# Explicit Personalization and Local Training: Double Communication Acceleration in Federated Learning

**Kai Yi** *kai.yi@kaust.edu.sa*
*Department of Computer Science*
*King Abdullah University of Science and Technology (KAUST)*

**Laurent Condat** *laurent.condat@kaust.edu.sa*
*Department of Computer Science*
*King Abdullah University of Science and Technology (KAUST)*
*SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA-KAUST AI)*

**Peter Richtárik** *peter.richtarik@kaust.edu.sa*
*Department of Computer Science*
*King Abdullah University of Science and Technology (KAUST)*
*SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA-KAUST AI)*

**Reviewed on OpenReview:** *https://openreview.net/forum?id=qVUEuhlaEa*

## Abstract

Federated Learning is an evolving machine learning paradigm, in which multiple clients perform computations based on their individual private data, interspersed by communication with a remote server. A common strategy to curtail communication costs is *Local Training*, which consists in performing multiple local stochastic gradient descent steps between successive communication rounds. However, the conventional approach to local training overlooks the practical necessity for client-specific *personalization*, a technique to tailor local models to individual needs. We introduce Scafflix, a novel algorithm that efficiently integrates explicit personalization with local training. This innovative approach benefits from these two techniques, thereby achieving doubly accelerated communication, as we demonstrate both in theory and practice. The code is publicly available at https://github.com/WilliamYi96/Scafflix.

## 1 Introduction

Due to privacy concerns and limited computing resources on edge devices, centralized training with all data first gathered in a datacenter is often impossible in many real-world applications. So, **Federated Learning (FL)** has gained increasing interest as a framework that enables multiple clients to do local computations, based on their personal data kept private, and to communicate back and forth with a server. FL is classically formulated as an empirical risk minimization problem of the form

$$\min_{x\in\mathbb{R}^d} \left[ f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right], \tag{ERM}$$

where $f_i$ is the local objective on client $i$, $n$ is the total number of clients, $x$ is the global model.

Thus, the usual approach is to solve (ERM) and then to deploy the obtained globally optimal model $x^\star := \arg\min_{x\in\mathbb{R}^d} f(x)$ to all clients. To reduce communication costs between the server and the clients, the practice of updating the local parameters multiple times before aggregation, known as **Local Training (LT)** (Povey et al., 2014; Moritz et al., 2016; McMahan et al., 2017; Li et al., 2020b; Haddadpour & Mahdavi, 2019; Khaled et al., 2019; 2020; Karimireddy et al., 2020; Gorbunov et al., 2020a; Mitra et al., 2021), is widely used in FL. LT, in its most modern form, is a communication-acceleration mechanism, as we detail in Section A.1.

Meanwhile, there is a growing interest in providing **personalization** to the clients, by providing them more-or-less customized models tailored to their individual needs and heterogeneous data, instead of the one-size-fits-all model $x^\star$. We review existing approaches to personalization in Section A.2. If personalization is pushed to the extreme, every client just uses its private data to learn its own locally-optimal model

$$x_i^\star \coloneqq \arg\min_{x\in\mathbb{R}^d} f_i(x)$$

and no communication at all is needed. Thus, intuitively, more personalization means less communication needed to reach a given accuracy. In other words, personalization is a communication-acceleration mechanism, like LT.

Therefore, we raise the following question: *Is it possible to achieve double communication acceleration in FL by jointly leveraging the acceleration potential of personalization and local training?*

For this purpose, we first have to formulate personalized FL as an optimization problem. A compelling interpretation of LT (Hanzely & Richtárik, 2020) is that it amounts to solve an implicit personalization objective of the form:

$$\min_{x_1,\ldots,x_n\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n f_i(x_i) + \frac{\lambda}{2n}\sum_{i=1}^n \|\bar{x}-x_i\|^2, \tag{1}$$

where $x_i \in \mathbb{R}^d$ denotes the local model at client $i \in [n] \coloneqq \{1,\ldots,n\}$, $\bar{x} \coloneqq \frac{1}{n}\sum_{i=1}^n x_i$ is the average of these local models, and $\lambda \geq 0$ is the implicit personalization parameter that controls the amount of personalization. When $\lambda$ is small, the local models tend to be trained locally. On the other hand, a larger $\lambda$ puts more penalty on making the local models $x_i$ close to their mean $\bar{x}$, or equivalently in making all models close to each other, by pushing towards averaging over all clients. Thus, LT is not only compatible with personalization, but can be actually used to implement it, though implicitly: there is a unique parameter $\lambda$ in equation 1 and it is difficult evaluate the amount of personalization for a given value of $\lambda$.

The more accurate FLIX model for personalized FL was proposed by Gasanov et al. (2022). It consists for every client $i$ to first compute locally its personally-optimal model $x_i^\star$, and then to solve the problem

$$\min_{x\in\mathbb{R}^d} \tilde{f}(x) \coloneqq \frac{1}{n}\sum_{i=1}^n f_i\big(\alpha_i x + (1-\alpha_i)x_i^\star\big), \tag{FLIX}$$

where $\alpha_i \in [0,1]$ is the explicit and individual personalization factor for client $i$. At the end, the personalized model used by client $i$ is the explicit mixture

$$\tilde{x}_i^\star \coloneqq \alpha_i x^\star + (1-\alpha_i)x_i^\star,$$

where $x^\star$ is the solution to (FLIX). A smaller value of $\alpha_i$ gives more weight to $x_i^\star$, which means more personalization. The extreme case $\alpha_i = 0$ simply means using fully personalized model without any communication. On the other hand, if $\alpha_i = 1$, the client $i$ uses the global model $x^\star$ without personalization. Thus, if all $\alpha_i$ are equal to 1, there is no personalization at all and (FLIX) reverts to (ERM). So, (FLIX) is a more general formulation of FL than (ERM). The functions in (FLIX) inherit smoothness and strong convexity from the $f_i$, so every algorithm appropriate for (ERM) can also be applied to solve (FLIX). Gasanov et al. (2022) proposed an algorithm also called FLIX to solve (FLIX), which is simply vanilla distributed gradient descent (GD) applied to (FLIX).

In this paper, we first redesign and generalize the recent Scaffnew algorithm (Mishchenko et al., 2022), which features LT and has an accelerated communication complexity, and propose Individualized-Scaffnew (i-Scaffnew), wherein the clients can have different properties. We then apply and tune i-Scaffnew for the problem (FLIX) and propose our new algorithm for personalized FL, which we call Scafflix. We answer positively to the above question and prove that Scafflix enjoys a doubly accelerated communication complexity, by jointly harnessing the acceleration potential of LT and personalization. That is, its communication complexity

**Algorithm 1** Scafflix for (FLIX)

```
1:  input: stepsizes $\gamma_1 > 0, \dots, \gamma_n > 0$; probability $p \in (0, 1]$; initial estimates $x_1^0, \dots, x_n^0 \in \mathbb{R}^d$ and $h_1^0, \dots, h_n^0 \in \mathbb{R}^d$ such that $\sum_{i=1}^n h_i^0 = 0$, personalization weights $\alpha_1, \dots, \alpha_n$
2:  at the server, $\gamma := \left(\frac{1}{n}\sum_{i=1}^n \alpha_i^2 \gamma_i^{-1}\right)^{-1}$    ◇ $\gamma$ is used by the server at Step 11
3:  at clients in parallel, $x_i^\star := \arg\min f_i$    ◇ not needed if $\alpha_i = 1$
4:  for $t = 0, 1, \dots$ do
5:    flip a coin $\theta^t := \{1$ with probability $p$, 0 otherwise$\}$
6:    for $i = 1, \dots, n$, at clients in parallel, do
7:      $\tilde{x}_i^t := \alpha_i x_i^t + (1 - \alpha_i) x_i^\star$    ◇ estimate of the personalized model $\tilde{x}_i^\star$
8:      compute an estimate $g_i^t$ of $\nabla f_i(\tilde{x}_i^t)$
9:      $\hat{x}_i^t := x_i^t - \frac{\gamma_i}{\alpha_i}\left(g_i^t - h_i^t\right)$    ◇ local SGD step
10:     if $\theta^t = 1$ then
11:       send $\frac{\alpha_i^2}{\gamma_i}\hat{x}_i^t$ to the server, which aggregates $\bar{x}^t := \frac{\gamma}{n}\sum_{j=1}^n \frac{\alpha_i^2}{\gamma_i}\hat{x}_j^t$ and broadcasts it to all clients    ◇ communication, but only with small probability $p$
12:       $x_i^{t+1} := \bar{x}^t$
13:       $h_i^{t+1} := h_i^t + \frac{p\alpha_i}{\gamma_i}\left(\bar{x}^t - \hat{x}_i^t\right)$    ◇ update of the local control variate $h_i^t$
14:     else
15:       $x_i^{t+1} := \hat{x}_i^t$
16:       $h_i^{t+1} := h_i^t$
17:     end if
18:   end for
19: end for
```

depends on the square root of the condition number of the functions $f_i$ and on the $\alpha_i$. In addition to establishing the new state of the art for personalized FL with our theoretical guarantees, we show by extensive experiments that Scafflix is efficient in real-world learning setups and outperforms existing algorithms.

Our approach is novel and its good performance is built on a solid theoretical foundation. We stress that our convergence theorem for Scafflix holds under standard assumptions, without bounded variance or any other restriction. By way of comparison with recent works, pFedGate (Chen et al., 2023) bases its theorem on the bounded diversity assumption, which is often unrealistic for non-iid FL. Neither FedCR (Zhang et al., 2023) nor FedGMM (Wu et al., 2023) comes with a conventional convergence theory. pFedGraph (Ye et al., 2023) and FED-PUB (Baek et al., 2023) also lack a solid convergence analysis.

## 2 Proposed Algorithm Scafflix and Convergence Analysis

We generalize Scaffnew (Mishchenko et al., 2022) and propose Individualized-Scaffnew (i-Scaffnew), shown as Algorithm 2 in the Appendix. Its novelty with respect to Scaffnew is to make use of different stepsizes $\gamma_i$ for the local SGD steps, in order to exploit the possibly different values of $L_i$ and $\mu_i$, as well as the different properties $A_i$ and $C_i$ of the stochastic gradients. This change is not straightforward and requires to rederive the whole proof with a different Lyapunov function and to formally endow $\mathbb{R}^d$ with a different inner product at every client.

We then apply and tune i-Scaffnew for the problem (FLIX) and propose our new algorithm for personalized FL, which we call Scafflix, shown as Algorithm 1.

We analyze Scafflix in the strongly convex case, because the analysis of linear convergence rates in this setting gives clear insights and allows us to deepen our theoretical understanding of LT and personalization. And to the best of our knowledge, there is no analysis of Scaffnew in the nonconvex setting. But we conduct several nonconvex deep learning experiments to show that our theoretical findings also hold in practice.

**Assumption 1** (Smoothness and strong convexity)**.** In the problem (FLIX) (and (ERM) as the particular case $\alpha_i \equiv 1$), we assume that for every $i \in [n]$, the function $f_i$ is $L_i$-smooth and $\mu_i$-strongly convex,[1] for some $L_i \geq \mu_i > 0$. This implies that the problem is strongly convex, so that its solution $x^\star$ exists and is unique.

We also make the two following assumptions on the stochastic gradients $g_i^t$ used in Scafflix (and i-Scaffnew as a particular case with $\alpha_i \equiv 1$).

**Assumption 2** (Unbiasedness)**.** We assume that for every $t \geq 0$ and $i \in [n]$, $g_i^t$ is an unbiased estimate of $\nabla f_i(\tilde{x}_i^t)$; that is,

$$\mathbb{E}\left[g_i^t \mid \tilde{x}_i^t\right] = \nabla f_i(\tilde{x}_i^t).$$

To characterize unbiased stochastic gradient estimates, the modern notion of *expected smoothness* is well suited (Gower et al., 2019; Gorbunov et al., 2020b):

**Assumption 3** (Expected smoothness)**.** We assume that, for every $i \in [n]$, there exist constants $A_i \geq L_i$ [2] and $C_i \geq 0$ such that, for every $t \geq 0$,

$$\mathbb{E}\left[\left\|g_i^t - \nabla f_i(\tilde{x}_i^\star)\right\|^2 \mid \tilde{x}_i^t\right] \leq 2A_i D_{f_i}(\tilde{x}_i^t, \tilde{x}_i^\star) + C_i, \tag{2}$$

where $D_\varphi(x, x') := f(x) - f(x') - \langle \nabla f(x'), x - x' \rangle \geq 0$ denotes the Bregman divergence of a function $\varphi$ at points $x, x' \in \mathbb{R}^d$.

Thus, unlike the analysis in Mishchenko et al. (2022, Assumption 4.1), where the same constants are assumed for all clients, since we consider personalization, we individualize the analysis: we consider that each client can be different and use stochastic gradients characterized by its own constants $A_i$ and $C_i$. This is more representative of practical settings. Assumption 3 is general and covers in particular the following two important cases (Gower et al., 2019):

1. (bounded variance) If $g_i^t$ is equal to $\nabla f_i(\tilde{x}_i^t)$ plus a zero-mean random error of variance $\sigma_i^2$ (this covers the case of the exact gradient $g_i^t = \nabla f_i(\tilde{x}_i^t)$ with $\sigma_i = 0$), then Assumption 3 is satisfied with $A_i = L_i$ and $C_i = \sigma_i^2$.

