POLITECNICO DI TORINO
Repository ISTITUZIONALE

Federated Learning under Heterogeneous and Correlated Client Availability

Original
Federated Learning under Heterogeneous and Correlated Client Availability / Rodio, A., Faticanti, F., Marfoq, O., Neglia, G., Leonardi, E.. - ELETTRONICO. - (2023), pp. 1-10. (IEEE INFOCOM 2023 - IEEE Conference on Computer Communications New York (USA) 17-20 May 2023) [10.1109/infocom53939.2023.10228876].

Availability:
This version is available at: 11583/2996282 since: 2025-01-07T08:13:36Z

Publisher:
IEEE

Published
DOI:10.1109/infocom53939.2023.10228876

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

28 November 2026

Federated Learning under Heterogeneous and Correlated Client Availability

Angelo Rodio*, Francescomaria Faticanti*, Othmane Marfoq*†, Giovanni Neglia*, Emilio Leonardi‡
*Inria, Université Côte d'Azur, France. Email: {firstname.lastname}@inria.fr,
†Accenture Labs, Sophia-Antipolis, France. Email: {firstname.lastname}@accenture.com,
‡Politecnico di Torino, Turin, Italy. Email: {firstname.lastname}@polito.it

arXiv:2301.04632v1 [cs.LG] 11 Jan 2023

***Abstract*—The enormous amount of data produced by mobile and IoT devices has motivated the development of federated learning (FL), a framework allowing such devices (or clients) to collaboratively train machine learning models without sharing their local data. FL algorithms (like `FedAvg`) iteratively aggregate model updates computed by clients on their own datasets. Clients may exhibit different levels of participation, often correlated over time and with other clients. This paper presents the first convergence analysis for a `FedAvg`-like FL algorithm under heterogeneous and correlated client availability. Our analysis highlights how correlation adversely affects the algorithm's convergence rate and how the aggregation strategy can alleviate this effect at the cost of steering training toward a biased model. Guided by the theoretical analysis, we propose `CA-Fed`, a new FL algorithm that tries to balance the conflicting goals of maximizing convergence speed and minimizing model bias. To this purpose, `CA-Fed` dynamically adapts the weight given to each client and may ignore clients with low availability and large correlation. Our experimental results show that `CA-Fed` achieves higher time-average accuracy and a lower standard deviation than state-of-the-art `AdaFed` and `F3AST`, both on synthetic and real datasets.**

***Index Terms*—Federated Learning, Distributed Optimization.**

I. Introduction

The enormous amount of data generated by mobile and IoT devices motivated the emergence of distributed machine learning training paradigms [1], [2]. Federated Learning (FL) [3]–[6] is an emerging framework where geographically distributed devices (or clients) participate in the training of a shared Machine Learning (ML) model without sharing their local data. FL was proposed to reduce the overall cost of collecting a large amount of data as well as to protect potentially sensitive users' private information. In the original Federated Averaging algorithm (`FedAvg`) [4], a central server selects a random subset of clients from the set of available clients and broadcasts them the shared model. The sampled clients perform a number of independent Stochastic Gradient Descent (SGD) steps over their local datasets and send their local model updates back to the server. Then, the server aggregates the received client updates to produce a new global model, and a new training round begins. At each iteration of `FedAvg`, the server typically samples randomly a few hundred devices to participate [7], [8].

This research was supported by the French government through the 3IA Côte d'Azur Investments in the Future project by the National Research Agency (ANR) with reference ANR-19-P3IA-0002, and by Groupe La Poste, sponsor of Inria Foundation, in the framework of FedMalin Inria Challenge.
A first version of this work has been accepted at IEEE INFOCOM 2023.

In real-world scenarios, the availability/activity of clients is dictated by exogenous factors that are beyond the control of the orchestrating server and hard to predict. For instance, only smartphones that are idle, under charge, and connected to broadband networks are commonly allowed to participate in the training process [4], [9]. These eligibility requirements can make the availability of devices correlated over time and space [7], [10]–[12]. For example, *temporal correlation* may origin from a smartphone being under charge for a few consecutive hours and then ineligible for the rest of the day. Similarly, the activity of a sensor powered by renewable energy may depend on natural phenomena intrinsically correlated over time (e.g., solar light). *Spatial correlation* refers instead to correlation across different clients, which often emerges as consequence of users' different geographical distribution. For instance, clients in the same time zone often exhibit similar availability patterns, e.g., due to time-of-day effects.

Temporal correlation in the data sampling procedure is known to negatively affect the performance of ML training even in the centralized setting [13], [14] and can potentially lead to *catastrophic forgetting*: the data used during the final training phases can have a disproportionate effect on the final model, "erasing" the memory of previously learned information [15], [16]. Catastrophic forgetting has also been observed in FL, where clients in the same geographical area have more similar local data distributions and clients' participation follows a cyclic daily pattern (leading to spatial correlation) [7], [10], [11], [17]. Despite this evidence, a theoretical study of the convergence of FL algorithms under both temporally and spatially correlated client participation is still missing.

This paper provides the first convergence analysis of `FedAvg` [4] under heterogeneous and correlated client availability. We assume that clients' temporal and spatial availability follows an arbitrary finite-state Markov chain: this assumption models a realistic scenario in which the activity of clients is correlated and, at the same time, still allows the analytical tractability of the system. Our theoretical analysis (i) quantifies the negative effect of correlation on the algorithm's convergence rate through an additional term, which depends on the spectral properties of the Markov chain; (ii) points out a trade-off between two conflicting objectives: slow convergence to the optimal model, or fast convergence to a biased model, i.e., a model that minimizes an objective function different from the initial target. Guided by insights from the theoretical analysis,

we propose CA-Fed, an algorithm which dynamically assigns weights to clients and achieves a good trade-off between maximizing convergence speed and minimizing model bias. Interestingly, CA-Fed can decide to ignore clients with low availability and high temporal correlation. Our experimental results demonstrate that excluding such clients is a simple, but effective approach to handle the heterogeneous and correlated client availability in FL. Indeed, while CA-Fed achieves a comparable maximum accuracy as the state-of-the-art methods F3AST [18] and AdaFed [19], its test accuracy exhibits higher time-average and smaller variability over time.

The remainder of this paper is organized as follows. Section II describes the problem of correlated client availability in FL and discusses the main related works. Section III provides a convergence analysis of FedAvg under heterogeneous and correlated client participation. CA-Fed, our correlation-aware FL algorithm, is presented in Section IV. We evaluate CA-Fed in Section V, comparing it with state-of-the-art methods on synthetic and real-world data. Section VII concludes the paper.

II. Background and Related Works

We consider a finite set $\mathcal{K}$ of N clients. Each client $k \in \mathcal{K}$ holds a local dataset D_k. Clients aim to jointly learn the parameters $\boldsymbol{w} \in W \subseteq \mathbb{R}^d$ of a global ML model (e.g., the weights of a neural network architecture). During training, the quality of the model with parameters $\boldsymbol{w}$ on a data sample $\xi \in D_k$ is measured by a loss function $f(\boldsymbol{w};\xi)$. The clients solve, under the orchestration of a central server, the following optimization problem:

$$\min_{\boldsymbol{w}\in W\subseteq\mathbb{R}^d} \left[F(\boldsymbol{w}) \coloneqq \sum_{k\in\mathcal{K}} \alpha_k F_k(\boldsymbol{w}) \right], \tag{1}$$

where $F_k(\boldsymbol{w}) \coloneqq \frac{1}{|D_k|}\sum_{\xi\in D_k} f(\boldsymbol{w};\xi)$ is the average loss computed on client k's local dataset, and $\boldsymbol{\alpha} = (\alpha_k)_{k\in\mathcal{K}}$ are positive coefficients such that $\sum_k \alpha_k = 1$. They represent the *target importance* assigned by the central server to each client k. Typically $(\alpha_k)_{k\in\mathcal{K}}$ are set proportional to the clients' dataset size $|D_k|$, such that the objective function F in (1) coincides with the average loss computed on the union of the clients' local datasets $D = \cup_{k\in\mathcal{K}} D_k$.

Under proper assumptions, precised in Section III, Problem (1) admits a unique solution. We use $\boldsymbol{w}^*$ (resp. F^*) to denote the minimizer (resp. the minimum value) of F. Moreover, for $k\in\mathcal{K}$, F_k admits a unique minimizer on W. We use $\boldsymbol{w}_k^*$ (resp. F_k^*) to denote the minimizer (resp. the minimum value) of F_k.

Problem (1) is commonly solved through iterative algorithms [4], [8] requiring multiple communication rounds between the server and the clients. At round $t > 0$, the server broadcasts the latest estimate of the global model $\boldsymbol{w}_{t,0}$ to the set of available clients (A_t). Client $k \in A_t$ updates the global model with its local data through $E \geq 1$ steps of local Stochastic Gradient Descent (SGD):

$$\boldsymbol{w}_{t,j+1}^k = \boldsymbol{w}_{t,j}^k - \eta_t \nabla F_k(\boldsymbol{w}_{t,j}^k, \mathcal{B}_{t,j}^k) \quad j = 0,\dots,E-1, \tag{2}$$

where $\eta_t > 0$ is an appropriately chosen learning rate, referred to as *local learning rate*; $\mathcal{B}_{t,j}^k$ is a random batch sampled from client k' local dataset at round t and step j; $\nabla F_k(\cdot,\mathcal{B}) \coloneqq \frac{1}{|\mathcal{B}|}\sum_{\xi\in\mathcal{B}} \nabla f(\cdot,\xi)$ is an unbiased estimator of the local gradient ∇F_k. Then, each client sends its local model update $\Delta_t^k \coloneqq \boldsymbol{w}_{t,E}^k - \boldsymbol{w}_{t,0}^k$ to the server. The server computes $\Delta_t \coloneqq \sum_{k\in A_t} q_k \cdot \Delta_t^k$, a weighted average of the clients' local updates with non-negative *aggregation weights* $\boldsymbol{q} = (q_k)_{k\in\mathcal{K}}$. The choice of the aggregation weights defines an aggregation strategy (we will discuss different aggregation strategies later). The aggregated update Δ_t can be interpreted as a proxy for $-\nabla F(\boldsymbol{w}_{t,0})$; the server applies it to the global model:

$$\boldsymbol{w}_{t+1,0} = \mathbf{Proj}_W(\boldsymbol{w}_{t,0} + \eta_s \cdot \Delta_t) \tag{3}$$

where $\mathbf{Proj}_W(\cdot)$ denotes the projection over the set W, and $\eta_s > 0$ is an appropriately chosen learning rate, referred to as the *server learning rate*.[1]

The aggregate update Δ_t is, in general, a biased estimator of $-\nabla F(\boldsymbol{w}_{t,0})$, where each client k is taken into account proportionally to its frequency of appearance in the set A_t and to its aggregation weight q_k. Indeed, under proper assumptions specified in Section III, one can show (see Theorem 2) that the update rule described by (2) and (3) converges to the unique minimizer of a biased global objective F_B, which depends both on the clients' availability (i.e., on the sequence $(A_t)_{t>0}$) and on the aggregation strategy (i.e., on $\boldsymbol{q} = (q_k)_{k\in\mathcal{K}}$):

$$F_B(\boldsymbol{w}) \coloneqq \textstyle\sum_{k=1}^N p_k F_k(\boldsymbol{w}), \text{ with } p_k \coloneqq \frac{\pi_k q_k}{\sum_{h=1}^N \pi_h q_h}, \tag{4}$$

where $\pi_k \coloneqq \lim_{t\to\infty} \mathbb{P}(k \in A_t)$ is the asymptotic availability of client k. The coefficients $\boldsymbol{p} = (p_k)_{k\in\mathcal{K}}$ can be interpreted as the *biased importance* the server is giving to each client k during training, in general different from the *target importance* $\boldsymbol{\alpha}$. In what follows, $\boldsymbol{w}_B^*$ (resp. F_B^*) denotes the minimizer (resp. the minimum value) of F_B.

