Figure 1: Illustration of the time notations introduced in Section 2.2 with $M = 2$ clients. The frequency of the updates of Client 1 (C1) is twice the one of Client 2 (C2). If the server (S) creates the new global model after every fixed waiting time ($\Delta t^n = \Delta t$), C1 contributes at every optimization round, while C2 contributes once every two rounds. This aggregation policy define the federated learning strategy FEDFIX (Section 4.4)

With FEDAVG, the server waits for every client to send its contribution $\boldsymbol{\Delta}_i(n)$ to create the new global model. To allow for partial computation within the server aggregation scheme, we introduce the aggregation weight $d_i(n)$ corresponding to the weight given by the server to client i at optimization round n. We can then define the stochastic aggregation weight $\omega_i(n)$ given to client i at optimization step n as

$$\omega_i(n) \coloneqq \mathbb{I}(T_i^n \leq \Delta t^n) d_i(n), \tag{4}$$

with $\omega_i(n) = d_i(n)$ if client i updated its work at optimization round n and $\omega_i(n) = 0$ otherwise. In the general setting, client i receives $\boldsymbol{\theta}^{\rho_i(n)}$ and its contribution is $\boldsymbol{\Delta}_i(\rho_i(n)) = \boldsymbol{\theta}_i^{(\rho_i(n),K)} - \boldsymbol{\theta}^{\rho_i(n)}$. By weighting each delayed contribution $\boldsymbol{\Delta}_i(\rho_i(n))$ with its stochastic aggregation weight $\omega_i(n)$, we propose the following aggregation scheme

$$\boldsymbol{\theta}^{n+1} \coloneqq \boldsymbol{\theta}^n + \eta_g \sum_{i=1}^{M} \omega_i(n) \boldsymbol{\Delta}_i(\rho_i(n)), \tag{5}$$

where η_g is a global learning rate that the server can use to mitigate the disparity in clients contributions (Reddi et al., 2021; Karimireddy et al., 2020; Wang et al., 2020b). Equation (5) generalizes FedAvg aggregation scheme (3) ($\eta_g = 1$ and $\Delta t^n = \max_i T_i$), and the one of Fraboni et al. (2022) based on client sampling.

We introduce with Algorithm 1 the implementation of the optimization schemes satisfying aggregation scheme (5) with stochastic aggregation weights satisfying equation (4).

Algorithm 1 Asynchronous Federated Learning based on equation (5)

Require: server learning rate η_g, aggregation weights $\{d_i(n)\}$, number of SGD K, learning rate η_l, batch size B, aggregation time policy Δt^n.

1: The server sends to the M clients the learning parameters (K, η_l, B) and the initial global model $\boldsymbol{\theta}^0$.
2: **for** $n \in \{0, ..., N-1\}$ **do**
3: Clients in $S_n = \{i : T_i^n \leq \Delta t^n\}$ send their contribution $\boldsymbol{\Delta}_i(\rho_i(n)) = \boldsymbol{\theta}_i^{\rho_i(n)+1} - \boldsymbol{\theta}^{\rho_i(n)}$ to the server.
4: The server creates the new global model $\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \eta_g \sum_{i \in S_n} d_i(n) \boldsymbol{\Delta}_i(\rho_i(n))$, equation (5).
5: The global model $\boldsymbol{\theta}^{n+1}$ is sent back to the clients in S_n.
6: **end for**

2.4 Expressing FL as cumulative GD steps

To obtain the tightest possible convergence bound, we consider a convergence framework similar to the one of Li et al. (2020b) and Khaled et al. (2020). We introduced the aggregation rule for the

server global models $\{\boldsymbol{\theta}^n\}$ in Section 2.3, and we generalize it in this section by introducing the virtual sequence of global models $\boldsymbol{\theta}^{n,k}$. This sequence corresponds to the *virtual* global model that would be obtained with the clients contribution at optimization round n computed on $k \leq K$ SGD, i.e.

$$\boldsymbol{\theta}^{n,k} \coloneqq \boldsymbol{\theta}^n + \eta_g \sum_{i=1}^{M} \omega_i(n) \left[\boldsymbol{\theta}_i^{(\rho_i(n),k)} - \boldsymbol{\theta}^{\rho_i(n)} \right].$$

We retrieve $\boldsymbol{\theta}^{n,0} = \boldsymbol{\theta}^n$ and $\boldsymbol{\theta}^{n,K} = \boldsymbol{\theta}^{n+1,0} = \boldsymbol{\theta}^{n+1}$. The server has not access to $\boldsymbol{\theta}^{n,k}$ when $k \neq 0$ or $k \neq K$. Hence the name virtual for the model $\boldsymbol{\theta}^{n,k}$.

The difference between two consecutive global models in our virtual sequence depends on the sum of the differences between local models $\boldsymbol{\theta}_i^{\rho_i(n),k+1} - \boldsymbol{\theta}_i^{\rho_i(n),k} = -\eta_l \nabla \mathcal{L}_i(\boldsymbol{\theta}_i^{\rho_i(n),k}, \boldsymbol{\xi}_i)$, where $\boldsymbol{\xi}_i$ is a random batch of data samples of client i. Hence, we can rewrite the aggregation process as a GD step with

$$\boldsymbol{\theta}^{n,k+1} = \boldsymbol{\theta}^{n,k} - \eta_g \eta_l \sum_{i=1}^{M} \omega_i(n) \nabla \mathcal{L}_i(\boldsymbol{\theta}_i^{\rho_i(n),k}, \boldsymbol{\xi}_i).$$

2.5 Asynchronous FL as a Sequence of Optimization Problems

For the rest of this work, we define $q_i(n) \coloneqq \mathbb{E}\left[\omega_i(n)\right]$, the expected aggregation weight of client i at optimization round n. No assumption is made on $q_i(n)$ which can vary across optimization rounds. The expected clients contribution $\sum_{i=1}^{M} q_i(n) \boldsymbol{\Delta}_i(n)$ help minimizing the optimization problem $\mathcal{L}^n$ defined as

$$\mathcal{L}^n(\boldsymbol{\theta}) \coloneqq \sum_{i=1}^{M} q_i(n) \mathcal{L}_i(\boldsymbol{\theta}).$$

We denote by $\bar{\boldsymbol{\theta}}^n$ the optimum of $\mathcal{L}^n$ and by $\boldsymbol{\theta}^*$ the optimum of the optimization problem $\mathcal{L}$ defined in equation (2). Finally, we define by $q_i = \frac{1}{N} \sum_{n=0}^{N-1} q_i(n)$ the expected importance given to client i over the N server aggregations during the FL process, and by $\tilde{q}_i(n)$ the normalized expected importance $\tilde{q}_i(n) = q_i(n) / (\sum_{i=1}^{M} q_i(n))$. We define by $\bar{\mathcal{L}}$ the associated optimization problem

$$\bar{\mathcal{L}}(\boldsymbol{\theta}) \coloneqq \sum_{i=1}^{M} q_i \mathcal{L}_i(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=0}^{N-1} \mathcal{L}^n(\boldsymbol{\theta}), \tag{6}$$

and we denote by $\bar{\boldsymbol{\theta}}$ the associated optimum.

Finally, we introduce the following expected convergence residual, which quantifies the variance at the optimum in function of the relative clients importance $q_i(n)$

$$\Sigma \coloneqq \sum_{i=1}^{M} q_i \, \mathbb{E}_{\boldsymbol{\xi}_i} \left[\left\| \nabla \mathcal{L}_i(\bar{\boldsymbol{\theta}}, \boldsymbol{\xi}_i) \right\|^2 \right].$$

The convergence guarantees provided in this work (Section 3) are proportional to the expected convergence residual. Σ is positive and null only when clients have the same loss function and perform GD steps for local optimization.

2.6 Formalizing Heterogeneity across Clients

We assume the existence of $J \leq M$ different clients feature spaces $\mathcal{Z}_i$ and, without loss of generality, assume that the first J clients feature spaces are different. This formalism allows us to represent the heterogeneity of data distribution across clients. In DL problems, we have $J < M$ when the same dataset split is accessible to many clients. When clients share the same distribution, we assume that their optimization problem is equivalent. In this case, we call $F_j(\boldsymbol{\theta})$ their loss function with optimum $\boldsymbol{\theta}_j^*$. The federated problem of equation (2) can thus be formalized with respect to the discrepancy between the clients feature spaces $\mathcal{Z}_i$. To this end, we define Q_j the set of clients with the same feature space of client j, i.e. $Q_j \coloneqq \{i : \mathcal{Z}_i = \mathcal{Z}_j\}$. Each feature space as thus importance

	Client i	Sample distribution j
Importance	p_i	r_j
Stochastic aggregation weight	$\omega_i(n)$	-
Aggregation weight	$d_i(n)$	-
Expected agg. weight	$q_i(n)$	$s_j(n)$
Normalized expected agg. weight	$\tilde{q}_i(n)$	$\tilde{s}_j(n)$
Expected agg. weight over N rounds	q_i	s_j

Table 1: The different weights used to account for the importance of clients or data distributions at every optimization round and during the full FL process.

$r_j = \sum_{i \in Q_j} p_i$, and expected importance $s_j(n) = \sum_{i \in Q_j} q_i(n)$ such that

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{j=1}^{J} r_j F_j(\boldsymbol{\theta}) \text{ and } \mathcal{L}^n(\boldsymbol{\theta}) = \sum_{j=1}^{J} s_j(n) F_j(\boldsymbol{\theta}).$$

As for $\tilde{q}_i(n)$, we define $\tilde{s}_j(n) = s_j(n) / \sum_{i=1}^{M} s_j(n)$.

In Table 1, we summarize the different weights used to adapt the federated optimization problem to account respectively for heterogeneity in clients importance and data distributions across rounds.

3 Convergence of Federated Problem (2)

In this section, we prove the convergence of the optimization based on the stochastic aggregation scheme defined in equation (5), with implementation given in Algorithm 1. We first introduce in Section 3.1 the necessary assumptions and then prove with Theorem 1 the convergence of the sequence of optimized models (Section 3.2). We show in Section 3.3 the implications of Theorem 1 on the convergence of the federated problem (2), and propose sufficient conditions for the learnt model to be the associated optimum. Finally, with two additional assumptions, we propose in Section 3.4 simpler and practical sufficient conditions for FL convergence to the optimum of the federated problem (2).

3.1 Assumptions

We make the following assumptions regarding the Lipschitz smoothness and convexity of the clients local loss functions (Assumption 1 and 2), unbiased gradients estimators (Assumption 3), finite answering time for the clients (Assumption 4), and the clients aggregation weights (Assumption 5). Assumption 3 (Khaled et al., 2020) considers unbiased gradient estimators without assuming bounded variance, giving in turn more interpretable convergence bounds. Assumption 5 states that the covariance between two aggregation weights can be expressed as the product of their expected aggregation weight up to a positive multiplicative factor α. We show in Section 4 that Assumption 5 is not limiting as it is satisfied by all the standard FL optimization schemes considered in this work.

Assumption 1 (Smoothness)**.** *Clients local objective functions are L-Lipschitz smooth, that is,* $\forall i \in \{1, ..., n\},\ \|\nabla \mathcal{L}_i(\boldsymbol{x}) - \nabla \mathcal{L}_i(\boldsymbol{y})\| \leq L \|\boldsymbol{x} - \boldsymbol{y}\|$.

Assumption 2 (Convexity)**.** *Clients local objective functions are convex.*

Assumption 3 (Unbiased Gradient)**.** *Every client stochastic gradient* $g_i(\boldsymbol{x}) = \nabla \mathcal{L}_i(\boldsymbol{x}, \boldsymbol{z}_i)$ *of a model with parameters* $\boldsymbol{x}$ *evaluated on batch* $\boldsymbol{z}_i$ *is an unbiased estimator of the local gradient, i.e.* $\mathbb{E}_{\boldsymbol{z}_i}[g_i(\boldsymbol{x})] = \nabla \mathcal{L}_i(\boldsymbol{x})$.

Assumption 4 (Finite Answering Time)**.** *The server receives a client local work in at most* $\tau := \max_{i,n}(n - \rho_i(n))$ *optimization steps, which satisfy* $\mathbb{P}(\tau < \infty) = 1$.

Assumption 5. *There exists* $\alpha \in (0, 1)$ *such that* $\mathbb{E}[\omega_i(n)\omega_j(n)] = \alpha q_i(n) q_j(n)$.

3.2 Convergence of Algorithm 1

We first prove with Theorem 1 the convergence of Algorithm 1.

Theorem 1. *Under Assumptions 1 to 5, with* $\eta_l \leq 1/48KL\min\left(1, 1/3\rho^2\eta_g(\tau+1)\right)$*, we obtain the following convergence bound:*

$$\frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{K}\sum_{k=0}^{K-1}\left[\mathbb{E}\left[\mathcal{L}^n(\boldsymbol{\theta}^{n,k})\right] - \mathcal{L}^n(\bar{\boldsymbol{\theta}}^n)\right] \leq R(\{\mathcal{L}^n\}) + \epsilon_F + \epsilon_K + \epsilon_\alpha + \epsilon_\beta,$$

where

$$R(\{\mathcal{L}^n\}) = \frac{1}{N}\sum_{n=0}^{N-1}\left[\mathcal{L}^n(\bar{\boldsymbol{\theta}}) - \mathcal{L}^n(\bar{\boldsymbol{\theta}}^n)\right], \qquad \epsilon_F = \frac{1}{\tilde{\eta}KN}\left\|\boldsymbol{\theta}^0 - \bar{\boldsymbol{\theta}}\right\|^2,$$

$$\epsilon_K = \mathcal{O}\left(\eta_l^2(K-1)^2\left[R(\{\mathcal{L}^n\}) + \Sigma_1\right]\right), \quad \epsilon_\alpha = \mathcal{O}\left(\alpha\left[\tilde{\eta} + \tilde{\eta}^2K^2\tau^2\right]\left[R(\{\mathcal{L}^n\}) + \max q_i(n)\Sigma\right]\right),$$

$$\epsilon_\beta = \mathcal{O}\left(\beta\left[\tilde{\eta} + \tilde{\eta}^2K^2\tau^2\right]\left[R(\{\mathcal{L}^n\}) + \Sigma\right]\right), \qquad \tilde{\eta} = \eta_g\eta_l, \qquad \beta \coloneqq \max\{d_i(n) - \alpha q_i(n)\},$$

and $\mathcal{O}$ *accounts for numerical constants and the loss function Lipschitz smoothness* L*.*

Theorem 1 is proven in Appendix A. The convergence guarantee provided in Theorem 1 is composed of 5 terms: $R(\{\mathcal{L}^n\})$, ϵ_F, ϵ_K, ϵ_α, ϵ_β. In the following, we describe these terms and explain their origin in a given optimization scheme.

Optimized expected residual $R(\{\mathcal{L}^n\})$. The residual $R(\{\mathcal{L}^n\})$ quantifies the sensitivity of $\mathcal{L}^n$ between its optimum $\bar{\boldsymbol{\theta}}^n$ and the optimum $\bar{\boldsymbol{\theta}}$ of the overall expected minimized problem across optimization rounds $\tilde{\mathcal{L}}$. As such, the residual accounts for the heterogeneity in the history of optimized problems, and is minimized to 0 when the same optimization problem is minimized at every round n, i.e. $\mathcal{L}^n = \tilde{\mathcal{L}}$. This condition is always satisfied when clients have identical data distributions, but requires for the server to set properly every client aggregation weight $d_i(n)$ in function of the server waiting time policy Δt^n and the clients hardware capabilities T_i^n in the general case (Section 3.3 and 3.4).

