Federated Learning with Partial Model Personalization

Krishna Pillutla [1] Kshitiz Malik [2] Abdelrahman Mohamed [2] Michael Rabbat [2] Maziar Sanjabi [2] Lin Xiao [2]

Abstract

We consider two federated learning algorithms for training partially personalized models, where the shared and personal parameters are updated either simultaneously or alternately on the devices. Both algorithms have been proposed in the literature, but their convergence properties are not fully understood, especially for the alternating variant. We provide convergence analyses of both algorithms in the general nonconvex setting with partial participation and delineate the regime where one dominates the other. Our experiments on real-world image, text, and speech datasets demonstrate that (a) partial personalization can obtain most of the benefits of full model personalization with a small fraction of personal parameters, and, (b) the alternating update algorithm outperforms the simultaneous update algorithm by a small but consistent margin.

1. Introduction

Federated Learning (McMahan et al., 2017) has emerged as a powerful paradigm for distributed and privacy-preserving machine learning (see Kairouz et al., 2021, and references therein). We consider a typical setting of Federated Learning (FL) with n devices (also called clients), where each device i has a training dataset of N_i samples $z_{i,1}, \cdots, z_{i,N_i}$. Let $w \in \mathbb{R}^d$ represent the parameters of a machine learning model and $f_i(w, z_{i,j})$ be the loss of the model on the training example $z_{i,j}$. Then the loss function associated with device i is $F_i(w) = (1/N_i)\sum_{j=1}^{N_i} f_i(w, z_{i,j})$. A common objective of FL is to find model parameters that minimize the weighted average loss across all devices

$$\underset{w}{\text{minimize}} \quad \sum_{i=1}^{n} \alpha_i F_i(w), \tag{1}$$

where the weights $\alpha_i > 0$ satisfy $\sum_{i=1}^n \alpha_i = 1$. A common practice is to choose $\alpha_i = N_i/N$ where $N = \sum_{i=1}^n N_i$, which corresponds to minimizing the average loss across all samples: $(1/N)\sum_{i=1}^n \sum_{j=1}^{N_i} f_i(w, z_{i,j})$.

The main motivation for minimizing the average loss over all devices is to leverage their collective statistical power for better generalization, because the amount of data on each device can be very limited. This is especially important for training modern deep learning models with large number of parameters. However, this argument assumes that the datasets from different devices are sampled from the same, or at least very similar, distributions. Given the diverse characteristics of the users and increasing trend of personalized on-device services, such an i.i.d. assumption may not hold in practice. Thus, the one-model-fits-all formulation in (1) can be ineffective and undesirable.

Several approaches have been proposed for personalized FL, including ones based on multi-task learning (Smith et al., 2017), meta learning (Fallah et al., 2020), and proximal methods (Dinh et al., 2020; Li et al., 2021). A simple formulation that captures their main idea is

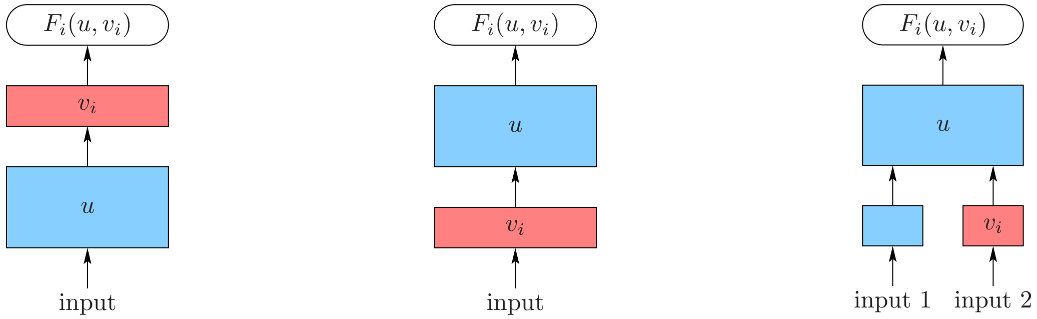
$$\underset{w_0, \{w_i\}_{i=1}^n}{\text{minimize}} \quad \sum_{i=1}^{n} \alpha_i \Big(F_i(w_i) + \frac{\lambda_i}{2}\|w_i - w_0\|^2 \Big), \tag{2}$$

where w_i for $i = 1, \ldots, n$ are personalized model parameters at the devices, w_0 is a reference model, and the λ_i's are regularization weights that control the extent of personalization. A major disadvantage of the formulation (2), which we call *full model personalization*, is that it requires twice the memory footprint of the full model, w_i and w_0 at each device, which severely limits the size of trainable models.

On the other hand, full model personalization may be unnecessary for modern deep learning models, which are composed of many simple functional units, typically organized into layers or a more general interconnected architecture. Personalizing the "right" components, selected with domain knowledge, may lead to substantial benefits with only a small increase in memory footprint. In addition, partial model personalization can be less susceptible to "catastrophic forgetting" (McCloskey & Cohen, 1989), where a large model finetuned on a small local dataset forgets the original (non-personalized) task, leading to degraded test performance.

[1] Paul G. Allen School of Computer Science & Engineering, University of Washington [2] Meta AI. Correspondence to: Krishna Pillutla <pillutla@cs.washington.edu>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).



(a) Personalized output layer(s). (b) Personalized input layer(s). (c) Personalized split input layer(s).

Figure 1. Three simple examples of partitioning deep learning models.

