Federated Learning Under Heterogeneous and Correlated Client Availability

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

***Abstract*—In Federated Learning (FL), devices – also referred to as clients – can exhibit heterogeneous availability patterns, often correlated over time and with other clients. This paper addresses the problem of heterogeneous and correlated client availability in FL. Our theoretical analysis is the first to demonstrate the negative impact of correlation on FL algorithms' convergence rate and highlights a trade-off between optimization error (related to convergence speed) and bias error (indicative of model quality). To optimize this trade-off, we propose Correlation-Aware FL (`CA-Fed`), a novel algorithm that dynamically balances the competing objectives of fast convergence and minimal model bias. `CA-Fed` achieves this by dynamically adjusting the aggregation weight assigned to each client and selectively excluding clients with high temporal correlation and low availability. Experimental evaluations on diverse datasets demonstrate the effectiveness of `CA-Fed` compared to state-of-the-art methods. Specifically, `CA-Fed` achieves the best trade-off between training time and test accuracy. By dynamically handling clients with high temporal correlation and low availability, `CA-Fed` emerges as a promising solution to mitigate the detrimental impact of correlated client availability in FL.**

***Index Terms*—Federated Learning, Correlated Client Availability, Markov Chains.**

I. Introduction

The enormous amount of data generated by mobile and IoT devices motivated the development of distributed machine learning training paradigms [2], [3]. Federated Learning (FL) [4]–[7] 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`) [5], 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 [8], [9].

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 was presented at IEEE INFOCOM 2023 [1].

In real-world scenarios, the availability 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 [5], [10]. These eligibility requirements can make the availability of devices correlated over time and space [8], [11]–[13]. 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 [14], [15] 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 [16], [17]. 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 also to spatial correlation) [8], [11], [12], [18]. 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 presents the first convergence analysis of `FedAvg` [5] under heterogeneous and correlated client availability. We assume that the clients' availability follows an arbitrary finite-state Markov chain, modeling both temporal and spatial correlation while maintaining analytical tractability. Our theoretical analysis provides valuable insights by (i) quantitatively measuring the negative impact of correlation on the algorithm's convergence rate through a novel additional term that depends on the spectral properties of the Markov chain, and (ii) highlighting an important trade-off between two conflicting objectives: slow convergence to the optimal model and fast convergence to a biased model that minimizes a different objective function from the initial target. To leverage this trade-off, we propose `CA-Fed`,

an algorithm which achieves an optimal balance between maximizing convergence speed and minimizing model bias through dynamic adjustment of aggregation weights assigned to clients. Depending on their contribution to the learning process, `CA-Fed` can decide to exclude clients exhibiting 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. Across synthetic and real datasets, `CA-Fed` consistently outperforms the state-of-the-art methods `F3AST` [19] and `AdaFed` [20] in terms of test accuracy. These results underscore the importance of optimizing the training process to leverage available client resources effectively and mitigate the impact of less available and correlated clients, a task successfully accomplished by `CA-Fed`.

The remainder of this paper is organized as follows. Section II introduces 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 availability. `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 VI concludes the paper. Supplementary material, comprising Appendices A–H, provides detailed proofs and further discussions on `CA-Fed` not included in the main text due to space constraints.

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. 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 [5], [9] 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 ($\mathcal{A}_t$). Client $k \in \mathcal{A}_t$ 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's 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 $\Delta_t \coloneqq \sum_{k\in\mathcal{A}_t} q_k \cdot \Delta_t^k$, 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:

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

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

The aggregate update Δ_t is generally a biased estimator of the pseudo-gradient $-\nabla F(\boldsymbol{w}_{t,0})$, to which each client k contributes proportionally to its frequency of appearance in the set $\mathcal{A}_t$ and its aggregation weight q_k. More specifically, under proper assumptions specified in Section III, we will prove in Theorem 2 that the update rule described by (2) and (3) converges to the unique minimizer of a biased global objective F_B. This objective function depends depends both on the clients' availability (i.e., on the sequence $(\mathcal{A}_t)_{t>0}$) and on the aggregation strategy (i.e., on $\boldsymbol{q} = (q_k)_{k\in\mathcal{K}}$):

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

where π_k represents the asymptotic availability of client k, defined as $\pi_k \coloneqq \lim_{t\to+\infty} \mathbb{P}(k \in \mathcal{A}_t)$. We denote $\boldsymbol{\pi} = (\pi_k)_{k\in\mathcal{K}}$. Moreover, the coefficients $\boldsymbol{p} = (p_k)_{k\in\mathcal{K}}$ in (4) 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 [6, 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.

[1] The aggregation rule (3) has been considered also in other works, e.g., [9], [21], [22]. In other FL algorithms, the server computes an average of clients' local models. This aggregation rule can be obtained with minor changes to (3).

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 [5], [18]–[20], [23]–[26] considered this problem and, under different assumptions on the clients' availability (i.e., on $(\mathcal{A}_t)_{t>0}$), designed aggregation strategies that unbias Δ_t through an appropriate choice of $\boldsymbol{q}$. Reference [23] 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 [5]), 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 [24]). However, references [5], [23], [24] ignore the variance induced by the clients stochastic availability. The authors of [25] reduce such variance by considering only the clients with important updates, as measured by the value of their norm. References [18] and [26] reduce the aggregation variance through clustered and soft-clustered sampling, respectively.

Some recent works [19], [20], [27] 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 [19] and `AdaFed` from [20].

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 [19], 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 [13]–[15], [28]: these works study a variant of stochastic gradient descent where samples are drawn according to a Markov chain. Reference [13] 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 [15].