In some large-scale FL applications, like training Google keyboard next-word prediction models, each client participates in training at most for one round. The orchestrator usually selects a few hundred clients at each round for a few thousand rounds (e.g., see [5, Table 2]), but the available set of clients may include hundreds of millions of Android devices. In this scenario, it is difficult to address the potential bias unless there is some a-priori information about each client's availability. Anyway, FL can be used by service providers with access to a much smaller set of clients (e.g., smartphone users that have installed a specific app). In this case, a client participates multiple times in training: the orchestrating server may keep track of each client's availability and try to compensate for the potentially dangerous heterogeneity in their participation.

Much previous effort on federated learning [4], [17]–[19], [22]–[25] considered this problem and, under different as-

[1] The aggregation rule (3) has been considered also in other works, e.g., [8], [20], [21]. In other FL algorithms, the server computes an average of clients' local models. This aggregation rule can be obtained with minor changes to (3).

sumptions on the clients' availability (i.e., on $(A_t)_{t>0}$), designed aggregation strategies that unbias Δ_t through an appropriate choice of $\boldsymbol{q}$. Reference [22] provides the first analysis of `FedAvg` on non-iid data under clients' partial participation. Their analysis covers both the case when active clients are sampled uniformly at random without replacement from $\mathcal{K}$ and assigned aggregation weights equal to their target importance (as assumed in [4]), and the case when active clients are sampled iid with replacement from $\mathcal{K}$ with probabilities $\boldsymbol{\alpha}$ and assigned equal weights (as assumed in [23]). However, references [4], [22], [23] ignore the variance induced by the clients stochastic availability. The authors of [24] reduce such variance by considering only the clients with important updates, as measured by the value of their norm. References [17] and [25] reduce the aggregation variance through clustered and soft-clustered sampling, respectively.

Some recent works [18], [19], [26] do not actively pursue the optimization of the unbiased objective. Instead, they derive bounds for the convergence error and propose heuristics to minimize those bounds, potentially introducing some bias. Our work follows a similar development: we compare our algorithm with `F3AST` from [18] and `AdaFed` from [19].

The novelty of our study is in considering the spatial and temporal correlation in clients' availability dynamics. As discussed in the introduction, such correlations are also introduced by clients' eligibility criteria, e.g., smartphones being under charge and connected to broadband networks. The effect of correlation has been ignored until now, probably due to the additional complexity in studying FL algorithms' convergence. To the best of our knowledge, the only exception is [18], which scratches the issue of spatial correlation by proposing two different algorithms for the case when clients' availabilities are uncorrelated and for the case when they are positively correlated (there is no smooth transition from one algorithm to the other as a function of the degree of correlation).

The effect of temporal correlation on *centralized* stochastic gradient methods has been addressed in [12]–[14], [27]: these works study a variant of stochastic gradient descent where samples are drawn according to a Markov chain. Reference [12] extends its analysis to a FL setting where each client draws samples according to a Markov chain. In contrast, our work does not assume a correlation in the data sampling but rather in the client's availability. Nevertheless, some of our proof techniques are similar to those used in this line of work and, in particular, we rely on some results in [14].

III. Analysis

A. Main assumptions

We consider a time-slotted system where a slot corresponds to one FL communication round. We assume that clients' availability over the timeslots $t \in \mathbb{N}$ follows a discrete-time Markov chain $(A_t)_{t\geq 0}$.[2]

[2]In Section III-D we will focus on the case where this chain is the superposition of N independent Markov chains, one for each client.

Assumption 1. *The Markov chain $(A_t)_{t\geq 0}$ on the finite state space $[M]$ is time-homogeneous, irreducible, and aperiodic. It has transition matrix $\boldsymbol{P}$ and stationary distribution $\boldsymbol{\pi}$.*

Markov chains have already been used in the literature to model the dynamics of stochastic networks where some nodes or edges in the graph can switch between active and inactive states [28], [29]. The previous Markovian assumption, while allowing a great degree of flexibility, still guarantees the analytical tractability of the system. The distance dynamics between current and stationary distribution of the Markov process can be characterized by the spectral properties of its transition matrix $\boldsymbol{P}$ [30]. Let $\lambda_2(\boldsymbol{P})$ denote the the second largest eigenvalue of $\boldsymbol{P}$ in absolute value. Previous works [14] have shown that:

$$\max_{i,j\in[M]} |[\boldsymbol{P}^t]_{i,j} - \pi_j| \leq C_P \cdot \lambda(\boldsymbol{P})^t, \quad \text{for } t \geq T_P, \tag{5}$$

where the parameter $\lambda(\boldsymbol{P}) \coloneqq (\lambda_2(\boldsymbol{P}) + 1)/2$, and C_P, T_P are positive constants whose values are reported in [14, Lemma 1].[3] Note that $\lambda(\boldsymbol{P})$ quantifies the correlation of the Markov process $(A_t)_{t\geq 0}$: the closer $\lambda(\boldsymbol{P})$ is to one, the slower the Markov chain converges to its stationary distribution.

In our analysis, we make the following additional assumptions. Let $\boldsymbol{w}^*, \boldsymbol{w}_B^*$ denote the minimizers of F and F_B on W, respectively.

Assumption 2. *The hypothesis class W is convex, compact, and contains in its interior the minimizers $\boldsymbol{w}^*, \boldsymbol{w}_B^*, \boldsymbol{w}_k^*$.*

The following assumptions concern clients' local objective functions $\{F_k\}_{k\in\mathcal{K}}$. Assumptions 3 and 4 are standard in the literature on convex optimization [31, Sections 4.1, 4.2]. Assumption 5 is a standard hypothesis in the analysis of federated optimization algorithms [8, Section 6.1].

Assumption 3 (L-smoothness)**.** *The local functions $\{F_k\}_{k=1}^N$ have L-Lipschitz continuous gradients: $F_k(\boldsymbol{v}) \leq F_k(\boldsymbol{w}) + \langle \nabla F_k(\boldsymbol{w}), \boldsymbol{v} - \boldsymbol{w}\rangle + \frac{L}{2}\|\boldsymbol{v} - \boldsymbol{w}\|_2^2$, $\forall \boldsymbol{v}, \boldsymbol{w} \in W$.*

Assumption 4 (Strong convexity)**.** *The local functions $\{F_k\}_{k=1}^N$ are μ-strongly convex: $F_k(\boldsymbol{v}) \geq F_k(\boldsymbol{w}) + \langle \nabla F_k(\boldsymbol{w}), \boldsymbol{v} - \boldsymbol{w}\rangle + \frac{\mu}{2}\|\boldsymbol{v} - \boldsymbol{w}\|_2^2$, $\forall \boldsymbol{v}, \boldsymbol{w} \in W$.*

Assumption 5 (Bounded variance)**.** *The variance of stochastic gradients in each device is bounded: $\mathbb{E}\|\nabla F_k(\boldsymbol{w}_{t,j}^k, \xi_{t,j}^k) - \nabla F_k(\boldsymbol{w}_{t,j}^k)\|^2 \leq \sigma_k^2$, $k = 1, \ldots, N$.*

Assumptions 2–5 imply the following properties for the local functions, described by Lemma 1 (proof in Appendix B).

Lemma 1. *Under Assumptions 2–5, there exist constants D, G, and $H > 0$, such that, for $\boldsymbol{w} \in W$ and $k \in \mathcal{K}$, we have:*

$$\|\nabla F_k(\boldsymbol{w})\| \leq D, \tag{6}$$

$$\mathbb{E}\|\nabla F_k(\boldsymbol{w}, \xi)\|^2 \leq G^2, \tag{7}$$

$$|F_k(\boldsymbol{w}) - F_k(\boldsymbol{w}_B^*)| \leq H. \tag{8}$$

[3]Note that (5) holds for different definitions of $\lambda(\boldsymbol{P})$ as far as $\lambda(\boldsymbol{P}) \in (\lambda_2(\boldsymbol{P}), 1)$. The specific choice for $\lambda(\boldsymbol{P})$ changes the constants C_P and T_P.

Similarly to other works [8], [22], [23], [32], we introduce a metric to quantify the heterogeneity of clients' local datasets:

$$\Gamma \coloneqq \max_{k\in\mathcal{K}}\{F_k(\boldsymbol{w}^*) - F_k^*\}. \tag{9}$$

If the local datasets are identical, the local functions $\{F_k\}_{k\in\mathcal{K}}$ coincide among them and with F, $\boldsymbol{w}^*$ is a minimizer of each local function, and $\Gamma = 0$. In general, Γ is smaller the closer the distributions the local datasets are drawn from.

B. Main theorems

Theorem 1 (proof in Appendix A) decomposes the error of the target global objective as the sum of an optimization error for the biased global objective and a bias error.

Theorem 1 (Decomposing the total error)**.** *Under Assumptions 2–4, the optimization error of the target global objective $\epsilon = F(\boldsymbol{w}) - F^*$ can be bounded as follows:*

$$\epsilon \leq \underbrace{2\kappa^2(F_B(\boldsymbol{w}) - F_B^*)}_{\coloneqq\epsilon_{opt}} + \underbrace{2\kappa^4\chi^2_{\boldsymbol{\alpha}\|\boldsymbol{p}}\Gamma}_{\coloneqq\epsilon_{bias}}, \tag{10}$$

where $\kappa \coloneqq L/\mu$, and $\chi^2_{\boldsymbol{\alpha}\|\boldsymbol{p}} \coloneqq \sum_{k=1}^N (\alpha_k - p_k)^2/p_k$.