We consider a general setting of FL with *partial model personalization*. Specifically, we partition the model parameters into two groups: the *shared* parameters $u \in \mathbb{R}^{d_0}$ and the *personal* parameters $v_i \in \mathbb{R}^{d_i}$ for $i = 1, \ldots, n$. The full model on device i is denoted as $w_i = (u, v_i)$, and the local loss function is $F_i(u, v_i) = (1/N_i) \sum_{j=1}^{N_i} f_i\big((u, v_i), z_{i,j}\big)$. Our goal is to solve the optimization problem

$$\underset{u,\, \{v_i\}_{i=1}^n}{\text{minimize}} \quad \sum_{i=1}^{n} \alpha_i F_i(u, v_i). \tag{3}$$

Notice that the dimensions of v_i can be different across the devices, allowing the personalized components to have different number of parameters or even different architecture.

We investigate two FL algorithms for solving problem (3): *FedSim*, a simultaneous update algorithm and *FedAlt*, an alternating update algorithm. Both algorithms follow the standard FL protocol. During each round, the server randomly selects a subset of the devices for update and broadcasts the current global version of the shared parameters to devices in the subset. Each selected device then performs one or more steps of (stochastic) gradient descent to update both the shared parameters and the personal parameters, and sends only the updated shared parameters to the server for aggregation. The updated personal parameters are kept locally at the device to serve as the initialization when the device is selected for another update. In FedSim, the shared and personal parameters are updated simultaneously during each local iteration. In FedAlt, the devices first update the personal parameters with the received shared parameters fixed and then update the shared parameters with the new personal parameters fixed. We provide convergence analysis and empirical evaluation of both methods.

Contributions. Our main contributions are as follows.

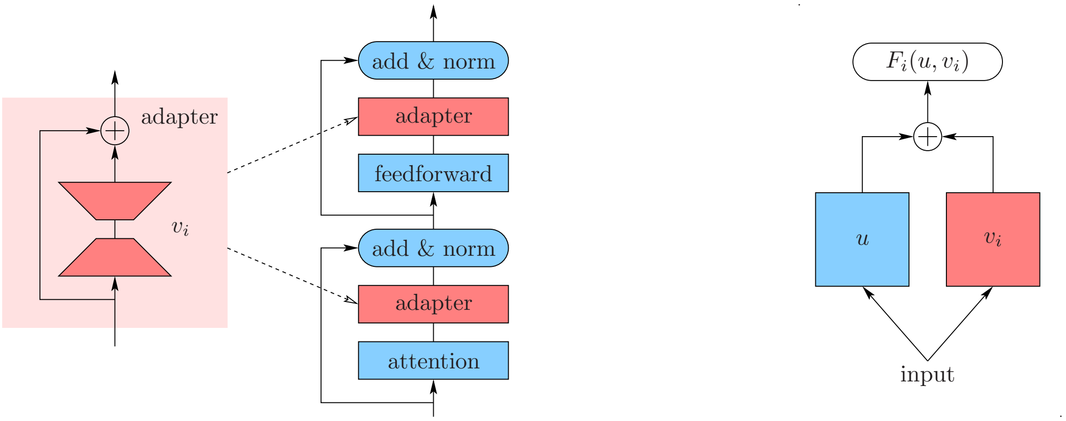
- We provide *convergence guarantees* for the FedAlt and FedSim methods in the general (smooth) *nonconvex setting with partial participation*. While both methods have appeared in the literature previously, they are either used without convergence analysis or with results on limited settings (assuming convexity or full participation). Our analysis focuses on the general nonconvex setting with partial participation, providing theoretical support for training modern deep learning models in practice. The analysis of FedAlt with partial participation is especially challenging. We decouple dependent random variables in FedAlt by introducing the technique of *virtual full participation* to overcome the difficulties.

- We conduct *extensive experiments* on realistic image, text, and speech tasks, exploring different model personalization strategies for each task, and comparing with strong baselines. Our results demonstrate that partial model personalization can obtain most of the benefit of full model personalization with only a small fraction of personalized parameters, and that FedAlt outperforms FedSim by a small but consistent margin.

- Our experiments also reveal that personalization (full or partial) may lead to *worse performance for some devices*, despite improving the average. Typical forms of regularization such as weight decay and dropout do not mitigate this issue. This phenomenon has been overlooked in previous work and calls for future research to improve both performance and fairness.

It is our hope that the generality of our theory together with strong empirical study can provide valuable guidelines for training partially personalized models in practice.

Related work. The ideas behind partial model personalization in federated learning can be traced back to seminal works on multi-task learning (Caruana, 1997; Baxter, 2000; Collobert & Weston, 2008). These works advocate for learning a shared representation across various tasks. These ideas were applied to the setting of federated learning by considering each client as a separate task by Arivazhagan et al. (2019) and Collins et al. (2021); see Figure 1a. Liang et al. (2019) instead propose to personalize the input layers to learn a personalized representation (Figure 1b).



(a) Transformer layer with two adapters.

(b) Generalized additive model.

Figure 2. More structured partial model personalization. (a) The adapter has a skip connection, thus it collapses to the identity mapping if $v_i = 0$; in addition, it has a bottleneck in the middle (Houlsby et al., 2019). (b) The generalized additive model can be further augmented with a shared input layer for representation learning.