III. Analysis

A. Main assumptions

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

Assumption 1. *The Markov chain $(\mathcal{A}_t)_{t\geq 0}$ on the M-finite state space $\mathcal{M}$ is time-homogeneous, irreducible, and aperiodic. It has transition matrix $\boldsymbol{P}$, stationary distribution $\boldsymbol{\rho}$, and has state distribution $\boldsymbol{\rho}$ at time $t = 0$.*

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 [29], [30]. The previous Markovian assumption, while allowing a great degree of flexibility, still guarantees the analytical tractability of the system. The distance dynamics between the current and the stationary distributions of the Markov process can be characterized in terms of the spectral properties of its transition matrix $\boldsymbol{P}$ [31]. Let $\bar{\lambda}_2(\boldsymbol{P})$ denote the the second largest module of the eigenvalues of $\boldsymbol{P}$. Previous work [15] has shown that:

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

where the parameters $\lambda(\boldsymbol{P}) \coloneqq (\bar{\lambda}_2(\boldsymbol{P}) + 1)/2$, C_P, and T_P are positive constants whose values are defined in [15, Lemma 1] and reported for completeness in Appendix B2, Lemma 16.[3] Note that $\lambda(\boldsymbol{P})$ quantifies the correlation of the Markov process $(\mathcal{A}_t)_{t\geq 0}$: 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.

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

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 [32, Sections 4.1, 4.2]. Assumption 5 is a standard hypothesis in the analysis of federated optimization algorithms [9, 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: $\mathbb{E}\|\nabla F_k(\boldsymbol{w}, \mathcal{B}) - \nabla F_k(\boldsymbol{w})\|^2 \leq \sigma_k^2$, $k = 1, \ldots, N$.*

[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.

[3] Note that (5) holds for different definitions of $\lambda(\boldsymbol{P})$ as long as $\lambda(\boldsymbol{P}) \in (\bar{\lambda}_2(\boldsymbol{P}), 1)$. The specific choice for $\lambda(\boldsymbol{P})$ changes the values of C_P and T_P.

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 all $\boldsymbol{w} \in W$ and $k \in \mathcal{K}$, we have:*

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

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

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

Similarly to other works [9], [23], [24], [33], we introduce a metric to quantify the heterogeneity of clients' local datasets, typically referred to as *statistical heterogeneity*:

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

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

$$\epsilon \leq 2\kappa^2 (\underbrace{F_B(\boldsymbol{w}) - F_B^*}_{\coloneqq \epsilon_{opt}} + \underbrace{F(\boldsymbol{w}_B^*) - F^*}_{\coloneqq \epsilon_{bias}}). \tag{10}$$

Moreover, let $\chi^2_{\boldsymbol{\alpha}\|\boldsymbol{p}} \coloneqq \sum_{k=1}^N (\alpha_k - p_k)^2/p_k$. Then:

$$\epsilon_{bias} \leq \kappa^2 \cdot \underbrace{\chi^2_{\boldsymbol{\alpha}\|\boldsymbol{p}} \cdot \Gamma}_{\coloneqq \bar{\epsilon}_{bias}}. \tag{11}$$

Theorem 1 (proof in Appendix A) decomposes the error of the target objective (ϵ) as the sum of an optimization error for the biased objective (ϵ_{opt}) and a bias error (ϵ_{bias}). The term ϵ_{opt}, evaluated on the trajectory determined by scheme (3), quantifies the optimization error associated with the biased objective F_B and asymptotically vanishes (see Theorem 2 below). The non-vanishing bias error ϵ_{bias} captures the discrepancy between $F(\boldsymbol{w}_B^*)$ and F^*. This term is bounded by 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 by 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 [6], [8]), and the previous asymptotic considerations may not apply. In this regime, the unbiased aggregation strategy can be sub-optimal, since the minimization of ϵ_{bias} not necessarily leads to the minimization of the total error $\epsilon \leq 2\kappa^2(\epsilon_{\text{opt}} + \epsilon_{\text{bias}})$. 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, σ_k, C_P, T_P, and $\lambda(\boldsymbol{P})$ defined above. Let $Q \coloneqq \sum_{k\in\mathcal{K}} q_k$. We require a diminishing step-size $\eta_t > 0$ satisfying:*

$$\eta_1 \leq \tfrac{1}{2L(1+2EQ)}, \ \sum_{t=1}^{+\infty} \eta_t = +\infty, \ \sum_{t=1}^{+\infty} \ln(t) \cdot \eta_t^2 < +\infty. \tag{12}$$

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

$$\mathbb{E}[F_B(\bar{\boldsymbol{w}}_{T,0}) - F_B^*] \leq \underbrace{\frac{\frac{1}{2}\boldsymbol{q}^\intercal \boldsymbol{\Sigma} \boldsymbol{q} + \upsilon}{\boldsymbol{\pi}^\intercal \boldsymbol{q}} + \psi + \frac{\phi}{\ln(1/\lambda(\boldsymbol{P}))}}{(\sum_{t=1}^T \eta_t)}}_{\coloneqq \bar{\epsilon}_{opt}}, \tag{13}$$

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*

$$\boldsymbol{\Sigma} \coloneqq \operatorname{diag}(2(E+1)\sigma_k^2 \pi_k \textstyle\sum_{t=1}^{+\infty} \eta_t^2),$$

$$\upsilon \coloneqq \tfrac{2}{E} \operatorname{diam}(W)^2 + \tfrac{1}{4} MQ \textstyle\sum_{t=1}^{+\infty} (\eta_t^2 + \frac{1}{t^2}),$$

$$\psi \coloneqq (4L(1+EQ)\Gamma + 2E^2G^2) \textstyle\sum_{t=1}^{+\infty} \eta_t^2 + H(\sum_{t=1}^{T_P - 1} \eta_t),$$

$$\mathcal{J}_t \coloneqq \min\left\{\max\left\{\left\lceil \ln\left(2C_P H t\right)/\ln\left(1/\lambda(\boldsymbol{P})\right)\right\rceil, T_P\right\}, t\right\},$$

$$\phi \coloneqq 2EDGQ \textstyle\sum_{t=1}^{+\infty} \ln(2C_P H t) \eta_{t-\mathcal{J}_t}^2.$$

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 (13) 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 (13) has a quadratic-over-linear fractional dependence on $\boldsymbol{q}$. Minimizing $\bar{\epsilon}_{\text{opt}}$ leads, in general, to a different choice of $\boldsymbol{q}$ than minimizing $\bar{\epsilon}_{\text{bias}}$.