Theorem 2 below proves that the optimization error ϵ_{opt} associated to the biased objective F_B, evaluated on the trajectory determined by scheme (3), asymptotically vanishes. The non-vanishing bias error ϵ_{bias} captures the discrepancy between $F(\boldsymbol{w})$ and $F_B(\boldsymbol{w})$. This latter term depends on the chi-square divergence $\chi^2_{\boldsymbol{\alpha}\|\boldsymbol{p}}$ between the target and biased probability distributions $\boldsymbol{\alpha} = (\alpha_k)_{k\in\mathcal{K}}$ and $\boldsymbol{p} = (p_k)_{k\in\mathcal{K}}$, and on Γ, that quantifies the degree of heterogeneity of the local functions. When all local functions are identical ($\Gamma = 0$), the bias term ϵ_{bias} also vanishes. For $\Gamma > 0$, the bias error can still be controlled by the aggregation weights assigned to the devices. In particular, the bias term vanishes when $q_k \propto \alpha_k/\pi_k, \forall k \in \mathcal{K}$. Since it asymptotically cancels the bias error, we refer to this choice as *unbiased aggregation strategy*.

However, in practice, FL training is limited to a finite number of iterations T (typically a few hundreds [5], [7]), and the previous asymptotic considerations may not apply. In this regime, the unbiased aggregation strategy can be suboptimal, since the minimization of ϵ_{bias} not necessarily leads to the minimization of the total error $\epsilon \leq \epsilon_{\text{opt}} + \epsilon_{\text{bias}}$. This motivates the analysis of the optimization error ϵ_{opt}.

Theorem 2 (Convergence of the optimization error ϵ_{opt})**.** *Let Assumptions 1–5 hold and the constants M, L, D, G, H, Γ, $\sigma_k, C_P, T_P, \lambda(\boldsymbol{P})$ be defined as above. Let $Q = \sum_{k\in\mathcal{K}} q_k$. Let the stepsizes satisfy:*

$$\textstyle\sum_t \eta_t = +\infty, \quad \sum_t \ln(t)\cdot\eta_t^2 < +\infty. \tag{11}$$

Let T denote the total communication rounds. For $T \geq T_P$, the expected optimization error can be bounded as follows:

$$\mathbb{E}[F_B(\bar{\boldsymbol{w}}_{T,0}) - F_B^*] \leq \frac{\frac{\frac{1}{2}\boldsymbol{q}^\intercal\boldsymbol{\Sigma}\boldsymbol{q}+\upsilon}{\boldsymbol{\pi}^\intercal\boldsymbol{q}} + \psi + \frac{\phi}{\ln(1/\lambda(\boldsymbol{P}))}}{(\sum_{t=1}^T \eta_t)}, \tag{12}$$

where $\bar{\boldsymbol{w}}_{T,0} \coloneqq \frac{\sum_{t=1}^T \eta_t \boldsymbol{w}_{t,0}}{\sum_{t=1}^T \eta_t}$, and

$$\begin{aligned}
\boldsymbol{\Sigma} &= \operatorname{diag}(\sigma_k^2\pi_k \textstyle\sum_t \eta_t^2),\\
\upsilon &= \tfrac{2}{E}\left\|\boldsymbol{w}_{0,0} - \boldsymbol{w}^*\right\|^2 + \tfrac{1}{4}MQ\textstyle\sum_t(\eta_t^2 + \frac{1}{t^2}),\\
\psi &= 4L(EQ+2)\Gamma\textstyle\sum_t \eta_t^2 + \frac{2}{3}(E-1)(2E-1)G^2\sum_t \eta_t^2,\\
\mathcal{J}_t &= \min\left\{\max\left\{\lceil \ln(2C_PHt)/\ln(1/\lambda(\boldsymbol{P}))\rceil, T_P\right\}, t\right\},\\
\phi &= 2EDGQ\textstyle\sum_t \ln(2C_PHt)\eta_{t-\mathcal{J}_t}^2.
\end{aligned}$$

Theorem 2 (proof in Appendix B) proves convergence of the expected biased objective F_B to its minimum F_B^* under correlated client participation. Our bound (12) captures the effect of correlation through the factor $\ln(1/\lambda(\boldsymbol{P}))$: a high correlation worsens the convergence rate. In particular, we found that the numerator of (12) has a quadratic-over-linear fractional dependence on $\boldsymbol{q}$. Minimizing ϵ_{opt} leads, in general, to a different choice of $\boldsymbol{q}$ than minimizing ϵ_{bias}.

C. Minimizing the total error $\epsilon \leq \epsilon_{opt} + \epsilon_{bias}$

Our analysis points out a trade-off between minimizing ϵ_{opt} or ϵ_{bias}. Our goal is to find the optimal aggregation weights $\boldsymbol{q}^*$ that minimize the upper bound on total error $\epsilon(\boldsymbol{q})$ in (10):

$$\begin{aligned}
\underset{\boldsymbol{q}}{\text{minimize}} \quad & \epsilon_{\text{opt}}(\boldsymbol{q}) + \epsilon_{\text{bias}}(\boldsymbol{q});\\
\text{subject to} \quad & \boldsymbol{q} \geq 0,\\
& \|\boldsymbol{q}\|_1 = Q.
\end{aligned} \tag{13}$$

In Appendix E we prove that (13) is a convex optimization problem, which can be solved with the method of Lagrange multipliers. However, the solution is not of practical utility because the constants in (10) and (12) (e.g., L, μ, Γ, C_P) are in general problem-dependent and difficult to estimate during training. In particular, Γ poses particular difficulties as it is defined in terms of the minimizer of the target objective F, but the FL algorithm generally minimizes the biased function F_B. Moreover, the bound in (10), similarly to the bound in [32], diverges when setting some q_k equal to 0, but this is simply an artifact of the proof technique. A result of more practical interest is the following (proof in Appendix C):

Theorem 3 (An alternative decomposition of the total error ϵ)**.** *Under the same assumptions of Theorem 1, let $\Gamma' \coloneqq \max_k\{F_k(\boldsymbol{w}_B^*) - F_k^*\}$. The following result holds:*

$$\epsilon \leq \underbrace{2\kappa^2(F_B(\boldsymbol{w}) - F_B^*)}_{\coloneqq\epsilon_{opt}} + \underbrace{8\kappa^4 d_{TV}^2(\boldsymbol{\alpha},\boldsymbol{p})\Gamma'}_{\coloneqq\epsilon'_{bias}}, \tag{14}$$

where $d_{TV}(\boldsymbol{\alpha},\boldsymbol{p}) \coloneqq \frac{1}{2}\sum_{k=1}^N|\alpha_k - p_k|$ is the total variation distance between the probability distributions $\boldsymbol{\alpha}$ and $\boldsymbol{p}$.

The new constant Γ' is defined in terms of $\boldsymbol{w}_B^*$, and then it is easier to evaluate during training. However, Γ' depends on $\boldsymbol{q}$, because it is evaluated at the point of minimum of F_B. This dependence makes the minimization of the right-hand side of (14) more challenging (for example, the corresponding problem is not convex). We study the minimization of the two terms ϵ_{opt} and ϵ'_{bias} separately and learn some insights, which we use to design the new FL algorithm `CA-Fed`.

D. Minimizing ϵ_{opt}

The minimization of ϵ_{opt} is still a convex optimization problem (Appendix D). In particular, at the optimum non-negative weights are set accordingly to $q_k^* = a(\lambda^* \pi_k - \theta^*)$ with a, λ^*, and θ^* positive constants (see (29)). It follows that clients with smaller availability get smaller weights in the aggregation. In particular, this suggests that clients with the smallest availability can be excluded from the aggregation, leading to the following guideline:

Guideline A: to speed up the convergence, we can exclude, i.e., set $q_k^* = 0$, the clients with lowest availability π_k.

This guideline can be justified intuitively: updates from clients with low participation may be too sporadic to allow the FL algorithm to keep track of their local objectives. They act as a noise slowing down the algorithm's convergence. It may be advantageous to exclude these clients from participating.

We observe that the choice of the aggregation weights $\boldsymbol{q}$ does not affect the clients' availability process and, in particular, $\lambda(\boldsymbol{P})$. However, if the algorithm excludes some clients, it is possible to consider the state space of the Markov chain that only specifies the availability state of the remaining clients, and this Markov chain may have different spectral properties. For the sake of concreteness, we consider here (and in the rest of the paper) the particular case when the availability of each client k evolves according to a two-states Markov chain $(A_t^k)_{t\geq 0}$ with transition probability matrix $\boldsymbol{P_k}$ and these Markov chains are all independent. In this case, the aggregate process is described by the product Markov chain $(A_t)_{t\geq 0}$ with transition matrix $\boldsymbol{P} = \bigotimes_{k\in\mathcal{K}} \boldsymbol{P_k}$ and $\lambda(\boldsymbol{P}) = \max_{k\in\mathcal{K}} \lambda(\boldsymbol{P_k})$, where $\boldsymbol{P_i} \bigotimes \boldsymbol{P_j}$ denotes the Kronecker product between matrices $\boldsymbol{P_i}$ and $\boldsymbol{P_j}$ [30, Exercise 12.6]. In this setting, it is possible to redefine the Markov chain $(A_t)_{t\geq 0}$ by taking into account the reduced state space defined by the clients with a non-null aggregation weight, i.e., $\boldsymbol{P'} = \bigotimes_{k'\in\mathcal{K}|q_{k'}>0} \boldsymbol{P_{k'}}$ and $\lambda(\boldsymbol{P'}) = \max_{k'\in\mathcal{K}|q_{k'}>0} \lambda(\boldsymbol{P_{k'}})$, which is potentially smaller than the case when all clients participate to the aggregation. These considerations lead to the following guideline:

Guideline B: to speed up the convergence, we can exclude, i.e., set $q_k^* = 0$, the clients with largest $\lambda(\boldsymbol{P_k})$.

Intuition also supports this guideline. Clients with large $\lambda(\boldsymbol{P_k})$ tend to be available or unavailable for long periods of time. Due to the well-known catastrophic forgetting problem affecting gradient methods [33], [34], these clients may unfairly steer the algorithm toward their local objective when they appear at the final stages of the training period. Moreover, their participation in the early stages may be useless, as their contribution will be forgotten during their long absence. The FL algorithm may benefit from directly neglecting such clients.

We observe that guideline B strictly applies to this specific setting where clients' dynamics are independent (and there is no spatial correlation). We do not provide a corresponding guideline for the case when clients are spatially correlated (we leave this task for future research). However, in this more general setting, it is possible to ignore guideline B but still draw on guidelines A and C, or still consider guideline B if clients are spatially correlated (see discussion in Section VI-B).