2. (sampling) If $f_i = \frac{1}{n_i}\sum_{j=1}^{n_i} f_{i,j}$ for some $L_i$-smooth functions $f_{i,j}$ and $g_i^t = \nabla f_{i,j^t}(\tilde{x}_i^t)$ for some $j^t$ chosen uniformly at random in $[n_i]$, then Assumption 3 is satisfied with $A_i = 2L_i$ and $C_i = \left(\frac{2}{n_i}\sum_{j=1}^{n_i}\left\|\nabla f_{i,j}(\tilde{x}_i^\star)\right\|^2\right) - 2\left\|\nabla f_i(\tilde{x}_i^\star)\right\|^2$ (this can be extended to minibatch and nonuniform sampling).

We now present our main convergence result:

**Theorem 4** (fast linear convergence)**.** *In (FLIX) and Scafflix, suppose that Assumptions 1, 2, 3 hold and that for every $i \in [n]$, $0 < \gamma_i \leq \frac{1}{A_i}$. For every $t \geq 0$, define the Lyapunov function*

$$\Psi^t := \frac{1}{n}\sum_{i=1}^{n} \frac{\gamma_{\min}}{\gamma_i}\left\|\tilde{x}_i^t - \tilde{x}_i^\star\right\|^2 + \frac{\gamma_{\min}}{p^2}\frac{1}{n}\sum_{i=1}^{n}\gamma_i\left\|h_i^t - \nabla f_i(\tilde{x}_i^\star)\right\|^2, \tag{3}$$

[1] A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be $L$-smooth if it is differentiable and its gradient is Lipschitz continuous with constant $L$; that is, for every $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$, $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, where, here and throughout the paper, the norm is the Euclidean norm. $f$ is said to be $\mu$-strongly convex if $f - \frac{\mu}{2}\|\cdot\|^2$ is convex. We refer to Bauschke & Combettes (2017) for such standard notions of convex analysis.

[2] We can suppose $A_i \geq L_i$. Indeed, we have the bias-variance decomposition $\mathbb{E}\left[\left\|g_i^t - \nabla f_i(\tilde{x}_i^\star)\right\|^2 \mid \tilde{x}_i^t\right] = \left\|\nabla f_i(\tilde{x}_i^t) - \nabla f_i(\tilde{x}_i^\star)\right\|^2 + \mathbb{E}\left[\left\|g_i^t - \nabla f_i(\tilde{x}_i^t)\right\|^2 \mid \tilde{x}_i^t\right] \geq \left\|\nabla f_i(\tilde{x}_i^t) - \nabla f_i(\tilde{x}_i^\star)\right\|^2$. Assuming that $L_i$ is the best known smoothness constant of $f_i$, we cannot improve the constant $L_i$ such that for every $x \in \mathbb{R}^d$, $\left\|\nabla f_i(x) - \nabla f_i(\tilde{x}_i^\star)\right\|^2 \leq 2L_i D_{f_i}(x, \tilde{x}_i^\star)$. Therefore, $A_i$ in equation 2 has to be $\geq L_i$.

*where $\gamma_{\min} \coloneqq \min_{i\in[n]} \gamma_i$. Then* Scafflix *converges linearly: for every $t \geq 0$,*

$$\mathbb{E}\big[\Psi^t\big] \leq (1-\zeta)^t \Psi^0 + \frac{\gamma_{\min}}{\zeta}\frac{1}{n}\sum_{i=1}^{n} \gamma_i C_i, \tag{4}$$

*where*

$$\zeta = \min\left(\min_{i\in[n]} \gamma_i \mu_i, p^2\right). \tag{5}$$

It is important to note that the range of the stepsizes $\gamma_i$, the Lyapunov function $\Psi^t$ and the convergence rate in equation 4–equation 5 do not depend on the personalization weights $\alpha_i$; they only play a role in the definition of the personalized models $\tilde{x}_i^t$ and $\tilde{x}_i^\star$. Indeed, the convergence speed essentially depends on the conditioning of the functions $x \mapsto f_i\big(\alpha_i x + (1-\alpha_i)x_i^\star\big)$, which are independent from the $\alpha_i$. More precisely, let us define, for every $i \in [n]$,

$$\kappa_i \coloneqq \frac{L_i}{\mu_i} \geq 1 \quad \text{and} \quad \kappa_{\max} = \max_{i\in[n]} \kappa_i,$$

and let us study the complexity of of Scafflix to reach $\epsilon$-accuracy, i.e. $\mathbb{E}[\Psi^t] \leq \epsilon$. If, for every $i \in [n]$, $C_i = 0$, $A_i = \Theta(L_i)$, and $\gamma_i = \Theta(\frac{1}{A_i}) = \Theta(\frac{1}{L_i})$, the iteration complexity of Scafflix is

$$\mathcal{O}\left(\left(\kappa_{\max} + \frac{1}{p^2}\right)\log(\Psi^0\epsilon^{-1})\right). \tag{6}$$

And since communication occurs with probability $p$, the communication complexity of Scafflix is

$$\mathcal{O}\left(\left(p\kappa_{\max} + \frac{1}{p}\right)\log(\Psi^0\epsilon^{-1})\right). \tag{7}$$

Note that $\kappa_{\max}$ can be much smaller than $\kappa_{\text{global}} \coloneqq \frac{\max_i L_i}{\min_i \mu_i}$, which is the condition number that appears in the rate of Scaffnew with $\gamma = \frac{1}{\max_i A_i}$. Thus, Scafflix is much more versatile and adapted to FL with heterogeneous data than Scaffnew.

**Corollary 5** (case $C_i \equiv 0$)**.** *In the conditions of Theorem 4, if $p = \Theta\big(\frac{1}{\sqrt{\kappa_{\max}}}\big)$ and, for every $i \in [n]$, $C_i = 0$, $A_i = \Theta(L_i)$, and $\gamma_i = \Theta(\frac{1}{A_i}) = \Theta(\frac{1}{L_i})$, the communication complexity of* Scafflix *is*

$$\mathcal{O}\left(\sqrt{\kappa_{\max}}\log(\Psi^0\epsilon^{-1})\right). \tag{8}$$

**Corollary 6** (general stochastic gradients)**.** *In the conditions of Theorem 4, if $p = \sqrt{\min_{i\in[n]} \gamma_i \mu_i}$ and, for every $i \in [n]$,*

$$\gamma_i = \min\left(\frac{1}{A_i}, \frac{\epsilon\mu_{\min}}{2C_i}\right) \tag{9}$$

*(or $\gamma_i \coloneqq \frac{1}{A_i}$ if $C_i = 0$), where $\mu_{\min} \coloneqq \min_{j\in[n]} \mu_j$, the iteration complexity of* Scafflix *is*

$$\mathcal{O}\!\left(\!\left(\max_{i\in[n]} \max\left(\frac{A_i}{\mu_i}, \frac{C_i}{\epsilon\mu_{\min}\mu_i}\right)\right)\log(\Psi^0\epsilon^{-1})\!\right) \!=\! \mathcal{O}\!\left(\max\left(\max_{i\in[n]}\frac{A_i}{\mu_i}, \max_{i\in[n]}\frac{C_i}{\epsilon\mu_{\min}\mu_i}\right)\log(\Psi^0\epsilon^{-1})\right) \tag{10}$$

*and its communication complexity is*

$$\mathcal{O}\left(\max\left(\max_{i\in[n]}\sqrt{\frac{A_i}{\mu_i}}, \max_{i\in[n]}\sqrt{\frac{C_i}{\epsilon\mu_{\min}\mu_i}}\right)\log(\Psi^0\epsilon^{-1})\right). \tag{11}$$

If $A_i = \Theta(L_i)$ uniformly, we have $\max_{i\in[n]}\sqrt{\frac{A_i}{\mu_i}} = \Theta(\sqrt{\kappa_{\max}})$. Thus, we see that thanks to LT, the communication complexity of Scafflix is accelerated, as it depends on $\sqrt{\kappa_{\max}}$ and $\frac{1}{\sqrt{\epsilon}}$.

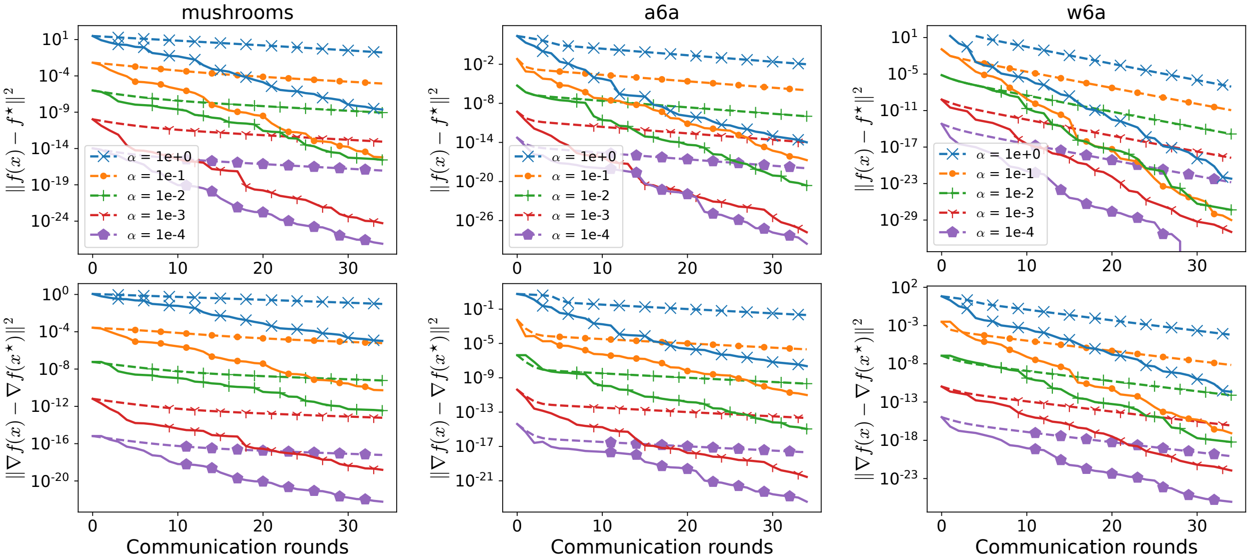


Figure 1: The objective gap $f(x^k) - f^\star$ and the squared gradient norm $\left\|\nabla f(x^k)\right\|^2$ against the number $k$ of communication rounds for Scafflix and GD on the problem (FLIX) on class-wise non-iid FL setting. We set all $\alpha_i$ to the same value for simplicity. The dashed line represents GD, while the solid line represents Scafflix. We observe the double communication acceleration achieved through explicit personalization and local training. Specifically, (a) for a given algorithm, smaller $\alpha_i$s (i.e. more personalized models) lead to faster convergence; (b) comparing the two algorithms, Scafflix is faster than GD, thanks to its local training mechanism.

In the expressions above, the acceleration effect of personalization is not visible: it is "hidden" in $\Psi^0$, because every client computes $x_i^t$ but what matters is its personalized model $\tilde{x}_i^t$, and $\left\|\tilde{x}_i^t - \tilde{x}_i^\star\right\|^2 = \alpha_i^2 \left\|x_i^t - x^\star\right\|^2$. In particular, assuming that $x_1^0 = \cdots = x_n^0 = x^0$ and $h_i^0 = \nabla f_i(\tilde{x}_i^0)$, we have

$$\Psi^0 \leq \frac{\gamma_{\min}}{n} \left\|x^0 - x^\star\right\|^2 \sum_{i=1}^n \alpha_i^2 \left(\frac{1}{\gamma_i} + \frac{\gamma_i L_i^2}{p^2}\right) \leq \left(\max_i \alpha_i^2\right) \frac{\gamma_{\min}}{n} \left\|x^0 - x^\star\right\|^2 \sum_{i=1}^n \left(\frac{1}{\gamma_i} + \frac{\gamma_i L_i^2}{p^2}\right),$$

and we see that the contribution of every client to the initial gap $\Psi^0$ is weighted by $\alpha_i^2$. Thus, the smaller the $\alpha_i$, the smaller $\Psi^0$ and the faster the convergence. This is why personalization is an acceleration mechanism in our setting.

## 3 Experiments

We first consider a convex logistic regression problem to show that the empirical behavior of Scafflix is in accordance with the theoretical convergence guarantees available in the convex case. Then, we make extensive experiments of training neural networks on large-scale distributed datasets.

### 3.1 Prelude: Convex Logistic Regression

We begin our evaluation by considering the standard convex logistic regression problem with an $l_2$ regularizer. This benchmark problem is takes the form (ERM) with

$$f_i(x) := \frac{1}{n_i} \sum_{j=1}^{n_i} \log\left(1 + \exp(-b_{i,j} x^T a_{i,j})\right) + \frac{\mu}{2}\|x\|^2,$$

where $\mu$ represents the regularization parameter, $n_i$ is the total number of data points present at client $i$; $a_{i,j}$ are the training vectors and the $b_{i,j} \in \{-1, 1\}$ are the corresponding labels. Every function $f_i$ is $\mu$-strongly convex and $L_i$-smooth with $L_i = \frac{1}{4n_i} \sum_{j=1}^{n_i} \left\|a_{i,j}\right\|^2 + \mu$. We set $\mu$ to 0.1 for this experiment. We employ the

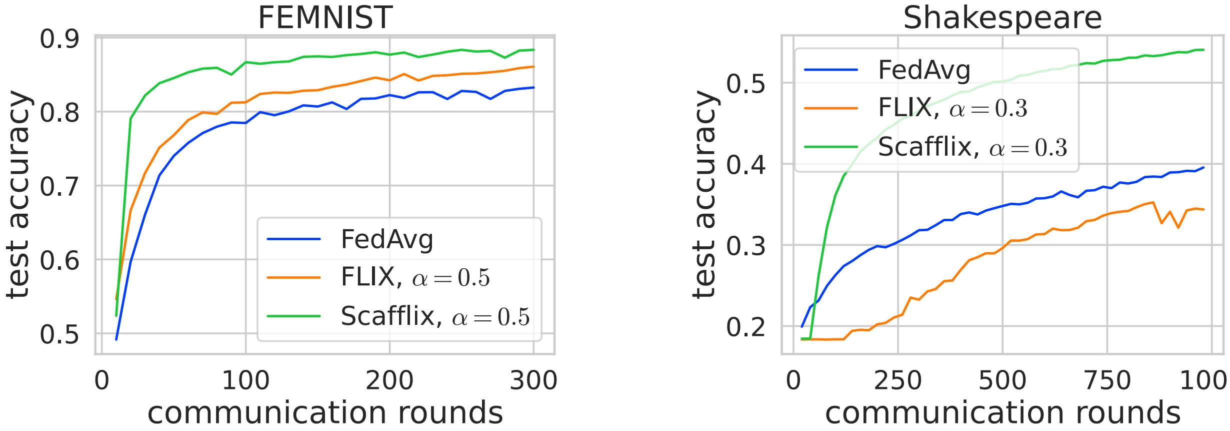


Figure 2: Comparative generalization analysis with baselines. We set the communication probability to $p = 0.2$. The left figure corresponds to the FEMNIST dataset with $\alpha = 0.5$, while the right figure corresponds to the Shakespeare dataset with $\alpha = 0.3$.