Initialization quality ϵ_F. ϵ_F only depends of the quality of the initial model $\boldsymbol{\theta}^0$ through its distance with respect to the optimum $\bar{\boldsymbol{\theta}}$ of the overall expected minimized problem across optimization rounds $\tilde{\mathcal{L}}$. This convergence term can only be minimized by performing as many serial SGD steps KN.

Clients data heterogeneity ϵ_K. This term accounts for the disparity in the clients updated models, and is proportional to the clients amount of local work K (quadratically) and to the heterogeneity of their data distributions $\mathcal{Z}_i$ through Σ_1. When $K = 1$, every client perform its SGD on the same model, which reduces the server aggregation to a traditional centralized SGD. We retrieve $\epsilon_K = 0$.

Gradient delay τ **through** ϵ_α **and** ϵ_β. Decreasing the server time policy Δt^n allows faster optimization rounds but decreases a client's participation probability $\mathbb{P}(T_i^n \leq \Delta t^n)$ resulting in an increased maximum answering time τ. In turn, we note that ϵ_α and ϵ_β are quadratically proportional to the maximum amount of serial SGD $K\tau$. This latter terms quantifies the maximum amount of SGD integrated in the global model $\boldsymbol{\theta}^n$.

3.3 Sufficient Conditions for Minimizing the Federated Problem (2)

Theorem 1 provides convergence guarantees for the history of optimized models $\{\mathcal{L}^n\}$. Under the same assumptions of Theorem 1, we can provide convergence guarantees for the original FL problem $\mathcal{L}(\boldsymbol{\theta})$ (proof in Appendix B).

Theorem 2. *Under the same conditions of Theorem 1, we have*

$$\frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\boldsymbol{\theta}^{n,k})\right\|^2\right] \leq \mathcal{O}\left(R(\{\mathcal{L}^n\})\right) + P(\{\mathcal{L}^n\}) + U(\{\mathcal{L}^n\}) + \mathcal{O}\left(\epsilon_F\right) + \epsilon_K + \epsilon_\alpha + \epsilon_\beta,$$

where

$$P(\{\mathcal{L}^n\}) = \mathcal{O}\left(\frac{1}{N}\sum_{n=0}^{N-1}\chi_n^2\sum_{j\in W_n}\tilde{s}_j(n)\left[F_j(\bar{\boldsymbol{\theta}}^n) - F_j(\boldsymbol{\theta}_j^*)\right]\right),$$

$$U(\{\mathcal{L}^n\}) = \mathcal{O}\left(\frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{K}\sum_{k=0}^{K-1}\sum_{j\notin W_n} r_j\left[\mathbb{E}\left[F_j(\boldsymbol{\theta}^{n,k})\right] - F_j(\boldsymbol{\theta}_j^*)\right]\right),$$

$\chi_n^2 = \sum_{j\in W_n}(r_j - \tilde{s}_j(n))^2/\tilde{s}_j(n)$, *and* $W_n = \{j : s_j(n) > 0\}$.

Theorem 2 provides convergence guarantees for the optimization problem (2). We retrieve the components of the convergence bound of Theorem 1. The terms ϵ_F to ϵ_τ can be mitigated by choosing an appropriate local learning rate η_l, but the same cannot be said for $R(\{\mathcal{L}^n\}), P(\{\mathcal{L}^n\}), U(\{\mathcal{L}^n\})$. Behind these three quantities, Theorem 2 shows that proper expected representation of every dataset type is needed, i.e. $s_j(n) = r_j$. Indeed, if a client is poorly represented, i.e. $s_j(n) \neq r_j$, then $R(\{\mathcal{L}^n\}) > 0$ and $P(\{\mathcal{L}^n\}) > 0$, while if a client is not represented at all, i.e. $s_j(n) = 0$, then $U(\{\mathcal{L}^n\}) > 0$. Therefore, we propose, with Corollary 1, sufficient conditions for any FL optimization scheme satisfying Algorithm 1 to converge to the optimum of the federated problem (2).

Corollary 1. *Under the conditions of Theorem 1, if every client data distribution satisfies* $\tilde{s}_j(n) = r_j$*, the following convergence bound for optimization problem (*2*) can be obtained*

$$\frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{K}\sum_{k=0}^{K-1}\left[\mathbb{E}\left[\mathcal{L}(\boldsymbol{\theta}^{n,k})\right] - \mathcal{L}(\boldsymbol{\theta}^*)\right] \leq \epsilon_F + \epsilon_K + \epsilon_\alpha + \epsilon_\beta.$$

Proof. Follows directly. $\tilde{s}_j(n) = r_j$ implies $\chi_n^2 = 0$, $W_n = \emptyset$, $\mathcal{L}^n = q(n)\mathcal{L}$, and $\bar{\boldsymbol{\theta}}^n = \boldsymbol{\theta}^*$. □

These theoretical results provide relevant insights for different FL scenarios.

iid data distributions, $\mathcal{Z}_i = \mathcal{Z}$. Consistently with the extensive literature on synchronous and asynchronous distributed learning, when clients have data points sampled from the same data distribution, FL always converges to its optimum (Corollary 1). Indeed, $\tilde{s}_j(n) = r_j = 1$ regardless of which clients are participating, and what importance p_i or aggregation weight $d_i(n)$ a client is given.

non-iid data distributions. The convergence of FL to the optimum requires to optimize by considering every data distribution type fairly at every optimization round, i.e. $\tilde{s}_j(n) = r_j$ (Corollary 1). This condition is weaker than requiring to treat fairly every client at every optimization round, i.e. $q_i(n) = p_i$. Ideally, only one client per data type needs to have a non-zero participating probability, i.e. $\mathbb{P}(T_i^n \leq \Delta t^n) > 0$, and an appropriate $d_i(n)$ such that $\tilde{s}_j(n) = r_j$ is satisfied. In practice, knowing the clients data distribution is not possible. Therefore, ensuring FL convergence to its optimum requires at every optimization round $\tilde{q}_i(n) = p_i$ (Wang et al., 2020a).

We provide in Example 1 an illustration on these results based on quadratic loss functions to show that considering fairly data distributions is sufficient for an optimization scheme satisfying Algorithm 1 to converge to the optimum of the optimization problem (2), since $\tilde{s}_j(n) = r_j$ is satisfied at every optimization round, while $\tilde{q}_i(n) \neq p_i$ may not be satisfied.

Example 1. *Let us consider four clients with data distributions such that their loss can be expressed as* $\mathcal{L}_i(\boldsymbol{\theta}) = \frac{1}{2}\left\|\boldsymbol{\theta} - \boldsymbol{\theta}_i^*\right\|^2$ *with* $\boldsymbol{\theta}_1^* = \boldsymbol{\theta}_2^*$ *(*$\mathcal{Z}_1$*),* $\boldsymbol{\theta}_3^* = \boldsymbol{\theta}_4^*$ *(*$\mathcal{Z}_2$*), and identical client importance, i.e.* $p_i = 1/4$*. Therefore, each data type has identical importance, i.e.* $r_j = 1/2$*, and the optimum satisfies* $\boldsymbol{\theta}^* = \frac{1}{2}[\boldsymbol{\theta}_1^* + \boldsymbol{\theta}_3^*]$*. We consider that clients with odd index participate at odd optimization rounds while the ones with even index at even optimization rounds, i.e.* $q_1^{2n+1} = q_3^{2n+1} = q_2^{2n} = q_4^{2n} = 1/2$ *and* $q_1^{2n} = q_2^{2n} = q_3^{2n+1} = q_4^{2n+1} = 0$ *which gives* $\tilde{s}_1(n) = \tilde{s}_2(n) = 1/2$ *and* $\tilde{q}_i(n) = 0$ *or* $\tilde{q}_i(n) = 1/2$ *but not* $\tilde{q}_i(n) = 1/4$*. With* $\eta_g = 1$*, equation (*5*) can be rewritten as*

$$\boldsymbol{\theta}^{n+2} = \boldsymbol{\theta}^{n+1} + \frac{1}{2}\left[(\boldsymbol{\theta}_1^{n+1} - \boldsymbol{\theta}^n) + (\boldsymbol{\theta}_3^{n+1} - \boldsymbol{\theta}^n)\right]. \tag{7}$$

Clients update can be rewritten as $\boldsymbol{\theta}_i^{n+1} - \boldsymbol{\theta}^n = \phi(\boldsymbol{\theta}_i^* - \boldsymbol{\theta}^n)$*, where* $\phi = 1 - (1-\eta_l)^K$*. Equation (*7*) can thus be rewritten as*

$$\boldsymbol{\theta}^{n+2} - \boldsymbol{\theta}^{n+1} + \phi\boldsymbol{\theta}^n = \phi\boldsymbol{\theta}^*. \tag{8}$$

*Solving equation (*8*) proves FL asymptotic convergence to the optimum* $\boldsymbol{\theta}^*$*.*

3.4 Relaxed Sufficient Conditions for Minimizing the Federated Problem (2)

Theorem 2 holds for any client's update time T_i and optimization scheme satisfying Algorithm 1, and provides finite convergence guarantees for the optimization problem (2). Corollary 1 shows that for the asymptotic convergence of FL, data distribution types should be treated fairly in expectation, i.e. $\tilde{s}_j(n) = r_j$. This sufficient condition is not necessarily realistic, since the server cannot know the clients data distributions and participation time, and thus needs to give to every client an aggregation weight $d_i(n)$ such that $\tilde{q}_i(n) = p_i$ without knowing T_i.

In Example 1, we note that we have $\frac{1}{2}\left[q_i^{2n} + q_i^{2n+1}\right] = p_i$. Therefore, every client is given proper consideration every two optimization rounds. Based on Example 1, in Theorem 3 we provide weaker sufficient conditions than the ones of Corollary 1. To this end, we assume that clients are considered with identical importance across W optimization rounds (Assumption 6) and that clients gradients are bounded (Assumption 7).

Assumption 6 (Window)**.** $\exists W \geq 1$ *such that* $\forall s,\ \frac{1}{W}\sum_{n=sW}^{(s+1)W-1} q_i(n) = q_i$.

With Assumption 6, we assume that over a cycle of W aggregations, the sum of the clients expected aggregation weights $q_i(n)$ are constant. By definition of q_i, Assumption 6 is always satisfied with $W = N$. Also, by construction, we have $W \geq \tau$. We note that Assumption 6 is made on a series of windows of size W and not for any window of size W.

Assumption 7 (Bounded Gradients)**.** $\exists B > 0$ *such that* $\mathbb{E}\left[\|\nabla \mathcal{L}_i(\boldsymbol{x})\|\right] \leq B$.

Gradient clipping is a typical operation performed during the optimization of deep learning models to prevent exploding gradients. A pre-determined gradient threshold B is introduced, and gradients with norm exceeding this threshold are clipped to the norm B. Therefore, using Assumption 7 and the subadditivity of the norm, the distance between two consecutive global models can be bounded by

$$\mathbb{E}\left[\left\|\boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^{n}\right\|\right] \leq \eta_g \sum_{i=1}^{M} q_i(n)\, \mathbb{E}\left[\left\|\boldsymbol{\theta}_i^{\rho_i(n)+1} - \boldsymbol{\theta}^{\rho_i(n)}\right\|\right] \leq \eta_g \eta_l q(n) K B,$$

which, thanks to the convexity of the clients loss function and to the Cauchy Schwartz inequality, gives

$$\mathbb{E}\left[\mathcal{L}_i(\boldsymbol{\theta}^{n+1})\right] - \mathbb{E}\left[\mathcal{L}_i(\boldsymbol{\theta}^{n})\right] \leq \mathbb{E}\left[\langle \nabla \mathcal{L}_i(\boldsymbol{\theta}^{n+1}), \boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^{n}\rangle\right] \leq \eta_g \eta_l q(n) B^2 K. \tag{9}$$

Finally, using equation (9) and Assumption 6, the performance history on the original optimized problem can be bounded as follows

$$\sum_{n=sW}^{(s+1)W-1} \sum_{k=0}^{K-1} q_i\, \mathbb{E}\left[\mathcal{L}_i(\boldsymbol{\theta}^{(n,k)})\right] \leq \sum_{n=sW}^{(s+1)W-1} \sum_{k=0}^{K-1} q_i(n) \left[\mathbb{E}\left[\bar{\mathcal{L}}(\boldsymbol{\theta}^{(n,k)})\right] + \eta_g \eta_l K (W-1) B^2\right]. \tag{10}$$

Theorem 3. *Under the conditions of Theorem 1, Assumptions 6 and 7, and considering that W is a divider of N, we get the following convergence bound for the optimization problem (6):*

$$\frac{1}{N}\sum_{n=0}^{N-1} \frac{1}{K}\sum_{k=0}^{K-1} \left[\mathbb{E}\left[\bar{\mathcal{L}}(\boldsymbol{\theta}^{n,k})\right] - \bar{\mathcal{L}}(\bar{\boldsymbol{\theta}})\right] \leq \epsilon \coloneqq \epsilon_F + \epsilon_K + \epsilon_\alpha + \epsilon_\beta + \epsilon_W,$$

where $\epsilon_W = \mathcal{O}(\eta_g \eta_l (W-1) K)$. Furthermore, we obtain the following convergence guarantees for the federated problem (2):

$$\frac{1}{N}\sum_{n=0}^{N-1} \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left[\left\|\nabla \mathcal{L}(\boldsymbol{\theta}^{n,k})\right\|^2\right] \leq \epsilon + \mathcal{O}(\chi^2[\bar{\mathcal{L}}(\bar{\boldsymbol{\theta}}) - \sum_{j=1}^{J} s_j F_j(\boldsymbol{\theta}_j^*)]),$$

where $\chi^2 = \sum_{j=1}^{J} \frac{(r_j - \tilde{s}_j)^2}{\tilde{s}_j}$.

Proof.

$$\frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{K}\sum_{k=0}^{K-1}\left[\mathbb{E}\left[\bar{\mathcal{L}}(\boldsymbol{\theta}^{n,k})\right]-\bar{\mathcal{L}}(\bar{\boldsymbol{\theta}})\right]$$
$$\leq \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{K}\sum_{k=0}^{K-1} q_i(n)\left[\mathbb{E}\left[\mathcal{L}_i(\boldsymbol{\theta}^{n,k})\right]+\tilde{\eta}K(W-1)B^2\right]-\bar{\mathcal{L}}(\bar{\boldsymbol{\theta}})$$
$$\leq R(\{\mathcal{L}^n\})+\epsilon+\frac{1}{N}\sum_{n=0}^{N-1}\mathcal{L}^n(\bar{\boldsymbol{\theta}}^n)-\bar{\mathcal{L}}(\bar{\boldsymbol{\theta}})=\epsilon,$$

where we use equation (10) for the first inequality and Theorem 1 for the second inequality.

Finally, we can obtain convergence guarantees on the optimization problem (2) with Theorem 2 by considering the minimization of the optimization problem $\bar{\mathcal{L}}$. Therefore, the bound of Theorem 2 can be simplified noting that $\mathcal{L}^n = \bar{\mathcal{L}}$, $\bar{\boldsymbol{\theta}}^n = \bar{\boldsymbol{\theta}}$, $W_n = \emptyset$, $\chi_n^2 = \chi^2$, and by adding ϵ_W, which completes the proof. □

Theorem 3 shows that the condition $\tilde{s}_j = r_j$ is sufficient to minimize the optimization problem (2). In practice, for privacy concerns, clients may not want to share their data distribution with the server, and thus the relaxed sufficient condition becomes $\tilde{q}_i = p_i$. This condition is weaker than the one obtained with Corollary 1, at the detriment of a looser convergence bound including an additional asymptotic term ϵ_W linearly proportional to the window size W. Therefore, for a given learning application, the maximum local work delay τ and the window size W need to be considered when selecting an FL optimization scheme satisfying Algorithm 1. Also, the server needs to properly allocate clients aggregation weight $d_i(n)$ such that Assumption 6 is satisfied while keeping at a minimum the window size W. We note that W depends of the considered FL optimization scheme and clients hardware capabilities. Based on the results of Theorem 3, in the following section, we introduce FEDFIX, a novel asynchronous FL setting based on a waiting policy over fixed time windows Δt^n.