Algorithm 1: `CA-Fed` (Correlation-Aware FL)

Input : $\boldsymbol{w}_{0,0}$, $\boldsymbol{\alpha}$, $\boldsymbol{q}^{(0)}$, $\{\eta_t\}_{t=1}^T$, η_s, E, β, τ
Initialize $\hat{\boldsymbol{F}}^{(0)}$, $\hat{\boldsymbol{F}}^*$, $\hat{\Gamma}'^{(0)}$, $\hat{\boldsymbol{\pi}}^{(0)}$, and $\hat{\boldsymbol{\lambda}}^{(0)}$;
for $t = 1, \dots, T$ **do**
  Receive set of active client A_t, loss vector $\boldsymbol{F}^{(t)}$;
  Update $\hat{\boldsymbol{F}}^{(t)}$, $\hat{\Gamma}'^{(t)}$, $\hat{\boldsymbol{\pi}}^{(t)}$, and $\hat{\boldsymbol{\lambda}}^{(t)}$;
  Initialize $\boldsymbol{q}^{(t)} = \frac{\boldsymbol{\alpha}}{\hat{\boldsymbol{\pi}}^{(t)}}$;
  $\boldsymbol{q}^{(t)} \leftarrow$ `get` $(\boldsymbol{q}^{(t)}, \boldsymbol{\alpha}, \hat{\boldsymbol{F}}^{(t)}, \hat{\boldsymbol{F}}^*, \hat{\Gamma}'^{(t)}, \hat{\boldsymbol{\pi}}^{(t)}, \hat{\boldsymbol{\lambda}}^{(t)})$;
  $\boldsymbol{q}^{(t)} \leftarrow$ `get` $(\boldsymbol{q}^{(t)}, \boldsymbol{\alpha}, \hat{\boldsymbol{F}}^{(t)}, \hat{\boldsymbol{F}}^*, \hat{\Gamma}'^{(t)}, \hat{\boldsymbol{\pi}}^{(t)}, -\hat{\boldsymbol{\pi}}^{(t)})$;
  for *client* $\{k \in A_t;\ q_k^{(t)} > 0\}$, *in parallel* **do**
    for $j = 0, \dots, E-1$ **do**
      $\boldsymbol{w}_{t,j+1}^k = \boldsymbol{w}_{t,j}^k - \eta_t \nabla F_k(\boldsymbol{w}_{t,j}^k, \mathcal{B}_{t,j}^k)$;
    $\Delta_t^k \leftarrow \boldsymbol{w}_{t,E} - \boldsymbol{w}_{t,0}$;
  $\boldsymbol{w}_{t+1,0} \leftarrow \mathbf{Proj}_W(\boldsymbol{w}_{t,0} + \eta_s \sum_{k\in A_t} q_k^{(t)} \cdot \Delta_t^k)$;
Function `get` $(\boldsymbol{q}, \boldsymbol{\alpha}, \boldsymbol{F}, \boldsymbol{F}^*, \Gamma, \boldsymbol{\pi}, \boldsymbol{\rho})$ **:**
  $\mathcal{K} \leftarrow$ sort by descending order in $\boldsymbol{\rho}$;
  $\hat{\epsilon} \leftarrow \langle \boldsymbol{F} - \boldsymbol{F}^*, \boldsymbol{\pi} \tilde{\odot} \boldsymbol{q} \rangle + d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{\pi} \tilde{\odot} \boldsymbol{q}) \cdot \Gamma$;
  for $k \in \mathcal{K}$ **do**
    $q_k^+ \leftarrow 0$;
    $\hat{\epsilon}^+ \leftarrow \langle \boldsymbol{F} - \boldsymbol{F}^*, \boldsymbol{\pi} \tilde{\odot} \boldsymbol{q}^+ \rangle + d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{\pi} \tilde{\odot} \boldsymbol{q}^+) \cdot \Gamma$;
    if $\hat{\epsilon} - \hat{\epsilon}^+ \geq \tau$ **then**
      $\hat{\epsilon} \leftarrow \hat{\epsilon}^+$;
      $\boldsymbol{q} \leftarrow \boldsymbol{q}^+$;
  return $\boldsymbol{q}$

E. Minimizing ϵ'_{bias}

The bias error ϵ'_{bias} in (14) vanishes when the total variation distance between the target importance $\boldsymbol{\alpha}$ and the biased importance $\boldsymbol{p}$ is zero, i.e., when $q_k \propto \alpha_k/\pi_k, \forall k \in \mathcal{K}$. Then, after excluding the clients that contribute the most to the optimization error and particularly slow down the convergence (guidelines A and B), we can assign to the remaining clients an aggregation weight inversely proportional to their availability, such that the bias error ϵ'_{bias} is minimized.

Guideline C: to reduce the bias error, we set $q_k^* \propto \alpha_k/\pi_k$ for the clients that are not excluded by the previous guidelines.

IV. Proposed Algorithm

Guidelines A and B in Section III suggest that the minimization of ϵ_{opt} can lead to the exclusion of some available clients from the aggregation step (3), in particular those with low availability and/or high correlation. For the remaining clients, guideline C proposes to set their aggregation weight inversely proportional to their availability to reduce the bias error ϵ'_{bias}. Motivated by these insights, we propose `CA-Fed`, a client sampling and aggregation strategy that takes into account the problem of correlated client availability in FL, described in

Algorithm 1. CA-Fed learns during training which are the clients to exclude and how to set the aggregation weights of the other clients to achieve a good trade-off between ϵ_{opt} and ϵ'_{bias}. While guidelines A and B indicate which clients to remove, the exact number of clients to remove at round t is identified by minimizing $\epsilon^{(t)}$ as a proxy for the bound in (14):[4]

$$\epsilon^{(t)} \coloneqq F_B(\boldsymbol{w}_{t,0}) - F_B^* + d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{p})\Gamma'. \tag{15}$$

A. CA-Fed's core steps

At each communication round t, the server sends the current model $\boldsymbol{w}_{t,0}$ to all active clients and each client k sends back a noisy estimate $F_k^{(t)}$ of the current loss computed on a batch of samples $\mathcal{B}_{t,0}^k$, i.e., $F_k^{(t)} = \frac{1}{|\mathcal{B}_{t,0}^k|}\sum_{\xi\in\mathcal{B}_{t,0}^k} f(\boldsymbol{w}_{t,0}, \xi)$ (line 3). The server uses these values and the information about the current set of available clients A_t to refine its own estimates of each client's loss ($\hat{\boldsymbol{F}}^{(t)} = (\hat{F}_k^{(t)})_{k\in\mathcal{K}}$), and each client's loss minimum value ($\hat{\boldsymbol{F}}^* = (\hat{F}_k^*)_{k\in\mathcal{K}}$), as well as of Γ', π_k, λ_k, and $\epsilon^{(t)}$, denoted as $\hat{\Gamma}'^{(t)}$, $\hat{\pi}_k^{(t)}$, $\hat{\lambda}_k^{(t)}$, and $\hat{\epsilon}^{(t)}$, respectively (possible estimators are described below) (line 4).

The server decides whether excluding clients whose availability pattern exhibits high correlation (high $\hat{\lambda}_k^{(t)}$) (line 6). First, the server considers all clients in descending order of $\hat{\boldsymbol{\lambda}}^{(t)}$ (line 14), and evaluates if, by excluding them (line 17), $\hat{\epsilon}^{(t)}$ appears to be decreasing by more than a threshold $\tau \geq 0$ (line 19). Then, the server considers clients in ascending order of $\hat{\boldsymbol{\pi}}^{(t)}$, and repeats the same procedure to possibly exclude some of the clients with low availability (low $\hat{\pi}_k^{(t)}$) (lines 7).

Once the participating clients (those with $q_k > 0$) have been selected, the server notifies them to proceed updating the current models (lines 9–10) according to (2), while the other available clients stay idle. Finally, model's updates are aggregated according to (3) (line 12).

B. Estimators

We now briefly discuss possible implementation of the estimators $\hat{F}_k^{(t)}$, $\hat{F}_k^*$, $\hat{\Gamma}'^{(t)}$, $\hat{\pi}_k^{(t)}$, and $\hat{\lambda}_k^{(t)}$. Server's estimates for the clients' local losses ($\hat{\boldsymbol{F}}^{(t)} = (\hat{F}_k^{(t)})_{k\in\mathcal{K}}$) can be obtained from the received active clients' losses ($\boldsymbol{F}^{(t)} = (F_k^{(t)})_{k\in A_t}$) through an auto-regressive filter with parameter $\beta \in (0, 1]$:

$$\hat{\boldsymbol{F}}^{(t)} = (\mathbf{1} - \beta\mathbb{1}_{A_t}) \odot \hat{\boldsymbol{F}}^{(t-1)} + \beta\mathbb{1}_{A_t} \odot \boldsymbol{F}^{(t)}, \tag{16}$$

where $\odot$ denotes the component-wise multiplication between vectors, and $\mathbb{1}_{A_t}$ is a N-dimensions binary vector whose k-th component equals 1 if and only if k is active at round t, i.e., $k \in A_t$. The server can keep track of the clients' loss minimum values and estimate F_k^* as $\hat{F}_k^* = \min_{s\in[0,t]} \hat{F}_k^{(s)}$. The values of $F_B(\boldsymbol{w}_{t,0})$, F_B^*, Γ', and $\epsilon^{(t)}$ can be estimated as follows:

$$\hat{F}_B^{(t)} - \hat{F}_B^* = \langle \hat{\boldsymbol{F}}^{(t)} - \hat{\boldsymbol{F}}^*, \hat{\boldsymbol{\pi}}^{(t)} \tilde{\odot} \boldsymbol{q}^{(t)} \rangle, \tag{17}$$

$$\hat{\Gamma}'^{(t)} = \max_{k\in\mathcal{K}} (\hat{F}_k^{(t)} - \hat{F}_k^*), \tag{18}$$

$$\hat{\epsilon}^{(t)} = \hat{F}_B^{(t)} - \hat{F}_B^* + d_{TV}^2(\boldsymbol{\alpha}, \hat{\boldsymbol{\pi}}^{(t)} \tilde{\odot} \boldsymbol{q}^{(t)}) \cdot \hat{\Gamma}'^{(t)}. \tag{19}$$

[4]Following (14), one could reasonably introduce a hyper-parameter to weigh the relative importance of the optimization and bias terms in the sum. We discuss this additional optimization of CA-Fed in Section VI-A.

where $\boldsymbol{\pi} \tilde{\odot} \boldsymbol{q} \in \mathbb{R}^N$, such that $\left(\boldsymbol{\pi} \tilde{\odot} \boldsymbol{q}\right)_k = \frac{\pi_k q_k}{\sum_{h=1}^N \pi_h q_h},\ k \in \mathcal{K}$.

For $\hat{\pi}_k^{(t)}$, the server can simply keep track of the total number of times client k was available up to time t and compute $\hat{\pi}_k^{(t)}$ using a Bayesian estimator with beta prior, i.e., $\hat{\pi}_k^{(t)} = (\sum_{s\leq t} \mathbb{1}_{k\in A_s} + n_k)/(t + n_k + m_k)$, where n_k and m_k are the initial parameters of the beta prior.