`mushrooms`, `a6a`, and `w6a` datasets from the LibSVM library (Chang & Lin, 2011) to conduct these tests. We consider several non-iid splits and present the results on feature-wise non-iid in Figure 1. We discuss the difference among non-iid settings and complementary results in Appendix E.1.

The data is distributed evenly across all clients, and the $\alpha_i$ are set to the same value. The results are shown in Figure 1. We can observe the double acceleration effect of our approach, which combines explicit personalization and accelerated local training. Lower $\alpha_i$ values, i.e. more personalization, yield faster convergence for both GD and Scafflix. Moreover, Scafflix is much faster than GD, thanks to its specialized local training mechanism.

### 3.2 Neural Network Training: Datasets and Baselines for Evaluation

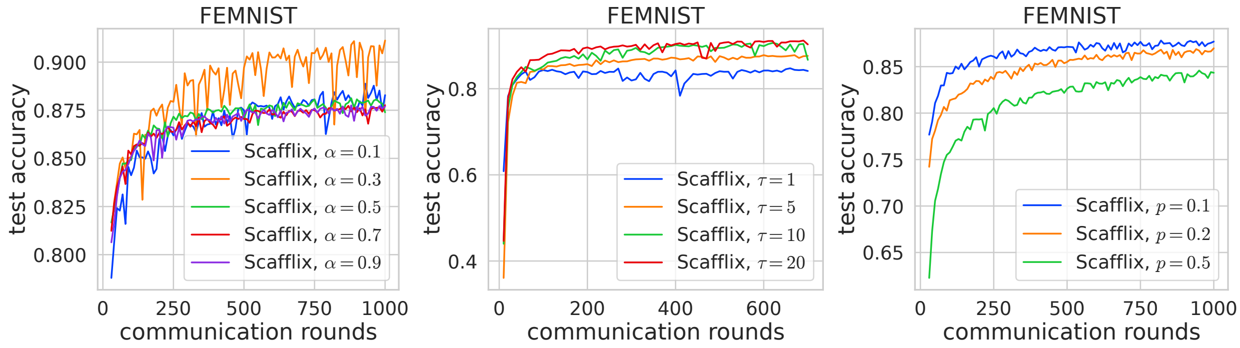
To assess the generalization capabilities of Scafflix, we undertake a comprehensive evaluation involving the training of neural networks using two widely-recognized large-scale FL datasets.

**Datasets.** Our selection comprises two notable large-scale FL datasets: Federated Extended MNIST (FEMNIST) (Caldas et al., 2018), and Shakespeare (McMahan et al., 2017). FEMNIST is a character recognition dataset consisting of 671,585 samples. In line with the methodology described in FedJax (Ro et al., 2021), we distributed these samples across 3,400 devices, with each device exhibiting a naturally non-iid characteristic. For all algorithms, we employ a Convolutional Neural Network (CNN) model, featuring two convolutional layers and one fully connected layer. The Shakespeare dataset, used for next character prediction tasks, contains a total of 16,068 samples, which we distribute randomly across 1,129 devices. For all algorithms applied to this dataset, we use a Recurrent Neural Network (RNN) model, comprising two Long Short-Term Memory (LSTM) layers and one fully connected layer.

**Baselines.** The performance of our proposed Scafflix algorithm is benchmarked against prominent baseline algorithms, specifically FLIX (Gasanov et al., 2022) and FedAvg (McMahan et al., 2016). The FLIX algorithm optimizes the FLIX objective utilizing the SGD method, while FedAvg is designed to optimize the ERM objective. We employ the official implementations for these benchmark algorithms. Comprehensive hyperparameter tuning is carried out for all algorithms, including Scafflix, to ensure optimal results. For both FLIX and Scafflix, local training is required to achieve the local minima for each client. By default, we set the local training batch size at 100 and employ SGD with a learning rate selected from the set $C_s \coloneqq \{10^{-5}, 10^{-4}, \cdots, 1\}$. Upon obtaining the local optimum, we execute each algorithm with a batch size of 20 for 1000 communication rounds. The model's learning rate is also selected from the set $C_s$. All the experiments were conducted on a single NVIDIA A100 GPU with 80GB of memory.

Table 1: Numerical comparison with baselines.

| Method | Local Training | Objective | FEMNIST | Shakespeare |
|---|---|---|---|---|
| FedAvg | Basic | (ERM) | 85.25 | 38.67 |
| FLIX | Basic | Personalized (FLIX) | 87.18 | 43.40 |
| Scaffnew | Accelerated | (ERM) | 87.73 | 51.66 |
| Scafflix | Accelerated | Personalized (FLIX) | 89.43 | 54.21 |



(a) $\alpha$s (b) $\tau$s (c) $p$s

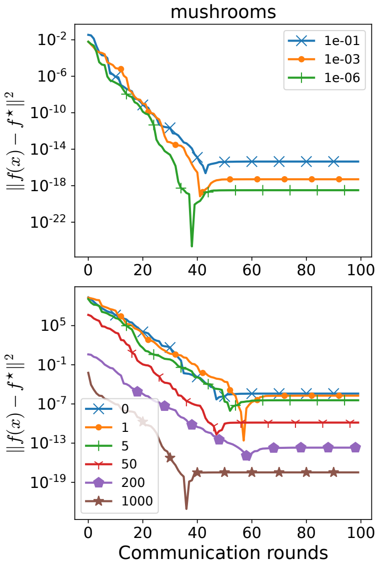
Figure 3: Key ablation studies: (a) evaluate the influence on personalization factor $\alpha$, (b) examinate the effect of different numbers of clients participating to communication, (c) compare different values of the communication probability $p$.

### 3.3 Analysis of Generalization with Limited Communication Rounds

In this section, we perform an in-depth examination of the generalization performance of Scafflix, particularly in scenarios with a limited number of training epochs. This investigation is motivated by our theoretical evidence of the double acceleration property of Scafflix. To that aim, we conduct experiments on both FEMNIST and Shakespeare. These two datasets offer a varied landscape of complexity, allowing for a comprehensive evaluation of our algorithm. In order to ensure a fair comparison with other baseline algorithms, we conducted an extensive search of the optimal hyperparameters for each algorithm. The performance assessment of the generalization capabilities was then carried out on a separate, held-out validation dataset. The hyperparameters that gave the best results in these assessments were selected as the most optimal set.

In order to examine the impact of personalization, we assume that all clients have same $\alpha_i \equiv \alpha$ and we select $\alpha$ in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. We present the results corresponding to $\alpha = 0.1$ in Figure 2. Additional comparative analyses with other values of $\alpha$ are available in the Appendix. As shown in Figure 2, it is clear that Scafflix outperforms the other algorithms in terms of generalization on both the FEMNIST and Shakespeare datasets. Interestingly, the Shakespeare dataset (next-word prediction) poses a greater challenge compared to the FEMNIST dataset (digit recognition). Despite the increased complexity of the task, Scafflix not only delivers significantly better results but also achieves this faster. Thus, Scafflix is superior both in speed and accuracy.

For additional clarity, we provide numerical results comparing our approach with FedAvg, FLIX, and Scaffnew, as summarized in Table 1. To align with Figure 2, we set $\alpha = 0.5$ for FEMNIST and $\alpha = 0.3$ for Shakespeare, reporting the former after 300 communication rounds and the latter after 1,000 rounds. Notably, the advantages of accelerated local training are even more pronounced on the Shakespeare dataset—an observation that warrants further investigation.



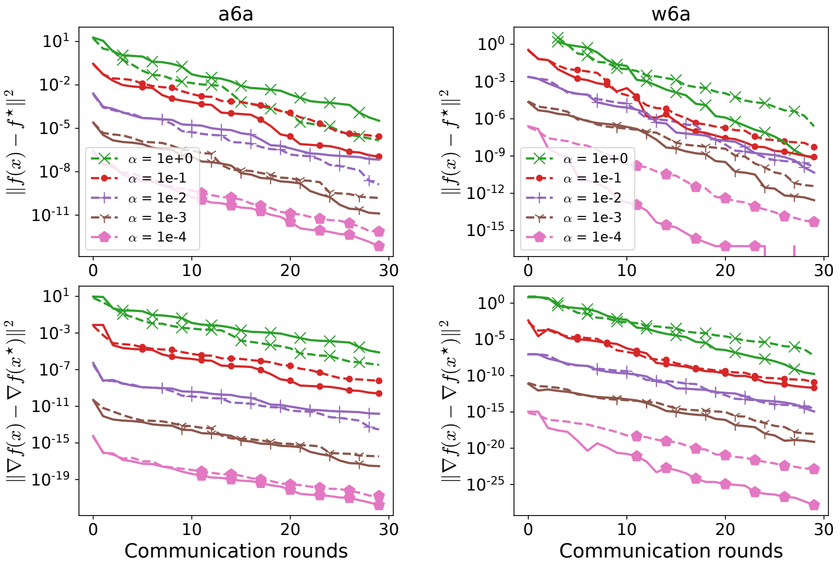


Figure 4: Inexact approximation of the local optimal.

Figure 5: Comparison between global stepsize (dashed lines) and individual stepsizes (solid lines).

### 3.4 Key Ablation Studies

In this section, we conduct several critical ablation studies to verify the efficacy of our proposed Scafflix method. These studies investigate the optimal personalization factor for Scafflix, assess the impact of the number of clients per communication round, and examine the influence of the communication probability $p$ in Scafflix.

**Optimal Personalization Factor.** In this experiment, we explore the effect of varying personalization factors on the FEMNIST dataset. The results are presented in Figure 3a. We set the batch size to 128 and determine the most suitable learning rate through a hyperparameter search. We consider linearly increasing personalization factors within the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. An exponential scale for $\alpha$ is also considered in the Appendix, but the conclusion remains the same.

We note that the optimal personalization factor for the FEMNIST dataset is 0.3. Interestingly, personalization factors that yield higher accuracy also display a slightly larger variance. However, the overall average performance remains superior. This is consistent with expectations as effective personalization may emphasize the representation of local data, and thus, could be impacted by minor biases in the model parameters received from the server.

**Number of Clients Communicating per Round.** In this ablation study, we examine the impact of varying the number of participating clients in each communication round within the Scafflix framework. By default, we set this number to 10. Here, we conduct extensive experiments with different client numbers per round, choosing $\tau$ from $\{1, 5, 10, 20\}$. The results are presented in Figure 3b. We can observe that Scafflix shows that for larger batch sizes, specifically $\tau = 10$ and 20, demonstrate slightly improved generalization performance.

**Selection of Communication Probability $p$.** In this ablation study, we explore the effects of varying the communication probability $p$ in Scafflix. We select $p$ from $\{0.1, 0.2, 0.5\}$, and the corresponding results are shown in Figure 3c. We can clearly see that a smaller value of $p$, indicating reduced communication, facilitates faster convergence and superior generalization performance. This highlights the benefits of LT, which not only makes FL faster and more communication-efficient, but also improves the learning quality.

**Inexact Local Optimal.** In FL, the primary challenge lies in minimizing communication overhead while effectively managing local computation times. Attaining a satisfactory local optimum (or approximation) for each client is both practical and similar to *pretraining* for finding a good initialization, a common practice in fields like computer vision and natural language processing. For instance, in our study of the Shakespeare dataset, distributed across 1,129 devices with over 16,000 samples, a mere *50* epochs of local training per client were necessary to achieve optimal results, as demonstrated in Figure 2. This efficiency stands in stark contrast to traditional methods, which often require more than 800 communication rounds, each involving multiple local updates.

We further conducted detailed ablation studies on logistic regression to assess the impact of inexact local optimum approximation. A threshold was set such that $\|\nabla f_i(x)\| < \epsilon$ indicates a client has reached its local optimum, with the default $\epsilon$ set to $1e-6$. Our investigation focused on the consequences of using higher $\epsilon$ values. Appendix Figure 12 details the expected number of local iterations for 100 clients. Notably, an $\epsilon$ value of $1e-1$ is found to be 23.55 times more efficient than $\epsilon = 1e-6$. Additional results for 8 workers with $\alpha = 0.1$ are presented in Figure 4, showing that $\epsilon = 1e-1$ provides a satisfactory approximation. (We anticipate an even lower computational cost for finding a local optimum approximation when the data per client is smaller.) Opting for $\epsilon = 1e-1$ is a viable strategy to reduce computation, while smaller $\epsilon$ values are advantageous for greater precision. To ensure that our initial $x_i^0$ is not already near the optimum, we initialized each element of $x_i^0$ to 100. Additionally, we explored the number of local iterations required for achieving the optimal setting, ranging from $[0, 1, 5, 200, 1000]$, as depicted in the right panel of Figure 4. These findings underscore the need for a balance between performance and computational costs. More comprehensive insights and results are provided in Appendix E.2.