Finally, the following example illustrates a practical application of the condition $\tilde{q}_i = p_i$.

Example 2. *We consider two clients, $i = 1, 2$, with $\mathcal{L}_i(\boldsymbol{\theta}) = \frac{1}{2}\left\|\boldsymbol{\theta}-\boldsymbol{\theta}_i^*\right\|^2$ where clients have identical importance, i.e. $p_1 = p_2 = 1/2$. Client 1 contributes at even optimization rounds and Client 2 at odd ones, i.e. $q_1^{2n} = q_1$, $q_2^{2n+1} = q_2$, and $q_1^{2n+1} = q_2^{2n} = 0$. Hence, we have*

$$\boldsymbol{\theta}^n \xrightarrow{n\to\infty} \frac{q_1\boldsymbol{\theta}_1^* + q_2\boldsymbol{\theta}_2^*}{q_1+q_2},$$

which converges to the optimum of problem (2) if and only if $\frac{1}{2}\left[\tilde{q}_i^{2n}+\tilde{q}_i^{2n+1}\right] = p_i$ (Theorem 3).

The conditions of Corollary 1 and Theorem 3 are equivalent when $W = 1$, where we retrieve $\epsilon_W = 0$. They are also equivalent when clients have the same data distributions, and we retrieve $\tilde{s}_j = r_j = 1$ at every optimization round, which also implies that $W = 1$.

The convergence guarantee proposed in Theorem 3 depends on the window size W, and to the maximum amount of optimizations needed for a client to update its work τ. We provide sufficient conditions in Corollary 2 for the parameters W, and τ, such that an optimization scheme satisfying Algorithm 1 converges to the optimum of the optimization problem (2).

Corollary 2. *Let us assume there exists $a \geq 0$ and $b \geq 0$ such that $W = \mathcal{O}(N^a)$, $\tau = \mathcal{O}(N^b)$, and $\eta_l \propto N^{-c}$. The convergence bound of Theorem 3 asymptotically converges to 0 if*

$$W = o(N), \tau = o(N), \text{ and } \max(a,b) < c < 1$$

Proof. The bound of Theorem 3 converges to 0 if the following quantities also do: $\eta_l W$, $\frac{1}{\eta_l N}$, $\tau\eta_l$, η_l. We get the following conditions on a, b, and c: $-c+a<0$, $c-1<0$, $b-c<0$, $-c<0$, which completes the proof. □

By construction and definition of q_i, Assumption 6 is always satisfied with $W = N$. However, Corollary 2 shows that when $W = N$, no learning rate η_l can be chosen such that the learning

process converges to $\boldsymbol{\theta}^*$. Also, Corollary 2 shows that Assumption 4 can be relaxed. Indeed, Assumption 4 implies that $\tau = \mathcal{O}(1)$ and Corollary 2 shows that $\tau = o(N)$ is sufficient. We show in Section 4 that all the considered optimization schemes satisfy $\tau = \mathcal{O}(1)$ and $W = \mathcal{O}(1)$, and also depend of the clients hardware capabilities and amount of participating clients M.

4 Applications

In this section, we show that the formalism of Section 2 can be applied to a wide-range of optimization schemes, demonstrating the validity of the conclusions of Corollary 1 and Theorem 3 (Section 3). When clients have identical data distributions, the sufficient conditions of Corollary 1 are always satisfied (Section 3). In the heterogeneous case, these conditions can also (theoretically) be satisfied. It suffices that every client has a non-null participation probability, i.e. $\mathbb{P}(T_i^n \leq \Delta t^n) > 0$ such that there exists an appropriate $d_i(n)$ satisfying $\tilde{q}_i(n) = p_i$. Yet, in practice clients generally may not even know their update time distribution $\mathbb{P}(T_i^n)$ making the computation of $d_i(n)$ intractable. In what follows, we thus focus on Theorem 3 to obtain the close-form of ϵ, which only requires from the server to know the clients time τ_i.

Theorem 3 provides a close-form for the convergence bound ϵ of an optimization scheme in function of the amount of server aggregation rounds N. We first introduce in Section 4.1 our considerations for the clients hardware and data to instead express ϵ in function of the training time T. The quantity ϵ also depends on the optimization scheme time policy Δt^n through α, β and τ, and on the clients data heterogeneity through $R(\{\mathcal{L}^n\})$ and W. We provide their close-form for synchronous FEDAVG (Section 4.2), asynchronous FEDAVG (Section 4.3), and FEDFIX (Section 4.4), a novel asynchronous optimization scheme motivated by Section 3.4. Finally, in Section 4.5, we show that the conclusions drawn for synchronous/asynchronous FEDAVG and FEDFIX can also be extended to other distributed optimization schemes with delayed gradients. Of course, similar bounds can seamlessly be derived for centralized learning and client sampling, which we differ to Appendix C to focus on asynchronous FL in this section.

4.1 Heterogeneity of clients hardware and data distributions

Clients importance. We restrict our investigation to the case where clients have identical aggregation weights during the learning process, i.e. $d_i(n) = d_i$. We also consider identical client importance $p_i = 1/M$. We can therefore define the averaged optimum residual Σ defined as the average of the clients SGD evaluated on the global optimum, i.e.

$$\Sigma \coloneqq \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}_{\boldsymbol{\xi}_i} \left[\left\| \nabla \mathcal{L}_i(\boldsymbol{\theta}^*, \boldsymbol{\xi}_i) \right\|^2 \right].$$

When clients have identical data distributions, Σ can be simplified as $\Sigma = \mathbb{E}_{\boldsymbol{\xi}} \left[\left\| \nabla \mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\xi}) \right\|^2 \right]$, and $\Sigma = 0$ when clients perform GD. We note that in the DL and FL literature Σ is often simplified by assuming bounded variance of the stochastic gradients, i.e. $\Sigma \leq \sigma^2$, where σ^2 is the bounded variance of any client SG.

Clients computation time. In the rest of this work, we consider that clients guarantee reliable computation and communication, although with heterogeneous hardware capabilities, i.e. $\exists \tau_i \in \mathbb{R}$, s.t. $T_i = \tau_i$. Without loss of generality, we assume that clients are ordered by increasing τ_i, i.e. $\tau_i \leq \tau_{i+1}$, where the unit of τ_i is such that τ_i is an integer. In what follows, we provide the close form of d_i for all the considered optimization schemes. This derivation still holds for applications where clients have unreliable hardware capabilities that can be modeled as an exponential distribution, i.e. $T_i \sim \exp(\tau_i^{-1})$ which gives $\mathbb{E}[T_i] = \tau_i$.

Clients data distributions. Unless stated otherwise, we will consider the FL setting where each client has its unique data distribution. Therefore, clients have heterogeneous hardware and non-iid data distributions. The obtained results can be simplified for the DL setting where a dataset is made available to M processors. In this special case, clients have iid data distributions ($\mathcal{Z}_i = \mathcal{Z}_1$) , and identical computation times ($\tau_i = \tau_1$, $W = M$, and $\tau = M - 1$).

Learning rates. For sake of clarity, we ignore the server learning rate when expressing the convergence bounds ϵ, i.e. $\eta_g = 1$. Also, we consider a local learning rate η_l inversely proportional to the

	Sync. FEDAVG	Async. FEDAVG	FEDFIX
d_i	$= p_i$	$= \left[\sum_{i=1}^{M} \frac{1}{\tau_i}\right] \tau_i p_i$	$= \lceil \tau_i / \Delta t \rceil \, p_i$
N	T/τ_M	$\sum_{i=1}^{M} T/\tau_i$	$T/\Delta t$
Δt	$= \max T_i^n$	$= \min T_i^n$	$= \Delta t$
α	1	0	1
β	0	$\max d_i \leq \tau_m/\tau_0$	0
τ	0	$\Omega(M)$, $\mathcal{O}(M\tau_M/\tau_0)$	0, $\lfloor \tau_m/\tau_0 \rfloor$
W	1	$\Omega(M)$, $\mathcal{O}(M(\tau_M)^M)$	1, $M \lceil \tau_m/\tau_0 \rceil^M$
$R(\{\mathcal{L}^n\})$	$= 0$	$= \frac{1}{M}\sum_{i=1}^{M} [\mathcal{L}_i(\boldsymbol{\theta}^*) - \mathcal{L}_i(\boldsymbol{\theta}_i^*)]$	$\leq \frac{1}{M}\sum_{i=1}^{M} [\mathcal{L}_i(\boldsymbol{\theta}^*) - \mathcal{L}_i(\boldsymbol{\theta}_i^*)]$

Table 2: The different variables used to account for the importance of clients or data distributions at every optimization round and during the full FL process. For τ and W, we give two values which correspond to their respective lower and upper bound.

serial amount of SGD included in the global model, i.e. $\eta_l \propto 1/\sqrt{KN}$, consistently with the rest of the distributed optimization literature.

We propose Table 2 to summarize the close form or bounds of the different parameters used in Section 3.

4.2 FEDAVG, Synchronous Federated Learning

As described for FEDAVG in Section 2.3, at every optimization round, the server sends to the clients the current global model to perform K SGD steps on their own data before returning the resulting model to the server. Once every client performs its local work, the new global model is created as the weighted average of the clients contribution. The time required for an optimization step is therefore the one of the slowest client ($\Delta t^n = \max_i(T_i^n)$), and every client is considered ($\mathbb{P}(T_i^n \leq \Delta t^n) = 1$). Hence, $\alpha = 1$, $\beta = 0$, and setting $d_i = p_i$ is sufficient to satisfy the conditions of Corollary 1 (and thus the ones of Theorem 3) ensuring that FL converges to its optimum (Wang et al., 2020a). The term ϵ then reduces to

$$\epsilon_{\text{FEDAVG}} = \frac{1}{\sqrt{KN}} \left\| \boldsymbol{\theta}^0 - \boldsymbol{\theta}^* \right\|^2 + \mathcal{O}\left(\frac{K-1}{N}\Sigma\right) + \mathcal{O}\left(\frac{1}{\sqrt{KN}}\frac{1}{M}\Sigma\right). \tag{11}$$

The second element of equation (11) accounts for the clients update disparity through their amount of local work K between two server aggregations, and is proportional to the SG variance Σ. The third element benefits of the distributed computation by being proportional to $1/M$. Equation (11) is consistent with literature on convex distributed optimization with FEDAVG including Wang et al. (2020a); Khaled et al. (2020).

4.3 Asynchronous FEDAVG

With FEDAVG, every client waits for the slowest one to perform its local work, and cannot contribute to the learning process during this waiting time. To remove this bottleneck, with asynchronous FEDAVG, the server creates a new global model whenever it receives a client contribution before sending it back to this client. For in depth discussion of Asynchronous FEDAVG, please refer to Xu et al. (2021).

With asynchronous FEDAVG, clients always compute their local work but each on a different global model, giving $\Delta t^n = \min_i T_i^n$, $\alpha = 0$, and $\beta = \max_i d_i$. In addition, while the slowest client updates its local work, other clients performs a fix amount of updates (up to $\lceil \tau_M/\tau_i \rceil$). By scaling this amount of updates by the amount of clients sending updates to the server, we have

$$\tau = \mathcal{O}\left(\frac{\tau_M}{\tau_0}(M-1)\right).$$

We define $lcm(\{x_i\})$ the function returning the least common multiplier of the set of integers $\{x_i\}$. Hence, after every $\nu := lcm(\{\tau_i\})$ time, each client has performed ν/τ_i optimization rounds and

the cycle of clients update repeats itself. Thus, the smallest window W satisfies

$$W = \sum_{i=1}^{M} \nu/\tau_i.$$

By construction, $\nu \geq \tau_M$ and thus $W = \Omega(M)$, with $W = M$ when clients have homogeneous hardware ($\tau_M = \tau_0$). In the worse case, every τ_i is a prime number, and we have $\nu/\tau_i \leq (\tau_M)^{M-1}$, which gives $W = \mathcal{O}(M\,(\tau_M)^{M-1})$. In a cycle of W optimization rounds, every client participates ν/τ_i times to the creation of a new global model. Therefore, we have $q_i(n) = d_i$ for the ν/τ_i participation of client i, and $q_i(n) = 0$ otherwise. Hence, the sufficient conditions of Theorem 3 are satisfied when

$$q_i = \frac{1}{W} \sum_{n=kW}^{(k+1)W-1} q_i(n) = \frac{1}{\sum_{i=1}^{M} \nu/\tau_i} \frac{\nu}{\tau_i} d_i = p_i \Rightarrow d_i = \left[\sum_{i=1}^{M} \frac{1}{\tau_i} \right] \tau_i p_i. \tag{12}$$

The client weight calculated in equation (12) is constant and only depends on the client importance p_i (set and thus known by the server), and on the clients computation time τ_i (eventually estimated by the server after some clients updates). The condition on d_i can be further simplified by accounting for the server learning rate η_g. Coupling equation (5) with equation (12) gives $\eta_g d_i \propto \tau_i p_i$, which is sufficient to guarantee the convergence of asynchronous FL to its optimum. Finally, by bounding τ_i, we also have $\beta = \max_i d_i \leq \tau_M/\tau_0$, bounded the hardware computation time heterogeneity.

The disparity between the optimized objectives $R(\{\mathcal{L}^n\})$ at different optimization rounds also slows down the learning process. Indeed, at every optimization round, only a single client can participate with probability 1. As such, we have $\mathcal{L}^n = d_i \mathcal{L}_i$ which, thanks to the assumption $p_i = 1/M$, yields

$$R(\{\mathcal{L}^n\}) = \frac{1}{M} \sum_{i=1}^{M} \left[\mathcal{L}_i(\boldsymbol{\theta}^*) - \mathcal{L}_i(\boldsymbol{\theta}_i^*) \right].$$

Finally, we simplify the close-form of ϵ (Theorem 3) for asynchronous FEDAVG to get

$$\begin{aligned} \epsilon_{Async} = & \frac{1}{\sqrt{KN}} \left\| \boldsymbol{\theta}^0 - \boldsymbol{\theta}^* \right\|^2 + \mathcal{O}\left(\frac{K-1}{N} \Sigma \right) + \mathcal{O}\left(\frac{\tau_M}{\tau_0} \frac{1}{\sqrt{KN}} \left[R(\{\mathcal{L}^n\}) + \Sigma \right] \right) \\ & + \mathcal{O}\left(\left(\frac{\tau_M}{\tau_0} \right)^3 \frac{K}{N} M^2 \left[R(\{\mathcal{L}^n\}) + \Sigma \right] \right) + \mathcal{O}\left(\frac{1}{\sqrt{KN}} (W-1) \right). \end{aligned} \tag{13}$$

With equation (13), we can compare synchronous and asynchronous FEDAVG. The first and second asymptotic terms are identical for the two learning algorithms, while the third asymptotic term is scaled by the hardware characteristics τ_M/τ_0 instead of $1/M$ in FEDAVG, with the addition of a non null residual $R(\{\mathcal{L}^n\})$ for asynchronous FEDAVG. However, the fourth and fifth term are unique to asynchronous FEDAVG, and explains why its convergence gets more challenging as the amount of clients M increases. The impact of the hardware heterogeneity is also identified through the importance of τ_M/τ_0 in the third term. With no surprise, for a given optimization round, synchronous FEDAVG outperforms its asynchronous counterpart. However, in T time, the server performs

$$N = \sum_{i=1}^{M} T/\tau_i$$

aggregations with asynchronous FEDAVG against T/τ_M for synchronous FEDAVG. With asynchronous FEDAVG, the server thus performs at least M times more aggregations than with synchronous FEDAVG. As a result, the first two terms of equation (13), which are proportional to how good the initial model is $\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|$, decrease faster with asynchronous FEDAVG at the detriment of an higher convergence residual coming for the two last terms.