C. Minimizing the total error $\epsilon \leq 2\kappa^2(\bar{\epsilon}_{opt} + \bar{\epsilon}_{bias})$

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

$$\begin{aligned} \underset{\boldsymbol{q}}{\text{minimize}} \quad & \bar{\epsilon}_{\text{opt}}(\boldsymbol{q}) + \bar{\epsilon}_{\text{bias}}(\boldsymbol{q}); \\ \text{subject to} \quad & \boldsymbol{q} \geq 0, \\ & \|\boldsymbol{q}\|_1 = Q. \end{aligned} \tag{14}$$

In Appendix D we prove that (14) is a convex optimization problem, which can be solved with the method of Lagrange multipliers. However, its solution lacks practical utility because the constants in (10) and (13) (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), as well as the bound in [33], diverges when setting some q_k values equal to 0, but this divergence is merely an artifact of the proof technique. For more practical considerations, we present the following result (proof in Appendix C):

Theorem 3 (An alternative bound on the bias error ϵ_{bias})**.** *Under the same assumptions of Theorem 1, define* $\Gamma' := \max_k\{F_k(\boldsymbol{w}_B^*) - F_k^*\}$. *The following result holds:*

$$\epsilon_{bias} \leq 4\kappa^2 \cdot \underbrace{d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{p}) \cdot \Gamma'}_{:=\bar{\epsilon}'_{bias}}, \tag{15}$$

where $d_{TV}(\boldsymbol{\alpha}, \boldsymbol{p}) := \frac{1}{2}\sum_{k=1}^{N}|\alpha_k - p_k|$ *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 (15) more challenging (for example, the corresponding problem is not convex). We study the minimization of the two terms $\bar{\epsilon}_{\text{opt}}$ and $\bar{\epsilon}'_{\text{bias}}$ separately and learn some insights, which we use to design the new FL algorithm `CA-Fed`.

D. Minimizing $\bar{\epsilon}_{opt}$

The minimization of $\bar{\epsilon}_{\text{opt}}$ is still a convex optimization problem (Appendix E). In particular, at the optimum, non-negative weights are set accordingly to $q_k^* = a(\iota^*\pi_k - \theta^*)$ with a and ι^* positive constants (Appendix E2). 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 accelerate convergence, we can exclude clients with low availability π_k *by setting* $q_k^* = 0$.

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. Their updates act as a noise slowing down the algorithm's convergence. It may then be advantageous to exclude these clients.

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, unless otherwise specified, we consider from now on the particular case when the availability of each client k evolves according to a Markov chain $(\mathcal{A}_t^k)_{t\geq 0}$ with transition probability matrix $\boldsymbol{P}_k$ and these Markov chains are all independent [31, Exercise 12.6]. In this case, the aggregate process is described by the product Markov chain $(\mathcal{A}_t)_{t\geq 0}$ 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 $\boldsymbol{P}_j$ (Appendix F2). In this setting, it is possible to redefine the Markov chain $(\mathcal{A}_t)_{t\geq 0}$ 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 w.r.t. the case when all clients participate to the aggregation. These considerations lead to the following guideline:

Guideline B: to accelerate convergence, we can exclude clients with high correlation (high $\lambda(\boldsymbol{P}_k)$*) by setting their* $q_k^* = 0$.

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 [34], [35], 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 the spatially correlated clients can be grouped in clusters, each cluster evolving as an independent Markov chain (see Section V-B, Paragraph e).

E. Minimizing $\bar{\epsilon}'_{bias}$

The bias error $\bar{\epsilon}'_{\text{bias}}$ in (15) 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 $\bar{\epsilon}'_{\text{bias}}$ is minimized.

Guideline C: to minimize the bias error, we assign $q_k^* \propto \alpha_k/\pi_k$ *to the clients not excluded by the previous guidelines.*

IV. Proposed Algorithm

Guidelines A and B in Section III suggest that minimizing $\bar{\epsilon}_{\text{opt}}$ 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 setting their aggregation weight inversely proportional to their availability to reduce the bias error $\bar{\epsilon}'_{\text{bias}}$. Motivated by these insights, we propose `CA-Fed`, a client aggregation strategy that considers the problem of correlated client availability in FL, described in Algorithm 1. `CA-Fed` learns during training which clients to exclude and how to set the aggregation weights of the remaining clients to achieve a good trade-off between $\bar{\epsilon}_{\text{opt}}$ and $\bar{\epsilon}'_{\text{bias}}$. 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 bounds in (10) and (15):

$$\epsilon^{(t)} := \underbrace{F_B(\boldsymbol{w}_{t,0}) - F_B^*}_{\epsilon_{\text{opt}}} + 4\bar{\kappa}^2 \cdot \underbrace{d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{p})\Gamma'}_{\bar{\epsilon}'_{\text{bias}}}, \tag{16}$$

where $\bar{\kappa}^2 \geq 0$ is a hyper-parameter that weights the relative importance of the optimization and bias error (see Sec. IV-C).

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 $\mathcal{A}_t$ 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, $\lambda(\boldsymbol{P}_k)$, 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).

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

Input : $\boldsymbol{w}_{0,0}$, $\boldsymbol{\alpha}$, $\boldsymbol{q}^{(0)}$, $\{\eta_t\}_{t=1}^T$, $\bar{\eta}$, E, $\bar{\kappa}^2$, β, τ