For $\hat{\lambda}_k^{(t)}$, the server can assume the client's availability evolves according to a Markov chain with two states (available and unavailable), track the corresponding number of state transitions, and estimate the transition matrix $\hat{\boldsymbol{P}}_k^{(t)}$ through a Bayesian estimator similarly to what done for $\hat{\pi}_k^{(t)}$. Finally, $\hat{\lambda}_k^{(t)}$ is obtained computing the eigenvalues of $\hat{\boldsymbol{P}}_k^{(t)}$.

C. CA-Fed's computation/communication cost

CA-Fed aims to improve training convergence and not to reduce its computation and communication overhead. Nevertheless, excluding some available clients reduces the overall training cost, as we will discuss in this section referring, for the sake of concreteness, to neural networks' training.

The available clients not selected for training are only requested to evaluate their local loss on the current model once on a single batch instead than performing E gradient updates, which would require roughly $2 \times E - 1$ more calculations (because of the forward and backward pass). For the selected clients, there is no extra computation cost as computing the loss corresponds to the forward pass they should, in any case, perform during the first local gradient update.

In terms of communication, the excluded clients only transmit the loss, a single scalar, much smaller than the model update. Conversely, participating clients transmit the local loss and the model update. Still, this additional overhead is negligible and likely fully compensated by the communication savings for the excluded clients.

V. Experimental Evaluation

A. Experimental Setup

a) Federated system simulator: In our experiments, we simulate the clients' availability dynamics featuring different levels of temporal correlations. We model the activity of each client as a two-state homogeneous Markov process with state space $\mathcal{S} = \{$"*active*","*inactive*"$\}$. We use $p_{k,s}$ to denote the probability that client $k \in \mathcal{K}$ remains in state $s \in \mathcal{S}$.

In order to simulate the statistical heterogeneity present in the federated learning system, we consider an experimental setting with two disjoint groups of clients $G_i,\ i = 1, 2$, to which we associate two different data distributions $\mathcal{P}_i,\ i = 1, 2$, to be precised later. Let $r_i = |G_i|/N,\ i = 1, 2$ denote the fraction of clients in group $i = 1, 2$. In order to simulate the heterogeneity of clients' availability patterns in realistic federated systems, we split the clients of each group in two classes uniformly at random: "more available" clients whose steady-state probability to be active is $\pi_{k,\text{active}} = 1/2 + g$ and "less available" clients with $\pi_{k,\text{active}} = 1/2 - g$, where $g \in$

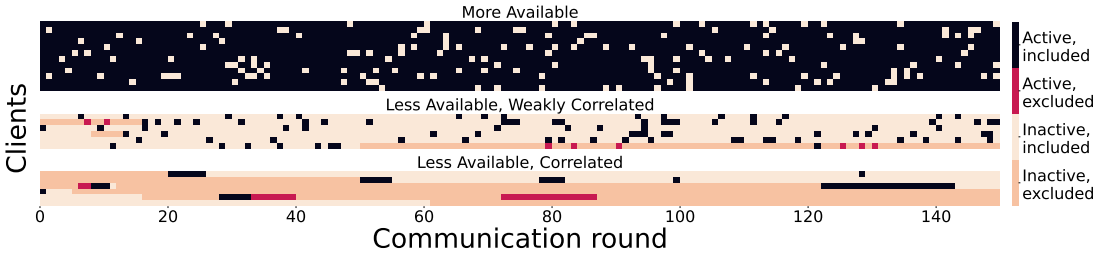


Fig. 1: Clients' activities and CA-Fed's clients selection on the synthetic dataset.

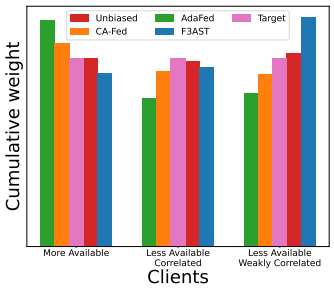
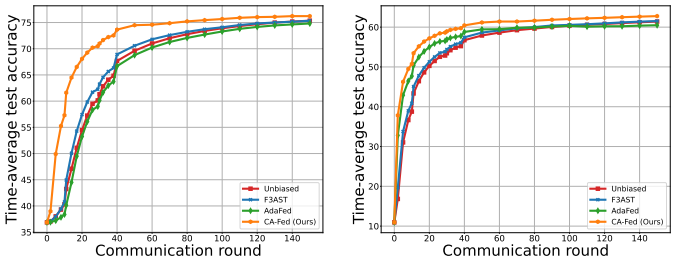


Fig. 2: Importance given to the clients by the different algorithms throughout a whole training process on the synthetic dataset.



(a) Synthetic (b) MNIST

Fig. 3: Test accuracy vs number of communication rounds.

$(0, 1/2)$ is a parameter controlling the heterogeneity of clients availability. We furthermore split each class of clients in two sub-classes uniformly at random: "correlated" clients that tend to persist in the same state ($\lambda_k = \nu$ with values of ν close to 1), and "weakly correlated" clients that are almost as likely to keep as to change their state ($\lambda_k \sim \mathcal{N}(0, \varepsilon^2)$, with ε close to 0). In our experiments, we suppose that $r_1 = r_2 = 1/2$, $g = 0.4$, $\nu = 0.9$, and $\varepsilon = 10^{-2}$.

b) Datasets and models: All experiments are performed on a binary classification synthetic dataset (described in Appendix F) and on the real-world MNIST dataset [35], using $N = 24$ clients. For MNIST dataset, we introduce statistical heterogeneity across the two groups of clients (i.e., we make the two distributions $\mathcal{P}_1$ and $\mathcal{P}_2$ different), following the same approach in [36]: 1) every client is assigned a random subset of the total training data; 2) the data of clients from the second group is modified by randomly swapping two pairs of labels. We maintain the original training/test data split of MNIST and use 20% of the training dataset as validation dataset. We use a linear classifier with a ridge penalization of parameter 10^{-2}, which is a strongly convex objective function, for both the synthetic and the real-world MNIST datasets.

c) Benchmarks: We compare CA-Fed, defined in Algorithm 1, with the Unbiased aggregation strategy, where all the active clients participate and receive a weight inversely proportional to their availability, and with the state-of-the-art FL algorithms discussed in Section II: F3AST [18] and AdaFed [19]. We tuned the learning rates η, η_s via grid search, on the grid $\eta : \{10^{-3}, 10^{-2.5}, 10^{-2}, 10^{-1.5}, 10^{-1}\}$, $\eta_s : \{10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}, 10^{0}\}$. For CA-Fed, we used $\tau = 0$, $\beta = 0.2$. We assume all algorithms can access an oracle providing the true availability parameters for each client. In practice, Unbiased, AdaFed, and F3AST rely on the exact knowledge of $\pi_{k,\text{active}}$, and CA-Fed on $\pi_{k,\text{active}}$ and λ_k. [5]

B. Experimental Results

Figure 1 shows the availability of each client during a training run on the synthetic dataset. Clients selected (resp. excluded) by CA-Fed are highlighted in black (resp. red). We observe that excluded clients tend to be those with low average availability or high correlation.

Figure 2 shows the importance p_k (averaged over time) given by different algorithms to each client k during a full training run. We observe that all the algorithms, except Unbiased, depart from the target importance $\boldsymbol{\alpha}$. As suggested by guidelines A and B, CA-Fed tends to favor the group of "more available" clients, at the expense of the "less available" clients.

Figure 3 shows the time-average accuracy up to round t of the learned model averaged over three different runs. On both datasets, CA-Fed achieves the highest accuracy, which is about a percentage point higher than the second best algorithm (F3AST). Table I shows for each algorithm: the average over three runs of the maximum test accuracy achieved during training, the time-average test accuracy achieved during training, together with its standard deviation within the second half of the training period. Results show that while CA-Fed achieves a maximum accuracy which is comparable to the Unbiased baseline and state-of-the-art AdaFed and F3AST, it gets a higher time-average accuracy (1.24 percentage points) in comparison to the second best (F3AST), and a smaller standard deviation (1.5×) in comparison to the second best (F3AST).

[5]The authors have provided public access to their code and data at: https://github.com/arodio/CA-Fed.

TABLE I: Maximum and time-average test accuracy, together with their standard deviations, on the Synthetic / MNIST datasets.

	Test Accuracy		
	Maximum	Time-average	Standard deviation
Unbiased	78.94 / 64.87	75.32 / 61.39	0.48 / 1.09
F3AST	78.97 / 64.91	75.33 / 61.52	0.40 / 0.94
AdaFed	78.69 / 63.77	74.81 / 60.48	0.59 / 1.37
CA-Fed	**79.03** / **64.94**	**76.22** / **62.76**	**0.28** / **0.61**

VI. Discussion

In this section, we discuss some general concerns and remarks on our algorithm.

A. Controlling the number of excluded clients

Theorems 1 and 3 suggest that the condition number κ^2 can play a meaningful role in the minimization of the total error ϵ. Our algorithm uses a proxy ($\epsilon^{(t)}$) of the total error. To take into account the effect of κ^2, we can introduce a hyper-parameter that weights the relative importance of the optimization and bias error in (15):

$$\epsilon'^{(t)} \coloneqq F_B(\boldsymbol{w}_{t,0}) - F_B^* + \bar{\kappa}^2 \cdot d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{p})\Gamma'.$$

A small value of $\bar{\kappa}^2$ penalizes the bias term in favor of the optimization error, resulting in a larger number of clients excluded by CA-Fed. On the other hand, CA-Fed tends to include more clients for a large value of $\bar{\kappa}^2$. Asymptotically, for $\bar{\kappa}^2 \to +\infty$, CA-Fed reduces to the Unbiased baseline. To further improve the performance of CA-Fed, a finer tuning of the values of $\bar{\kappa}^2$ can be performed.

B. CA-Fed in presence of spatial correlation

Although CA-Fed is mainly designed to handle temporal correlation, it does not necessarily perform poorly in presence of spatial correlation, as well.

Consider the following spatially-correlated scenario: clients are grouped in clusters, each cluster $c \in \mathcal{C}$ is characterized by an underlying Markov chain, which determines when all clients in the cluster are available/unavailable, the Markov chains of different clusters are independent. Let λ_c denote the second largest eigenvalue in module of cluster-c's Markov chain. In this case, one needs to exclude all clients in the cluster $\bar{c} = \arg\max_{c \in \mathcal{C}} \lambda_c$ to reduce the eigenvalue of the aggregate Markov chain.

In this setting, CA-Fed would associate similar eigenvalue estimates to all clients in the same cluster, then it would correctly start considering for exclusion the clients in cluster $\bar{c}$ and potentially remove sequentially all clients in the same cluster. These considerations suggest that CA-Fed may still operate correctly even in presence of spatial correlation.