**Individual Stepsizes for Each Client.** In our experiments, we initially assumed a uniform learning rate for all clients for simplicity. However, to more accurately represent the personalized approach of our method and to align closely with Algorithm 1, we explored different stepsizes for each client. Specifically, we set $\gamma_i = 1/L_i$, where $L_i$ denotes the smoothness constant of the function $f_i$ optimizing (FLIX). The impact of this variation is demonstrated in Figure 5, which presents results using the mushrooms dataset. We observed that employing individual stepsizes generally enhances performance. This approach, along with a global stepsize (indicated by dashed lines in the figure), both contribute to improved outcomes.

### 3.5 Comparisons with Personalized FL Baselines

While our research primarily seeks to ascertain the impact of explicit personalization and local training on communication costs, we recognize the interest of the community for a broad comparative scope. Accordingly, we have included extensive baseline comparisons with other recent FL and particularly personalized FL (pFL) methodologies. A comparative performance analysis on popular datasets like CIFAR100 and FMNIST is presented below:

Table 2: Results of additional baselines.

| Method | Ditto | FedSR-FT | FedPAC | FedCR | Scafflix |
|---|---|---|---|---|---|
| CIFAR100 | 58.87 | 69.95 | 69.31 | 78.49 | 72.37 |
| FMNIST | 85.97 | 87.08 | 89.49 | 93.77 | 89.62 |

We utilized the public code and adopted the optimal hyper-parameters from FedCR (Zhang et al., 2023), subsequently re-running and documenting all baseline performances under the 'non-iid' setting. Our proposed Scafflix algorithm was reported with a communication probability of $p = 0.3$ and spanned 500 communication rounds. We set the personalization factor $\alpha$ at 0.3. Based on the results, when focusing solely on the generalization (testing) performance of the final epoch, our method is on par with state-of-the-art approaches such as FedPAC (Xu et al., 2023) and FedCR (Zhang et al., 2023). However, our primary emphasis lies in demonstrating accelerated convergence.

# 4 Discussion

## 4.1 Significance and Novelty

While our approach builds on existing “5th-generation" local training methods—many of which employ control variates and exhibit (sub)linear convergence-it introduces two key innovations that yield significant theoretical and practical benefits. First, individualized stepsizes allow each client to adapt updates to its own data distribution and local smoothness/strong convexity constants, thus capturing heterogeneity more precisely than prior work employing a single global stepsize. Second, explicit personalization via FLIX is seamlessly integrated into these local updates, ensuring that client-specific models converge faster while also maintaining a strong global consensus.

At first glance, these modifications may appear incremental. However, historical precedents-such as the adoption of Polyak's adaptive stepsize (Polyak, 1964), dropout (Srivastava et al., 2014), or batch normalization (Ioffe & Szegedy, 2015)-demonstrate that small algorithmic changes often unlock disproportionately large improvements. Analogously, our refinements enable a double communication acceleration, evidenced by communication complexity depending on $\sqrt{\kappa_{\max}}$ rather than $\kappa_{\max}$ . Beyond achieving faster theoretical rates, this synergy of local training and personalization simultaneously reduces the initial error gap, enhancing real-world performance. We believe these contributions-both in handling heterogeneous data and attaining provably accelerated convergence—distinguish our work from prior state-of-the-art federated learning algorithms.

## 4.2 Learning Rate Design

Most existing work on 5th-generation local training methods (e.g., (Mishchenko et al., 2022; Malinovsky et al., 2022; Condat et al., 2023b; Meinhardt et al., 2024)) relies on a single, global learning rate applied uniformly across all clients. While effective, such an approach does not adapt to varying local data geometry or allow for personalization at the client level. In contrast, our method employs *individualized* stepsizes $\gamma_i = {}^1\!/_{A_i}$, where $A_i$ is a client-specific smoothness parameter (Theorem 4). This design is grounded in theoretical insights indicating that per-client rates can better capture heterogeneity and potentially yield faster convergence in practice.

We acknowledge that this choice of stepsize may appear straightforward; however, it represents a deliberate divergence from the convention of a uniform, global learning rate. Although more sophisticated adaptive strategies (e.g., adaptive gradient methods) could be considered in future work, here we show that even a relatively simple rule—rooted in individual smoothness constants—establishes a strong theoretical and empirical baseline. Our findings suggest that principled, client-specific learning rates open new avenues for further personalization, and we anticipate subsequent studies to refine or extend these ideas.

## 4.3 Nonconvex Setting and Partial Client Participation

In Theorem 4, we establish linear convergence in the convex setting. To the best of our knowledge, there is no existing analysis of accelerated local training methods under standard assumptions in the nonconvex regime. Addressing this nonconvex case remains a significant challenge and is beyond the scope of this work. Instead, our primary objective is to evaluate the practical effectiveness of our approach. Therefore, we conduct experiments in both convex and nonconvex settings, and the empirical results confirm the efficiency of our method.

Theoretical analysis of partial client participation in accelerated local training methods is similarly challenging. For more details on this topic, we refer interested readers to Grudzień et al. (2022); Condat et al. (2023b). In this paper, we focus on whether double acceleration can be achieved through explicit personalization and local training. In Section 3.4, we examine the effect of varying the number of participating clients per round and observe that increasing client participation slightly improves the generalization performance of Scafflix.

## 5 Conclusion

In the contemporary era of artificial intelligence, improving federated learning to achieve faster convergence and reduce communication costs is crucial to enhance the quality of models trained on huge and heterogeneous datasets. For this purpose, we introduced Scafflix, a novel algorithm that achieves double communication acceleration by redesigning the objective to support explicit personalization for individual clients, while leveraging a state-of-the-art local training mechanism. We provided complexity guarantees in the convex setting, and also validated the effectiveness of our approach in the nonconvex setting through extensive experiments and ablation studies. Thus, our work is a step forward on the important topic of communication-efficient federated learning and offers valuable insights for further investigation in the future. Future work could profitably explore the algorithm's robustness against adversarial attacks and its adaptability to differential privacy in federated settings, expanding its applicability and ensuring its resilience in more challenging scenarios. Additionally, exploring the potential of Scafflix in federated fine-tuning of large language models could yield significant insights and contributions, further advancing the field.

### Acknowledgments

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST): i) KAUST Baseline Research Scheme, ii) Center of Excellence for Generative AI, under award number 5940, iii) SDAIA-KAUST Center of Excellence in Artificial Intelligence and Data Science.

## References

S. Alam, L. Liu, M. Yan, and M. Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. preprint arXiv:2212.01548, 2022.

M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary. Federated learning with personalization layers. preprint arXiv:1912.00818, 2019.

J. Baek, W. Jeong, J. Jin, J. Yoon, and S. J. Hwang. Personalized subgraph federated learning. In *Proc. of 40th Int. Conf. Machine Learning (ICML), PMLR 202*, pp. 1396–1415, 2023.

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.* Springer, New York, 2nd edition, 2017.

D. Bui, K. Malik, J. Goetz, H. Liu, S. Moon, A. Kumar, and K. G. Shin. Federated user representation learning. preprint arXiv:1909.12535, 2019.

S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar. LEAF: A benchmark for federated settings. preprint arXiv:1812.01097, 2018.

C.-C. Chang and C.-J. Lin. LibSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

D. Chen, L. Yao, D. Gao, B. Ding, and Y. Li. Efficient personalized federated learning via sparse model-adaptation. preprint arXiv:2305.02776, 2023.

Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020.

L. Condat and P. Richtárik. RandProx: Primal-dual optimization algorithms with randomized proximal updates. In *Proc. of Int. Conf. Learning Representations (ICLR)*, 2023.

L. Condat, I. Agarský, and P. Richtárik. Provably doubly accelerated federated learning: The first theoretically successful combination of local training and compressed communication. *preprint arXiv:2210.13277*, 2022.

L. Condat, I. Agarský, G. Malinovsky, and P. Richtárik. TAMUNA: Doubly accelerated federated learning with local training, compression, and partial participation. *preprint arXiv:2302.09832*, 2023a.

L. Condat, G. Malinovsky, and P. Richtárik. TAMUNA: Accelerated federated learning with local training and partial participation. *preprint arXiv:2302.09832*, 2023b.

C. T. Dinh, N. H. Tran, and T. D. Nguyen. Personalized federated learning with Moreau envelopes. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 21394–21405, 2020.

E. Gasanov, A. Khaled, S. Horváth, and P. Richtárik. Flix: A simple and communication-efficient alternative to local methods in federated learning. In *Proc. of 24th Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2022.

E. Gorbunov, F. Hanzely, and P. Richtárik. Local SGD: Unified theory and new efficient methods. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2020a.

E. Gorbunov, F. Hanzely, and P. Richtárik. A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent. In *Proc. of 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR 108*, 2020b.

R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik. SGD: General analysis and improved rates. In *Proc. of 36th Int. Conf. Machine Learning (ICML), PMLR 97*, pp. 5200–5209, 2019.

M. Grudzień, G. Malinovsky, and P. Richtárik. Can 5th Generation Local Training Methods Support Client Sampling? Yes! In *Proc. of Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, April 2023.

Michał Grudzień, Grigory Malinovsky, and Peter Richtárik. Can 5th generation local training methods support client sampling? yes! *preprint arXiv:2212.14370*, 2022.

F. Haddadpour and M. Mahdavi. On the convergence of local descent methods in federated learning. *preprint arXiv:1910.14425*, 2019.

F. Hanzely and P. Richtárik. Federated learning of a mixture of global and local models. preprint arXiv:2002.05516, 2020.

S. Horvath, S. Laskaridis, M. Almeida, I. Leontiadis, S. Venieris, and N. Lane. FjORD: Fair and accurate federated learning under heterogeneous targets with ordered dropout. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 12876–12889, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

P. Kairouz et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2019.

S. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. Suresh. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. In *Proc. of Int. Conf. Machine Learning (ICML)*, 2020.

A. Khaled, K. Mishchenko, and P. Richtárik. First analysis of local GD on heterogeneous data. paper arXiv:1909.04715, presented at NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality, 2019.

A. Khaled, K. Mishchenko, and P. Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *Proc. of 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2020.

D. Li and J. Wang. Fedmd: Heterogenous federated learning via model distillation. preprint arXiv:1910.03581, 2019.

Q. Li, B. He, and D. Song. Model-contrastive federated learning. In *Proc. of IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 10713–10722, 2021.

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*, 2:429–450, 2020a.

X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of FedAvg on non-IID data. In *Proc. of Int. Conf. Learning Representations (ICLR)*, 2020b.

G. Malinovsky, D. Kovalev, E. Gasanov, L. Condat, and P. Richtárik. From local SGD to local fixed point methods for federated learning. In *Proc. of 37th Int. Conf. Machine Learning (ICML)*, 2020.

G. Malinovsky, K. Yi, and P. Richtárik. Variance reduced Proxskip: Algorithm, theory and application to federated learning. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2022.

A. Maranjyan, M. Safaryan, and P. Richtárik. GradSkip: Communication-accelerated local gradient methods with better computational complexity. preprint arXiv:2210.16402, 2022.

B. McMahan, E. Moore, D. Ramage, and B. Agüera y Arcas. Federated learning of deep networks using model averaging. preprint arXiv:1602.05629, 2016.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Y. Mei, P. Guo, M. Zhou, and V. Patel. Resource-adaptive federated learning with all-in-one neural composition. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2022.

Georg Meinhardt, Kai Yi, Laurent Condat, and Peter Richtárik. Prune at the clients, not the server: Accelerated sparse training in federated learning. *arXiv preprint arXiv:2405.20623*, 2024.

K. Mishchenko, G. Malinovsky, S. Stich, and P. Richtárik. ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally! In *Proc. of 39th Int. Conf. Machine Learning (ICML)*, 2022.

A. Mitra, R. Jaafar, G. Pappas, and H. Hassani. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2021.

P. Moritz, R. Nishihara, I. Stoica, and M. I. Jordan. SparkNet: Training deep networks in Spark. In *Proc. of Int. Conf. Learning Representations (ICLR)*, 2016.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

D. Povey, X. Zhang, and S. Khudanpur. Parallel training of DNNs with natural gradient and parameter averaging. preprint arXiv:1410.7455, 2014.

J. H. Ro, A. T. Suresh, and K. Wu. FedJAX: Federated learning simulation with JAX. preprint arXiv:2108.02117, 2021.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

J. Wang et al. A field guide to federated optimization. preprint arXiv:2107.06917, 2021.

Y. Wu, S. Zhang, W. Yu, Y. Liu, Q. Gu, D. Zhou, H. Chen, and W. Cheng. Personalized federated learning under mixture of distributions. 2023. preprint arXiv:2305.01068.

J. Xu, X. Tong, and S.-L. Huang. Personalized federated learning with feature alignment and classifier collaboration. preprint arXiv:2306.11867, 2023.

H. Yang, H. He, W. Zhang, and X. Cao. Fedsteg: A federated transfer learning framework for secure image steganalysis. *IEEE Trans. Network Science and Engineering*, 8(2):1084–1094, 2020.

R. Ye, Z. Ni, F. Wu, S. Chen, and Y. Wang. Personalized federated learning with inferred collaboration graphs. In *Proc. of 40th Int. Conf. Machine Learning (ICML), PMLR 202*, 2023.