1 Initialize $\hat{\boldsymbol{F}}^{(0)}$, $\hat{\boldsymbol{F}}^*$, $\hat{\Gamma}'^{(0)}$, $\hat{\boldsymbol{\pi}}^{(0)}$, and $\hat{\boldsymbol{\lambda}}^{(0)}$;
2 **for** $t = 1,\dots,T$ **do**
3 Receive set of active client $\mathcal{A}_t$, loss vector $\boldsymbol{F}^{(t)}$;
4 Update $\hat{\boldsymbol{F}}^{(t)}$, $\hat{\Gamma}'^{(t)}$, $\hat{\boldsymbol{\pi}}^{(t)}$, and $\hat{\boldsymbol{\lambda}}^{(t)}$;
5 Initialize $\boldsymbol{q}^{(t)} = \frac{\boldsymbol{\alpha}}{\hat{\boldsymbol{\pi}}^{(t)}}$;
6 $\boldsymbol{q}^{(t)} \leftarrow$ `get` $(\boldsymbol{q}^{(t)}, \boldsymbol{\alpha}, \hat{\boldsymbol{F}}^{(t)}, \hat{\boldsymbol{F}}^*, \hat{\Gamma}'^{(t)}, \hat{\boldsymbol{\pi}}^{(t)}, \hat{\boldsymbol{\lambda}}^{(t)})$;
7 $\boldsymbol{q}^{(t)} \leftarrow$ `get` $(\boldsymbol{q}^{(t)}, \boldsymbol{\alpha}, \hat{\boldsymbol{F}}^{(t)}, \hat{\boldsymbol{F}}^*, \hat{\Gamma}'^{(t)}, \hat{\boldsymbol{\pi}}^{(t)}, -\hat{\boldsymbol{\pi}}^{(t)})$;
8 **for** *client* $\{k \in \mathcal{A}_t;\ q_k^{(t)} > 0\}$, *in parallel* **do**
9 **for** $j = 0,\dots,E-1$ **do**
10 $\boldsymbol{w}_{t,j+1}^k = \boldsymbol{w}_{t,j}^k - \eta_t \nabla F_k(\boldsymbol{w}_{t,j}^k, \mathcal{B}_{t,j}^k)$;
11 $\Delta_t^k \leftarrow \boldsymbol{w}_{t,E} - \boldsymbol{w}_{t,0}$;
12 $\boldsymbol{w}_{t+1,0} \leftarrow \mathbf{Proj}_W(\boldsymbol{w}_{t,0} + \bar{\eta}\sum_{k\in\mathcal{A}_t} q_k^{(t)} \cdot \Delta_t^k)$;
13 **Function** `get` $(\boldsymbol{q}, \boldsymbol{\alpha}, \boldsymbol{F}, \boldsymbol{F}^*, \Gamma, \boldsymbol{\pi}, \boldsymbol{\rho})$**:**
14 Sort $\mathcal{K}$ by descending order in $\boldsymbol{\rho}$;
15 $\hat{\epsilon} \leftarrow \langle \boldsymbol{F} - \boldsymbol{F}^*, \boldsymbol{\pi}\tilde{\odot}\boldsymbol{q}\rangle + 4\bar{\kappa}^2 \cdot d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{\pi}\tilde{\odot}\boldsymbol{q})\Gamma$;
16 **for** $k \in \mathcal{K}$ **do**
17 $q_k^+ \leftarrow 0$;
18 $\hat{\epsilon}^+ \leftarrow \langle \boldsymbol{F} - \boldsymbol{F}^*, \boldsymbol{\pi}\tilde{\odot}\boldsymbol{q}^+\rangle + 4\bar{\kappa}^2 \cdot d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{\pi}\tilde{\odot}\boldsymbol{q}^+)\Gamma$;
19 **if** $\hat{\epsilon} - \hat{\epsilon}^+ \geq \tau$ **then**
20 $\hat{\epsilon} \leftarrow \hat{\epsilon}^+$;
21 $\boldsymbol{q} \leftarrow \boldsymbol{q}^+$;
22 **return** $\boldsymbol{q}$

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 ($\boldsymbol{F}^{(t)} = (F_k^{(t)})_{k\in\mathcal{A}_t}$) through an auto-regressive filter with parameter $\beta \in (0,1]$:

$$\hat{\boldsymbol{F}}^{(t)} = (\mathbf{1} - \beta\mathbb{1}_{\mathcal{A}_t}) \odot \hat{\boldsymbol{F}}^{(t-1)} + \beta\mathbb{1}_{\mathcal{A}_t} \odot \boldsymbol{F}^{(t)}, \tag{17}$$

where $\odot$ denotes the component-wise multiplication between vectors, and $\mathbb{1}_{\mathcal{A}_t}$ is a N-dimensions binary vector whose k-th component equals 1 if and only if client k is active at round t, i.e., $k \in \mathcal{A}_t$. The server can estimate client-k's loss minimum value 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:

$$\hat{F}_B^{(t)} - \hat{F}_B^* = \langle \hat{\boldsymbol{F}}^{(t)} - \hat{\boldsymbol{F}}^*, \hat{\boldsymbol{\pi}}^{(t)}\tilde{\odot}\boldsymbol{q}^{(t)}\rangle, \tag{18}$$

$$\hat{\Gamma}'^{(t)} = \max_{k\in\mathcal{K}}(\hat{F}_k^{(t)} - \hat{F}_k^*), \tag{19}$$

$$\hat{\epsilon}^{(t)} = \hat{F}_B^{(t)} - \hat{F}_B^* + 4\bar{\kappa}^2 \cdot d_{TV}^2(\boldsymbol{\alpha}, \hat{\boldsymbol{\pi}}^{(t)}\tilde{\odot}\boldsymbol{q}^{(t)})\hat{\Gamma}'^{(t)}. \tag{20}$$

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 (active and inactive), 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 $\hat{\boldsymbol{P}}_k^{(t)}$.

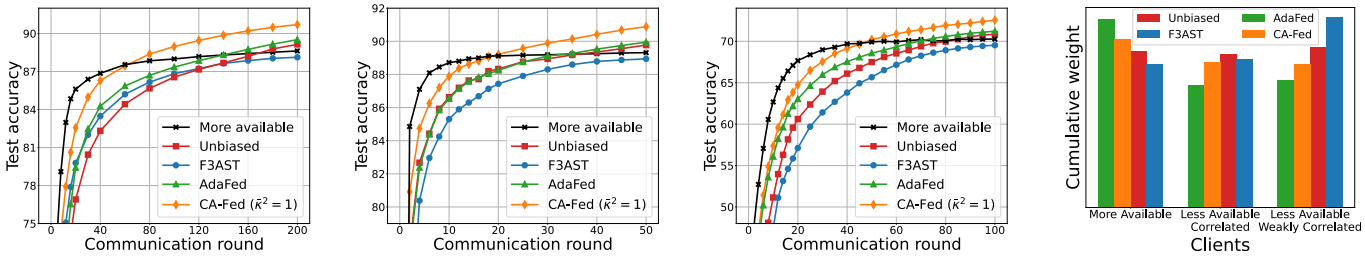
C. The role of the hyper-parameter $\bar{\kappa}^2$

Theorems 1 and 3 suggest that the condition number κ^2 has a significant impact on the minimization of the total error ϵ. Our algorithm uses a proxy ($\epsilon^{(t)}$) for the total error (see (16)). To account for the effect of κ^2, we introduced the hyper-parameter $\bar{\kappa}^2 \geq 0$, which weights the relative importance of the optimization and bias error in (16). In practice, $\bar{\kappa}^2$ controls the number of excluded clients by CA-Fed. A small value of $\bar{\kappa}^2$ penalizes the bias term in favor of the optimization error, resulting in a larger number of excluded clients. Conversely, the bias term dominates for large values of $\bar{\kappa}^2$, and CA-Fed tends to include more clients. Asymptotically, for $\bar{\kappa}^2 \to \infty$, CA-Fed reduces to the *unbiased aggregation strategy*.