Both optimization algorithms — FedSim and FedAlt— have appeared in the literature previously, but the scope of their convergence analyses is limited. Specifically, Liang et al. (2019), Arivazhagan et al. (2019) and Hanzely et al. (2021) use FedSim, while Collins et al. (2021) and Singhal et al. (2021) proposed variants of FedAlt. Notably, Hanzely et al. (2021) establish convergence of FedSim with participation of all devices in each round in the convex and non-convex cases, while Collins et al. (2021) prove the linear convergence of FedAlt for a two-layer linear network where $F_i(\cdot, v_i)$ and $F_i(u, \cdot)$ are both convex for fixed v_i and u respectively. We analyze both FedAlt and FedSim in the general nonconvex case with partial device participation where only a sample of devices participate in each round, hence addressing a more practical setting.

While we primarily consider problem (3) in the context of partial model personalization, it can serve as a general formulation that covers many other problems. Hanzely et al. (2021) demonstrate that various full model personalization formulations based on regularization (Dinh et al., 2020; Li et al., 2021), including (2), as well as interpolation (Deng et al., 2020a; Mansour et al., 2020) are special cases of this problem. The rates of convergence we prove in §3 are competitive with or better than those in previous works for full model personalization methods in the non-convex case.

2. Partially Personalized Models

Modern deep learning models all have a multi-layer architecture. While a complete understanding of why they work so well is still out of reach, a general insight is that the lower layers (close to the input) are responsible for feature extraction and the upper layers (close to the output) focus on complex pattern recognition. Depending on the application domain and scenarios, we may personalize either the input layer(s) or the output layer(s) of the model; see Figure 1.

In Figure 1c, the input layers are split horizontally into two parts, one shared and the other personal. They process different chunks of the input vector and their outputs are concatenated before feeding to the upper layers of the model. As demonstrated by Bui et al. (2019), this partitioning can help protect user-specific private features (input 2 in Figure 1c) as the corresponding feature embedding (through v_i) are personalized and kept local at the device. Similar architectures have also been proposed in context-dependent language models (e.g., Mikolov & Zweig, 2012).

A more structured partitioning is illustrated in Figure 2a, where a typical transformer layer (Vaswani et al., 2017) is augmented with two adapters. This architecture is proposed by Houlsby et al. (2019) for finetuning large language models. Similar residual adapter modules are proposed by Rebuffi et al. (2017) for image classification models in the context of multi-task learning. In the context of FL, we treat the adapter parameters as personal and the rest of the model parameters as shared.

Figure 2b shows a generalized additive model, where the outputs of two separate models, one shared and the other personalized, are fused to generate a prediction. Suppose the shared model is $h(u, \cdot)$ and the personal model is $h_i(v_i, \cdot)$. For regression tasks with samples $z_{i,j} = (x_{i,j}, y_{i,j})$, where $x_{i,j}$ is the input and $y_{i,j}$ is the output, we let $F_i(u, v_i) = (1/N_i)\sum_{j=1}^{N_i} f_i\big((u, v_i), z_{i,j}\big)$ with

$$f_i\big((u, v_i), z_{i,j}\big) = \|y_{i,j} - h(u, x_{i,j}) - h_i(v_i, x_{i,j})\|^2 .$$

In this special case, the personal model fits the residual of the shared model and vice-versa (Evgeniou & Pontil,

Algorithm 1 FedAlt / FedSim

```
**Input:** Initial states $u^{(0)}, \{v_i^{(0)}\}_{i=1}^n$, number of communication rounds $T$, number of devices per round $m$
**for** $t = 0, 1, \cdots, T-1$ **do**
    Server samples $m$ devices $S^{(t)} \subset \{1, \ldots, n\}$
    Server broadcasts $u^{(t)}$ to each device in $S^{(t)}$
    **for** each device $i \in S^{(t)}$ in parallel **do**
        $u_i^{(t+1)}, v_i^{(t+1)} = \text{LocalAlt / LocalSim}\big(u^{(t)}, v_i^{(t)}\big)$
        Device sends $u_i^{(t+1)}$ back to server
    Server updates $u^{(t+1)} = (1/m)\sum_{i \in S^{(t)}} u_i^{(t+1)}$
```

2004; Agarwal et al., 2020). For classification tasks, $h(u, \cdot)$ and $h_i(v_i, \cdot)$ produce probability distributions over multiple classes. We can use the cross-entropy loss between $y_{i,j}$ and a convex combination of the two model outputs: $\theta h(u, x_{i,j}) + (1-\theta) h_i(v_i, x_{i,j})$, where $\theta \in (0, 1)$ is a learnable parameter.

Finally, we can cast full model personalization in (2) as a special case of (3) by letting $u \leftarrow w_0$, $v_i \leftarrow w_i$ and

$$F_i(u, v_i) \leftarrow F_i(v_i) + (\lambda_i/2)\|v_i - u\|^2.$$

Many other formulations of full model personalization can be reduced to (3) as well; see Hanzely et al. (2021).

3. Algorithms and Convergence Analysis

In this section, we present and analyze the FedAlt and FedSim algorithms for solving problem (3). To simplify presentation, we denote $V = (v_1, \ldots, v_n) \in \mathbb{R}^{d_1 + \cdots + d_n}$ and focus on the case of $\alpha_i = 1/n$, i.e.,

$$\text{minimize}_{u,\,V} \quad F(u, V) := \tfrac{1}{n}\textstyle\sum_{i=1}^n F_i(u, v_i). \tag{4}$$

This is equivalent to (3) if we scale F_i by $n\alpha_i$, thus does not lose generality. Moreover, we consider the more general setting with local functions $F_i(u, v_i) = \mathbf{E}_{z \sim \mathcal{D}_i}[f_i((u, v_i), z)]$, where $\mathcal{D}_i$ is the local data distribution.