C. About CA-Fed's fairness

A strategy that excludes clients from the training phase, such as CA-Fed, may naturally raise fairness concerns. The concept of fairness in FL does not have a unified definition in the literature [37, Chapter 8]: fairness goals can be captured by a suitable choice of the target weights in (1). For example, per-client fairness can be achieved by setting α_k equal for every client, while per-sample fairness by setting α_k proportional to the local dataset size $|D_k|$. If we assume that the global objective in (1) indeed reflects also fairness concerns, then CA-Fed is intrinsically fair, in the sense that it guarantees that the performance objective of the learned model is as close as possible to its minimum value.

VII. Conclusion

This paper presented the first convergence analysis for a FedAvg-like FL algorithm under heterogeneous and correlated client availability. The analysis quantifies how correlation adversely affects the algorithm's convergence rate and highlights a general bias-versus-convergence-speed trade-off. Guided by the theoretical analysis, we proposed CA-Fed, a new FL algorithm that tries to balance the conflicting goals of maximizing convergence speed and minimizing model bias. Our experimental results demonstrate that adaptively excluding clients with high temporal correlation and low availability is an effective approach to handle the heterogeneous and correlated client availability in FL.

Appendix

A. Proof of Theorem 1

We bound the optimization error of the target objective as the optimization error of the biased objective plus a bias term:

$$\begin{aligned} F(\boldsymbol{w}) - F^* &\overset{(a)}{\leq} \tfrac{1}{2\mu} \left\|\nabla F(\boldsymbol{w})\right\|^2 \overset{(b)}{\leq} \tfrac{L^2}{2\mu} \left\|\boldsymbol{w} - \boldsymbol{w}^*\right\|^2 \\ &\overset{(c)}{\leq} \tfrac{L^2}{\mu}(\left\|\boldsymbol{w} - \boldsymbol{w}_B^*\right\|^2 + \left\|\boldsymbol{w}_B^* - \boldsymbol{w}^*\right\|^2) \\ &\overset{(d)}{\leq} \underbrace{\tfrac{2L^2}{\mu^2}(F_B(\boldsymbol{w}) - F_B^*)}_{:=\epsilon_{\text{opt}}} + \underbrace{\tfrac{2L^2}{\mu^2}(F(\boldsymbol{w}_B^*) - F^*)}_{:=\epsilon_{\text{bias}}}, \end{aligned}$$

where (a), (b), and (d) follow from the Assumptions 3, 4, and the inequality (c) follows from $(a+b)^2 \leq 2a^2 + 2b^2$. In particular, (b) requires $\nabla F_k(\boldsymbol{w}_k^*) = 0$. Theorem 2 further develops the optimization error ϵ_{opt}. We now expand ϵ_{bias}:

$$\begin{aligned} \left\|\nabla F(\boldsymbol{w}_B^*)\right\| &\overset{(e)}{=} \left\|\textstyle\sum_{k=1}^N (\alpha_k - p_k)\nabla F_k(\boldsymbol{w}_B^*)\right\| \\ &\overset{(f)}{\leq} L \textstyle\sum_{k=1}^N |\alpha_k - p_k| \left\|\boldsymbol{w}_B^* - \boldsymbol{w}_k^*\right\| \\ &\overset{(g)}{\leq} L\sqrt{\tfrac{2}{\mu}} \textstyle\sum_{k=1}^N \tfrac{|\alpha_k - p_k|}{\sqrt{p_k}} \sqrt{p_k(F_k(\boldsymbol{w}_B^*) - F_k^*)}, \end{aligned} \tag{20}$$

where (e) uses $\nabla F_B(\boldsymbol{w}_B^*) = 0$; (f) applies first the triangle inequality, then the L-smoothness, and (g) follows from the μ-strong convexity. In addition, (f) requires $\nabla F_k(\boldsymbol{w}_k^*) = 0$. Similarly to [32], in (g) we multiply numerator and denominator by $\sqrt{p_k}$. By direct calculations, it follows that:

$$\begin{aligned} \left\|\nabla F(\boldsymbol{w}_B^*)\right\|^2 &\overset{(h)}{\leq} \tfrac{2L^2}{\mu}\Big(\textstyle\sum_{k=1}^N \tfrac{|\alpha_k - p_k|}{\sqrt{p_k}} \sqrt{p_k(F_k(\boldsymbol{w}_B^*) - F_k^*)}\Big)^2 \\ &\overset{(i)}{\leq} \tfrac{2L^2}{\mu}\Big(\sum_{k=1}^N \tfrac{(\alpha_k - p_k)^2}{p_k}\Big)\Big(\sum_{k=1}^N p_k(F_k(\boldsymbol{w}_B^*) - F_k^*)\Big) \\ &\overset{(j)}{\leq} \tfrac{2L^2}{\mu}\chi^2_{\boldsymbol{\alpha}\|\boldsymbol{p}}\Gamma, \end{aligned}$$

where (i) uses the Cauchy–Schwarz inequality, and (j) used: $\sum_{k=1}^N p_k(F_k(\boldsymbol{w}_B^*) - F_k^*) \leq \sum_{k=1}^N p_k(F_k(\boldsymbol{w}^*) - F_k^*) \leq \Gamma$.

Finally, by strong convexity of F, we conclude that:

$$F(\boldsymbol{w}_B^*) - F^* \leq \frac{1}{2\mu} \left\|\nabla F(\boldsymbol{w}_B^*)\right\|^2 \leq \frac{L^2}{\mu^2}\chi^2_{\boldsymbol{\alpha}\|\boldsymbol{p}}\Gamma. \quad \square$$

B. Proof of Theorem 2

1) Additional notation: let $\boldsymbol{w}_{t,j}^k$ be the model parameter vector computed by device k at the global round t, local iteration j. We define:

$$g_t(A_t) = \textstyle\sum_{k\in A_t} q_k \sum_{j=0}^{E-1} \nabla F_k(\boldsymbol{w}_{t,j}^k, \xi_{t,j}^k),$$

and $\bar{g}_t(A_t) = \mathbb{E}_{\xi|A_t}[g_t(A_t)]$.

Following (2) and (3), the update rule of `CA-Fed` is:

$$\boldsymbol{w}_{t+1,0} = \mathbf{Proj}_W(\boldsymbol{w}_{t,0} - \eta_t g_t(A_t)). \tag{21}$$

2) Key lemmas and results: we provide useful lemmas and results to support the proof of the main theorem.

Proof of Lemma 1. The boundedness of W gives a bound on $(\boldsymbol{w}_{t,0})_{t\geq 0}$ based on the update rules in (2) and (3). From the convexity of $\{F_k\}_{k\in\mathcal{K}}$, it follows that:

$$D \coloneqq \sup_{\boldsymbol{w}\in W, k\in\mathcal{K}} \left\|\nabla F_k(\boldsymbol{w})\right\| < +\infty.$$

Items (6), (8) are directly derived from the previous observation. Item (7) follows combining (6) and Assumption 5:

$$\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}, \xi)\right\|^2 \leq D^2 + \max_{k\in\mathcal{K}}\{\sigma_k^2\} \coloneqq G^2. \quad \square$$

Lemma 2 (Convergence under heterogeneous client availability)**.** *Let the local functions $\{F_k\}_{k\in\mathcal{K}}$ be convex, Assumptions 3, 5 hold. If $\eta_t \leq \frac{1}{2L(EQ+1)}$, we have:*

$$\begin{aligned}\textstyle\sum_t \eta_t \,\mathbb{E}[\textstyle\sum_{k\in A_t} q_k \left(F_k(\boldsymbol{w}_{t,0}) - F_k(\boldsymbol{w}_B^*)\right)] \leq \\ + \tfrac{2}{E}\left\|\boldsymbol{w}_{0,0} - \boldsymbol{w}_B^*\right\|^2 + 2\textstyle\sum_{k=1}^N \pi_k q_k^2 \sigma_k^2 \sum_t \eta_t^2 \\ + \tfrac{2}{3}\textstyle\sum_{k=1}^N \pi_k q_k (E-1)(2E-1)G^2 \sum_t \eta_t^2 \\ + 2L(EQ+2)\textstyle\sum_{k=1}^N \pi_k q_k \Gamma \sum_t \eta_t^2 := C_1 < +\infty.\end{aligned}$$

Proof of Lemma 2.

$$\begin{aligned}\left\|\boldsymbol{w}_{t+1,0} - \boldsymbol{w}_B^*\right\|^2 &= \left\|\mathbf{Proj}_W(\boldsymbol{w}_{t,0} - \eta_t g_t) - \mathbf{Proj}_W(\boldsymbol{w}_B^*)\right\|^2 \\ &\leq \left\|\boldsymbol{w}_{t,0} - \eta_t g_t - \boldsymbol{w}_B^* + \eta_t\bar{g}_t - \eta_t\bar{g}_t\right\|^2 = A_1 + A_2 + A_3,\end{aligned}$$

where:

$$\begin{aligned}A_1 &= \left\|\boldsymbol{w}_{t,0} - \boldsymbol{w}_B^* - \eta_t\bar{g}_t\right\|^2, \\ A_2 &= 2\eta_t\langle \boldsymbol{w}_{t,0} - \boldsymbol{w}_B^* - \eta_t\bar{g}_t, \bar{g}_t - g_t\rangle, \\ A_3 &= \eta_t^2\left\|g_t - \bar{g}_t\right\|^2.\end{aligned}$$

Note $\mathbb{E}[A_2] = 0$. We bound A_1, A_3 using the key steps in [22]:

(1) the variance of $g_t(A_t)$ is bounded if the variance of the stochastic gradients at each device is bounded:

$$\begin{aligned}A_3 &= \mathbb{E}_{\mathcal{B}|A_t}\left\|g_t - \bar{g}_t\right\|^2 = \\ &= \textstyle\sum_{k\in A_t} q_k^2 \sum_{j=0}^{E-1} \mathbb{E}_{\mathcal{B}|A_t}\left\|\nabla F_k(\boldsymbol{w}_{t,j}^k, \xi_{t,j}^k) - \nabla F_k(\boldsymbol{w}_{t,j}^k)\right\|^2 \\ &\leq E\textstyle\sum_{k\in A_t} q_k^2\sigma_k^2;\end{aligned}$$