H. Zhang, C. Li, W. Dai, J. Zou, and H. Xiong. FedCR: Personalized federated learning based on across-client common representation with conditional mutual information regularization. In *Proc. of 40th Int. Conf. Machine Learning (ICML), PMLR 202*, 2023.

## Contents

# A Related Work

## A.1 Local Training Methods in Federated Learning

Theoretical evolutions of LT in FL have been long-lasting, spanning five generations from empirical results to accelerated communication complexity. The celebrated FedAvg algorithm proposed by McMahan et al. (2017) showed the feasibility of communication-efficient learning from decentralized data. It belongs to the first generation of LT methods, where the focus was on empirical results and practical validations (Povey et al., 2014; Moritz et al., 2016; McMahan et al., 2017).

The second generation of studies on LT for solving (ERM) was based on homogeneity assumptions, such as bounded gradients $\left(\exists c < +\infty, \|\nabla f_i(x)\| \leq c, x \in \mathbb{R}^d,\ i \in [n]\right)$ (Li et al., 2020b) and bounded gradient diversity $\left(\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x)\|^2 \leq c\|\nabla f(x)\|^2\right)$ (Haddadpour & Mahdavi, 2019). However, these assumptions are too restrictive and do not hold in practical FL settings (Kairouz et al., 2019; Wang et al., 2021).

The third generation of approaches, under generic assumptions on the convexity and smoothness, exhibited sublinear convergence (Khaled et al., 2019; 2020) or linear convergence to a neighborhood (Malinovsky et al., 2020).

Recently, popular algorithms have emerged, such as Scaffold (Karimireddy et al., 2020), S-Local-GD (Gorbunov et al., 2020a), and FedLin (Mitra et al., 2021), successfully correcting for the client drift and enjoying linear convergence to an exact solution under standard assumptions. However, their communication complexity remains the same as with GD, namely $\mathcal{O}(\kappa \log \epsilon^{-1})$, where $\kappa := L/\mu$ is the condition number.

Finally, Scaffnew was proposed by Mishchenko et al. (2022), with accelerated communication complexity $\mathcal{O}(\sqrt{\kappa} \log \epsilon^{-1})$. This is a major achievement, which proves for the first time that LT is a communication acceleration mechanism. Thus, Scaffnew is the first algorithm in what can be considered the fifth generation of LT-based methods with accelerated convergence. Subsequent works have further extended Scaffnew with features such as variance-reduced stochastic gradients (Malinovsky et al., 2022), compression (Condat et al., 2022), partial client participation (Condat et al., 2023a), asynchronous communication of different clients (Maranjyan et al., 2022), and to a general primal–dual framework (Condat & Richtárik, 2023). The fifth generation of LT-based methods also includes the 5GCS algorithm (Grudzień et al., 2023), based on a different approach: the local steps correspond to an inner loop to compute a proximity operator inexactly. Our proposed algorithm Scafflix generalizes Scaffnew and enjoys even better accelerated communication complexity, thanks to a better dependence on the possibly different condition numbers of the functions $f_i$.

## A.2 Personalization in FL

We can distinguish three main approaches to achieve personalization:

a) One-stage training of a single global model using personalization algorithms. One common scheme is to design a suitable regularizer to balance between current and past local models (Li et al., 2021) or between global and local models (Li et al., 2020a; Hanzely & Richtárik, 2020). The FLIX model (Gasanov et al., 2022) achieves explicit personalization by balancing the local and global model using interpolation. Meta-learning is also popular in this area, as evidenced by Dinh et al. (2020), who proposed a federated meta-learning framework using Moreau envelopes and a regularizer to balance personalization and generalization.

b) Training a global model and fine-tuning every local client or knowledge transfer/distillation. This approach allows knowledge transfer from a source domain trained in the FL manner to target domains (Li & Wang, 2019), which is especially useful for personalization in healthcare domains (Chen et al., 2020; Yang et al., 2020).

c) Collaborative training between the global model and local models. The basic idea behind this approach is that each local client trains some personalized parts of a large model, such as the last few layers of a neural network. Parameter decoupling enables learning of task-specific representations for better personalization (Arivazhagan et al., 2019; Bui et al., 2019), while channel sparsity encourages each local client to train the neural network with sparsity based on their limited computation resources (Horvath et al., 2021; Alam et al., 2022; Mei et al., 2022).

**Algorithm 2** i-Scaffnew for (ERM)

```
1:  input: stepsizes $\gamma_1 > 0, \dots, \gamma_n > 0$; probability $p \in (0,1]$; initial estimates $x_1^0, \dots, x_n^0 \in \mathbb{R}^d$ and
    $h_1^0, \dots, h_n^0 \in \mathbb{R}^d$ such that $\sum_{i=1}^n h_i^0 = 0$.
2:  at the server, $\gamma := \left(\frac{1}{n}\sum_{i=1}^n \gamma_i^{-1}\right)^{-1}$                ◇ $\gamma$ is used by the server for Step 9
3:  for $t = 0, 1, \dots$ do
4:    flip a coin $\theta^t := \{1$ with probability $p$, $0$ otherwise$\}$
5:    for $i = 1, \dots, n$, at clients in parallel, do
6:      compute an estimate $g_i^t$ of $\nabla f_i(x_i^t)$
7:      $\hat{x}_i^t := x_i^t - \gamma_i\big(g_i^t - h_i^t\big)$                                  ◇ local SGD step
8:      if $\theta^t = 1$ then
9:        send $\frac{1}{\gamma_i}\hat{x}_i^t$ to the server, which aggregates $\bar{x}^t := \frac{\gamma}{n}\sum_{j=1}^n \frac{1}{\gamma_i}\hat{x}_j^t$ and broadcasts it to all clients   ◇
          communication, but only with small probability $p$
10:       $x_i^{t+1} := \bar{x}^t$
11:       $h_i^{t+1} := h_i^t + \frac{p}{\gamma_i}\big(\bar{x}^t - \hat{x}_i^t\big)$                ◇ update of the local control variate $h_i^t$
12:     else
13:       $x_i^{t+1} := \hat{x}_i^t$
14:       $h_i^{t+1} := h_i^t$
15:     end if
16:   end for
17: end for
```

Despite the significant progress made in FL personalization, many approaches only present empirical results. Our approach benefits from the simplicity and efficiency of the FLIX framework and enjoys accelerated convergence.

## B Proposed i-Scaffnew Algorithm

We consider solving (ERM) with the proposed i-Scaffnew algorithm, shown as Algorithm 2 (applying i-Scaffnew to (FLIX) yields Scafflix, as we discuss subsequently in Section C).

**Theorem 7** (fast linear convergence)**.** *In (ERM) and i-Scaffnew, suppose that Assumptions 1, 2, 3 hold and that for every $i \in [n]$, $0 < \gamma_i \le \frac{1}{A_i}$. For every $t \ge 0$, define the Lyapunov function*

$$\Psi^t := \sum_{i=1}^n \frac{1}{\gamma_i}\left\|x_i^t - x^\star\right\|^2 + \frac{1}{p^2}\sum_{i=1}^n \gamma_i \left\|h_i^t - \nabla f_i(x^\star)\right\|^2. \tag{12}$$

*Then i-Scaffnew converges linearly: for every $t \ge 0$,*

$$\mathbb{E}\big[\Psi^t\big] \le (1-\zeta)^t \Psi^0 + \frac{1}{\zeta}\sum_{i=1}^n \gamma_i C_i, \tag{13}$$

*where*

$$\zeta = \min\left(\min_{i\in[n]} \gamma_i \mu_i, p^2\right). \tag{14}$$

*Proof.* To simplify the analysis of i-Scaffnew, we introduce vector notations: the problem (ERM) can be written as

$$\text{find } \mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathcal{X}} \ \mathbf{f}(\mathbf{x}) \quad \text{s.t.} \quad W\mathbf{x} = 0, \tag{15}$$

where $\mathcal{X} := \mathbb{R}^{d\times n}$, an element $\mathbf{x} = (x_i)_{i=1}^n \in \mathcal{X}$ is a collection of vectors $x_i \in \mathbb{R}^d$, $\mathbf{f} : \mathbf{x} \in \mathcal{X} \mapsto \sum_{i=1}^n f_i(x_i)$, the linear operator $W : \mathcal{X} \to \mathcal{X}$ maps $\mathbf{x} = (x_i)_{i=1}^n$ to $(x_i - \frac{1}{n}\sum_{j=1}^n \frac{\gamma}{\gamma_j} x_j)_{i=1}^n$, for given values $\gamma_1 > 0, \dots, \gamma_n > 0$ and their harmonic mean $\gamma = \left(\frac{1}{n}\sum_{i=1}^n \gamma_i^{-1}\right)^{-1}$. The constraint $W\mathbf{x} = 0$ means that $\mathbf{x}$ minus its weighted average is zero; that is, $\mathbf{x}$ has identical components $x_1 = \cdots = x_n$. Thus, equation 15 is indeed equivalent to (ERM). $\mathbf{x}^\star := (x^\star)_{i=1}^n \in \mathcal{X}$ is the unique solution to equation 15, where $x^\star$ is the unique solution to (ERM).

Moreover, we introduce the weighted inner product in $\mathcal{X}$: $(\mathbf{x}, \mathbf{y}) \mapsto \langle \mathbf{x}, \mathbf{y} \rangle_{\boldsymbol{\gamma}} := \sum_{i=1}^n \frac{1}{\gamma_i} \langle x_i, y_i \rangle$. Then, the orthogonal projector $P$ onto the hyperspace $\{\mathbf{y} \in \mathcal{X} \ : \ y_1 = \cdots = y_n\}$, with respect to this weighted inner product, is $P : \mathbf{x} \in \mathcal{X} \mapsto \bar{\mathbf{x}} = (\bar{x})_{i=1}^n$ with $\bar{x} = \frac{\gamma}{n} \sum_{i=1}^n \frac{1}{\gamma_i} x_i$ (because $\bar{x}$ minimizes $\|\bar{\mathbf{x}} - \mathbf{x}\|_{\boldsymbol{\gamma}}^2$, so that $\frac{1}{n} \sum_{i=1}^n \frac{1}{\gamma_i}(\bar{x} - x_i) = 0$). Thus, $P$, as well as $W = \mathrm{Id} - P$, where Id denotes the identity, are self-adjoint and positive linear operators with respect to the weighted inner product. Moreover, for every $\mathbf{x} \in \mathcal{X}$,

$$\|\mathbf{x}\|_{\boldsymbol{\gamma}}^2 = \|P\mathbf{x}\|_{\boldsymbol{\gamma}}^2 + \|W\mathbf{x}\|_{\boldsymbol{\gamma}}^2 = \|\bar{\mathbf{x}}\|_{\boldsymbol{\gamma}}^2 + \|W\mathbf{x}\|_{\boldsymbol{\gamma}}^2 = \frac{n}{\gamma} \|\bar{x}\|^2 + \|W\mathbf{x}\|_{\boldsymbol{\gamma}}^2,$$

where $\bar{\mathbf{x}} = (\bar{x})_{i=1}^n$ and $\bar{x} = \frac{\gamma}{n} \sum_{i=1}^n \frac{1}{\gamma_i} x_i$.

Let us introduce further vector notations for the variables of i-Scaffnew: for every $t \geq 0$, we define the *scaled* concatenated control variate $\mathbf{h}^t := (\gamma_i h_i^t)_{i=1}^n$, $\mathbf{h}^\star := (\gamma_i h_i^\star)_{i=1}^n$, with $h_i^\star := \nabla f_i(x^\star)$, $\bar{\mathbf{x}}^t := (\bar{x}^t)_{i=1}^n$, $\mathbf{w}^t := (w_i^t)_{i=1}^n$, with $w_i^t := x_i^t - \gamma_i g_i^t$, $\mathbf{w}^\star := (w_i^\star)_{i=1}^n$, with $w_i^\star := x_i^\star - \gamma_i \nabla f_i(x_i^\star)$, $\hat{\mathbf{h}}^t := \mathbf{h}^t - pW\hat{\mathbf{x}}^t$. Finally, we denote by $\mathcal{F}_0^t$ the $\sigma$-algebra generated by the collection of $\mathcal{X}$-valued random variables $\mathbf{x}^0, \mathbf{h}^0, \ldots, \mathbf{x}^t, \mathbf{h}^t$ and by $\mathcal{F}^t$ the $\sigma$-algebra generated by these variables, as well as the stochastic gradients $g_i^t$.

We can then rewrite the iteration of i-Scaffnew as:

```
$\hat{\mathbf{x}}^t := \mathbf{w}^t + \mathbf{h}^t$
if $\theta^t = 1$ then
  $\mathbf{x}^{t+1} := \bar{\mathbf{x}}^t$
  $\mathbf{h}^{t+1} := \mathbf{h}^t - pW\hat{\mathbf{x}}^t$
else
  $\mathbf{x}^{t+1} := \hat{\mathbf{x}}^t$
  $\mathbf{h}^{t+1} := \mathbf{h}^t$
end if
```

We suppose that $\sum_{i=1}^n h_i^0 = 0$. Then, it follows from the definition of $\bar{x}^t$ that $\frac{\gamma}{n} \sum_{j=1}^n \frac{1}{\gamma_i}(\bar{x}^t - \hat{x}_j^t) = 0$, so that for every $t \geq 0$, $\sum_{i=1}^n h_i^t = 0$; that is, $W\mathbf{h}^t = \mathbf{h}^t$.