The FedAlt and FedSim algorithms share a common outer-loop description given in Algorithm 1. They differ only in the local update procedures LocalAlt and LocalSim, which are given in Algorithms 2 and 3 respectively. We use $\widetilde{\nabla}_u$ and $\widetilde{\nabla}_v$ to represent stochastic gradients with respect to w and v_i respectively. In LocalAlt (Algorithm 2), the personal parameters are updated first with the received shared parameters fixed, then the shared parameters are updated with the new personal parameters fixed. In LocalSim (Algorithm 3), the personal variables v_i and local version of the shared parameters u_i are updated simultaneously, with their partial gradients evaluated at the same point. They are analogous respectively to the Gauss-Seidel and Jacobi update in numerical linear algebra (e.g., Demmel, 1997, §6.5).

The rest of the section is devoted to the convergence analysis. We start with the assumptions in §3.1. In §3.2, we outline the key technical difficulty of dependent random variables in the analysis of FedAlt and describe how we overcome it with virtual full participation. Finally, we compare the convergence rates of FedAlt and FedSim in §3.3.

3.1. Assumptions

We make some assumptions for the convergence analysis.

Assumption 1 (Smoothness)**.** *For each $i = 1, \ldots, n$, the function F_i is continuously differentiable. There exist constants L_u, L_v, L_{uv}, L_{vu} such that for each $i = 1, \ldots, n$:*

- *$\nabla_u F_i(u, v_i)$ is L_u–Lipschitz with respect to u and L_{uv}–Lipschitz with respect to v_i, and*
- *$\nabla_v F_i(u, v_i)$ is L_v–Lipschitz with respect to v_i and L_{vu}–Lipschitz with respect to u.*

We summarize the relative cross-sensitivity of $\nabla_u F_i$ with respect to v_i and $\nabla_v F_i$ with respect to u with the scalar

$$\chi := \max\{L_{uv},\, L_{vu}\} \big/ \sqrt{L_u L_v}. \tag{5}$$

Assumption 2 (Bounded Variance)**.** *The stochastic gradients in Algorithm 3 and Algorithm 2 are unbiased and have bounded variance. That is, for all u and v_i,*

$$\mathbf{E}\big[\widetilde{\nabla}_u F_i(u, v_i)\big] = \nabla_u F_i(u, v_i),$$
$$\mathbf{E}\big[\widetilde{\nabla}_v F_i(u, v_i)\big] = \nabla_v F_i(u, v_i)\,.$$

Furthermore, there exist constants σ_u and σ_v such that

$$\mathbf{E}\big[\big\|\widetilde{\nabla}_u F_i(u, v_i) - \nabla_u F_i(u, v_i)\big\|^2\big] \le \sigma_u^2\,,$$
$$\mathbf{E}\big[\big\|\widetilde{\nabla}_v F_i(u, v_i) - \nabla_v F_i(u, v_i)\big\|^2\big] \le \sigma_v^2\,.$$

This is a standard bounded variance assumption on the per-device stochastic gradients (Bottou et al., 2018). We have another source of stochasticity in our setting due to partial device participation. We can view $\nabla_u F_i(u, v_i)$, when i is randomly sampled from $\{1, \ldots, n\}$, as a stochastic partial gradient of $F(u, V)$. The next assumption imposes a constant variance bound.

Assumption 3 (Partial Gradient Diversity)**.** *There exist a constant $\delta \ge 0$ such that for all u and V,*

$$\tfrac{1}{n}\textstyle\sum_{i=1}^n \big\|\nabla_u F_i(u, v_i) - \nabla_u F(u, V)\big\|^2 \le \delta^2\,.$$

Throughout this paper, we assume F is bounded below by $F^\star$ and denote $\Delta F_0 = F\left(u^{(0)}, V^{(0)}\right) - F^\star$. Further, we use the shorthands $V^{(t)} = (v_1^{(t)}, \ldots, v_n^{(t)})$,

$$\Delta_u^{(t)} = \Big\|\nabla_u F\big(u^{(t)}, V^{(t)}\big)\Big\|^2, \quad \text{and}$$

$$\Delta_v^{(t)} = \frac{1}{n}\sum_{i=1}^n \big\|\nabla_v F_i\big(u^{(t)}, v_i^{(t)}\big)\big\|^2\,.$$

Algorithm 2 LocalAlt(u, v_i)

```
Input: Number of steps $\tau_v, \tau_u$, and step sizes $\gamma_v, \gamma_u$
Initialize $v_{i,0} = v_i$
for $k = 0, 1, \cdots, \tau_v - 1$ do
    $v_{i,k+1} = v_{i,k} - \gamma_v \widetilde{\nabla}_v F_i(u, v_{i,k})$
Update $v_i^+ = v_{i,\tau_v}$ and initialize $u_{i,0} = u$
for $k = 0, 1, \cdots, \tau_u - 1$ do
    $u_{i,k+1} = u_{i,k} - \gamma_u \widetilde{\nabla}_u F_i(u_{i,k}, v_i^+)$
Update $u_i^+ = u_{i,\tau_u}$
Return $(u_i^+, v_i^+)$
```

Algorithm 3 LocalSim(u, v_i)

```
Input: Number of steps $\tau$, and step sizes $\gamma_v, \gamma_u$
Initialize $v_{i,0} = v_i$
Initialize $u_{i,0} = u$
for $k = 0, 1, \cdots, \tau - 1$ do
    $v_{i,k+1} = v_{i,k} - \gamma_v \widetilde{\nabla}_v F_i(u_{i,k}, v_{i,k})$
    $u_{i,k+1} = u_{i,k} - \gamma_u \widetilde{\nabla}_u F_i(u_{i,k}, v_{i,k})$
Update $v_i^+ = v_{i,\tau}$
Update $u_i^+ = u_{i,\tau}$
Return $(u_i^+, v_i^+)$
```

For smooth and nonconvex loss functions F_i, we obtain convergence in expectation to a stationary point of F if the expected values of these two sequences converge to zero.