V. Experimental Evaluation

A. Experimental Setup

a) Federated system simulator: In our experiments, we consider a population of $N = |\mathcal{K}| = 100$ clients. We model the activity of each client $k \in \mathcal{K}$ as a two-state homogeneous Markov process with state space $\mathcal{S} = \{$"*active*", "*inactive*"$\}$, characterized by a transition matrix $\boldsymbol{P}_k$, a stationary distribution $\boldsymbol{\pi}^{(k)}$, and a second largest absolute eigenvalue $\bar{\lambda}_2(\boldsymbol{P}_k)$ (see Appendix F3 for details). Our goal is to simulate realistic dynamics of federated systems featuring varying levels of



(a) Synthetic (b) MNIST (c) CIFAR-10 (d) Cumulative importance

Fig. 1: Average test accuracy among $N = 100$ clients achieved by the algorithms on the Synthetic, MNIST, and CIFAR-10 datasets. Cumulative importance assigned by the algorithms to the clients after $T = 200$ rounds on the Synthetic dataset.

clients' availability and correlation. To introduce heterogeneity in clients' availability patterns, we divide the population in two equally-sized classes: the "more available" clients with a steady-state probability of being active $\pi_{k,\text{active}} = 1/2+g$, and the "less available" clients with $\pi_{k,\text{active}} = 1/2 - g$. Here, the parameter $g \in (0, 1/2)$ controls the degree of heterogeneity in clients' availability. We furthermore divide each class of clients in two equally-sized sub-classes: clients exhibiting a largely correlated time behavior (in the following referred to as "correlated" clients) that tend to persist in the same state for rather long periods ($\lambda_k = \nu$ with values of ν close to 1), and clients exhibiting a weakly correlated time behavior (referred to as "weakly correlated" clients) that are almost as likely to keep as to change their state at every t ($\lambda_k \sim \mathcal{N}(0, \varepsilon^2)$, with ε close to 0). We use $g = 0.4$, $\nu = 0.9$, and $\varepsilon = 10^{-2}$.

b) Datasets and models: We conduct experiments on the LEAF Synthetic dataset [36], a benchmark for multinomial classification tasks, and on the real-world MNIST [37] and CIFAR-10 [38] datasets, respectively for handwritten digits and image recognition tasks. To simulate the statistical heterogeneity present in the federated learning system, we use common approaches in the literature. For the Synthetic dataset, we tune the parameters (γ, δ), which control data heterogeneity among clients [23]. For MNIST and CIFAR-10, we distribute samples from the same class across the clients according to a symmetric Dirichlet distribution with parameter ς, following the same approach as [39]. Unless otherwise indicated, we set $\gamma = \delta = \varsigma = 0.5$. We use the original training/test data split of MNIST and reserve 20% of the training dataset as the validation dataset. For Synthetic and MNIST, we use a linear classifier with a ridge penalization of parameter 10^{-2}, which corresponds to a strongly convex objective function. For CIFAR-10, we use a neural network with two convolutional and one fully connected layers.

c) Benchmarks: We compare `CA-Fed`, defined in Algorithm 1, with four baselines including two state-of-the-art FL algorithms discussed in Section II: 1) `Unbiased`, which aggregates the active clients $k \in \mathcal{A}_t$ with weights $q_k = \alpha_k/\pi_k$; 2) `More available`, which considers only the "more available" clients and always excludes the "less available" ones; 3) `AdaFed` [20], which, similarly to `Unbiased`, aggregates all active clients, but normalizes their aggregation weights (i.e., it considers $q_k = \frac{\alpha_k/\pi_k}{\sum_{k\in\mathcal{A}_t} \alpha_k/\pi_k}$); 4) `F3AST` [19], which, oppositely to `More available`, favors the "less available" clients. For all algorithms, we tuned the learning rates η, $\bar{\eta}$ via grid search. For `CA-Fed`, we use $\beta = \tau = 0$. Unless otherwise specified, we assume that the algorithms can access an oracle providing the true availability parameters for each client: in practice, all the algorithms rely on the exact knowledge of $\pi_{k,\text{active}}$; in addition, `CA-Fed` also receives $\lambda(\boldsymbol{P}_k)$. In Section V-B, Paragraph d, we will relax this assumption by considering the estimators $\hat{\pi}_k^{(t)}$ and $\hat{\lambda}_k^{(t)}$. The code for this paper is available at: https://github.com/arodio/CA-Fed.