Comparison with asynchronous DL and FEDAVG literature. The convergence rates obtained in the convex distributed optimization literature relies on additional assumptions to ours, with which we retrieve their proposed convergence rate. To the best of our knowledge, only Zinkevich et al. (2009) considers non-iid data distributions for the clients. When assuming $W = \mathcal{O}(\tau)$ and $\eta_l \propto 1/\sqrt{\tau N}$, we retrieve a convergence rate $\sqrt{\tau/N}$.

We also match convergence rates for literature with iid client data distributions and $K = 1$. With $M = \mathcal{O}(\sqrt{N})$, then we have $\mathcal{O}(1/\sqrt{N})$ (Agarwal and Duchi, 2011; Lian et al., 2015). When $\eta_l = \mathcal{O}(1/\tau\sqrt{KN})$, we retrieve $\tau/N + 1/\sqrt{N}$ (Stich and Karimireddy, 2020; Stich et al., 2021).

4.4 FEDFIX

The analysis of asynchronous FEDAVG (Section 4.3) and its comparison with synchronous FEDAVG (Section 4.2), shows that asynchronous FEDAVG is not scalable to large cohort of clients. We thus propose FEDFIX combining the strong points of synchronous and asynchronous FEDAVG, where the server creates the new global model at a fixed time t^n with the contributions received since t^{n-1}. Therefore, the server does not wait for every client, contrarily to synchronous FEDAVG, and considers more than one client per aggregation to have more stable aggregations, contrarily to asynchronous FEDAVG. We provide in Figure 1 an illustration of FEDFIX with two clients.

With FEDFIX, an iteration time $\Delta t^n = t^{n+1} - t^n$ is decided by the server and is independent from the clients remaining update time T_i^n. For sake of convenience, we further assume that the time between optimization rounds is identical, i.e. $\Delta t^n = \Delta t$, but the following results can be derived for other fixed time policies $\{\Delta t^n\}$. Therefore, T_i^n and T_j^n are independent, and so are ω_i and ω_j, which gives $\alpha = 1$ and $\beta = 0$.

Every client sends an update to the server in $N_i' = \lceil T_i/\Delta t \rceil$ optimization steps. Contrarily to asynchronous FEDAVG, we thus have $\tau = \lceil \tau_m/\Delta t \rceil = \mathcal{O}(1)$, which is independent from the amount of participating clients M. In this case, the smallest window W satisfies $W = lcm(\{N_i'\})$, and clients update W/N_i' times their work to the server during the window W. Therefore, satisfying the conditions of Theorem 3 requires

$$d_i = \left\lceil \frac{\tau_i}{\Delta t} \right\rceil p_i. \tag{14}$$

With equation (14), we can notice the relationship between FEDFIX and synchronous or asynchronous FEDAVG. When $\Delta t \geq \tau_i$, client i participates to every optimization round and is thus considered synchronously, which gives $d_i = p_i$. When $\Delta_t \geq \tau_M$, we retrieve synchronous FL and $d_i = p_i$ for every client. On the contrary, for asynchronous FL, when $\Delta t \ll \tau_i$, we obtain $\lceil \tau_i/\Delta t \rceil \approx \tau_i/\Delta t$ and we retrieve $\eta_g d_i = \eta_g \lceil \tau_i/\Delta t \rceil p_i \propto \tau_i p_i$.

Regarding the disparity between the local objectives $R\{\mathcal{L}^n\}$, we know that a client participates to an optimization round with $q_i(n) = d_i$. We thus have $\mathcal{L}^n = \sum_{i \in S_n} d_i \mathcal{L}_i$, where S_n is the set of the participating clients at optimization step n. Considering that $\mathcal{L}^n(\bar{\boldsymbol{\theta}}^n) \geq \sum_{i \in S_n} d_i \mathcal{L}_i(\boldsymbol{\theta}_i^*)$, the close form of FEDFIX is bounded by the one of of asynchronous FEDAVG, i.e.

$$R(\{\mathcal{L}^n\}) \leq \frac{1}{M} \sum_{i=1}^{M} \left[\mathcal{L}_i(\boldsymbol{\theta}^*) - \mathcal{L}_i(\boldsymbol{\theta}_i^*) \right].$$

Finally, we simplify the close-form of ϵ (Theorem 3) for FEDFIX to get

$$\begin{aligned} \epsilon_{\text{FEDFIX}} = & \frac{1}{\sqrt{KN}} \mathbb{E}\left[\left\| \boldsymbol{\theta}^0 - \boldsymbol{x} \right\|^2 \right] + \mathcal{O}\left(\frac{K-1}{N} \left[R(\{\mathcal{L}^n\}) + \Sigma \right] \right) \\ & + \mathcal{O}\left(\left[\frac{1}{\sqrt{KN}} + \frac{K}{N} \left\lceil \frac{\tau_m}{\Delta t} \right\rceil^2 \right] \left[R(\{\mathcal{L}^n\}) + \left\lceil \frac{\tau_m}{\Delta t} \right\rceil \frac{1}{M} \Sigma \right] \right) + \mathcal{O}\left(\frac{1}{\sqrt{KN}} (W-1) \right) \end{aligned} \tag{15}$$

The first two elements of equation (15) are identical for FEDFIX, synchronous and asynchronous FEDAVG. However, thanks to lower values for the different variables (cf Table 2), the last two asymptotic terms of the convergence bound are smaller for FEDFIX than for asynchronous FEDAVG, equation (15). Similarly, these two terms are larger with FEDFIX than with synchronous FEDAVG. The hardware heterogeneity and the amount of participating clients still impacts the convergence bound through $\lceil \tau_M/\Delta t \rceil$ and W, but can be mitigated with proper selection of Δt. Therefore, after N optimization rounds, synchronous FEDAVG outperforms FEDFIX which outperforms in turn asynchronous FEDAVG. However, in T time, the server performs $N = T/\Delta t$ aggregations with FEDFIX against T/τ_M for synchronous FEDAVG. With asynchronous FEDAVG, the server thus performs at least $\tau_M/\Delta t$ times more aggregations than with synchronous FEDAVG. Overall, Δt can be considered as the level of asynchronicity given to Algorithm 1, with FEDAVG when $\Delta t = \tau_M$ and asynchronous FEDAVG when $\Delta t \geq \tau_M$.

In the DL case, clients have identical computation time ($\tau_1 = \tau_m$), and we retrieve the convergence bound of synchronous FEDAVG.

In addition, we can increase the waiting time for the clients update, since the learning process converges and gets closer to the optimum of optimization problem (2), to reach a behavior similar to the one of synchronous FL. Indeed, for Theorem 3 to hold, we only need the same optimization time rounds Δt over a window W

4.5 Generalization

Coupled with the theoretical method developed in Wang et al. (2020a), the proof of Theorem 1 can account for FL regularization methods (Li et al., 2020a, 2019; Acar et al., 2021), other SGD solvers (Kingma and Ba, 2015; Ward et al., 2019; Li and Orabona, 2019; Yu et al., 2019a,b; Haddadpour et al., 2019), and/or gradient compression/quantization (Reisizadeh et al., 2020; Basu et al., 2019; Wang et al., 2018; Koloskova* et al., 2020).

We also note that Theorem 3 can be applied to other distributed optimization schemes using different waiting time policy Δt^n. With FEDBUFF (Nguyen et al., 2021), the server waits for m client updates to create the new global model. The server then communicates to these clients the new global model, while the other clients keep performing local work on the global model they received.

In this section, the sufficient conditions of Theorem 3 regarding the expected aggregation weights $q_i(n)$ were applied to obtain proper aggregation weight d_i. We keep identical clients local learning rate η_l and amount of local work K. We could instead get the close-form of a client specific learning rate $\eta_l(i)$ or amount of local work $K(i)$ using the gradient formalization of Wang et al. (2020a).

5 Experiments

In this section, we experimentally demonstrate the theoretical claims of Section 3 and 4. We first introduce the information needed to understand how the experiments are run in Section 5.1. Finally, in Section 5.2, we provide our experiments and their interpretation.

5.1 Experimental Setting

We introduce in this subsection the dataset and the predictive models used for federated optimization, the hardware scenarios proposed to simulate hardware heterogeneity, the clients aggregation weights strategies, and how the different hyperparameters are set.

Optimization Problems. We consider learning a predictive model for optimization problem (2) where clients have identical importance ($p_i = 1/M$) based on the following datasets with their associated learning scenarios.

- **MNIST** (Lecun et al., 1998) and **MNIST-shard**. MNIST is a dataset of 28x28 pixel grayscale images of handwritten single digits between 0 and 9 composed of 60 000 training and 10 000 testing samples split equally among the clients. We use a logistic regression to predict the images class. Clients are randomly allocated digits to match their number of samples. With MNIST-shard, we split instead data samples among clients using a Dirichlet distribution of parameter 0.1, i.e. $Dir(0.1)$. Therefore, with MNIST and MNIST-shard, we evaluate our theory on a convex optimization problem.
- **CIFAR-10** (Krizhevsky, 2009). The dataset consists of 10 classes of 32x32 images with three RGB channels. There are 50000 training and 10000 testing examples. The model architecture was taken from (McMahan et al., 2017) which consists of two convolutional layers and a linear transformation layer to produce logits. Clients get the same amount of samples but their percentage for each class vary and is determined with a Dirichlet distribution of parameter 0.1, i.e. $Dir(0.1)$ (Harry Hsu et al., 2019).
- **Shakespeare** (Caldas et al., 2018). We study a LSTM model for next character prediction on the dataset of *The Complete Works of William Shakespeare*. The model architecture is composed of a two-layer LSTM classifier containing 100 hidden units with an 8 dimensional embedding layer taken from (Li et al., 2020a). The model takes as an input a sequence of 80 characters, embeds each of the characters into a learned 8-dimensional space

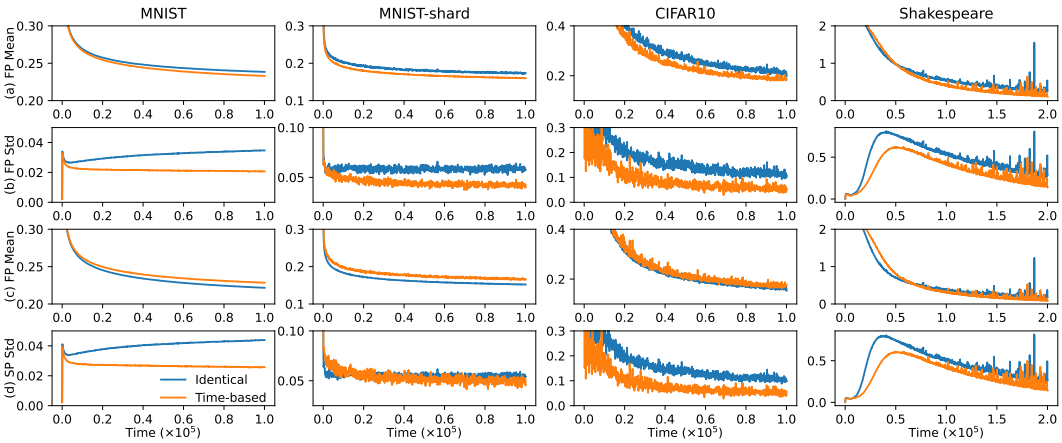

Figure 2: We consider the loss evolution over time of federated problem (2) (FP) and surrogate problem (6) (SP) for MNIST, MNIST-shard, CIFAR10, and Shakespeare; and the respective standard deviation of the loss over clients in (b) and (d). We consider $M = 10$ for a time scenario $F80$ with $K = 1$.

and outputs one character per training sample after 2 LSTM layers and a fully connected one.

Hardware Scenarios. In the following experimental scenarios, clients computation time are obtained according to the time policy FX. We consider that clients have fixed update times that can be up to X% slower than the fastest client. Clients computation time are uniformly distributed from the upper to the lower bound. Clients have thus identical hardware with $F0$. To simulate heterogeneous clients hardware, we consider the time scenario $F80$.

Clients Aggregation Weights. To compare asynchronous FL with and without the close-form of d_i provided in Section 4, we introduce IDENTICAL where $d_i = 1$ for every client regardless of the time scenario FX, and TIME-BASED where d_i satisfies equation (12) derived in Section 4.

Hyperparameters. Unless specified otherwise, we consider a global learning rate $\eta_g = 1$. We finetune the local learning rate η_l with values ranging from 10^{-5} to 1. We consider a batch size $B = 64$ for every dataset. We report mean and standard deviation on 5 random seeds. Every comparison of IDENTICAL with TIME-BASED is done using the same local learning rate. We give an advantage to IDENTICAL by finetuning the learning rate on this clients aggregation weight scenario.

5.2 Experimental Results

We experimentally show that asynchronous FL has better performances with TIME-BASED than with IDENTICAL, and thus we demonstrate the correctness of Theorem 3 with Figure 2 in Section 5.2.1. We however show in Figure 3 that TIME-BASED is less stable than IDENTICAL to the change in amount of local work K. Finally, we compare synchronous FEDAVG and asynchronous FEDAVG in Figure 4.

5.2.1 Impact of the Clients Aggregation Weights on Asynchronous FEDAVG

Figure 2(a) experimentally shows the interest of coupling asynchronous FL with TIME-BASED instead of IDENTICAL for different applications (MNIST, MNIST-shard, CIFAR10, and Shakespeare). The learnt model with TIME-BASED has better minima on the federated problem (2). In addition, Figure 2(b) shows that losses across clients are more homogeneous with TIME-BASED, resulting in generally lower standard deviations.

Focusing on MNIST and MNIST-shard, we see the impact of data heterogeneity on the learnt model performances. With IDENTICAL, asynchronous FL converges to a suboptimum point and the differences between the learnt model losses is twice as large for MNIST-shard than for MNIST, Figure 2(a). Figure 2(b) shows a similar result concerning the clients loss heterogeneity. Therefore, data

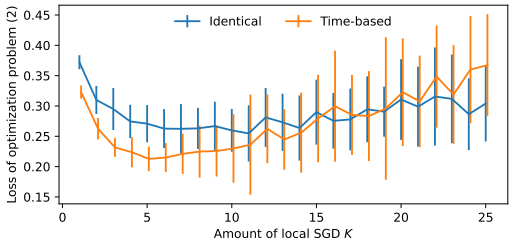

Figure 3: Evolution of the loss of optimization problem (2) for CIFAR10 with $M = 10$, time scenario $F80$, $\eta_g = 1$, $\eta_l = 0.005$, and varying amount of local work K ranging from 1 to 25.

heterogeneity degrades the suboptimum loss and cannot be ignored in asynchronous FL applications. Indeed, IDENTICAL and TIME-BASED curves are significantly different even for the simplest application on MNIST, where the dataset is uniformly distributed across $M = 10$ clients. Hence, the assumption of identical data distributions should generally not be made and the aggregation scheme TIME-BASED should be used instead for any asynchronous FL (or DL).