3.2. Challenges of FedAlt and Virtual Full Participation

To convey the salient ideas, we assume full gradients on each device ($\sigma_u^2 = 0 = \sigma_v^2$) and a single local update per device ($\tau_u = 1 = \tau_v$). The only stochasticity in the algorithm comes from partial participation, i.e., sampling m devices in each round.

Dependent Random Variables. Consider the iterates $(u^{(t)}, V^{(t)})$ generated by FedAlt (Algorithm 1 with local updates from Algorithm 2). In order to analyze the effect of the u-update, we invoke the smoothness of $F(\cdot\,, V^{(t+1)})$ as

$$F\big(u^{(t+1)}, V^{(t+1)}\big) - F\big(u^{(t)}, V^{(t+1)}\big) \le \qquad (6)$$
$$\big\langle \nabla_u F\big(u^{(t)}, V^{(t+1)}\big),\, u^{(t+1)} - u^{(t)} \big\rangle + \frac{L_u}{2} \big\| u^{(t+1)} - u^{(t)} \big\|^2 .$$

Standard convergence proofs of stochastic gradient methods rely on the fact that when we take expectation w.r.t. the sampling $S^{(t)}$ over the first order term (within the inner product), we obtain simplifications because the gradient is usually independent of $S^{(t)}$. This is true for FedSim and the v-step of FedAlt. However, this is not the case for the u-step of FedAlt since

$$\mathbf{E}_t \Big[\big\langle \nabla_u F\big(u^{(t)}, V^{(t+1)}\big),\, u^{(t+1)} - u^{(t)} \big\rangle \Big] \neq$$
$$\big\langle \mathbf{E}_t[\nabla_u F\big(u^{(t)}, V^{(t+1)}\big)],\, \mathbf{E}_t[u^{(t+1)} - u^{(t)}] \big\rangle$$

in general, where $\mathbf{E}_t = \mathbf{E}[\,\cdot\,|u^{(t)}, V^{(t)}]$ denotes the expectation w.r.t. $S^{(t)}$. Indeed, $V^{(t+1)}$ is already updated based on $S^{(t)}$, so both $V^{(t+1)}$ and $u^{(t+1)}$ are dependent random variables, due to their mutual dependence on the sampling $S^{(t)}$; see Figure 3 (left). Therefore, directly taking expectation w.r.t. $S^{(t)}$ in (6) does not lead to a useful result.

Virtual Full Participation. We decouple the dependent random variables with virtual full participation. Define $\widetilde{V}^{(t+1)}$ as the result of local v-updates as if *every* device had participated. This iterate is *virtual*, meaning that it is a tool of the analysis but is not required by the algorithm. We introduce $\widetilde{V}^{(t+1)}$ on the right hand side of (6) to get

$$F\big(u^{(t+1)}, V^{(t+1)}\big) - F\big(u^{(t)}, V^{(t+1)}\big) \le E^{(t)} +$$
$$\big\langle \nabla_u F(u^{(t)}, \widetilde{V}^{(t+1)}),\, u^{(t+1)} - u^{(t)} \big\rangle + \frac{L_u}{2} \big\| u^{(t+1)} - u^{(t)} \big\|^2 ,$$

where $E^{(t)}$ is the error term from replacing $V^{(t+1)}$ with $\widetilde{V}^{(t+1)}$. Since $\widetilde{V}^{(t+1)}$ is deterministic when conditioned on $(u^{(t)}, V^{(t)})$, we can now take an expectation w.r.t. the sampling $S^{(t)}$ over $u^{(t+1)}$ only, cf. Figure 3 (right). This allows us to simplify the first order term as

$$\begin{aligned} &\mathbf{E}_t \Big[\big\langle \nabla_u F\big(u^{(t)}, \widetilde{V}^{(t+1)}\big),\, u^{(t+1)} - u^{(t)} \big\rangle \Big] \\ &= \big\langle \nabla_u F\big(u^{(t)}, \widetilde{V}^{(t+1)}\big),\, \mathbf{E}_t[u^{(t+1)} - u^{(t)}] \big\rangle \\ &= -\frac{\gamma_u}{n} \sum_{i=1}^{n} \mathbf{E}_t \|\nabla_u F(u^{(t)}, \tilde{v}^{(t+1)})\|^2 . \end{aligned}$$

Finally, we bound the error term $\mathbf{E}_t[E^{(t)}] \le O(L_u \gamma_u^2 + \chi^2 L_v \gamma_v^2)$, which can be made small by choosing appropriately small learning rates.

The technique of virtual full participation is distinct from shadow iterates $\bar{u}_k^{(t)} = (1/n) \sum_{i=1}^n u_{i,k}^{(t)}$ typically used in decentralized (Yuan et al., 2016) and federated optimization (Wang et al., 2021), and could be of independent interest. We refer to Appendix A.2 for additional details.

3.3. Comparing FedAlt and FedSim

We first present our main result for FedAlt (Algorithm 1 with LocalAlt). The proof relies on the technique of virtual full participation and is proved in Appendix A.3.