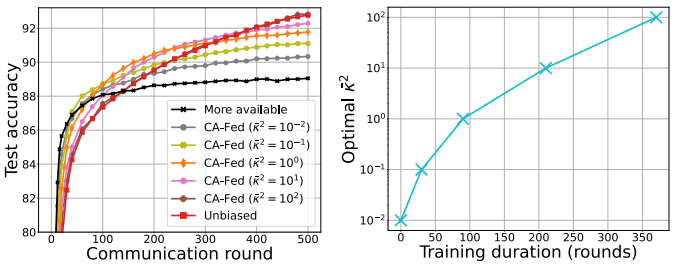
B. Experimental Results

a) `CA-Fed` vs. baselines: Figure 1 compares the test accuracy achieved by `CA-Fed` ($\bar{\kappa}^2 = 1$) and the baselines on the Synthetic (Fig. 1a), MNIST (Fig. 1b), and CIFAR-10 (Fig. 1c) datasets over 10 different runs. Across all three datasets, `CA-Fed` consistently outperforms the baselines, achieving higher test accuracy (+1.56 pp on Synthetic; +0.94 pp on MNIST; +1.32 pp on CIFAR-10) compared to the second best performing method, `AdaFed`. These results demonstrate that `CA-Fed` achieves the best balance between convergence speed and test accuracy. For deeper insights into the algorithms' behavior, Figure 1d illustrates the cumulative aggregation weights $\{\frac{1}{T}\sum_{t=1}^{T} q_k^{(t)}\}_{k\in\mathcal{K}}$, representing the cumulative importance that the algorithms assigned to the clients at the end of the training. In Figure 1d, we grouped the clients into three categories: "more available", "less available, weakly correlated", and "less available, correlated". By setting the aggregation weights inversely proportional to the clients' availabilities, `Unbiased` equalizes the importance for all clients (see Fig. 1d), but achieves a slower convergence (as shown in Figs. 1a, 1b, and 1c). On the contrary, by excluding all the "less available" clients, `More available` achieves a faster convergence but introduces a non-vanishing bias error ϵ_{bias}, which, in practice, leads to poor accuracy performance. The state-of-the-art algorithm `AdaFed`, similarly to `Unbiased`, considers all the active clients, but normalizes their aggregation weights at each communication round. As a result, similarly to `CA-Fed`, `AdaFed` indeed prioritizes the "more available" clients (as shown in Fig. 1d), and then a convergence speed-up could be expected. However, `AdaFed`

does not exclude the "less available and correlated" clients, and therefore their presence causes a convergence slowdown. Finally, F3AST favors the "less available, correlated" clients and achieves a slower convergence with a non-vanishing bias error, which corresponds to lower accuracy performance. By opportunely excluding some of the "less available and correlated" clients, CA-Fed achieves the best test accuracy by the end of the training time.

b) Convergence speed vs. Bias error: The trade-off between ϵ_{opt} or ϵ_{bias} discussed in Section III is visible in our experiments. In particular, Figure 2a compares the test accuracy achieved by More available, Unbiased, and CA-Fed on the Synthetic dataset for $T = 500$ communication rounds. As expected, by targeting the minimization of ϵ_{opt} and thus excluding the "less available" clients, More available achieves the fastest convergence at the expense of a large non-vanishing bias error ϵ_{bias}. On the other hand, by targeting the minimization of ϵ_{bias} and thus equalizing the clients' importance, Unbiased asymptotically removes this error and ultimately achieves the highest test accuracy at communication round $T = 500$, but suffers from slower convergence due to the presence of the "correlated" clients. Our algorithm, CA-Fed, leverages the trade-off between convergence speed and model bias and achieves fast convergence to the neighborhood of the target objective. To explore this trade-off, in Figure 2a, we varied the value of the hyper-parameter $\bar{\kappa}^2$ in the range $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. CA-Fed tends to exclude more clients for low values of $\bar{\kappa}^2$ and achieves a similar convergence rate as More available for $\bar{\kappa}^2 = 10^{-2}$. For intermediate values of $\bar{\kappa}^2$, CA-Fed trades a small accuracy decrease for faster convergence (refer, for example, to the curves $\bar{\kappa}^2 = 10^{0}, 10^{1}$). For $\bar{\kappa}^2 = 10^{2}$, CA-Fed reduces to Unbiased (their curves overlap in Fig. 2a). Moreover, we observe that the optimal value of $\bar{\kappa}^2$ depends on the available time for training. Low values of $\bar{\kappa}^2$ speed-up convergence and then they can be beneficial for short training durations (e.g., CA-Fed ($\bar{\kappa} = 10^{-1}$) achieves a higher test accuracy of +2.8 pp with respect to Unbiased at communication round $t = 40$). For longer training periods, a larger value of $\bar{\kappa}^2$ may be preferable as it reduces the bias error and increases the test accuracy (e.g., CA-Fed ($\bar{\kappa} = 10^{2}$) improves of +3.8 pp with respect to More available at communication round $t = 500$). Figure 2b illustrates the optimal value of $\bar{\kappa}^2$ for different durations of the training period T.

c) Effect of statistical heterogeneity: The bias error bounds $\bar{\epsilon}_{\text{bias}}$ and $\bar{\epsilon}'_{\text{bias}}$ in Theorems 1 and 3 are influenced by the degree of heterogeneity among local functions, commonly known as *statistical heterogeneity*, characterized by the constants Γ and Γ' in (11) and (15), respectively. To control statistical heterogeneity, we manipulate the dissimilarity among the clients' local datasets, specifically through the parameters γ and δ in the case of the Synthetic dataset, as explained in Section V-A. Figure 3 illustrates the impact of γ and δ on the test accuracy achieved by CA-Fed after $T = 200$ communication rounds on the Synthetic dataset. As expected,



(a) Test accuracy (b) Optimal $\bar{\kappa}^2$

Fig. 2: *Convergence speed vs. Model bias* trade-off for different values of $\bar{\kappa}^2$ on the Synthetic dataset, for $\gamma = \delta = 0.5$.

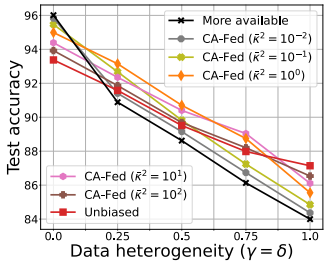
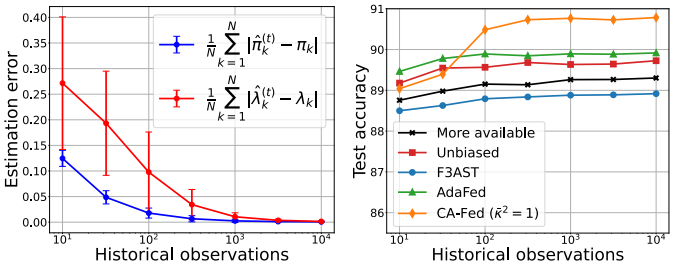


Fig. 3: Effects of *data heterogeneity* on the Synthetic dataset after $T = 200$ rounds.