Let $t \geq 0$. We have

$$\mathbb{E}\Big[\big\|\mathbf{x}^{t+1} - \mathbf{x}^\star\big\|_{\boldsymbol{\gamma}}^2 \mid \mathcal{F}^t\Big] = p\,\big\|\bar{\mathbf{x}}^t - \mathbf{x}^\star\big\|_{\boldsymbol{\gamma}}^2 + (1-p)\,\big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|_{\boldsymbol{\gamma}}^2,$$

with

$$\big\|\bar{\mathbf{x}}^t - \mathbf{x}^\star\big\|_{\boldsymbol{\gamma}}^2 = \big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|_{\boldsymbol{\gamma}}^2 - \big\|W\hat{\mathbf{x}}^t\big\|_{\boldsymbol{\gamma}}^2.$$

Moreover,

$$\begin{aligned}
\big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|_{\boldsymbol{\gamma}}^2 &= \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|_{\boldsymbol{\gamma}}^2 + \big\|\mathbf{h}^t - \mathbf{h}^\star\big\|_{\boldsymbol{\gamma}}^2 + 2\langle \mathbf{w}^t - \mathbf{w}^\star, \mathbf{h}^t - \mathbf{h}^\star \rangle_{\boldsymbol{\gamma}} \\
&= \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|_{\boldsymbol{\gamma}}^2 - \big\|\mathbf{h}^t - \mathbf{h}^\star\big\|_{\boldsymbol{\gamma}}^2 + 2\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, \mathbf{h}^t - \mathbf{h}^\star \rangle_{\boldsymbol{\gamma}} \\
&= \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|_{\boldsymbol{\gamma}}^2 - \big\|\mathbf{h}^t - \mathbf{h}^\star\big\|_{\boldsymbol{\gamma}}^2 + 2\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, \hat{\mathbf{h}}^t - \mathbf{h}^\star \rangle_{\boldsymbol{\gamma}} - 2\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, \hat{\mathbf{h}}^t - \mathbf{h}^t \rangle_{\boldsymbol{\gamma}} \\
&= \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|_{\boldsymbol{\gamma}}^2 - \big\|\mathbf{h}^t - \mathbf{h}^\star\big\|_{\boldsymbol{\gamma}}^2 + 2\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, \hat{\mathbf{h}}^t - \mathbf{h}^\star \rangle_{\boldsymbol{\gamma}} + 2p\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, W\hat{\mathbf{x}}^t \rangle_{\boldsymbol{\gamma}} \\
&= \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|_{\boldsymbol{\gamma}}^2 - \big\|\mathbf{h}^t - \mathbf{h}^\star\big\|_{\boldsymbol{\gamma}}^2 + 2\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, \hat{\mathbf{h}}^t - \mathbf{h}^\star \rangle_{\boldsymbol{\gamma}} + 2p\,\big\|W\hat{\mathbf{x}}^t\big\|_{\boldsymbol{\gamma}}^2.
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathbb{E}\Big[\big\|\mathbf{x}^{t+1} - \mathbf{x}^\star\big\|_{\boldsymbol{\gamma}}^2 \mid \mathcal{F}^t\Big] &= \big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|_{\boldsymbol{\gamma}}^2 - p\,\big\|W\hat{\mathbf{x}}^t\big\|_{\boldsymbol{\gamma}}^2 \\
&= \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|_{\boldsymbol{\gamma}}^2 - \big\|\mathbf{h}^t - \mathbf{h}^\star\big\|_{\boldsymbol{\gamma}}^2 + 2\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, \hat{\mathbf{h}}^t - \mathbf{h}^\star \rangle_{\boldsymbol{\gamma}} + p\,\big\|W\hat{\mathbf{x}}^t\big\|_{\boldsymbol{\gamma}}^2.
\end{aligned}$$

On the other hand, we have

$$\mathbb{E}\Big[\left\|\mathbf{h}^{t+1}-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 \mid \mathcal{F}^t\Big] = p\left\|\hat{\mathbf{h}}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 + (1-p)\left\|\mathbf{h}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2$$

and

$$\begin{aligned}
\left\|\hat{\mathbf{h}}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 &= \left\|(\mathbf{h}^t-\mathbf{h}^\star)+(\hat{\mathbf{h}}^t-\mathbf{h}^t)\right\|_{\boldsymbol{\gamma}}^2 \\
&= \left\|\mathbf{h}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 + \left\|\hat{\mathbf{h}}^t-\mathbf{h}^t\right\|_{\boldsymbol{\gamma}}^2 + 2\langle \mathbf{h}^t-\mathbf{h}^\star, \hat{\mathbf{h}}^t-\mathbf{h}^t\rangle_{\boldsymbol{\gamma}} \\
&= \left\|\mathbf{h}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 - \left\|\hat{\mathbf{h}}^t-\mathbf{h}^t\right\|_{\boldsymbol{\gamma}}^2 + 2\langle \hat{\mathbf{h}}^t-\mathbf{h}^\star, \hat{\mathbf{h}}^t-\mathbf{h}^t\rangle_{\boldsymbol{\gamma}} \\
&= \left\|\mathbf{h}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 - \left\|\hat{\mathbf{h}}^t-\mathbf{h}^t\right\|_{\boldsymbol{\gamma}}^2 - 2p\langle \hat{\mathbf{h}}^t-\mathbf{h}^\star, W(\hat{\mathbf{x}}^t-\mathbf{x}^\star)\rangle_{\boldsymbol{\gamma}} \\
&= \left\|\mathbf{h}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 - p^2\left\|W\hat{\mathbf{x}}^t\right\|_{\boldsymbol{\gamma}}^2 - 2p\langle W(\hat{\mathbf{h}}^t-\mathbf{h}^\star), \hat{\mathbf{x}}^t-\mathbf{x}^\star\rangle_{\boldsymbol{\gamma}} \\
&= \left\|\mathbf{h}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 - p^2\left\|W\hat{\mathbf{x}}^t\right\|_{\boldsymbol{\gamma}}^2 - 2p\langle \hat{\mathbf{h}}^t-\mathbf{h}^\star, \hat{\mathbf{x}}^t-\mathbf{x}^\star\rangle_{\boldsymbol{\gamma}}.
\end{aligned}$$

Hence,

$$\begin{aligned}
&\mathbb{E}\Big[\left\|\mathbf{x}^{t+1}-\mathbf{x}^\star\right\|_{\boldsymbol{\gamma}}^2 \mid \mathcal{F}^t\Big] + \frac{1}{p^2}\mathbb{E}\Big[\left\|\mathbf{h}^{t+1}-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 \mid \mathcal{F}^t\Big] \\
&\qquad = \left\|\mathbf{w}^t-\mathbf{w}^\star\right\|_{\boldsymbol{\gamma}}^2 - \left\|\mathbf{h}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 + 2\langle \hat{\mathbf{x}}^t-\mathbf{x}^\star, \hat{\mathbf{h}}^t-\mathbf{h}^\star\rangle_{\boldsymbol{\gamma}} + p\left\|W\hat{\mathbf{x}}^t\right\|_{\boldsymbol{\gamma}}^2 \\
&\qquad\quad + \frac{1}{p^2}\left\|\mathbf{h}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 - p\left\|W\hat{\mathbf{x}}^t\right\|_{\boldsymbol{\gamma}}^2 - 2\langle \hat{\mathbf{h}}^t-\mathbf{h}^\star, \hat{\mathbf{x}}^t-\mathbf{x}^\star\rangle_{\boldsymbol{\gamma}} \\
&\qquad = \left\|\mathbf{w}^t-\mathbf{w}^\star\right\|_{\boldsymbol{\gamma}}^2 + \frac{1}{p^2}\left(1-p^2\right)\left\|\mathbf{h}^t-\mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2. \qquad (16)
\end{aligned}$$

Moreover, for every $i \in [n]$,

$$\begin{aligned}
\left\|w_i^t-w_i^\star\right\|^2 &= \left\|x_i^t-x^\star-\gamma_i\big(g_i^t-\nabla f_i(x^\star)\big)\right\|^2 \\
&= \left\|x_i^t-x^\star\right\|^2 - 2\gamma_i\langle x_i^t-x^\star, g_i^t-\nabla f_i(x^\star)\rangle + \gamma_i^2\left\|g_i^t-\nabla f_i(x^\star)\right\|^2,
\end{aligned}$$

and, by unbiasedness of $g_i^t$ and Assumption 2,

$$\begin{aligned}
\mathbb{E}\Big[\left\|w_i^t-w_i^\star\right\|^2 \mid \mathcal{F}_0^t\Big] &= \left\|x_i^t-x^\star\right\|^2 - 2\gamma_i\langle x_i^t-x^\star, \nabla f_i(x_i^t)-\nabla f_i(x^\star)\rangle \\
&\quad + \gamma_i^2\mathbb{E}\Big[\left\|g_i^t-\nabla f_i(x^\star)\right\|^2 \mid \mathcal{F}^t\Big] \\
&\le \left\|x_i^t-x^\star\right\|^2 - 2\gamma_i\langle x_i^t-x^\star, \nabla f_i(x_i^t)-\nabla f_i(x^\star)\rangle + 2\gamma_i^2 A_i D_{f_i}(x_i^t,x^\star) \\
&\quad + \gamma_i^2 C_i.
\end{aligned}$$

It is easy to see that $\langle x_i^t-x^\star, \nabla f_i(x_i^t)-\nabla f_i(x^\star)\rangle = D_{f_i}(x_i^t,x^\star)+D_{f_i}(x^\star,x_i^t)$. This yields

$$\begin{aligned}
\mathbb{E}\Big[\left\|w_i^t-w_i^\star\right\|^2 \mid \mathcal{F}_0^t\Big] &\le \left\|x_i^t-x^\star\right\|^2 - 2\gamma_i D_{f_i}(x^\star,x_i^t) - 2\gamma_i D_{f_i}(x_i^t,x^\star) + 2\gamma_i^2 A_i D_{f_i}(x_i^t,x^\star) \\
&\quad + \gamma_i^2 C_i.
\end{aligned}$$

In addition, the strong convexity of $f_i$ implies that $D_{f_i}(x^\star,x_i^t) \ge \frac{\mu_i}{2}\left\|x_i^t-x^\star\right\|^2$, so that

$$\mathbb{E}\Big[\left\|w_i^t-w_i^\star\right\|^2 \mid \mathcal{F}_0^t\Big] \le (1-\gamma_i\mu_i)\left\|x_i^t-x^\star\right\|^2 - 2\gamma_i(1-\gamma_i A_i)D_{f_i}(x_i^t,x^\star) + \gamma_i^2 C_i,$$

and since we have supposed $\gamma_i \leq \frac{1}{A_i}$,

$$\mathbb{E}\Big[\left\|w_i^t - w_i^\star\right\|^2 \mid \mathcal{F}_0^t\Big] \leq (1-\gamma_i\mu_i)\left\|x_i^t - x^\star\right\|^2 + \gamma_i^2 C_i.$$

Therefore,

$$\mathbb{E}\Big[\left\|\mathbf{w}^t - \mathbf{w}^\star\right\|_{\boldsymbol{\gamma}}^2 \mid \mathcal{F}_0^t\Big] \leq \max_{i\in[n]}(1-\gamma_i\mu_i)\left\|\mathbf{x}^t - \mathbf{x}^\star\right\|_{\boldsymbol{\gamma}}^2 + \sum_{i=1}^n \gamma_i C_i$$

and

$$\begin{aligned}
\mathbb{E}\big[\Psi^{t+1} \mid \mathcal{F}_0^t\big] &= \mathbb{E}\Big[\left\|\mathbf{x}^{t+1} - \mathbf{x}^\star\right\|_{\boldsymbol{\gamma}}^2 \mid \mathcal{F}_0^t\Big] + \frac{1}{p^2}\mathbb{E}\Big[\left\|\mathbf{h}^{t+1} - \mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 \mid \mathcal{F}_0^t\Big] \\
&\leq \max_{i\in[n]}(1-\gamma_i\mu_i)\left\|\mathbf{x}^t - \mathbf{x}^\star\right\|_{\boldsymbol{\gamma}}^2 + \frac{1}{p^2}\left(1-p^2\right)\left\|\mathbf{h}^t - \mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2 + \sum_{i=1}^n \gamma_i C_i \\
&\leq (1-\zeta)\left(\left\|\mathbf{x}^t - \mathbf{x}^\star\right\|_{\boldsymbol{\gamma}}^2 + \frac{1}{p^2}\left\|\mathbf{h}^t - \mathbf{h}^\star\right\|_{\boldsymbol{\gamma}}^2\right) + \sum_{i=1}^n \gamma_i C_i \\
&= (1-\zeta)\Psi^t + \sum_{i=1}^n \gamma_i C_i, \qquad (17)
\end{aligned}$$

where

$$\zeta = \min\left(\min_{i\in[n]} \gamma_i\mu_i, p^2\right).$$

Using the tower rule, we can unroll the recursion in equation 17 to obtain the unconditional expectation of $\Psi^{t+1}$. □

## C From i-Scaffnew to Scafflix

We suppose that Assumptions 1, 2, 3 hold. We define for every $i \in [n]$ the function $\tilde{f}_i : x \in \mathbb{R}^d \mapsto f_i\big(\alpha_i x + (1-\alpha_i)x_i^\star\big)$. Thus, (FLIX) takes the form of (ERM) with $f_i$ replaced by $\tilde{f}_i$.

We want to derive Scafflix from i-Scaffnew applied to (ERM) with $f_i$ replaced by $\tilde{f}_i$. For this, we first observe that for every $i \in [n]$, $\tilde{f}_i$ is $\alpha_i^2 L_i$-smooth and $\alpha_i^2\mu_i$-strongly convex. This follows easily from the fact that $\nabla\tilde{f}_i(x) = \alpha_i\nabla f_i\big(\alpha_i x + (1-\alpha_i)x_i^\star\big)$.