Theorem 1 (**Convergence of FedAlt**). *Suppose Assumptions 1, 2 and 3 hold and the learning rates in FedAlt are chosen as $\gamma_u = \eta/(L_u \tau_u)$ and $\gamma_v = \eta/(L_v \tau_v)$. For a choice of η depending on the problem parameters $L_u, L_v, \chi^2, \sigma_u^2, \sigma_v^2, \delta^2, m, n$, and the number of rounds T,*

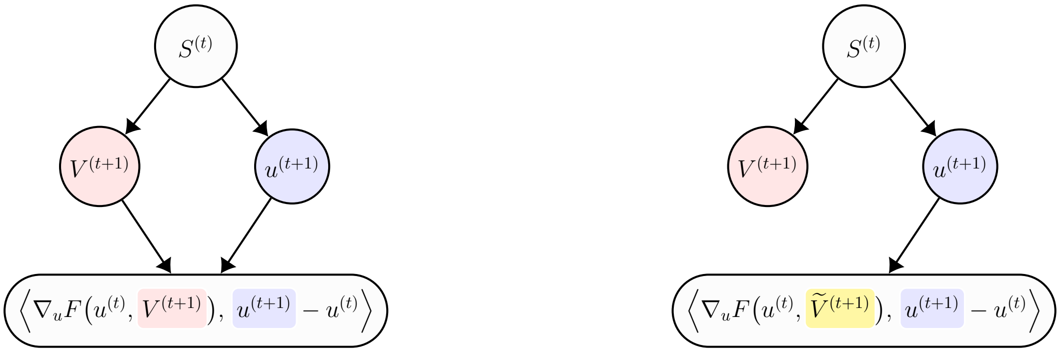

$S^{(t)}$
$V^{(t+1)}$
$u^{(t+1)}$
$\left\langle \nabla_u F\big(u^{(t)}, V^{(t+1)}\big),\ u^{(t+1)} - u^{(t)} \right\rangle$
$S^{(t)}$
$V^{(t+1)}$
$u^{(t+1)}$
$\left\langle \nabla_u F\big(u^{(t)}, \widetilde{V}^{(t+1)}\big),\ u^{(t+1)} - u^{(t)} \right\rangle$

Figure 3. **Left**: Graphical model depicting the problem of dependent random variables in the analysis of FedAlt. We cannot take an expectation of the bottom-most inner product term w.r.t. the device sampling $S^{(t)}$ because both $V^{(t+1)}$ and $u^{(t+1)}$ depend on it. **Right**: Virtual full participation overcomes this problem, since the virtual iterates $\widetilde{V}^{(t+1)}$ are statistically independent of the sampling $S^{(t)}$. The expectation can now pass through the inner product, as required by standard stochastic gradient analyses.

we have (ignoring absolute constants),

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{1}{L_u}\mathbf{E}\big[\Delta_u^{(t)}\big] + \frac{m}{nL_v}\mathbf{E}\big[\Delta_v^{(t)}\big]\right) \le \frac{\left(\Delta F_0\, \sigma^2_{\mathrm{alt},1}\right)^{1/2}}{\sqrt{T}} + \frac{\left(\Delta F_0^2\, \sigma^2_{\mathrm{alt},2}\right)^{1/3}}{T^{2/3}} + O\left(\frac{1}{T}\right), \tag{7}$$

where we define effective variance terms

$$\sigma^2_{\mathrm{alt},1} = \frac{\delta^2}{L_u}\left(1 - \frac{m}{n}\right) + \frac{\sigma_u^2}{L_u} + \frac{\sigma_v^2(m + \chi^2(n-m))}{L_v n},$$
$$\sigma^2_{\mathrm{alt},2} = \frac{\sigma_u^2 + \delta^2}{L_u}(1 - \tau_u^{-1}) + \frac{\sigma_v^2 m}{L_v n}(1 - \tau_v^{-1}) + \frac{\chi^2\sigma_v^2}{L_v},$$

and $O(\cdot)$ hides problem constants independent of T.

The left-hand side of (7) is the average over time of a weighted sum of $\mathbf{E}\big[\Delta_u^{(t)}\big]$ and $\mathbf{E}\big[\Delta_v^{(t)}\big]$. Convergence is measured in the rate at which this quantity decays to zero and depends on effective noise variances $\sigma^2_{\mathrm{alt},1}, \sigma^2_{\mathrm{alt},2}$; these are weighed sums of the variances δ^2, σ_u^2, and σ_v^2 contributed by the three sources of stochasticity. The right side contains a standard $T^{-1/2}$ term with effective noise variance $\sigma^2_{\mathrm{alt},1}$ and a lower order $T^{-2/3}$ term with variance $\sigma^2_{\mathrm{alt},2}$.

Next, we present our main result for FedSim (Algorithm 1 with LocalSim), proved in Appendix A.4.

Theorem 2 (Convergence of FedSim). *Suppose Assumptions 1, 2 and 3 hold and the learning rates in FedSim are chosen as $\gamma_u = \eta/(L_u\tau)$ and $\gamma_v = \eta/(L_v\tau)$. Then, for a η depending on the problem parameters and the number of rounds T, the bound (7) holds where the effective variance terms $\sigma^2_{\mathrm{alt},1}, \sigma^2_{\mathrm{alt},2}$ are respectively replaced by*

$$\sigma^2_{\mathrm{sim},1} = (1+\chi^2)\left(\frac{\delta^2}{L_u}\left(1 - \frac{m}{n}\right) + \frac{\sigma_u^2}{L_u} + \frac{\sigma_v^2 m}{L_v n}\right),$$
$$\sigma^2_{\mathrm{sim},2} = (1+\chi^2)\left(\frac{\delta^2}{L_u} + \frac{\sigma_u^2}{L_u} + \frac{\sigma_v^2}{L_v}\right)(1 - \tau^{-1}).$$

The bound of FedSim is analogous to that of FedAlt, with the only difference in the noise terms $\sigma^2_{\mathrm{sim},1}$ and $\sigma^2_{\mathrm{sim},2}$.