With Figure 2(c), we can also appreciate the performances of the learning procedure on the surrogate problem (6) based on the clients computation times T_i. Due to clients hardware heterogeneity, in the scenario $F80$, clients communicate with the server up to 5 more times than the slowest one. TIME-BASED balances this amount of updates disparity across clients. As a result, IDENTICAL has better performances than TIME-BASED on the surrogate problem (6) for MNIST, MNIST-shard, and CIFAR10, while for Shakespeare, TIME-BASED shows better performances. We attribute this fact to the depth of the predictive model enabling overfitting. As such, TIME-BASED outperforms IDENTICAL on the federated problem (2), Figure 2(a), while preventing catastrophic forgetting and thus leading to better losses on fast clients, Figure 2(c). Finally, Figure 2(d) shows that in addition, the weighted standard deviation of the surrogate loss is always worse for IDENTICAL.

5.2.2 Impact of the amount of local work on asynchronous FL convergence

With Figure 3, we consider the impact of the amount of local work on the convergence speed of Asynchronous FL with CIFAR10, time scenario $F80$, and $M = 10$ clients. For every simulation, we consider $\eta_l = 0.0005$, the optimal local learning rate for $K = 1$ with IDENTICAL. The server aggregates the clients contribution over $T = 25000$ units of time, and we report in Figure 3 mean and standard deviation over the 5% last server optimization rounds of the loss of the optimization problem (2).

Figure 3 shows that increasing the amount of local work K first decreases the loss of optimization problem (2) evaluated on the expected learnt model before increasing it. This point justifies asking to clients to perform $K > 1$ SGDs but requires proper finetuning of the amount of local work K. In particular, we notice that the variance strictly increases with K, which shows that the learning procedure becomes less stable.

These behaviors of asynchronous FL are due to the disparity between clients contributions induced by clients data heterogeneity, and can only be mitigated with a smaller global learning rate η_g, local learning rate η_l, and amount of local work K. Figure 3 shows that TIME-BASED is however more sensitive to an increase in amount of local work K than IDENTICAL. TIME-BASED is associated with higher variance after $K = 8$, and higher mean after $K = 15$, while IDENTICAL has very similar mean and standard deviation from $K = 8$ to $K = 16$.

This difference in convergence behavior is due to the FL aggregation scheme (5), and to the difference between the clients aggregation weights d_i for IDENTICAL and TIME-BASED. We have indeed $d_i = 1$ for every client with IDENTICAL, while with TIME-BASED fast clients are given lower aggregation weights $d_i < 1$, and slow clients higher weights $d_i > 1$. Therefore, whenever a slow client contributes, the new global model is more perturbed with TIME-BASED than with IDENTICAL, which makes TIME-BASED convergence speed more sensitive to a small change in the choice of K and other hyperparameters. This point can also be noticed in Figure 2(a) where IDENTICAL first converges faster than TIME-BASED. Still, IDENTICAL converges to a suboptimum.

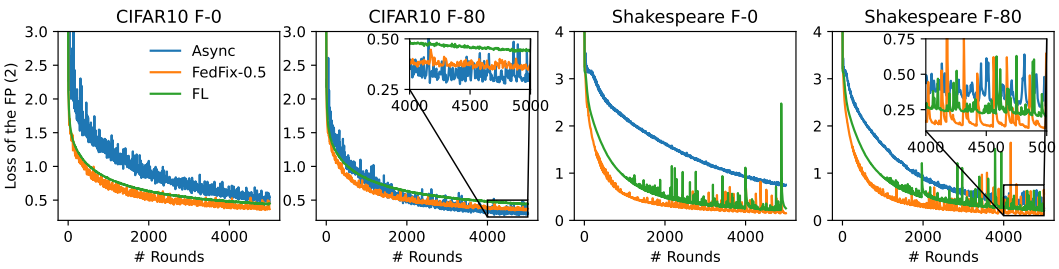

Figure 4: Evolution of federated problem (2) loss for CIFAR10 and Shakespeare and time scenario $F0$ and $F80$, with $M = 20$ (a) and $M = 50$ (b). We consider $\eta_g = 1$, $K = 10$, and $\Delta t = 0.5$ for FEDFIX.

IDENTICAL outperforms TIME-BASED in Figure 3 because we consider $\eta_l = 0.0005$. We note that considering our time budget T, doing a grid search for η_l would always provide a learnt model with better optimization loss for TIME-BASED.

5.2.3 Partial Asynchronicity with FEDFIX

The theory derived in Section 3 can be applied to asynchronous FL but also synchronous FL, FEDAVG, and other asynchronous FL schemes like FEDFIX (Section 4). We show with Figure 4 that allowing asynchronicity does not necessarily provide faster learning processes, e.g. comparison between synchronous and asynchronous FEDAVG above, but FEDFIX always outperforms FEDAVG by balancing convergence speed and stability.

With a small enough learning rate η_l, asynchronous FEDAVG outperforms FEDFIX, which outperforms synchronous FL (see Figure 5 and 7 in Appendix D). Indeed, in this case, global models change slowly and we can consider that the server receives contributions with no gradient delay. As such, the learning procedure including the most serial contributions in the global model is the fastest. However, in the other cases, the learning rate η_l does not mitigate the discrepancy between clients update, which slows down convergence for asynchronous FL, and can even prevent it.

Identifying the fastest optimization scheme must be done by comparing optimization schemes based on their best local learning rate η_l (Figure 4). Synchronous FL always outperforms asynchronous FL when clients have heterogeneous hardware ($F0$). Even with heterogeneous hardware ($F80$), synchronous FL can outperform asynchronous FL (Shakespeare). Indeed, the server needs to reduce its amount of aggregations to balance convergence speed and convergence stability. We see that FEDFIX-0.5 provides this trade-off and outperforms synchronous FL in every scenario.

We note that, even for synchronous FL, FL convergence is not monotonous. Indeed, for synchronous FL to have a better convergence speed than asynchronous FL, the server needs to consider a high local learning rate leading to convergence instability. Figure 4 shows this instability for Shakespeare and $t > 4000$, and Figure 5 to 7 in Appendix D provides the evolution of this instability as the learning rate η_l increases.

We note that even when clients have homogeneous hardware ($F0$), FEDFIX outperforms synchronous FL. This can be explained by the close-form of FEDFIX weights d_i, equation (14), which accounts for server aggregations where no client participates. As a result, FEDFIX-0.5 behaves as asynchronous FL but with an higher server learning rate $\eta_g = 2$ which provides faster convergence.

6 Discussion

This work introduces equation (5) which generalizes the expression of FEDAVG aggregation scheme by introducing stochastic aggregation weights $\omega_i(n)$ to account for asynchronous client updates. With a simple assumption for clients aggregation weights covariance, Assumption 5, we prove the convergence of FL schemes satisfying equation (5). A similar aggregation scheme has been derived in Fraboni et al. (2022) for unbiased client sampling, which this work generalizes. In addition, we show that aggregation scheme (5) and Assumption 5 are satisfied by asynchronous FL, FEDFIX, and FEDBUFF, Section 4. Finally, we assume fixed clients update time T_i such that we can consider

$d_i(n) = d_i$, and give in Section 4 its close-form to ensure any FL optimization scheme converges to the optimum of problem (2). Our work remains relevant for applications with $d_i(n) = d_i$ but we let the specific derivations to the reader.

This work shows theoretically and experimentally that asynchronous FEDAVG does not always outperform its synchronous counterpart. By creating the new global model with the contribution of only one client, asynchronous FEDAVG convergence speed is very sensitive to the choice of learning rate and amount of local work K. These two hyperparameters need to be fine-tuned to properly balance convergence speed and stability. Due to the hardware constraints inherent to the FL setting, fine-tuning is a challenging step for FL and is not necessarily feasible. Therefore, we proposed FEDFIX, an FL algorithm where the server, after a fixed amount of time, creates the new global model with the contribution of all the participating clients. We prove the convergence of FEDFIX with our theoretical framework, and experimentally demonstrate its improvement over FEDAVG in all the considered scenarios.

A Proof of Theorem 1

We first provide in Section A.1 the basic inequalities used in our proofs, and in Section A.2 the basic notations used to provide clearer proofs.

A.1 Basic Inequalities

We provide the following basic inequalities used in our proofs.

Let us consider f a L-Lipschitz smooth and convex function with optimum $\boldsymbol{x}^*$. For any vector $\boldsymbol{x}$ and $\boldsymbol{y}$, we have

$$\|\nabla f(\boldsymbol{x})\|^2 \leq 2L[f(\boldsymbol{x}) - f(\boldsymbol{x}^*)], \text{ and } \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|^2 \leq L^2 \|\boldsymbol{x} - \boldsymbol{y}\|^2 .$$

Let us consider g a convex function and d vectors $\{\boldsymbol{x}_k\}$ each with importance p_k such that $\sum_{k=1}^d p_k = 1$. With Jensen inequality, we have

$$g(\sum_{k=1}^{d} p_k \boldsymbol{x}_k) \leq \sum_{k=1}^{d} p_k g(\boldsymbol{x}_k).$$

Let us consider the random variable X, we have

$$\mathbb{E}\left[\|X - \mathbb{E}[X]\|^2\right] \leq \mathbb{E}\left[\|X\|^2\right].$$

A.2 Additional Notation

In Table 1, we synthesize the different random variables associated to the clients aggregation weights. In Table 3, we synthesize the remaining random variables.

We introduce the following notations to provide clear and compact proofs. Whenever considering a function $f(n,k)$, we define $f(n) \coloneqq 1/K \sum_{k=0}^{K-1} f(n,k)$, and $\bar{f}(N) \coloneqq 1/N \sum_{n=0}^{N-1} f(n)$. We introduce the following quantities

$$D(\boldsymbol{x},n,k) \coloneqq \mathbb{E}\left[\langle \sum_{i=1}^{M} q_i(n) \nabla \mathcal{L}_i(\boldsymbol{\theta}_i^{\rho_i(n),k}), \boldsymbol{\theta}^{n,k} - \boldsymbol{x}\rangle\right], \quad Q(n) \coloneqq \mathbb{E}\left[\left\|\boldsymbol{\theta}^{n+1,0} - \boldsymbol{\theta}^{n,0}\right\|^2\right],$$

$$R(n,k) \coloneqq \mathbb{E}\left[\left\|\sum_{i=1}^{M} \tilde{q}_i(n) g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})\right\|^2\right], \qquad S(n,k) \coloneqq \sum_{i=1}^{M} \tilde{q}_i(n)\, \mathbb{E}\left[\left\|g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})\right\|^2\right],$$

$$Z(n,k) = \mathcal{L}^n(\boldsymbol{\theta}^{n,k}) - \mathcal{L}^n(\bar{\boldsymbol{\theta}}^n), \qquad \Delta(n,k) \coloneqq \mathbb{E}\left[\left\|\boldsymbol{\theta}^{n,k+1} - \boldsymbol{x}\right\|^2\right] - \mathbb{E}\left[\left\|\boldsymbol{\theta}^{n,k} - \boldsymbol{x}\right\|^2\right],$$

Table 3: Common Notation Summary (addition to Table 1).

Symbol	Description
M	Number of clients.
K	Number of local SGD.
η_g, η_l	Global/Local learning rate.
$\tilde{\eta}$	Effective learning rate, $\tilde{\eta} = \eta_l \eta_g$.
$\boldsymbol{\theta}^n$	Global model at server iteration n.
$\boldsymbol{\theta}_i^{n+1}$	Local update of client i on model $\boldsymbol{\theta}^n$.
$\boldsymbol{\theta}^*$, $\boldsymbol{\theta}_i^*$	Optimum of the federated problem (2)/client i.
$\boldsymbol{\theta}^{(n,k)}$, $\boldsymbol{\theta}_i^{(n,k)}$	Global/Local update after k SGD on global model $\boldsymbol{\theta}^n$.
α	Covariance parameter.
β	Defined in Theorem 1.
$\mathcal{L}(\cdot), \mathcal{L}_i(\cdot)$	Federated/local loss function.
$g_i(\cdot)$	SG. We have $\mathbb{E}_{\xi_i}[g_i(\cdot)] = \nabla \mathcal{L}_i(\cdot)$ with Assumption 3.
$\boldsymbol{\xi}_i$	Random batch of samples from client i of size B.
L	Lipschitz smoothness parameter, Assumption 1.
T_i	Computation time of client i.
t^n	Time at aggregation n.
T_i^n	Remaining computation time of client i at time t^n.
Δt^n	Time elapsed between two server aggregations.
$\rho_i(n)$	Last index at which a client i received its global model.
ρ	Highest sum of aggregation weights, i.e. $\rho \coloneqq \max(1, q(n))$

$$\phi(n,k) \coloneqq \sum_{i=1}^M \tilde{q}_i(n)\, \mathbb{E}\left[\left\|\boldsymbol{\theta}_i^{\rho_i(n),k} - \boldsymbol{\theta}^{n,k}\right\|^2\right], \quad \sigma_1(n) \coloneqq \sum_{i=1}^M \tilde{q}_i(n)\, \mathbb{E}\left[\left\|\nabla \mathcal{L}_i(\bar{\boldsymbol{\theta}}^n, \boldsymbol{\xi}_i)\right\|^2\right],$$

$$\sigma_2(n) \coloneqq \sum_{i=1}^M \tilde{q}_i^2(n)\, \mathbb{E}\left[\left\|\nabla \mathcal{L}_i(\bar{\boldsymbol{\theta}}^n, \boldsymbol{\xi}_i)\right\|^2\right], \text{ and } \Xi(n,k) = \mathcal{L}^n(\boldsymbol{\theta}^{n,k}) - \mathcal{L}^n(\boldsymbol{x}).$$

Finally, we define $g_i(\boldsymbol{y}) = \nabla \mathcal{L}_i(\boldsymbol{y}, \boldsymbol{\xi}_i)$ the SG of client i evaluated on model parameters $\boldsymbol{y}$ and batch $\boldsymbol{\xi}_i$. We will thus write $g_i(\boldsymbol{\theta}_{i,k}^{\rho_i(n)})$ instead of $\nabla \mathcal{L}_i(\boldsymbol{\theta}_{i,k}^{\rho_i(n)}, \boldsymbol{\xi}_{i,k}^{\rho_i(n)})$.

A.3 Useful Lemmas

Lemma 1. *Let us consider n vectors $\boldsymbol{x}_i, ..., \boldsymbol{x}_n$ and assume Assumption 5. We have*

$$\mathbb{E}_{S_n}\left[\left\|\sum_{i=1}^M \omega_i(n)\boldsymbol{x}_i\right\|^2\right] = \sum_{i=1}^M \gamma_i(n)\left\|\boldsymbol{x}_i\right\|^2 + \alpha\left\|\sum_{i=1}^M q_i(n)\boldsymbol{x}_i\right\|^2,$$

where $\gamma_i(n) = \mathbb{E}_{S_n}\left[\omega_i^2(n)\right] - \alpha q_i^2(n) \geq 0$, and $\gamma_i(n) \leq \beta q_i(n)$ with $\beta \coloneqq \max\{d_i(n) - \alpha q_i(n)\}$.