Second, for every $t \geq 0$ and $i \in [n]$, $g_i^t$ is an unbiased estimate of $\nabla f_i(\tilde{x}_i^t) = \alpha_i^{-1}\nabla\tilde{f}_i(x_i^t)$. Therefore, $\alpha_i g_i^t$ is an unbiased estimate of $\nabla\tilde{f}_i(x_i^t)$ satisfying

$$\mathbb{E}\Big[\left\|\alpha_i g_i^t - \nabla\tilde{f}_i(x^\star)\right\|^2 \mid x_i^t\Big] = \alpha_i^2\mathbb{E}\Big[\left\|g_i^t - \nabla f_i(\tilde{x}_i^\star)\right\|^2 \mid \tilde{x}_i^t\Big] \leq 2\alpha_i^2 A_i D_{f_i}(\tilde{x}_i^t, \tilde{x}_i^\star) + \alpha_i^2 C_i.$$

Moreover,

$$\begin{aligned}
D_{f_i}(\tilde{x}_i^t, \tilde{x}_i^\star) &= f_i(\tilde{x}_i^t) - f_i(\tilde{x}_i^\star) - \langle\nabla f_i(\tilde{x}_i^\star), \tilde{x}_i^t - \tilde{x}_i^\star\rangle \\
&= \tilde{f}_i(x_i^t) - \tilde{f}_i(x^\star) - \langle\alpha_i^{-1}\nabla\tilde{f}_i(x^\star), \alpha_i(x_i^t - x^\star)\rangle \\
&= \tilde{f}_i(x_i^t) - \tilde{f}_i(x^\star) - \langle\nabla\tilde{f}_i(x^\star), x_i^t - x^\star\rangle \\
&= D_{\tilde{f}_i}(x_i^t, x^\star).
\end{aligned}$$

Thus, we obtain Scafflix by applying i-Scaffnew to solve (FLIX), viewed as (ERM) with $f_i$ replaced by $\tilde{f}_i$, and further making the following substitutions in the algorithm: $g_i^t$ is replaced by $\alpha_i g_i^t$, $h_i^t$ is replaced by $\alpha_i h_i^t$ (so that $h_i^t$ in Scafflix converges to $\nabla f_i(\tilde{x}_i^\star)$ instead of $\nabla\tilde{f}_i(x^\star) = \alpha_i\nabla f_i(\tilde{x}_i^\star)$), $\gamma_i$ is replaced by $\alpha_i^{-2}\gamma_i$ (so that the $\alpha_i$ disappear in the theorem).

Accordingly, Theorem 4 follows from Theorem 7, with the same substitutions and with $A_i$, $C_i$ and $\mu_i$ replaced by $\alpha_i^2 A_i$, $\alpha_i^2 C_i$ and $\alpha_i^2 \mu_i$, respectively. Finally, the Lyapunov function is multiplied by $\gamma_{\min}/n$ to make it independent from $\epsilon$ when scaling the $\gamma_i$ by $\epsilon$ in Corollary 6.

We note that i-Scaffnew is recovered as a particular case of Scafflix if $\alpha_i \equiv 1$, so that Scafflix is indeed more general.

# D Proof of Corollary 6

We place ourselves in the conditions of Theorem 4. Let $\epsilon > 0$. We want to choose the $\gamma_i$ and the number of iterations $T \geq 0$ such that $\mathbb{E}\big[\Psi^T\big] \leq \epsilon$. For this, we bound the two terms $(1-\zeta)^T \Psi^0$ and $\frac{\gamma_{\min}}{\zeta n} \sum_{i=1}^n \gamma_i C_i$ in equation 4 by $\epsilon/2$.

We set $p = \sqrt{\min_{i\in[n]} \gamma_i \mu_i}$, so that $\zeta = \min_{i\in[n]} \gamma_i \mu_i$. We have

$$T \geq \frac{1}{\zeta} \log(2\Psi^0 \epsilon^{-1}) \Rightarrow (1-\zeta)^T \Psi^0 \leq \frac{\epsilon}{2}. \tag{18}$$

Moreover,

$$(\forall i \in [n] \text{ s.t. } C_i > 0)\ \gamma_i \leq \frac{\epsilon \mu_{\min}}{2C_i} \Rightarrow \frac{\gamma_{\min}}{\zeta n} \sum_{i=1}^n \gamma_i C_i \leq \frac{\epsilon}{2} \frac{\left(\min_{j\in[n]} \gamma_j\right)\left(\min_{j\in[n]} \mu_j\right)}{\min_{j\in[n]} \gamma_j \mu_j} \leq \frac{\epsilon}{2}.$$

Therefore, we set for every $i \in [n]$

$$\gamma_i \coloneqq \min\left(\frac{1}{A_i}, \frac{\epsilon \mu_{\min}}{2C_i}\right)$$

(or $\gamma_i \coloneqq \frac{1}{A_i}$ if $C_i = 0$), and we get from equation 18 that $\mathbb{E}\big[\Psi^T\big] \leq \epsilon$ after

$$\mathcal{O}\left(\left(\max_{i\in[n]} \max\left(\frac{A_i}{\mu_i}, \frac{C_i}{\epsilon \mu_{\min} \mu_i}\right)\right) \log(\Psi^0 \epsilon^{-1})\right)$$

iterations.

# E Additional Experimental Results

## E.1 Evaluating Logistic Regression under Non-IID Conditions

Our thorough evaluation investigates the potential for achieving double acceleration through both explicit personalization and efficient local training under varying data distributions. We consider the scenarios outlined below:

- *IID:* Data is uniformly distributed across all clients with identical weighting factors, denoted as $\alpha_i$.
- *Label-wise Non-IID:* We induce imbalances in label distribution among clients. The data is bifurcated into positive and negative samples, followed by a tailored sampling technique that incrementally augments the ratio of positive samples relative to negative ones. We define these ratios as $r_{\text{pos}} = (i+1)/n$ and $r_{\text{neg}} = 1 - r_{\text{pos}}$, where $i$ represents the client index, and $n$ is the number of clients.
- *Feature-wise Non-IID:* Variations in feature distribution across clients are introduced by segmenting the features into clusters with the k-means algorithm. The number of clusters corresponds to the client count.
- *Quantity-wise Non-IID:* Data volume variance among clients is realized. The distribution of data samples per client follows a Dirichlet distribution, with a default setting of $\alpha = 0.5$. Notably, a higher $\alpha$ leads to a more uniform distribution. At $\alpha = 1$, it resembles a uniform distribution, while at $\alpha < 1$, the distribution becomes skewed, resulting in a disparate data volume across workers.

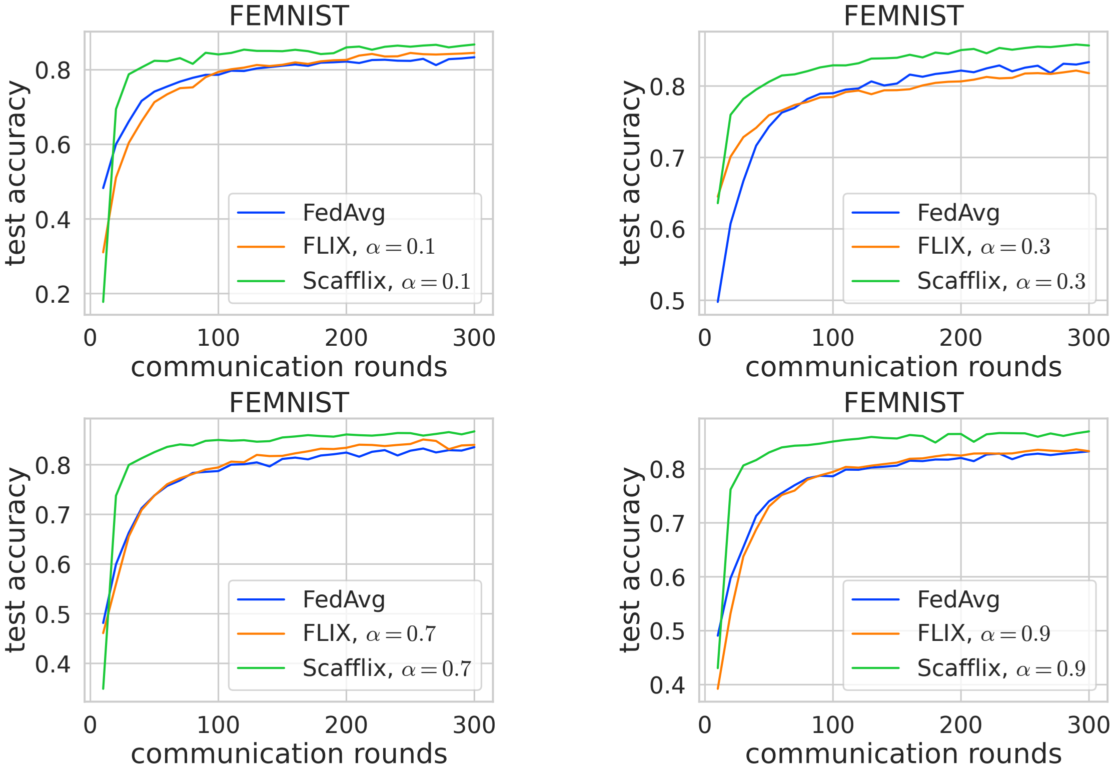


Figure 6: As part of our experimentation on the FEMNIST dataset, we performed complementary ablations by incorporating various personalization factors, represented as $\alpha$. In the main section, we present the results obtained specifically with $\alpha = 0.5$. Furthermore, we extend our analysis by highlighting the outcomes achieved with $\alpha$ values spanning from 0.1 to 0.9.

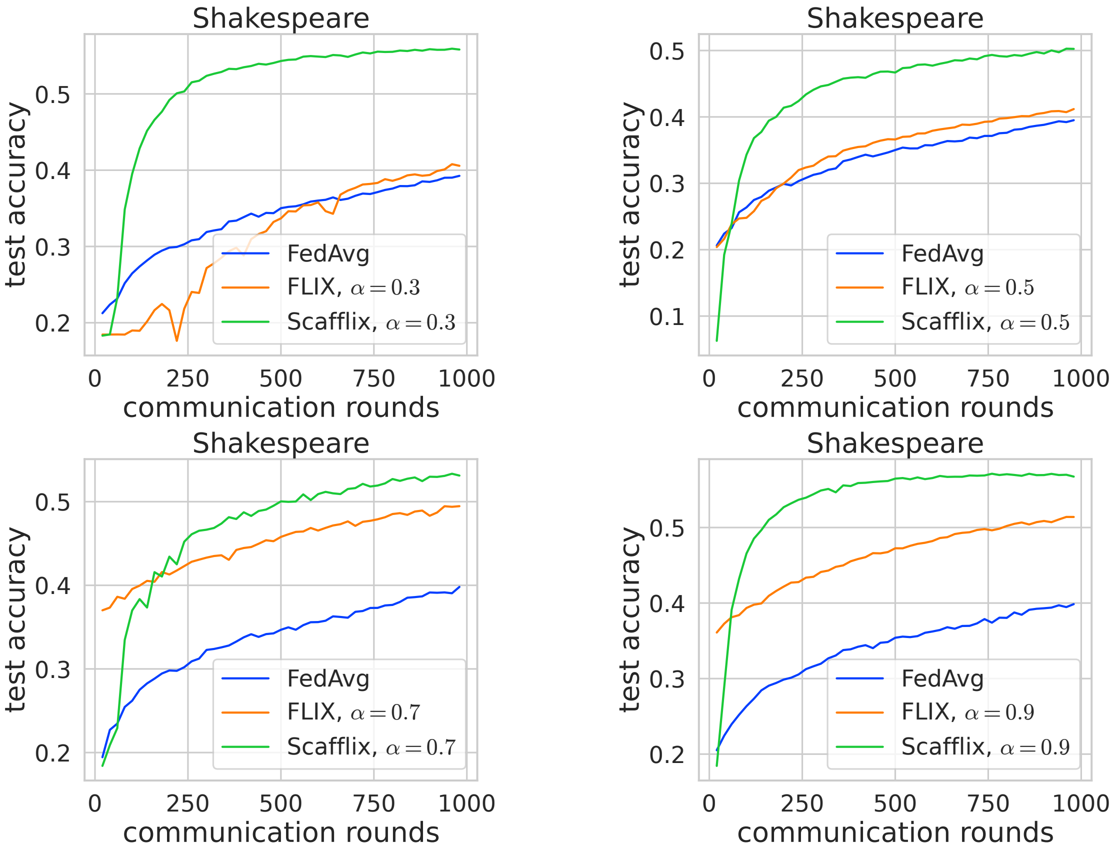


Figure 7: In our investigation of the Shakespeare dataset, we carried out complementary ablations, considering a range of personalization factors denoted as $\alpha$. The selection strategy for determining the appropriate $\alpha$ values remains consistent with the methodology described in the above figure.