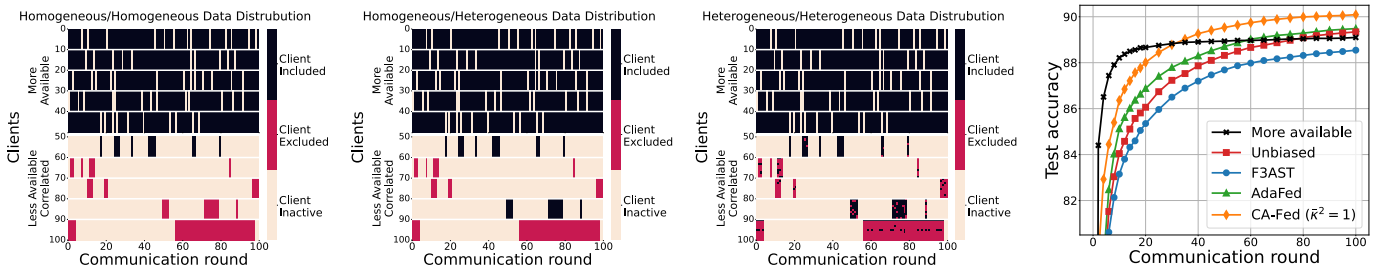


(a) Estimation error (b) Test accuracy

Fig. 4: Estimation of the *clients' activities* $(\hat{\pi}_k^{(t)}, \hat{\lambda}_k^{(t)})$ for different priors $t \in \{10^{1}, 10^{1.5}, 10^{2}, 10^{2.5}, 10^{3}, 10^{3.5}, 10^{4}\}$ and test accuracy after $T = 50$ rounds on the MNIST dataset.

in the extreme IID setting (when $\gamma = \delta = 0$), Γ and Γ' are small, and the bias error ϵ_{bias} is negligible. As a result, More available and CA-Fed ($\bar{\kappa}^2 = 10^{-2}$) reach the highest test accuracy, whereas CA-Fed ($\bar{\kappa}^2 = 10^{2}$) and Unbiased present slow convergence. Nevertheless, More available and CA-Fed ($\bar{\kappa}^2 = 10^{-2}$) perform poorly as the statistical heterogeneity increases (i.e., $\gamma = \delta \geq 0.25$). In the extreme non-IID setting (when $\gamma = \delta = 1$), Γ and Γ' are large, and ϵ_{bias} dominates. In this case, CA-Fed ($\bar{\kappa}^2 = 10^{2}$) and Unbiased should be preferred. For $\gamma = \delta = \{0.25, 0.5, 0.75\}$, CA-Fed (with $\bar{\kappa}^2 = 1$ or $\bar{\kappa}^2 = 10$) achieves the highest test accuracy (+1.6 pp, +1.2 pp, and +1.0 pp with respect to Unbiased).

d) Estimation of the clients' availability and correlation: In this experiment, CA-Fed utilizes estimators $\hat{\pi}_k^{(t)}$ and $\hat{\lambda}_k^{(t)}$ to estimate the clients' π_k and λ_k values. We employ a Bayesian estimator with a beta prior to estimate $\hat{\boldsymbol{P}}_k^{(t)}$, which we generate by observing the evolution of the Markov



(a) Homogeneous/Homogeneous (b) Homogeneous/Heterogeneous (c) Heterogeneous/Heterogeneous (d) Test accuracy

Fig. 5: Clients' activities and CA-Fed's inclusion/exclusion decisions in the presence of *spatial correlation* for different degrees of *intra-cluster/inter-cluster* data distributions. Average test accuracy after $T = 100$ rounds on the MNIST dataset.

chain defined by $\boldsymbol{P}_k$ over t' time-steps. We compute $\hat{\pi}_k^{(t)}$ and $\hat{\lambda}_k^{(t)}$ analytically, following the methodology explained in Section IV-B and described in detail in Appendix F3. Figure 4a shows the estimation errors $\frac{1}{N}\sum_{k\in\mathcal{K}}|\hat{\pi}_k^{(t)} - \pi_k|$ and $\frac{1}{N}\sum_{k\in\mathcal{K}}|\hat{\lambda}_k^{(t)} - \lambda_k|$ as a function of the number of historical observations t'. As expected, both errors decrease with an increasing number of observations, and the estimation error for λ_k is larger than that for π_k. Furthermore, Figure 4b compares the final test accuracy obtained by CA-Fed and the baselines for varying numbers of historical observations $t' \in \{10^1, 10^{1.5}, 10^2, 10^{2.5}, 10^3, 10^{3.5}, 10^4\}$ when training for $T = 50$ rounds on the MNIST dataset. In this setting, CA-Fed outperforms the baselines for $t' \geq 100$. This value is reasonable, because estimating λ_k requires a number of observations comparable to the expected hitting time for the slowest Markov chain, which is given by $\max_{k\in\mathcal{K}} \frac{1}{(1-\lambda_k)\pi_k} = 100$.

e) CA-Fed with Spatial Correlation: Although CA-Fed is primarily designed to handle temporal correlation (as discussed in Section III-D), we also evaluate its performance in the presence of spatial correlation. In the considered spatially correlated scenario, clients are grouped into clusters, and each cluster $c \in \mathcal{C}$ is characterized by an underlying Markov chain that determines when all clients in the cluster are available or unavailable. The Markov chains of different clusters are independent. Let λ_c denote the second-largest eigenvalue in magnitude of cluster c's Markov chain. To reduce the eigenvalue of the aggregate Markov chain, CA-Fed needs to exclude all clients in the cluster $\bar{c} = \arg\max_{c\in\mathcal{C}} \lambda_c$. In this experiment, we consider a population of $N = 100$ clients grouped into $|\mathcal{C}| = 10$ clusters. We equally split the clients, or equivalently, the clusters, into two categories: "more available" with $\pi_c = 0.9$ and $\lambda_c = 0$ for $c = 0, \ldots, 4$, and "less available, correlated" with $\pi_c = 0.1$ and $\lambda_c = c/10$ for $c = 5, \ldots, 9$. In Figures 5a, 5b, and 5c, each pixel represents, for each client $k \in \mathcal{K}$ and for each communication round, the client's activity (active/inactive) and CA-Fed's decision (included/excluded in training). From the experiments, we observe that CA-Fed's decisions depend on the degree of statistical heterogeneity among clients within a cluster (i.e., *intra-cluster*) and among clusters (i.e., *inter-cluster*). When both the intra-cluster and inter-cluster clients' data distributions are homogeneous, CA-Fed starts considering the clients in cluster $\bar{c} = 9$ with $\lambda_{\bar{c}} = 0.9$, and sequentially excludes, in order, all clients from clusters $\{9, 8, 7, 6\}$ (as shown in Fig. 5a). When the clients' data distributions are homogeneous within clusters, but heterogeneous among clusters (Fig. 5b), CA-Fed still excludes all clients from clusters $c = \{9, 7, 6\}$, but decides to include clients from cluster $c = 8$. This is because these clients happen to have a lower value of $\hat{F}_k^{(t)} - \hat{F}_k^*$, and despite having a large λ_c, CA-Fed decides to include them. Finally, when both the intra-cluster and inter-cluster clients' data distributions are heterogeneous (Fig. 5c), CA-Fed can partially include clients from the more correlated clusters, even though their λ_c is large. Figure 5d compares the test accuracy achieved by CA-Fed and the baselines with spatial correlation in the same setting as in Figure 5c. The experimental results show that CA-Fed can operate correctly in the presence of spatial correlation and still outperforms the baselines (+0.6 pp w.r.t. AdaFed).