Proof.

$$\begin{aligned}\mathbb{E}_{S_n}\left[\left\|\sum_{i=1}^M \omega_i(n)\boldsymbol{x}_i\right\|^2\right] &= \sum_{i=1}^M \mathbb{E}_{S_n}\left[\omega_i^2(n)\right]\left\|\boldsymbol{x}_i\right\|^2 + \sum_{i=1}^M \sum_{\substack{j=1 \\ j\neq i}}^M \mathbb{E}_{S_n}\left[\omega_i(n)\omega_j(n)\right]\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \\ &= \sum_{i=1}^M \mathbb{E}_{S_n}\left[\omega_i^2(n)\right]\left\|\boldsymbol{x}_i\right\|^2 + \sum_{i=1}^M \sum_{\substack{j=1 \\ j\neq i}}^M \alpha q_i(n) q_j(n)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle. \end{aligned} \tag{16}$$

In addition, we have

$$\sum_{i=1}^M \sum_{\substack{j=1 \\ j\neq i}}^M \langle q_i(n)\boldsymbol{x}_i, q_j(n)\boldsymbol{x}_j\rangle = \left\|\sum_{i=1}^M q_i(n)\boldsymbol{x}_i\right\|^2 - \sum_{i=1}^M q_i^2(n)\left\|\boldsymbol{x}_i\right\|^2. \tag{17}$$

Substituting equation (17) in equation (16) completes the first claim.

Considering that $\mathbb{E}_{S_n}\left[\omega_i^2(n)\right] = \mathrm{Var}\left[\omega_i(n)\right] + q_i^2(n) \geq q_i^2(n)$ and $\alpha \leq 1$, we have $\gamma_i(n) \geq 0$ which completes the second claim.

Finally, the third claim follows directly from the close-form of the clients aggregation weights, equation (4).

Remark. We can also provide the following lower bound for equation (17) using Jensen inequality

$$\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\langle q_i(n)\boldsymbol{x}_i, q_j(n)\boldsymbol{x}_j\rangle \geq \left\|\sum_{i=1}^{M} q_i(n)\boldsymbol{x}_i\right\|^2 - \frac{\max q_i(n)}{q(n)}\left\|\sum_{i=1}^{M} q_i(n)\boldsymbol{x}_i\right\|^2 \geq 0.$$

Therefore, $\mathbb{E}_{S_n}\left[\left\|\sum_{i=1}^{M}\omega_i(n)\boldsymbol{x}_i\right\|^2\right]$ is linearly proportional to α.

□

Lemma 2. *Under Assumption 5, the following equation holds for any vector $\boldsymbol{x}$:*

$$\Delta(n) \leq -2\tilde{\eta}D(\boldsymbol{x}, n) + \tilde{\eta}^2\alpha q^2(n)R(n) + \tilde{\eta}^2\beta q(n)S(n).$$

Proof. We consider S_n, the set of participating clients at optimization round n, i.e. $S_n = \{n : T_i^n \leq \Delta t^n\}$. We have

$$\begin{aligned}\mathbb{E}_{S_n}\left[\left\|\boldsymbol{\theta}^{n,k+1} - \boldsymbol{\theta}^*\right\|^2\right] &= \mathbb{E}_{S_n}\left[\left\|(\boldsymbol{\theta}^{n,k+1} - \boldsymbol{\theta}^{n,k}) + (\boldsymbol{\theta}^{n,k} - \boldsymbol{\theta}^*)\right\|^2\right]\\ &= \left\|\boldsymbol{\theta}^{n,k} - \boldsymbol{\theta}^*\right\|^2 + 2\langle\mathbb{E}_{S_n}\left[\boldsymbol{\theta}^{n,k+1} - \boldsymbol{\theta}^{n,k}\right], \boldsymbol{\theta}^{n,k} - \boldsymbol{\theta}^*\rangle\\ &\quad + \mathbb{E}_{S_n}\left[\left\|\boldsymbol{\theta}^{n,k+1} - \boldsymbol{\theta}^{n,k}\right\|^2\right]. \end{aligned} \tag{18}$$

By construction, we have $\boldsymbol{\theta}^{n,k+1} - \boldsymbol{\theta}^{n,k} = -\tilde{\eta}\sum_{i=1}^{M}\omega_i(n)g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})$. Taking the expectation over S_n, we can simplify the second term of equation (18) with $\mathbb{E}_{S_n}\left[\boldsymbol{\theta}^{n,k+1} - \boldsymbol{\theta}^{n,k}\right] = -\tilde{\eta}\sum_{i=1}^{M} q_i(n)g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})$. Finally, using Lemma 1, we can bound the third term. Therefore, we have

$$\begin{aligned}\mathbb{E}_{S_n}\left[\left\|\boldsymbol{\theta}^{n,k+1} - \boldsymbol{\theta}^*\right\|^2\right] &= \left\|\boldsymbol{\theta}^{n,k} - \boldsymbol{\theta}^*\right\|^2 + 2\tilde{\eta}\langle\sum_{i=1}^{M} q_i(n)g_i(\boldsymbol{\theta}_i^{\rho_i(n),k}), \boldsymbol{\theta}^n - \boldsymbol{\theta}^*\rangle\\ &\quad + \tilde{\eta}^2\sum_{i=1}^{M}\gamma_i(n)\left\|g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})\right\|^2 + \tilde{\eta}^2\alpha\left\|\sum_{i=1}^{M} q_i(n)g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})\right\|^2.\end{aligned}$$

Considering $\gamma_i(n) \leq \beta q_i(n)$, taking the expected value over the iteration random batches $\boldsymbol{\xi}^{\rho_i(n),k}$, and finally taking the expected value over the remaining random variables gives

$$\Delta(n,k) \leq -2\tilde{\eta}D(\boldsymbol{x}, n, k) + \tilde{\eta}^2\alpha q^2(n)R(n,k) + \tilde{\eta}^2\beta q(n)S(n,k).$$

Taking the mean over K completes the proof.

□

Lemma 3. *Under Assumption 3 and 1, and $D \coloneqq 6\eta_l^2(K-1)^2L^2 \leq 1/2$, we have*

$$\phi(n) \leq 4q(n)\tau\sum_{s=1}^{\tau}Q(n-s) + 4D\frac{1}{L}q^{-1}(n)Z(n) + 6\eta_l^2(K-1)^2\sigma_1(n),$$

$$\textit{and } S(n) \leq 12q(n)L^2\tau\sum_{s=1}^{\tau}Q(n-s) + 12Lq^{-1}(n)Z(n) + 6\sigma_1(n).$$

Proof. Let us decompose the difference $\boldsymbol{\theta}_i^{\rho_i(n),k} - \boldsymbol{\theta}^{n,k}$ as

$$\boldsymbol{\theta}_i^{\rho_i(n),k} - \boldsymbol{\theta}^{n,k} = \left[\boldsymbol{\theta}^{\rho_i(n)} - \eta_l \sum_{l=0}^{k-1} g_i(\boldsymbol{\theta}_i^{\rho_i(n),l})\right] - \left[\boldsymbol{\theta}^{n} - \eta_l \sum_{l=0}^{k-1}\sum_{i=1}^{M} \tilde{q}_i(n) g_i(\boldsymbol{\theta}_i^{\rho_i(n),l})\right].$$

Using Jensen inequality, we split the difference between the global models and the one between the gradients to get

$$\left\|\boldsymbol{\theta}_i^{\rho_i(n),k} - \boldsymbol{\theta}^{n,k}\right\|^2 \leq 2\left\|\boldsymbol{\theta}^{\rho_i(n)} - \boldsymbol{\theta}^{n}\right\|^2 + 2\eta_l^2 k \sum_{l=0}^{k-1}\left\|g_i(\boldsymbol{\theta}_i^{\rho_i(n),l}) - \sum_{i=1}^{M}\tilde{q}_i(n) g_i(\boldsymbol{\theta}_i^{\rho_i(n),l})\right\|^2. \quad (19)$$

Therefore, by taking the expectations of equation (19) and summing over M gives

$$\begin{aligned}
\phi(n,k) &\leq 2\sum_{i=1}^{M}\tilde{q}_i(n)\,\mathbb{E}\left[\left\|\boldsymbol{\theta}^{\rho_i(n)} - \boldsymbol{\theta}^{n}\right\|^2\right] \\
&\quad + 2\eta_l^2 k \sum_{l=0}^{k-1}\sum_{i=1}^{M}\tilde{q}_i(n)\,\mathbb{E}\left[\left\|g_i(\boldsymbol{\theta}_i^{\rho_i(n),l}) - \sum_{i=1}^{M}\tilde{q}_i(n) g_i(\boldsymbol{\theta}_i^{\rho_i(n),l})\right\|^2\right] \\
&\leq 2\sum_{i=1}^{M}\tilde{q}_i(n)\,\mathbb{E}\left[\left\|\boldsymbol{\theta}^{\rho_i(n)} - \boldsymbol{\theta}^{n}\right\|^2\right] + 2\eta_l^2 k \sum_{l=0}^{k-1}\sum_{i=1}^{M}\tilde{q}_i(n)\,\mathbb{E}\left[\left\|g_i(\boldsymbol{\theta}_i^{\rho_i(n),l})\right\|^2\right], \quad (20)
\end{aligned}$$

where we see that $S(n,l)$ appears in the second term of equation (20). We consider now bounding $S(n,k)$, and first note that a stochastic gradient can be bounded as follow

$$\begin{aligned}
\mathbb{E}\left[\left\|g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})\right\|^2\right] &\leq 3\,\mathbb{E}\left[\left\|\nabla\mathcal{L}_i(\boldsymbol{\theta}_i^{\rho_i(n),k}, \boldsymbol{\xi}_{i,k}^{\rho_i(n)}) - \nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k}, \boldsymbol{\xi}_{i,k}^{\rho_i(n)})\right\|^2\right] \\
&\quad + 3\,\mathbb{E}\left[\left\|\nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k}, \boldsymbol{\xi}_i) - \nabla\mathcal{L}_i(\bar{\boldsymbol{\theta}}^{n}, \boldsymbol{\xi}_i)\right\|^2\right] + 3\,\mathbb{E}\left[\left\|\nabla\mathcal{L}_i(\bar{\boldsymbol{\theta}}^{n}, \boldsymbol{\xi}_i)\right\|^2\right]. \quad (21)
\end{aligned}$$

When summing equation (21) over M, and considering the clients loss functions Lipschitz smoothness, Assumption 1, we have

$$S(n,k) = \sum_{i=1}^{M}\tilde{q}_i(n)\,\mathbb{E}\left[\left\|g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})\right\|^2\right] \leq 3L^2\phi(n,k) + 6Lq^{-1}(n)Z(n,k) + 3\sigma_1(n). \quad (22)$$

We also note the following intermediary results

$$\sum_{k=0}^{K-1} k \sum_{l=0}^{k-1} x_l \leq (K-1)\sum_{k=1}^{K-1}\sum_{l=0}^{k-1} x_l \leq (K-1)^2 \sum_{k=0}^{K-2} x_k \leq (K-1)^2 \sum_{k=0}^{K-1} x_k. \quad (23)$$

We substitute equation (22) in equation (20) such that D appears, take the mean over K to introduce $\phi(n)$ on the two sides of the equation, and use equation (23). We have

$$\phi(n) \leq 2\sum_{i=1}^{M}\tilde{q}_i(n)\,\mathbb{E}\left[\left\|\boldsymbol{\theta}^{\rho_i(n)} - \boldsymbol{\theta}^{n}\right\|^2\right] + D\phi(n) + 2D\frac{1}{L}q^{-1}(n)Z(n) + 6\eta_l^2(K-1)^2\sigma_1(n).$$

Finally, reminding that $D \leq 1/2$, which gives $1 - D \geq 1/2$, and using Assumption 4 to bound $\mathbb{E}\left[\left\|\boldsymbol{\theta}^{\rho_i(n)} - \boldsymbol{\theta}^{n}\right\|^2\right]$ with Jensen inequality completes the first claim for $\phi(n)$, i.e.

$$\mathbb{E}\left[\left\|\boldsymbol{\theta}^{\rho_i(n)} - \boldsymbol{\theta}^{n}\right\|^2\right] \leq \tau\sum_{s=1}^{\tau}\mathbb{E}\left[\left\|\boldsymbol{\theta}^{n-s+1} - \boldsymbol{\theta}^{n-s}\right\|^2\right] = \tau\sum_{s=1}^{\tau} Q(n-s).$$

Substituting the close-form of $\phi(n)$ in equation (22) completes the claim for $S(n,k)$.

□

Lemma 4. *Under Assumption 2 and 3, we have*

$$-2D(\boldsymbol{x},n) \leq -2\Xi(n) + 4Lq(n)\tau \sum_{s=1}^{\tau} Q(n-s) + 4DZ(n) + 6\eta_l^2(K-1)^2 q(n) L\sigma_1(n).$$

Proof. Follows directly from using Lemma 12 in Khaled et al. (2020) on $D(\boldsymbol{x},n,k)$, taking the mean over K, and using Lemma 3 to bound $\phi(n)$ completes the proof.

□

Lemma 5. *Under Assumption 1 and 3, and considering $D \leq 1/2$, we have*

$$R(n) \leq 12L^2\tau \sum_{s=1}^{\tau} Q(n-s) + 24Lq^{-1}(n)Z(n) + 3D\sigma_1(n) + 6\sigma_2(n).$$

Proof.

$$\begin{aligned} R(n,k) \leq\, & 3\,\mathbb{E}\left[\left\|\sum_{i=1}^{M} \tilde{q}_i(n)\left[g_i(\boldsymbol{\theta}_i^{\rho_i(n),k}) - \nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k}, \boldsymbol{\xi}_{i,k}^{\rho_i(n)})\right]\right\|^2\right] \\ & + 3\,\mathbb{E}\left[\left\|\sum_{i=1}^{M} \tilde{q}_i(n)\left[\nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k}, \boldsymbol{\xi}_i) - \nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k})\right]\right\|^2\right] \\ & + 3\,\mathbb{E}\left[\left\|\sum_{i=1}^{M} \tilde{q}_i(n)\nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k})\right\|^2\right]. \end{aligned} \tag{24}$$

We respectively call the three terms of equation (24), $a(n,k)$, $b(n,k)$, and $c(n,k)$. Using the local loss functions Lipschitz smoothness, Assumption 1, and Jensen inequality, we can bound $a(n,k)$ as

$$a(n,k) \leq 3\sum_{i=1}^{M} \tilde{q}_i(n)\,\mathbb{E}\left[\left\|g_i(\boldsymbol{\theta}_i^{\rho_i(n),k}) - \nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k}, \boldsymbol{\xi}_{i,k}^{\rho_i(n)})\right\|^2\right] \leq 3L^2\phi(n,k). \tag{25}$$

Using the unbiasedness of the gradient estimator, Assumption 3, and the local loss function Lipschitz smoothness, Assumption 1, we can bound $b(n,k)$ as

$$\begin{aligned} b(n,k) &= 3\sum_{i=1}^{M} \tilde{q}_i^2(n)\,\mathbb{E}\left[\left\|\nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k}, \boldsymbol{\xi}_i) - \nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k})\right\|^2\right] \\ &\leq 3\sum_{i=1}^{M} \tilde{q}_i^2(n)\,\mathbb{E}\left[\left\|\nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k}, \boldsymbol{\xi}_i)\right\|^2\right] \\ &\leq 6\sum_{i=1}^{M} \tilde{q}_i^2(n)\left[\mathbb{E}\left[\left\|\nabla\mathcal{L}_i(\boldsymbol{\theta}^{n,k}, \boldsymbol{\xi}_i) - \nabla\mathcal{L}_i(\bar{\boldsymbol{\theta}}^{n}, \boldsymbol{\xi}_i)\right\|^2\right] + \mathbb{E}\left[\left\|\nabla\mathcal{L}_i(\bar{\boldsymbol{\theta}}^{n}, \boldsymbol{\xi}_i)\right\|^2\right]\right] \\ &\leq 12L\max_i(\tilde{q}_i(n))\left[\tilde{\mathcal{L}}^n(\boldsymbol{\theta}^{n,k}) - \tilde{\mathcal{L}}^n(\bar{\boldsymbol{\theta}}^n)\right] + 6\sum_{i=1}^{M} \tilde{q}_i^2(n)\,\mathbb{E}\left[\left\|\nabla\mathcal{L}_i(\bar{\boldsymbol{\theta}}^{n}, \boldsymbol{\xi}_i))\right\|^2\right]. \end{aligned} \tag{26}$$

Using the Lipschitz smoothness of the local loss functions, Assumption 1 and Jensen inequality, we can bound $c(n,k)$ as

$$c(n,k) \leq 3\,\mathbb{E}\left[\left\|\nabla\tilde{\mathcal{L}}^n(\boldsymbol{\theta}^{n,k}) - \nabla\tilde{\mathcal{L}}^n(\bar{\boldsymbol{\theta}}^n)\right\|^2\right] \leq 6L\left[\tilde{\mathcal{L}}^n(\boldsymbol{\theta}^{n,k}) - \tilde{\mathcal{L}}^n(\bar{\boldsymbol{\theta}}^n)\right]. \tag{27}$$

Substituting equation (25), equation (26), and equation (27) in equation (24), considering that $\max_i(\tilde{q}_i(n)) \leq 1$, and summing over K gives

$$R(n) \leq 3L^2\phi(n) + 18Lq^{-1}(n)Z(n) + 6\sigma_2(n)$$

Using Lemma 3 to replace $\phi(n)$, and considering that $D \leq 1/2 < 1$ completes the proof.