FedAlt vs. FedSim: Two Regimes. Comparing the variances $\sigma^2_{\mathrm{alt},1}$ and $\sigma^2_{\mathrm{sim},1}$ in the leading $1/\sqrt{T}$ term, we identify two regimes in terms of problem parameters. The regime where FedAlt dominates FedSim is characterized by the condition

$$\frac{\sigma_v^2}{L_v}\left(1 - \frac{2m}{n}\right) < \frac{\sigma_u^2 + \delta^2(1 - m/n)}{mL_u}.$$

A practically relevant scenario where this is true is $\sigma_v^2 \approx 0$ and $\sigma_u^2 \approx 0$ from using a large or full batch on a small number of samples per device. In this case, the rate of FedAlt is better than FedSim by a factor of $(1+\chi^2)^{1/2}$, indicating that the rate of FedAlt is less affected by the coupling χ^2 between the personal and shared parameters. Our experiments in §4 corroborate the practical relevance of this regime.

Extensions and Discussion. Theorems 1 and 2 are also interesting because of the broad generality of the optimization model (3), as we discussed in §2 and as pointed out by Hanzely et al. (2021). In particular, Theorems 1 and 2 also give rates for full personalization schemes without convergence guarantees in the nonconvex case such as FedRes (Agarwal et al., 2020), Mapper (Mansour et al., 2020), and Ditto (Li et al., 2021). Furthermore, our rates are better than those of (Dinh et al., 2020) for their pFedMe objective.

Table 1. Summary of datasets and models. A histogram of data per device is given in Figure 6 (Appendix B).

Task	Dataset	#Classes	Model	# Model Params	#Devices	#Data per device	
						Mean	Max
Next-word prediction	StackOverflow	10000	4-layer transformer	$6M$	1000	4964	15520
Landmark recognition	GLDv2	2028	ResNet-18	$12M$	823	88	1000
Character recognition	EMNIST	63	ResNet-18	$11M$	1114	298	418
Speech recognition	LibriSpeech	N/A	6-layer transformer	$15M$	902	8.3 min	15 min

We give fully non-asymptotic versions of these theorems under more general assumptions in Appendix A. The $O(1/T)$ term is lower order and can be ignored for $T \geq \Omega((n/m)^2)$ for FedAlt and $T \geq \Omega(n/m)$ for FedSim.

4. Experiments

We experimentally compare different model personalization schemes using FedAlt and FedSim. Further details about the experiments and hyperparameters as well as additional experimental results are provided in the appendices. The code to reproduce the experimental results is publicly available.[1]

Datasets, Tasks and Models. We consider four learning tasks, summarized in Table 1.

(a) *Next-Word Prediction*: We use the StackOverflow dataset, where each device corresponds to the questions and answers of one user on `stackoverflow.com`. This is representative of mobile keyboard predictions. We use a 4-layer transformer model (Vaswani et al., 2017) trained with the cross entropy loss and evaluated with top-1 accuracy of next word prediction.

(b) *Landmark Recognition*: We use GLDv2 (Weyand et al., 2020), a large-scale image dataset of global landmarks. Each device corresponds to a Wikipedia contributor and is representative of smartphone users capturing images while traveling. We use ResNet-18 (He et al., 2016). with group norm instead of batch norm (Hsieh et al., 2020) and images are reshaped to 224 × 224. It is trained with the cross entropy loss and evaluated with the classification accuracy.

(c) *Character Recognition*: We use the EMNIST dataset (Cohen et al., 2017), where the input is a 28 × 28 grayscale image of a handwritten character and the output is its label (0-9, a-z, A-Z). Each device corresponds to a writer of the character. We use a ResNet-18 model with input and output layers modified to accommodate the smaller image size and number of classes.

(d) *Speech Recognition (ASR)*: We construct a federated version of the LibriSpeech dataset (Panayotov et al., 2015), partitioned by the speaker of the audio. The input is an audio clip of English speech represented by log-mel filterbank coefficients and the output is its text transcription. We use a 6-layer transformer model trained with the connectionist temporal classification (CTC) criterion (Graves et al., 2006) and report the word error rate for evaluation.

Model Partitioning for Partial Personalization. We consider three partitioning schemes.

(a) *Input layer personalization*: This architecture personalizes the input layer to learn personal representations, while the rest of the model is shared (Figure 1b). For next-word prediction, we personalize the first transformer layer instead of the embedding layer.

(b) *Output layer personalization*: This architecture learns a shared representation but personalizes the prediction layer (Figure 1a). We personalize the last transformer layer for a transformer model instead of the output layer.

(c) *Adapter personalization*: Each device adds personal adapter modules to a shared model (Figure 2a). We use the transformer adapters of Houlsby et al. (2019) and the residual adapters of Rebuffi et al. (2017).

Algorithms and Experimental Pipeline. We consider three full personalization baselines: (i) *Finetune*, where each device finetunes its personal full model starting from a learned common model, (ii) *Ditto* (Li et al., 2021), which is finetuning with ℓ_2 regularization, and, (iii) *pFedMe* (Dinh et al., 2020) which minimizes the objective (2). All methods, including FedAlt, FedSim and the baselines are initialized with a global model trained with FedAvg.