VI. Conclusion

This paper presents the first convergence analysis of a FedAvg-like federated learning (FL) algorithm in presence of heterogeneous and correlated client availability. The analysis reveals the detrimental effect of correlation on the convergence rate and highlights a fundamental trade-off between convergence speed and model bias. To navigate this tradeoff, we introduce CA-Fed, a novel FL algorithm, which adaptively manages the conflicting aims of enhancing convergence speed and reducing model bias, with the ultimate objective of maximizing model quality within the constraints of the training time available. CA-Fed achieves this goal by dynamically excluding clients who exhibit high temporal correlation and limited availability, contingent on their data distributions. Indeed, model updates from such clients may act as noise, increasing variance and slowing down the algorithm's convergence. CA-Fed disregards such clients unless their local datasets notably enhance the quality of the final model. The experimental results validate the effectiveness of our strategy, demonstrating that CA-Fed is a versatile and resilient FL algorithm, well-suited to address real-world scenarios characterized by heterogeneous and correlated client availability. Further discussions on the computation and communication costs, and fairness of CA-Fed can be found in Appendix H.

REFERENCES

[1] A. Rodio, F. Faticanti, O. Marfoq, G. Neglia, and E. Leonardi, "Federated Learning under Heterogeneous and Correlated Client Availability," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, May 2023, pp. 1–10.

[2] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen *et al.*, "A Survey on Distributed Machine Learning," *ACM Computing Surveys*, vol. 53, no. 2, pp. 30:1–30:33, Mar. 2020.

[3] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya *et al.*, "When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, Apr. 2018, pp. 63–71.

[4] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh *et al.*, "Federated Learning: Strategies for Improving Communication Efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, Apr. 2017, pp. 1273–1282.

[6] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis *et al.*, "Advances and Open Problems in Federated Learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, Jun. 2021.

[7] 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, May 2020.

[8] H. Eichner, T. Koren, B. Mcmahan, N. Srebro, and K. Talwar, "Semi-Cyclic Stochastic Gradient Descent," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, pp. 1764–1773.

[9] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan *et al.*, "A Field Guide to Federated Optimization," *arXiv:2107.06917*, Jul. 2021.

[10] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman *et al.*, "Towards Federated Learning at Scale: System Design," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 374–388, Apr. 2019.

[11] Y. Ding, C. Niu, Y. Yan, Z. Zheng, F. Wu *et al.*, "Distributed Optimization over Block-Cyclic Data," *arXiv:2002.07454*, Feb. 2020.

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

[13] T. T. Doan, "Local Stochastic Approximation: A Unified View of Federated Learning and Distributed Multi-Task Reinforcement Learning Algorithms," *arXiv:2006.13460*, Jun. 2020.

[14] 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:2002.02873*, Oct. 2020.

[15] T. Sun, Y. Sun, and W. Yin, "On Markov Chain Gradient Descent," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[16] 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.

[17] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins *et al.*, "Overcoming Catastrophic Forgetting in Neural Networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[18] M. Tang, X. Ning, Y. Wang, J. Sun, Y. Wang *et al.*, "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.

[19] M. Ribero, H. Vikalo, and G. de Veciana, "Federated Learning Under Intermittent Client Availability and Time-Varying Communication Constraints," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 1, pp. 98–111, Jan. 2023.

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

[21] A. Nichol, J. Achiam, and J. Schulman, "On First-Order Meta-Learning Algorithms," *arXiv:1803.02999*, Oct. 2018.

[22] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush *et al.*, "Adaptive Federated Optimization," in *International Conference on Learning Representations*, 2021.

[23] 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.

[24] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar *et al.*, "Federated Optimization in Heterogeneous Networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, Mar. 2020.

[25] W. Chen, S. Horváth, and P. Richtárik, "Optimal Client Sampling for Federated Learning," *Transactions on Machine Learning Research*, Aug. 2022.

[26] Y. Fraboni, R. Vidal, L. Kameni, and M. Lorenzi, "Clustered Sampling: Low-Variance and Improved Representativity for Clients Selection in Federated Learning," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, Jul. 2021, pp. 3407–3416.

[27] 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.

[28] T. T. Doan, L. M. Nguyen, N. H. Pham, and J. Romberg, "Finite-Time Analysis of Stochastic Gradient Descent under Markov Randomness," *arXiv:2003.10973*, Apr. 2020.

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

[30] 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.

[31] D. A. Levin and Y. Peres, *Markov Chains and Mixing Times: Second Edition*. American Mathematical Soc., 2017, vol. 107.

[32] 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.

[33] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 7611–7623.

[34] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks," *arXiv:1312.6211*, Mar. 2015.

[35] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring Catastrophic Forgetting in Neural Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018.

[36] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný *et al.*, "LEAF: A Benchmark for Federated Settings," *arXiv:1812.01097*, Dec. 2019.

[37] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research," *IEEE Signal Processing Magazine*, 2012.

[38] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," University of Toronto, Toronto, Tech. Rep., 2009.

[39] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated Learning with Matched Averaging," in *International Conference on Learning Representations*, 2020.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004.

[41] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2001.

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