□

Lemma 6. *Under Assumption 1 and 3, considering that $\gamma_i(n) \leq \beta q_i(n)$, and considering $12\rho^2 [\alpha + \beta] \tilde{\eta}^2 K^2 \tau^2 L^2 \leq 1/2$, we have*

$$\bar{Q}(N) \leq 24\rho [2\alpha + \beta] \tilde{\eta}^2 K^2 L \bar{Z}(N) + 6\rho^2 [\alpha D + 2\beta] \tilde{\eta}^2 K^2 \Sigma_1(N) + 12\rho^2 \alpha \tilde{\eta}^2 K^2 \Sigma_2(N).$$

Proof. Considering the proof of Lemma 2, using the fact that $\gamma_i(n) \leq \beta q_i(n)$, and Jensen inequality, we have

$$\begin{aligned} Q(n) &\leq q^2(n)\alpha\tilde{\eta}^2 \, \mathbb{E}\left[\left\|\sum_{i=1}^{M} \tilde{q}_i(n) \sum_{k=0}^{K-1} g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})\right\|^2\right] + q(n)\beta\tilde{\eta}^2 \sum_{i=1}^{M} \tilde{q}_i(n) \, \mathbb{E}\left[\left\|\sum_{k=0}^{K-1} g_i(\boldsymbol{\theta}_i^{\rho_i(n),k})\right\|^2\right] \\ &\leq q^2(n)\alpha\tilde{\eta}^2 K^2 R(n) + q(n)\beta\tilde{\eta}^2 K^2 S(n) \end{aligned}$$

Using Lemma 5 to bound $R(n)$ and Lemma 3 to bound $S(n)$, we can thus bound $Q(n)$ with the previous global model distances to the optimum $Q(s)$, where $\max(0, n-\tau) \leq s \leq n-1$, we thus have

$$\begin{aligned} \frac{1}{\rho\tilde{\eta}^2 K^2} Q(n) \leq{}& 12\rho [\alpha + \beta] \tau L^2 \sum_{s=1}^{\tau} Q(n-s) + 12 [2\alpha + \beta] L Z(n) \\ &+ 3\rho [\alpha D + 2\beta] \sigma_1(n) + 6\rho\alpha\sigma_2(n). \end{aligned} \tag{28}$$

We can thus define $A(n)$ and $B(n)$ such that the bound of of equation (28) can be rewritten as in equation (29), with its associated implications when taking the mean over N, reordering, and considering that $\tau A(n) \leq 1/2$:

$$Q(n) \leq A(n) \sum_{s=1}^{\tau} Q(n-s) + B(n) \Rightarrow \bar{Q}(N) = \frac{1}{N} \sum_{n=0}^{N-1} Q(n) \leq 2 \frac{1}{N} \sum_{n=0}^{N-1} B(n). \tag{29}$$

Therefore, considering $12\rho^2 [\alpha + \beta] \tilde{\eta}^2 K^2 \tau^2 L^2 \leq 1/2$ completes the proof.

□

A.4 Proof of Theorem 1

Proof. Using Lemma 2, we have

$$\frac{1}{\tilde{\eta}} \Delta(n) \leq -2D(\boldsymbol{x}, n) + \rho^2 \alpha \tilde{\eta} R(n) + \rho\beta\tilde{\eta} S(n)$$

Using Lemma 4 to bound $D(\boldsymbol{x}, n)$, Lemma 5 to bound $R(n)$, Lemma 3 to bound $S(n)$, and $3\rho [\alpha + \beta] \tilde{\eta} L \leq 1$, we get

$$\begin{aligned} \frac{1}{\tilde{\eta}} \Delta(n) \leq{}& -2\Xi(n) + 8\rho\tau L \sum_{s=1}^{\tau} Q(n-s) + 4DZ(n) + 6\rho\eta_l^2 (K-1)^2 L \sigma_1(n) \\ &+ 12 [2\alpha + \beta] \rho\tilde{\eta} L Z(n) + 3\rho^2 \tilde{\eta} [\alpha D + 2\beta] \sigma_1(n) + 6\rho^2 \alpha \tilde{\eta} \sigma_2(n). \end{aligned}$$

When considering the following intermediary result

$$\sum_{n=0}^{N-1} K\Delta(n) = \mathbb{E}\left[\left\|\boldsymbol{\theta}^{KN} - \boldsymbol{x}\right\|^2\right] - \left\|\boldsymbol{\theta}^0 - \boldsymbol{x}\right\|^2 \geq -\left\|\boldsymbol{\theta}^0 - \boldsymbol{x}\right\|^2,$$

reordering the terms, and taking the mean over N, we get

$$\begin{aligned} 2\bar{\Xi}(N) \leq{}& \frac{1}{\tilde{\eta} K N} \mathbb{E}\left[\left\|\boldsymbol{\theta}^0 - \boldsymbol{x}\right\|^2\right] + 8\rho L \tau^2 \bar{Q}(N) + 4D\bar{Z}(N) + 6\rho\eta_l^2 (K-1)^2 L \Sigma_1(N) \\ &+ 12\rho [2\alpha + \beta] \tilde{\eta} L \bar{Z}(N) + 3\rho^2 [\alpha D + 2\beta] \tilde{\eta} \Sigma_1(N) + 6\rho^2 \alpha \tilde{\eta} \Sigma_2(N). \end{aligned}$$

Using Lemma 6 to bound $\bar{Q}(N)$, and with $\nu = 16\rho L$, we have

$$\begin{aligned} 2\bar{\Xi}(N) \leq{}& \frac{1}{\tilde{\eta} K N} \mathbb{E}\left[\left\|\boldsymbol{\theta}^0 - \boldsymbol{x}\right\|^2\right] + 4D\bar{Z}(N) + 6\rho\eta_l^2 (K-1)^2 L \Sigma_1(N) \\ &+ 12\rho [2\alpha + \beta] \left[\tilde{\eta} + \nu\tilde{\eta}^2 K^2 \tau^2\right] L \bar{Z}(N) + 3\rho^2 [\alpha D + 2\beta] \left[\tilde{\eta} + \nu\tilde{\eta}^2 K^2 \tau^2\right] \Sigma_1(N) \\ &+ 6\rho^2 \alpha \left[\tilde{\eta} + \nu\tilde{\eta}^2 K^2 \tau^2\right] \Sigma_2(N). \end{aligned}$$

We note that when $\bar{\Xi}(N) \leq 0$, the claim follows directly. Therefore, we consider $\bar{\Xi}(N) \geq 0$ for the rest of this proof. We first note that

$$\bar{Z}(N) = \bar{\Xi}(N) + R(\{\mathcal{L}^n\}), \tag{30}$$

and consider η_l such that

$$2 - 4D - 12\rho\left[2\alpha + \beta\right]\left[\tilde{\eta} + \nu\tilde{\eta}^2K^2\tau^2\right]L \geq 1,$$

which gives

$$\begin{aligned}\bar{\Xi}(N) \leq \frac{1}{\tilde{\eta}KN}\mathbb{E}\left[\left\|\boldsymbol{\theta}^0 - \boldsymbol{x}\right\|^2\right] + 4DR(\{\mathcal{L}^n\}) + 6\rho\eta_l^2(K-1)^2L\Sigma_1(N) \\ + 12\rho\left[2\alpha+\beta\right]\left[\tilde{\eta} + \nu\tilde{\eta}^2K^2\tau^2\right]LR(\{\mathcal{L}^n\}) + 3\rho^2\left[\alpha D + 2\beta\right]\left[\tilde{\eta} + \nu\tilde{\eta}^2K^2\tau^2\right]\Sigma_1(N) \\ + 6\rho^2\alpha\left[\tilde{\eta} + \nu\tilde{\eta}^2K^2\tau^2\right]\Sigma_2(N).\end{aligned}$$

The 5th term can be simplified with the third one. Indeed, we consider a local learning rate such that $3\rho^2\tilde{\eta}L \leq 1$, $48\rho^3\tilde{\eta}^2K^2\tau^2L^2 \leq 1$, and we remind that $\alpha \leq 1$. We thus have

$$\begin{aligned}\bar{\Xi}(N) \leq \frac{1}{\tilde{\eta}KN}\mathbb{E}\left[\left\|\boldsymbol{\theta}^0 - \boldsymbol{x}\right\|^2\right] + \mathcal{O}\left(\eta_l^2(K-1)^2\left[R(\{\mathcal{L}^n\}) + \Sigma_1(N)\right]\right) \\ + \mathcal{O}\left(\alpha\left[\tilde{\eta} + \tilde{\eta}^2K^2\tau^2\right]\left[R(\{\mathcal{L}^n\}) + \Sigma_2(N)\right]\right) \\ + \mathcal{O}\left(\beta\left[\tilde{\eta} + \tilde{\eta}^2K^2\tau^2\right]\left[R(\{\mathcal{L}^n\}) + \Sigma_1(N)\right]\right).\end{aligned} \tag{31}$$

With

$$\left\|\nabla\mathcal{L}_i(\boldsymbol{\theta},\xi)\right\|^2 \leq 2\left\|\nabla\mathcal{L}_i(\boldsymbol{\theta},\xi) - \nabla\mathcal{L}_i(\bar{\boldsymbol{\theta}},\xi)\right\|^2 + 2\left\|\nabla\mathcal{L}_i(\bar{\boldsymbol{\theta}},\xi)\right\|^2,$$

we have

$$\Sigma_2(N) \leq \max q_i(n)\Sigma_1(N) \leq \max q_i(n)\left[4LR(\mathcal{L}^n) + 2\Sigma\right]. \tag{32}$$

Finally, substituting equation (30) and (32) in equation (31) completes the proof.

□

A.5 Simplifying the constraint on the learning rate

The constraints on the learning rate can be summarized as $D = 6\eta_l^2(K-1)^2L^2 \leq 1/2$ (Lemma 3), $12\rho^2\left[\alpha+\beta\right]\tilde{\eta}^2K^2\tau^2L^2 \leq 1/2$ (Lemma 6), $3\rho\left[\alpha+\beta\right]\tilde{\eta}L \leq 1$ (Theorem 1), $2 - 4D - 12\rho\left[2\alpha+\beta\right]\left[\tilde{\eta} + \nu\tilde{\eta}^2K^2\tau^2\right]L \geq 1$ (Theorem 1), $3\rho^2\tilde{\eta}L \leq 1$ (Theorem 1), and $48\rho^3\tilde{\eta}^2K^2\tau^2L^2 \leq 1$ (Theorem 1).

We note that $\alpha \leq 1$, and $\beta \leq 1$. We thus propose the following sufficient conditions to satisfy the conditions above

$$48\eta_l^2(K-1)^2L^2 \leq 1, 144\rho^2\tilde{\eta}L \leq 1, \text{ and } 2304\rho^3\tilde{\eta}^2K^2\tau^2L^2 \leq 1,$$

which can further be simplified with

$$\eta_l \leq \frac{1}{48KL}\min\left(1, \frac{1}{3\rho^2\eta_g(\tau+1)}\right).$$

B Proof of Theorem 2

In this proof, we consider $\tilde{\mathcal{L}}^n = q^{-1}(n)\mathcal{L}^n$.

B.1 Useful Lemma

Lemma 7. *The difference between the gradients of $\mathcal{L}(\boldsymbol{\theta})$ and $\mathcal{L}(\boldsymbol{\theta})$ can be bounded as follow*

$$\left\|\nabla\mathcal{L}(\boldsymbol{\theta}) - \nabla\tilde{\mathcal{L}}^n(\boldsymbol{\theta})\right\|^2 \leq 4L\chi_n^2[\tilde{\mathcal{L}}^n(\boldsymbol{\theta}) - \sum_{j\in W_n}\tilde{s}_j(n)\mathcal{L}_j(\boldsymbol{\theta}_j^*)] + 4L\sum_{j\notin W_n}r_j[\mathcal{L}_j(\boldsymbol{\theta}) - \mathcal{L}_j(\boldsymbol{\theta}_j^*)],$$

where $W_n = \{j : s_j(n) > 0\}$ and $\chi_n^2 = \sum_{j\in W_n}(r_j - \tilde{s}_j(n))^2/\tilde{s}_j(n)$.

Proof. We have $\sum_{j=1}^{J} s_j(n) = \sum_{i=1}^{M} q_i(n) = q(n)$. Hence, by definition of $\mathcal{L}(\boldsymbol{\theta})$ and $\mathcal{L}^n(\boldsymbol{\theta})$, we have

$$\begin{aligned}\nabla\mathcal{L}(\boldsymbol{\theta}) - \nabla\tilde{\mathcal{L}}^n(\boldsymbol{\theta}) &= \sum_{j=1}^{J}(r_j - \tilde{s}_j(n))\nabla\mathcal{L}_j(\boldsymbol{\theta}) \\ &= \sum_{j\in W_n} \frac{r_j - \tilde{s}_j(n)}{\sqrt{\tilde{s}_j(n)}}\sqrt{\tilde{s}_j(n)}\nabla\mathcal{L}_j(\boldsymbol{\theta}) + \sum_{j\notin W_n} r_j\nabla\mathcal{L}_j(\boldsymbol{\theta}).\end{aligned}$$

Applying Jensen and Cauchy-Schwartz inequality gives

$$\begin{aligned}\left\|\nabla\mathcal{L}(\boldsymbol{\theta}) - \nabla\tilde{\mathcal{L}}^n(\boldsymbol{\theta})\right\|^2 &\leq 2\left\|\sum_{j\in W_n} \frac{r_j - \tilde{s}_j(n)}{\sqrt{\tilde{s}_j(n)}}\sqrt{\tilde{s}_j(n)}\nabla\mathcal{L}_j(\boldsymbol{\theta})\right\|^2 + 2\left\|\sum_{j\notin W_n} r_j\nabla\mathcal{L}_j(\boldsymbol{\theta})\right\|^2 \\ &\leq 2\left[\sum_{j\in W_n}\frac{(r_j - \tilde{s}_j(n))^2}{\tilde{s}_j(n)}\right]\sum_{j=1}^{J}\tilde{s}_j(n)\left\|\nabla\mathcal{L}_j(\boldsymbol{\theta})\right\|^2 \\ &+ 2\left[\sum_{j\notin W_n} r_j\right]\sum_{j\notin W_n} r_j\left\|\nabla\mathcal{L}_j(\boldsymbol{\theta})\right\|^2\end{aligned}$$

Considering the Lipschitz smoothness of the clients loss function, and $\sum_{j\notin W_n} r_j \leq 1$ completes the proof.