(2) the distance of the local model $\boldsymbol{w}_{t,E}^k$ from the global model $\boldsymbol{w}_{t,0}$ is bounded since the expected squared norm of the stochastic gradients is bounded:

$$\begin{aligned}&\mathbb{E}_{\mathcal{B}|A_t}\textstyle\sum_{k\in A_t} q_k \sum_{j=0}^{E-1}\left\|\boldsymbol{w}_{t,j}^k - \boldsymbol{w}_{t,0}\right\|^2 = \\ &= \mathbb{E}_{\mathcal{B}|A_t}\textstyle\sum_{k\in A_t} q_k \sum_{j=1}^{E-1}\eta_t^2\left\|\sum_{j'=0}^{j-1}\nabla F_k(\boldsymbol{w}_{t,j'}^k, \xi_{t,j'}^k)\right\|^2 \\ &\leq \eta_t^2\textstyle\sum_{k\in A_t} q_k \sum_{j=1}^{E-1} j \sum_{j'=0}^{j-1}\mathbb{E}_{\mathcal{B}|A_t}\left\|\nabla F_k(\boldsymbol{w}_{t,j'}^k, \xi_{t,j'}^k)\right\|^2 \\ &\leq \eta_t^2\textstyle\sum_{k\in A_t} q_k G^2 \sum_{j=1}^{E-1} j^2 \\ &= \tfrac{1}{6}\eta_t^2\textstyle\sum_{k\in A_t} q_k E(E-1)(2E-1)G^2. \quad \square\end{aligned}$$

Lemma 3 (Optimization error after $\mathcal{J}_t$ steps)**.** *Let Assumptions 1, 2 hold, the local functions $\{F_k\}_{k\in\mathcal{K}}$ be convex, D, H be defined as in* (6)*,* (8)*, and $\mathcal{J}_t$ defined as in Theorem 2. Then:*

$$\begin{aligned}&\textstyle\sum_t \eta_t\,\mathbb{E}[\textstyle\sum_{k\in A_t} q_k(F_k(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_k(\boldsymbol{w}_{t,0}))] \\ &\leq EDGQ\textstyle\sum_t \mathcal{J}_t\eta_{t-\mathcal{J}_t}^2\sum_{k=1}^N \pi_k q_k := \frac{C_3}{\ln(1/\lambda(\boldsymbol{P}))} < +\infty.\end{aligned}$$

For the proof of Lemma 3, we introduce the following results:

$$\left|F_k(\boldsymbol{v}) - F_k(\boldsymbol{w})\right| \leq D\cdot\left\|\boldsymbol{v} - \boldsymbol{w}\right\|, \ \forall \boldsymbol{v}, \boldsymbol{w} \in W, \tag{22}$$

$$\mathbb{E}_{\mathcal{B}_{t,0}^k,\ldots,\mathcal{B}_{t,E-1}^k}\left\|\boldsymbol{w}_{t+1,0} - \boldsymbol{w}_{t,0}\right\| \leq \eta_t GE(\textstyle\sum_{k\in A_t} q_k). \tag{23}$$

Equation (22) is due to convexity of $\{F_k\}_{k\in\mathcal{K}}$, which gives:

$$\langle\nabla F_k(\boldsymbol{v}), \boldsymbol{v} - \boldsymbol{w}\rangle \leq \left\|F_k(\boldsymbol{v}) - F_k(\boldsymbol{w})\right\| \leq \langle\nabla F_k(\boldsymbol{w}), \boldsymbol{v} - \boldsymbol{w}\rangle;$$

the Cauchy–Schwarz inequality concludes:

$$\begin{aligned}\left|F_k(\boldsymbol{v}) - F_k(\boldsymbol{w})\right| &\leq \max\{\left\|\nabla F_k(\boldsymbol{v})\right\|, \left\|\nabla F_k(\boldsymbol{w})\right\|\}\left\|\boldsymbol{v} - \boldsymbol{w}\right\| \\ &\leq D\cdot\left\|\boldsymbol{v} - \boldsymbol{w}\right\|.\end{aligned}$$

Equation (23) follows combining equations (7) and (21):

$$\begin{aligned}&\mathbb{E}_{\mathcal{B}|A_t}\left\|\boldsymbol{w}_{t+1,0} - \boldsymbol{w}_{t,0}\right\| \leq \\ &\leq \eta_t\,\mathbb{E}_{\mathcal{B}|A_t}\left\|\textstyle\sum_{k\in A_t} q_k\sum_{j=0}^{E-1}\nabla F_k(\boldsymbol{w}_{t,j}^k, \xi_{t,j}^k)\right\| \\ &\leq \eta_t\textstyle\sum_{k\in A_t} q_k\sum_{j=0}^{E-1}\mathbb{E}_{\mathcal{B}|A_t}\left\|\nabla F_k(\boldsymbol{w}_{t,j}^k, \xi_{t,j}^k)\right\| \\ &\leq \eta_t GE(\textstyle\sum_{k\in A_t} q_k).\end{aligned}$$

Proof of Lemma 3. The evolution of the local objectives after $\mathcal{J}_t$ communication rounds is bounded:

$$\begin{aligned}&\textstyle\sum_t\eta_t\,\mathbb{E}[\textstyle\sum_{k\in A_t} q_k(F_k(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_k(\boldsymbol{w}_{t,0}))] \\ &\overset{(a)}{\leq} D\textstyle\sum_t \eta_t\,\mathbb{E}[\textstyle\sum_{k\in A_t} q_k\,\mathbb{E}_B\left\|\boldsymbol{w}_{t-\mathcal{J}_t,0} - \boldsymbol{w}_{t,0}\right\|] \\ &\overset{(b)}{\leq} D\textstyle\sum_t \eta_t\sum_{d=t-\mathcal{J}_t}^{t-1}\mathbb{E}[\textstyle\sum_{k\in A_t} q_k\,\mathbb{E}_B\left\|\boldsymbol{w}_{d,0} - \boldsymbol{w}_{d+1,0}\right\|] \\ &\overset{(c)}{\leq} EDG\textstyle\sum_t\sum_{d=t-\mathcal{J}_t}^{t-1}\eta_t\eta_d\,\mathbb{E}[\textstyle\sum_{k\in A_t} q_k\sum_{k'\in A_d} q_{k'}] \\ &\overset{(d)}{\leq} \tfrac{EDG}{2}\textstyle\sum_t\sum_{d=t-\mathcal{J}_t}^{t-1}(\eta_t^2 + \eta_d^2)\,\mathbb{E}[\textstyle\sum_{k\in A_t} q_k\sum_{k'\in A_d} q_{k'}] \\ &\overset{(e)}{\leq} EDGQ\textstyle\sum_t \mathcal{J}_t\eta_{t-\mathcal{J}_t}^2\sum_{k=1}^N \pi_k q_k := \frac{C_3}{\ln(1/\lambda(\boldsymbol{P}))},\end{aligned}$$

where (a) follows from (22); (b) applies the triangle inequality; (c) uses (23); (d) applies the Cauchy–Schwarz inequality; (e) uses $\eta_t < \eta_d \leq \eta_{t-\mathcal{J}_t}$ and $\sum_{k=1}^N q_k = Q$. □

3) Core of the proof: The proof consists in two main steps:

1. $\sum_t \eta_t \sum_{k=1}^N \pi_k q_k \, \mathbb{E}[F_B(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_B^*)] \leq C_2 + \frac{C_3}{\ln(1/\lambda(\boldsymbol{P}))}$;

2. $\sum_t \eta_t \sum_{k=1}^N \pi_k q_k \, \mathbb{E}[F_B(\boldsymbol{w}_{t,0}) - F_B(\boldsymbol{w}_{t-\mathcal{J}_t,0})] \leq \frac{C_3}{\ln(1/\lambda(\boldsymbol{P}))}$.

Step 1. Combining Lemma 2 and 3, we get:

$$\textstyle\sum_t \eta_t \, \mathbb{E}[\sum\limits_{k\in A_t} q_k (F_k(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_k(\boldsymbol{w}_B^*))] \leq C_1 + \frac{C_3}{\ln(1/\lambda(\boldsymbol{P}))}.$$

The constant $\mathcal{J}_t$, introduced in [14], is an important parameter for the analysis and frequently used. Combining its definition in Theorem 2 and equation (5), it follows:

$$\left|[P^{\mathcal{J}_t}]_{i,j} - \pi_j\right| \leq C_P \lambda(\boldsymbol{P})^{\mathcal{J}_t} \leq \tfrac{1}{2Ht}, \quad \forall i,j \in [M]. \quad (24)$$

Assume $t \geq T_P$. We derive an important lower bound:

$$\begin{aligned}
&\mathbb{E}_{A_t|A_{t-\mathcal{J}_t}}[\textstyle\sum_{k\in A_t} q_k (F_k(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_k(\boldsymbol{w}_B^*))] \\
&\overset{(a)}{=} \textstyle\sum_{\mathcal{I}=1}^M \mathbb{P}(A_t{=}\mathcal{I}|A_{t-\mathcal{J}_t}) \sum_{k\in\mathcal{I}} q_k (F_k(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_k(\boldsymbol{w}_B^*)) \\
&\overset{(b)}{=} \textstyle\sum_{\mathcal{I}=1}^M \; [P^{\mathcal{J}_t}]_{A_{t-\mathcal{J}_t},\mathcal{I}} \; \sum_{k\in\mathcal{I}} q_k \left(F_k(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_k(\boldsymbol{w}_B^*)\right) \\
&\overset{(c)}{\geq} \textstyle\sum_{\mathcal{I}=1}^M \; \left(\pi(\mathcal{I}) - \frac{1}{2Ht}\right) \sum_{k\in\mathcal{I}} q_k (F_k(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_k(\boldsymbol{w}_B^*)) \\
&\overset{(d)}{\geq} (\textstyle\sum_{k=1}^N \pi_k q_k) \cdot (F_B(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_B^*) - \frac{1}{2t} MQ, \qquad (25)
\end{aligned}$$

where (a) is the definition of the conditional expectation, (b) uses the Markov property, (c) follows from (24), and (d) is due to (8). Taking total expectations:

$$\begin{aligned}
&(\textstyle\sum_{k=1}^N \pi_k q_k) \sum_t \eta_t \, \mathbb{E}[F_B(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_B^*] \\
&\quad \leq \textstyle\sum_t \eta_t \, \mathbb{E}[\sum_{k\in A_t} q_k (F_k(\boldsymbol{w}_{t-\mathcal{J}_t,0}) - F_k(\boldsymbol{w}_B^*))] \\
&\qquad + \tfrac{1}{4} MQ \textstyle\sum_t (\eta_t^2 + \frac{1}{t^2}) = C_2 + \frac{C_3}{\ln(1/\lambda(\boldsymbol{P}))}, \qquad (26)
\end{aligned}$$

where $C_2 = C_1 + \frac{1}{4} MQ \sum_t (\eta_t^2 + \frac{1}{t^2})$.