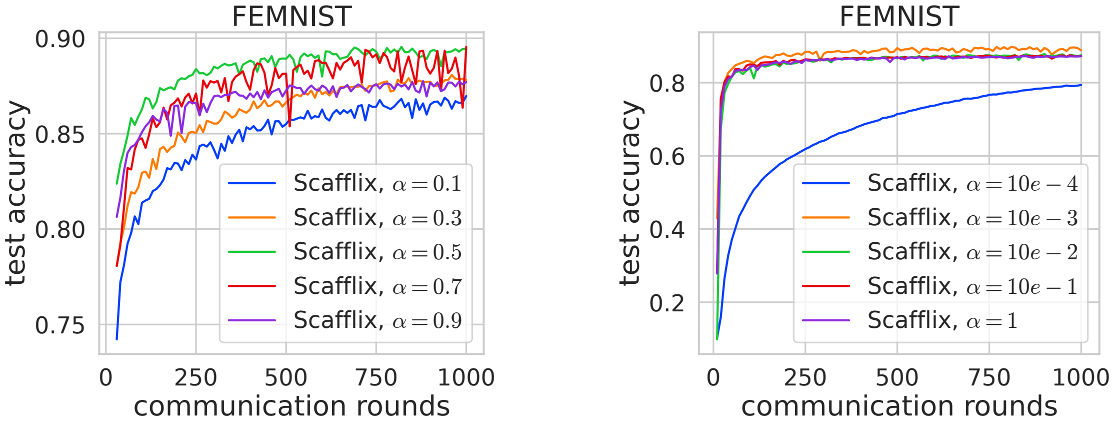


Figure 8: Ablation studies with different values of the personalization factor $\alpha$. The left figure is the complementary experiment of linearly increasing $\alpha$ with full batch size; the right is the figure with exponentially increasing $\alpha$ with default batch size of 20.

In the main text, Figure 1 illustrates the outcomes for *label-wise non-IID*. For the sake of completeness, we also include results in Figure 9, Figure 10, and Figure 11 depicting various data partitioning strategies. Across these figures, we consistently observe that Scafflix successfully achieves double acceleration.

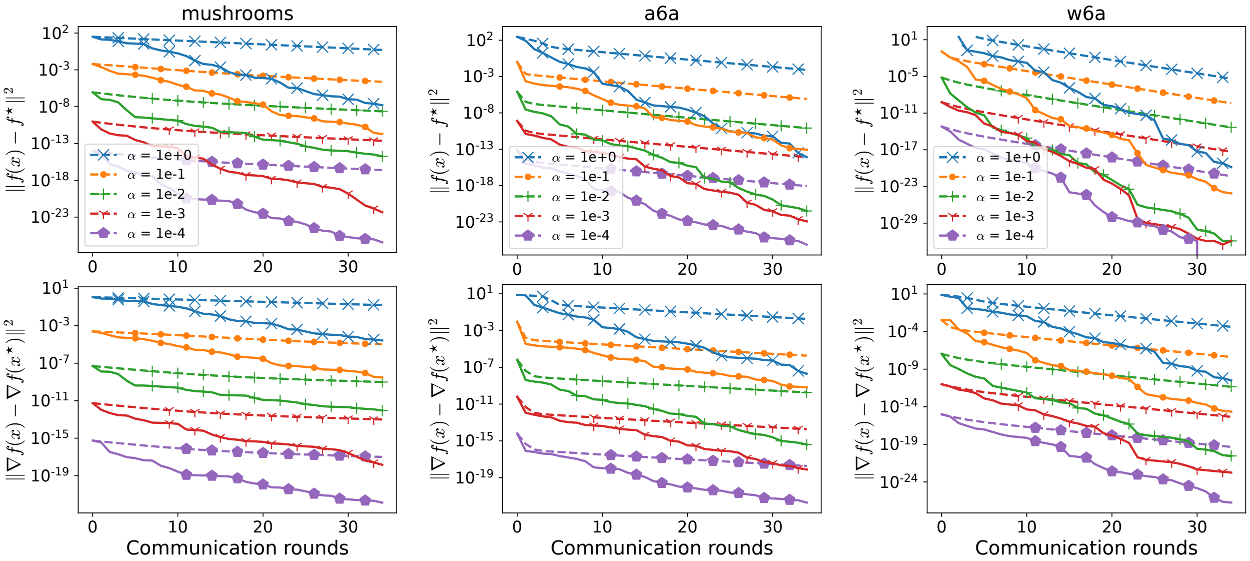


Figure 9: Results on IID splits.

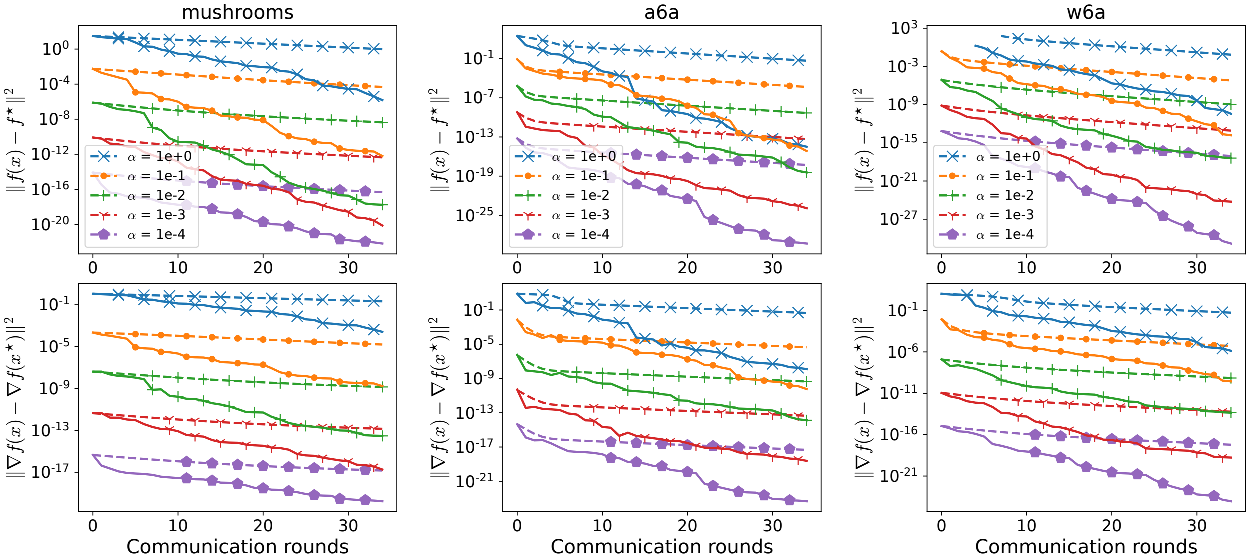


Figure 10: Feature-wise non-IID.

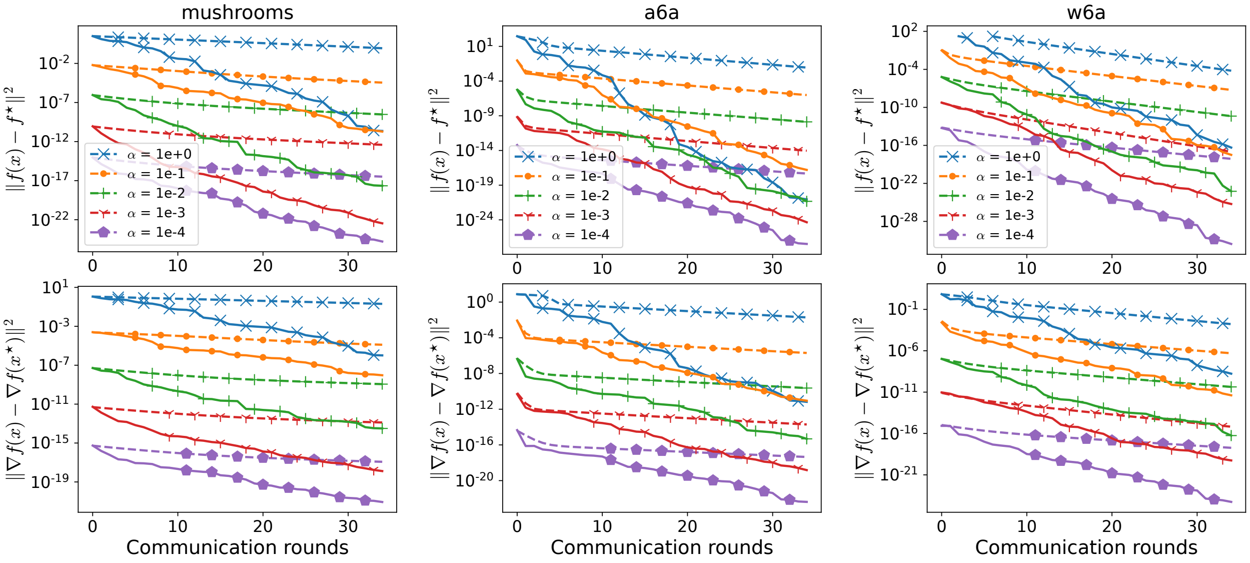


Figure 11: Quantity-wise non-IID.

### E.2 Inexact Approximation of Local Optimal

To visualize the cost of local communications, we present the expected number of local iterations to achieve an epsilon such that $\|\nabla f_i(x)\| < \epsilon$. We present the results in Figure 12. We can see there is a huge difference with respect to the different $\epsilon$. Since in FL, the communication cost is always the bottleneck, for scenarios that local computation is not that expensive, we can run more local iterations to obtain a smaller $\epsilon$. In Figure 4, we show on ablations that even choose $\epsilon = 1e-1$, which can still provide guidance leading to acceptable neighborhood. In general, there is a neighborhood here. Since in Figure 4, we consider the personalization factor $\alpha = 0.1$, here we conduct further ablations with $\alpha = 0.01$ with the results presented in Figure 13.

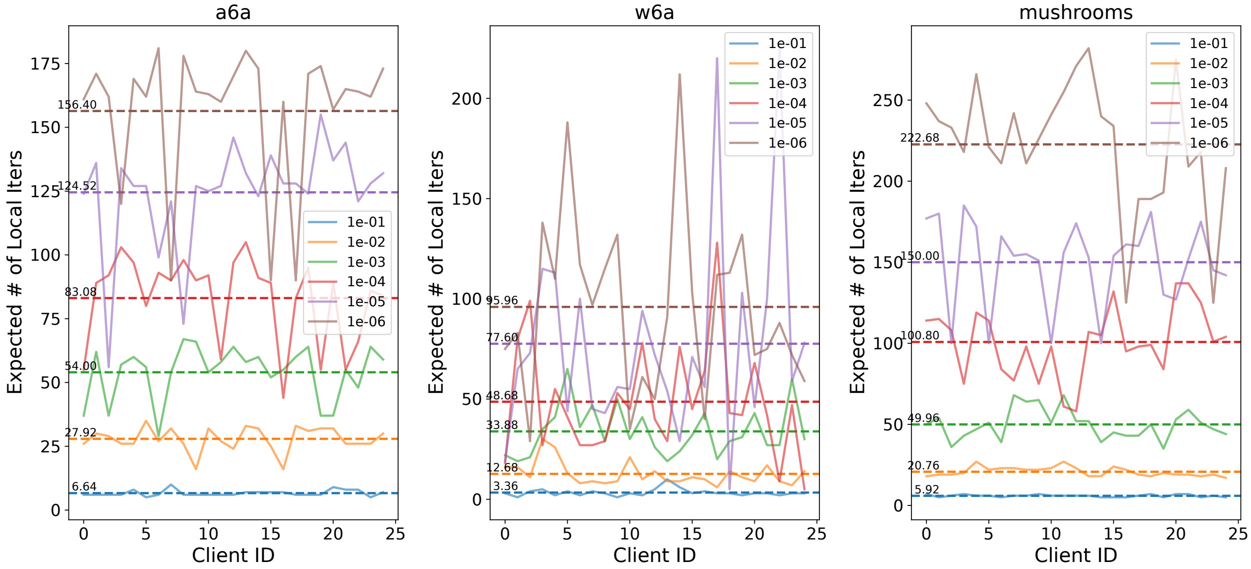


Figure 12: Number of local iterations per client for find an approximation $\bar{x}_i^\star$ of the local optimal $x_i^\star$ such that $\|\nabla f_i(x)\| < \epsilon$. The legend is $\epsilon$.

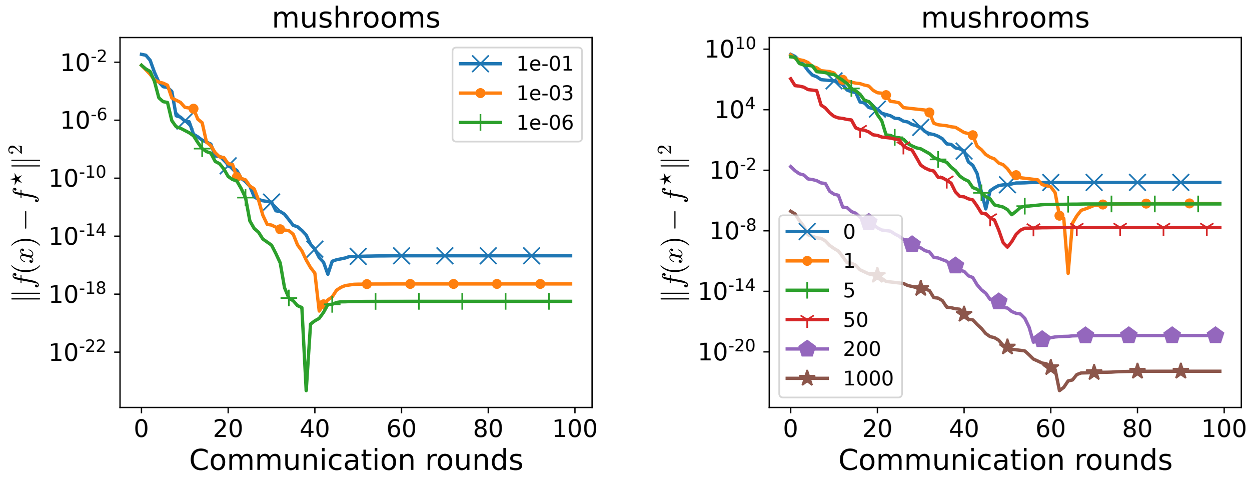


Figure 13: Inexact local optimal approximation with $\alpha = 0.01$.