□

B.2 Proof of Theorem 2

Proof. Using Jensen inequality and Lemma 7 gives

$$\begin{aligned}\left\|\nabla\mathcal{L}(\boldsymbol{\theta})\right\|^2 &\leq 2\left\|\nabla\mathcal{L}(\boldsymbol{\theta}) - \frac{1}{q(n)}\nabla\mathcal{L}^n(\boldsymbol{\theta})\right\|^2 + 2\left\|\frac{1}{q(n)}\nabla\mathcal{L}^n(\boldsymbol{\theta})\right\|^2 \\ &\leq 4L\left[\chi_n^2\frac{1}{q(n)} + \frac{1}{q^2(n)}\right][\mathcal{L}^n(\boldsymbol{\theta}) - \mathcal{L}^n(\bar{\boldsymbol{\theta}}^n)] \\ &+ \chi_n^2\frac{1}{q(n)}4L[\mathcal{L}^n(\bar{\boldsymbol{\theta}}^n) - \sum_{j\in W_n} s_j(n)\mathcal{L}_j(\boldsymbol{\theta}_j^*)] \\ &+ 4L\sum_{j\notin W_n} r_j[\mathcal{L}_j(\boldsymbol{\theta}) - \mathcal{L}_j(\boldsymbol{\theta}_j^*)]\end{aligned}$$

We take the maximum of χ_n^2 and $q(n)$, the mean over the KN serial SGD steps, and use Theorem 1 to complete the proof .

□

C Applying Theorem 3

This section extends Section 4, where we apply Theorem 3 to centralized learning (Section C.1) and synchronous FEDAVG with unbiased and biased client sampling (Section C.2 and C.3 respectively).

C.1 Centralized Learning

In this setting, one client, i.e. $M = 1$, learns a predictive model on its own data. In this case, we always have $\tilde{q}_1(n) = 1$, and the resulting optimization problem is always proportional to $\mathcal{L} = \mathcal{L}_1$ which thus gives $R(\{\mathcal{L}^n\}) \leq R(\mathcal{L}) = 0$. There is no gradient delay ($\tau = 1$), while the clients always participate at each optimization round ($\alpha = 1$ and $\beta = 0$), while the global learning rate

is redundant with the local learning rate ($\eta_g = 1$). The server performs KN SGD steps. All these considered elements give

$$\epsilon = \mathcal{O}\left(\frac{\left\|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\right\|^2}{\eta_l K N}\right) + \mathcal{O}\left(\eta_l \, \mathbb{E}_{\boldsymbol{\xi}}\left[\left\|\nabla \mathcal{L}(\boldsymbol{x}, \boldsymbol{\xi})\right\|^2\right]\right). \tag{33}$$

With equation (33), we retrieve standard convergence guarantees for centralized ML derived in Bottou et al. (2016).

C.2 Unbiased client sampling ($q_i(n) = p_i$)

We define by S_n the set of sampled clients performing their local work at optimization step n. Setting $\Delta t^n = \max_{i \in S_n} T_i$, with $T_i = \infty$ for the clients that are not sampled, and thus not in S_n, gives $\mathbb{P}(T_i \leq \Delta t^n) = \mathbb{P}(i \in S_n)$. S_n is independent from the clients hardware capabilities and is decided by the server. This allows to pre-compute $\mathbb{P}(T_i \leq \Delta t^n)$ and to allocate to each client the aggregation weight d_i such that $q_i = p_i$.

Standard unbiased client sampling schemes include sampling m clients uniformly without replacement (Li et al., 2020b) or sampling m clients according to a Multinomial distribution (Li et al., 2020a). Fraboni et al. (2022) shows that both Uniform and MD sampling satisfy Assumption 5. In particular, in those setting, the term $\alpha \leq 1$ is proportional to m, the amount of sampled clients, while $1 \geq \beta > 0$ is inversely proportional to m. We get

$$\epsilon = \mathcal{O}\left(\frac{1}{\eta_g \eta_l K N}\right) + \mathcal{O}\left(\eta_g \eta_l \alpha \frac{1}{M} \Sigma\right) + \mathcal{O}\left(\eta_l^2 (K-1)^2 \Sigma\right) + \mathcal{O}\left(\eta_g \eta_l \beta \Sigma\right).$$

The second term, proportional to α/M, is reduced at the expense of the introduction of a fourth term proportional to β. In turn, it still provides faster optimization rounds with $\Delta t^n = \max_{i \in S_n} T_i$ and $N = \mathcal{O}\left(T / \mathbb{E}\left[\max_{i \in S_n} T_i\right]\right)$. FedAvg with client sampling generalizes FedAvg with full client participation ($\alpha = 1$ and $\beta = 0$).

C.3 Biased client sampling ($q_i(n) \neq p_i$)

The condition $q_i(n) = p_i$ imposes the design of new client sampling based on the clients data heterogeneity. Nevertheless, we show convergence of biased client samplings where m clients are selected according to a deterministic criterion, e.g. when selecting the m clients with the highest loss (Cho et al., 2020), or when selecting the m clients with the most available computation resources (Nishio and Yonetani, 2019). In this case, $\mathbb{P}(i \in S_n) = 0/1$, with 1 if a client satisfies the criterion and 0 otherwise. In this case, no weighting scheme can make an optimization round unbiased. We also have $\mathbb{P}(\{i, j\} \in S_n) = \mathbb{P}(i \in S_n)\mathbb{P}(j \in S_n)$, which gives $\alpha = 1$ with $\beta = 0$. Without modification, this client sampling cannot satisfy the relaxed sufficient conditions of Theorem 3 and thus converges to a suboptimum point. This drawback can be mitigated by allocating a part of time in the window W to sample clients according to the criterion, and the rest of the window to consider clients such that $q_i = p_i$ is satisfied over W optimization rounds. By denoting ϵ_{FedAvg} the convergence guarantees (11), we have

$$\epsilon = \epsilon_{\text{FedAvg}} + \mathcal{O}\left(\eta_g \eta_l (W-1) K\right). \tag{34}$$

We note that equation (34) provides a looser bound than equation (11) in term of optimization rounds N. Still, this bound is informative and shows that, with minor changes, biased clients sampling based on a deterministic criterion can be proven to converge to the FL optimum.

D Additional Experiments

Acknowledgments

This work has been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002, and by the ANR JCJC project Fed-BioMed 19-CE45-0006-01. The project was also supported by Accenture. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

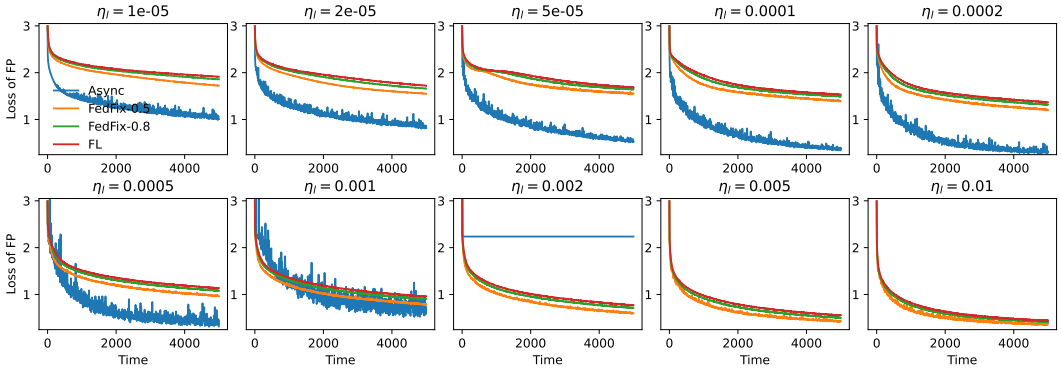

Figure 5: Evolution of federated problem (2) loss for CIFAR10 and time scenario $F80$ with $M = 20$ and $K = 10$.

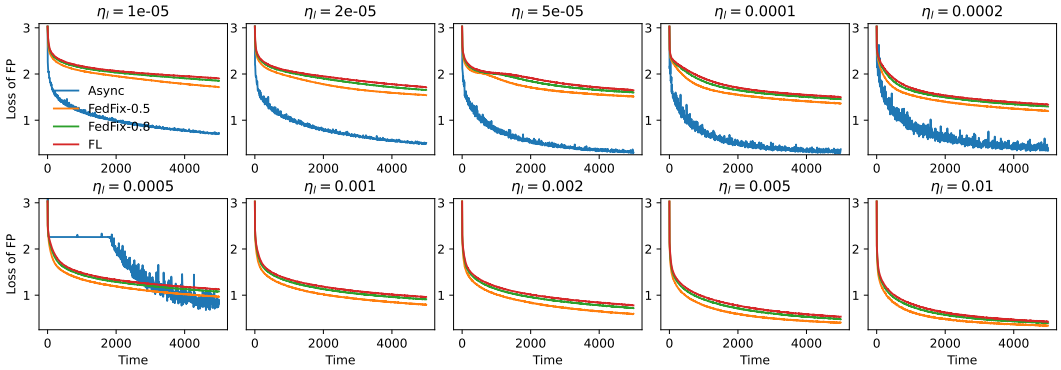

Figure 6: Evolution of federated problem (2) loss for CIFAR10 and time scenario $F80$ with $M = 50$ and $K = 10$.

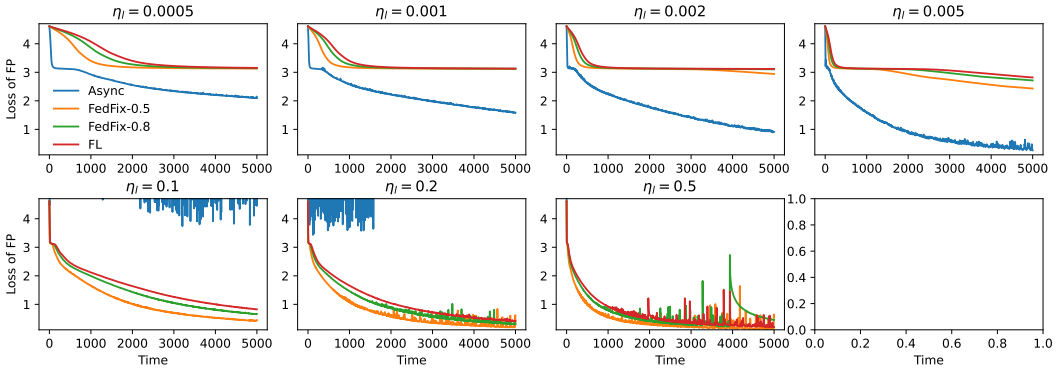

Figure 7: Evolution of federated problem (2) loss for Shakespeare and time scenario $F80$ with $M = 20$ and $K = 10$.

References

Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., and Saligrama, V. (2021). Federated learning based on dynamic regularization. In *International Conference on Learning Representations*.

Agarwal, A. and Duchi, J. C. (2011). Distributed delayed stochastic optimization. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Basu, D., Data, D., Karakus, C., and Diggavi, S. (2019). Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., USA.

Bottou, L., Curtis, F. E., and Nocedal, J. (2016). Optimization methods for large-scale machine learning. *SIAM Review*, 60.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. (2018). LEAF: A Benchmark for Federated Settings. (NeurIPS):1–9.

Cho, Y. J., Wang, J., and Joshi, G. (2020). Client selection in federated learning: Convergence analysis and power-of-choice selection strategies.

De Sa, C. M., Zhang, C., Olukotun, K., Ré, C., and Ré, C. (2015). Taming the wild: A unified analysis of hogwild-style algorithms. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Fraboni, Y., Vidal, R., Kameni, L., and Lorenzi, M. (2022). A general theory for client sampling in federated learning. In *International Workshop on Trustworthy Federated Learning in conjunction with IJCAI 2022 (FL-IJCAI'22)*.

Haddadpour, F., Kamani, M. M., Mahdavi, M., and Cadambe, V. (2019). Trading redundancy for communication: Speeding up distributed SGD for non-convex optimization. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2545–2554. PMLR.

Harry Hsu, T. M., Qi, H., and Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv*.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K. A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. (2019). Advances and open problems in federated learning. *CoRR*, abs/1912.04977.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR.

Khaled, A., Mishchenko, K., and Richtarik, P. (2020). Tighter theory for local sgd on identical and heterogeneous data. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4519–4529. PMLR.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Koloskova*, A., Lin*, T., Stich, S. U., and Jaggi, M. (2020). Decentralized deep learning with arbitrary communication compression. In *International Conference on Learning Representations*.

Koloskova, A., Stich, S., and Jaggi, M. (2019). Decentralized stochastic optimization and gossip algorithms with compressed communication. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3478–3487. PMLR.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020a). Federated optimization in heterogeneous networks. In Dhillon, I., Papailiopoulos, D., and Sze, V., editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450.

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

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. (2020b). On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*.

Li, X. and Orabona, F. (2019). On the convergence of stochastic gradient descent with adaptive stepsizes. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 983–992. PMLR.

Lian, X., Huang, Y., Li, Y., and Liu, J. (2015). Asynchronous parallel stochastic gradient for nonconvex optimization. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J., editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

Nedić, A., Bertsekas, D., and Borkar, V. (2001). Distributed asynchronous incremental subgradient methods. *Studies in Computational Mathematics*, 8(C):381–407.

Nguyen, J., Malik, K., Zhan, H., Yousefpour, A., Rabbat, M. G., Malekesmaeili, M., and Huba, D. (2021). Federated learning with buffered asynchronous aggregation. *ArXiv*, abs/2106.06639.

Nguyen, L., NGUYEN, P. H., van Dijk, M., Richtarik, P., Scheinberg, K., and Takac, M. (2018). SGD and hogwild! Convergence without the bounded gradients assumption. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3750–3758. PMLR.

Nishio, T. and Yonetani, R. (2019). Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. (2021). Adaptive federated optimization. In *International Conference on Learning Representations*.

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., and Pedarsani, R. (2020). Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2021–2031. PMLR.

Stich, S., Mohtashami, A., and Jaggi, M. (2021). Critical parameters for scalable distributed learning with large batches and asynchronous updates. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 4042–4050. PMLR.

Stich, S. U. and Karimireddy, S. P. (2020). The Error-Feedback framework: SGD with Delayed Gradients. *Journal of Machine Learning Research*, 21(237):1–36.

Wang, H., Sievert, S., Liu, S., Charles, Z., Papailiopoulos, D., and Wright, S. (2018). Atomo: Communication-efficient learning via atomic sparsification. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. (2020a). Tackling the objective inconsistency problem in heterogeneous federated optimization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wang, J., Tantia, V., Ballas, N., and Rabbat, M. (2020b). Slowmo: Improving communication-efficient distributed sgd with slow momentum. In *International Conference on Learning Representations*.

Ward, R., Wu, X., and Bottou, L. (2019). AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6677–6686. PMLR.

Xu, C., Qu, Y., Xiang, Y., and Gao, L. (2021). Asynchronous federated learning on heterogeneous devices: A survey.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2).

Yu, H., Jin, R., and Yang, S. (2019a). On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7184–7193. PMLR.

Yu, H., Yang, S., and Zhu, S. (2019b). Parallel Restarted SGD with Faster Convergence and Less Communication: Demystifying Why Model Averaging Works for Deep Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5693–5700.

Zinkevich, M., Langford, J., and Smola, A. (2009). Slow learners are fast. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.