Step 2. By direct calculation (similar to Lemma 3):

$$(\textstyle\sum_{k=1}^N \pi_k q_k) \sum_t \eta_t \, \mathbb{E}[F_B(\boldsymbol{w}_{t,0}) - F_B(\boldsymbol{w}_{t-\mathcal{J}_t,0})] \leq \frac{C_3}{\ln(1/\lambda(\boldsymbol{P}))}.$$

Summing Step 1 and 2, and applying Jensen's inequality:

$$\begin{aligned}
&(\textstyle\sum_{t=1}^T \eta_t)(\sum_{k=1}^N \pi_k q_k) \, \mathbb{E}[F_B(\bar{\boldsymbol{w}}_{T,0}) - F_B^*] \leq \\
&(\textstyle\sum_{k=1}^N \pi_k q_k) \sum_{t=1}^T \eta_t \, \mathbb{E}[F_B(\boldsymbol{w}_{t,0}) - F_B^*] \leq C_2 + \frac{2C_3}{\ln(1/\lambda(\boldsymbol{P}))},
\end{aligned}$$

where $\bar{\boldsymbol{w}}_{T,0} := \frac{\sum_{t=1}^T \eta_t \boldsymbol{w}_{t,0}}{\sum_{t=1}^T \eta_t}$, and the constants are in (12). □

C. Proof of Theorem 3

It follows the same lines of Theorem 1, developing (20) as:

$$\begin{aligned}
\|\nabla F(\boldsymbol{w}_B^*)\| &\leq L\sqrt{\tfrac{2}{\mu}} \textstyle\sum_{k=1}^N |\alpha_k - p_k| \sqrt{(F_k(\boldsymbol{w}_B^*) - F_k^*)} \\
&\leq 2L\sqrt{\tfrac{2}{\mu}} d_{TV}(\boldsymbol{\alpha}, \boldsymbol{p}) \sqrt{\Gamma'},
\end{aligned}$$

where $d_{TV}(\boldsymbol{\alpha}, \boldsymbol{p}) \coloneqq \frac{1}{2} \sum_{k=1}^N |\alpha_k - p_k|$ is the total variation distance between the probability measures $\boldsymbol{\alpha}$ and $\boldsymbol{p}$. □

D. Minimizing ϵ_{opt}

Equation 12 defines the following optimization problem:

$$\begin{aligned}
\underset{\boldsymbol{q}}{\text{minimize}} \quad & f(\boldsymbol{q}) = \tfrac{\frac{1}{2}\boldsymbol{q}^\intercal \boldsymbol{A}\boldsymbol{q} + B}{\boldsymbol{\pi}^\intercal \boldsymbol{q}} + C; \\
\text{subject to} \quad & \boldsymbol{q} \geq 0, \\
& \boldsymbol{\pi}^\intercal \boldsymbol{q} > 0, \\
& \|\boldsymbol{q}\|_1 = Q.
\end{aligned}$$

Let us rewrite the problem by adding a variable $s := 1/\boldsymbol{\pi}^\intercal \boldsymbol{q}$ and then replacing $\boldsymbol{y} := s\boldsymbol{q}$. Note that the objective function is the perspective of a convex function, and is therefore convex:

$$\begin{aligned}
\min_{\boldsymbol{y},s} \quad & f(\boldsymbol{y}, s) = \tfrac{1}{2s} \boldsymbol{y}^\intercal \boldsymbol{A}\boldsymbol{y} + Bs + C \qquad (27a) \\
\text{s.t.} \quad & \boldsymbol{y} \geq 0, \; s > 0, \; \boldsymbol{\pi}^\intercal \boldsymbol{y} = 1, \; \|\boldsymbol{y}\|_1 = Qs. \qquad (27b)
\end{aligned}$$

The Lagrangian function $\mathcal{L}$ is as follows:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{y}, s, \lambda, \theta, \boldsymbol{\mu}) &= \tfrac{1}{2s} \boldsymbol{y}^\intercal \boldsymbol{A}\boldsymbol{y} + Bs + C + \\
&+ \lambda(1 - \boldsymbol{\pi}^\intercal \boldsymbol{y}) + \theta(\|\boldsymbol{y}\|_1 - Qs) - \boldsymbol{\mu}^\intercal \boldsymbol{y}. \qquad (28)
\end{aligned}$$

Since the constraint $s > 0$ defines an open set, the set defined by the constraints in (27b) is not closed. However, the solution is never on the boundary $s = 0$ because $\mathcal{L}^* \to +\infty$ as $s \to 0^+$, and we can consider $s \geq 0$. The KKT conditions for y_k^* read:

$$\text{if } y_k^* > 0\text{: } y_k^* = \tfrac{s^*}{A[kk]}(\lambda^* \pi_k - \theta^*); \; y_k^* = 0 \text{ otherwise.} \quad (29)$$

Since $\lambda^* \geq 0$, the clients with smaller π_k may have $q_k^* = 0$.

E. Convexity of $\epsilon_{opt} + \epsilon_{bias}$

In Appendix D, we proved that $\epsilon_{\text{opt}}(\boldsymbol{q})$ is convex. To prove that $\epsilon_{\text{bias}}(\boldsymbol{q})$ is also convex, we need to study the convexity of $\chi^2_{\boldsymbol{\alpha}\|\boldsymbol{p}} = \sum_{k=1}^N (f_k \circ g_k)(\boldsymbol{q})$, where $f_k(p_k) = (p_k - \alpha_k)^2/p_k$, and $g_k(\boldsymbol{q}) = (\pi_k q_k)/\sum_{h=1}^N \pi_h q_h$. We observe that $f_k(p_k)$ is convex, and $g_k(\boldsymbol{q})$ is a particular case of linear-fractional function [38]. By direct inspection, it can be proved that $(f_k \circ g_k)(\boldsymbol{q})$ is convex in $\text{dom}(f_k \circ g_k) = \{\boldsymbol{q} : \|\boldsymbol{q}\|_1 = Q > 0\}$.

F. Synthetic dataset

Our synthetic datasets has been generated as follows:

1) For client $k \in \mathcal{K}$, sample group identity i_k from a Bernoulli distribution of parameter $1/2$;
2) Sample model parameters $\boldsymbol{w}^* \sim \mathcal{N}(0, I_d)$ from the d-dimensional normal distribution;
3) For client $k \in \mathcal{K}$ and sample index $j \in \{1, \ldots, 150\}$, sample clients input data $\boldsymbol{x}_k^{(j)} \sim \mathcal{N}(0, I_d)$ from the d-dimensional normal distribution;
4) For client $k \in \mathcal{K}$ such that $i_k = 0$ and sample index $j \in \{1, \ldots, 150\}$, sample the true labels $y_k^{(j)}$ from a Bernoulli distribution with parameter equal to $\text{sigmoid}(\langle \boldsymbol{w}^*, \boldsymbol{x}_k^{(j)} \rangle)$;
5) For client $k \in \mathcal{K}$ such that $i_k = 1$ and sample index $j \in \{1, \ldots, 150\}$, sample the true labels $y_k^{(j)}$ from a Bernoulli distribution with parameter equal to $0.8 \cdot \text{sigmoid}(\langle \boldsymbol{w}^*, \boldsymbol{x}_k^{(j)} \rangle) + 0.2 \cdot (1 - \text{sigmoid}(\langle \boldsymbol{w}^*, \boldsymbol{x}_k^{(j)} \rangle))$.

REFERENCES

[1] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–33, 2020.

[2] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 63–71.

[3] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016, https://arxiv.org/abs/1610.05492.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[7] H. Eichner, T. Koren, B. McMahan, N. Srebro, and K. Talwar, "Semi-cyclic stochastic gradient descent," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1764–1773.

[8] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data *et al.*, "A field guide to federated optimization," *arXiv preprint arXiv:2107.06917*, 2021.

[9] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale: System design," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 374–388, 2019.

[10] Y. Ding, C. Niu, Y. Yan, Z. Zheng, F. Wu, G. Chen, S. Tang, and R. Jia, "Distributed optimization over block-cyclic data," *arXiv preprint arXiv:2002.07454*, 2020.

[11] C. Zhu, Z. Xu, M. Chen, J. Konečnỳ, A. Hard, and T. Goldstein, "Diurnal or Nocturnal? Federated Learning from Periodically Shifting Distributions," in *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[12] T. T. Doan, "Local stochastic approximation: A unified view of federated learning and distributed multi-task reinforcement learning algorithms," *arXiv preprint arXiv:2006.13460*, 2020.

[13] T. T. Doan, L. M. Nguyen, N. H. Pham, and J. Romberg, "Convergence rates of accelerated Markov gradient descent with applications in reinforcement learning," *arXiv preprint arXiv:2002.02873*, 2020.

[14] T. Sun, Y. Sun, and W. Yin, "On Markov chain gradient descent," *Advances in neural information processing systems*, vol. 31, 2018.

[15] M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," in *Psychology of Learning and Motivation*, G. H. Bower, Ed. Academic Press, 1989, vol. 24, pp. 109–165.

[16] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[17] M. Tang, X. Ning, Y. Wang, J. Sun, Y. Wang, H. Li, and Y. Chen, "FedCor: Correlation-Based Active Client Selection Strategy for Heterogeneous Federated Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[18] M. Ribero, H. Vikalo, and G. De Veciana, "Federated Learning Under Intermittent Client Availability and Time-Varying Communication Constraints," *arXiv preprint arXiv:2205.06730*, 2022.

[19] L. Tan, X. Zhang, Y. Zhou, X. Che, M. Hu, X. Chen, and D. Wu, "AdaFed: Optimizing Participation-Aware Federated Learning with Adaptive Aggregation Weights," *IEEE Transactions on Network Science and Engineering*, 2022.

[20] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[21] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive Federated Optimization," in *International Conference on Learning Representations*, 2021.

[22] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the Convergence of FedAvg on Non-IID Data," in *International Conference on Learning Representations*, 2019.

[23] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[24] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *arXiv preprint arXiv:2010.13723*, 2020.

[25] Y. Fraboni, R. Vidal, L. Kameni, and M. Lorenzi, "Clustered sampling: Low-variance and improved representativity for clients selection in federated learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3407–3416.

[26] Y. Jee Cho, J. Wang, and G. Joshi, "Towards Understanding Biased Client Selection in Federated Learning," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 10 351–10 375.

[27] T. T. Doan, L. M. Nguyen, N. H. Pham, and J. Romberg, "Finite-time analysis of stochastic gradient descent under Markov randomness," *arXiv preprint arXiv:2003.10973*, 2020.

[28] A. Meyers and H. Yang, "Markov Chains for Fault-Tolerance Modeling of Stochastic Networks," *IEEE Transactions on Automation Science and Engineering*, 2021.

[29] H. Olle, P. Yuval, and E. S. Jeffrey, "Dynamical percolation," in *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, vol. 33, no. 4. Elsevier, 1997, pp. 497–528.

[30] D. A. Levin and Y. Peres, *Markov chains and mixing times*. American Mathematical Soc., 2017, vol. 107.

[31] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Review*, vol. 60, no. 2, pp. 223–311, 2018.

[32] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[33] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," in *International Conference on Learning Representations*, 2013, arXiv preprint arXiv:1312.6211.

[34] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[35] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[36] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[37] H. Ludwig and N. Baracaldo, *Federated Learning: A Comprehensive Overview of Methods and Applications*. Springer Cham, 2022.

[38] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.