4.1. Experimental Results

Partial personalization nearly matches full personalization and can sometimes outperform it. Table 2 shows the *average* test accuracy across all devices of different FL algorithms. We see that on the StackOverflow dataset, output layer personalization (25.05%) makes up nearly 90% of the gap between the non-personalized baseline (23.82%) and full personalization (25.21%). On EMNIST, adapter personalization exactly matches full personalization. Most surprisingly, on GLDv2, adapter personalization outperforms full personalization by 3.5pp (percentage points).

[1] `https://github.com/krishnap25/FL_partial_personalization`

Table 2. Comparison of partial model personalization with full model personalization in terms of the *average* test accuracy % across devices. The subscript denotes the standard deviation over 5 random runs. The boldfaced/highlighted numbers denote entries within one standard deviation of the maximum in each row. For partial personalization, we show the accuracy of FedAlt; see Table 4 for FedSim.

	Non-pers.	Full Model Personalization			Partial Model Personalization		
	FedAvg	Finetune	Ditto	pFedMe	Input Layer	Output Layer	Adapter
StackOverflow	23.82	$\mathbf{25.20}_{0.01}$	$\mathbf{25.20}_{0.01}$	$\mathbf{25.21}_{0.01}$	$24.44_{0.01}$	$25.05_{0.01}$	$24.82_{0.01}$
GLDv2	51.43	$62.85_{0.02}$	$62.85_{0.01}$	$62.92_{0.02}$	$53.94_{0.07}$	$56.64_{0.05}$	$\mathbf{66.41}_{0.06}$
EMNIST	93.18	$\mathbf{94.13}_{0.01}$	$\mathbf{94.13}_{0.01}$	$\mathbf{94.13}_{0.01}$	$93.62_{0.04}$	$93.57_{0.05}$	$\mathbf{94.13}_{0.03}$

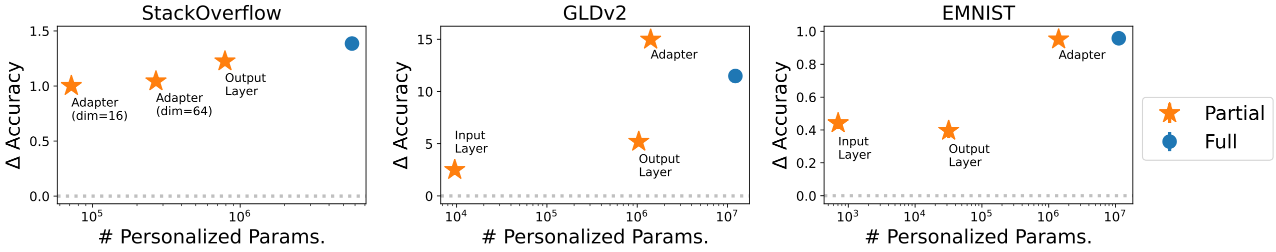


Figure 4. Absolute change in accuracy (percentage points) due to personalization plotted against number of personal parameters (i.e., dimensionality of v_i). Note that the x-axis is in log scale.

This success of adapter personalization can be explained partly by the nature of GLDv2. On average, the training data on each device contains 25 classes out of a possible 2028 while the testing data contains 10 classes not seen in its own training data. These unseen classes account for nearly 23% of all testing data. Personalizing the full model is susceptible to "forgetting" the original task (Kirkpatrick et al., 2017), making it harder to get these unseen classes right. Such *catastrophic forgetting* is worse when finetuning on a very small local dataset, as we often have in FL. On the other hand, personalizing the adapters does not suffer as much from this issue (Rebuffi et al., 2017).

Partial personalization only requires a fraction of the parameters to be personalized. Figure 4 shows that the number of personal parameters required to compete with full personalization is rather small. On StackOverflow, personalizing 1.2% of the parameters with adapters captures 72% of the accuracy boost from personalizing all $5.7M$ parameters; this can be improved to nearly 90% by personalizing 14% of the parameters (output layer). Likewise, we match full personalization on EMNIST and exceed it on GLDv2 with adapters, personalizing 11.5-12.5% of parameters.

The best personalized architecture is model and task dependent. Table 2 shows that personalizing the final transformer layer (denoted as "Output Layer") achieves the best performance for StackOverflow, while the residual adapter achieves the best performance for GLDv2 and EMNIST. In contrast, input layer personalization achieves the best performance for speech recognition, cf. Table 3.

Table 3. Comparison of finetuning and partial personalization for ASR on Librispeech. We report the word error rate (WER, %) on the test data, averaged across devices. Smaller values are better.

Finetune	Input Layer	Output Layer	Adapter
15.55	**15.13**	15.53	15.50

This variation is explained via the primary source of data heterogeneity across devices for each task. The choice of the next word after a context can vary between users, so the output layer is the right component to personalize for this task. Likewise, there is greater heterogeneity in the audio of LibriSpeech (accent, tone, and voice of the speaker) than the text (standard literary English), so input layer personalization works best in this case. This shows that the approach of personalizing a fixed model part, as in past works, is suboptimal. Our framework allows for the use of domain knowledge to determine customized personalization.

Finetuning is competitive with other full personalization methods. Full finetuning matches the performance of pFedMe and Ditto on StackOverflow and EMNIST. On GLDv2, however, pFedMe outperforms finetuning by 0.07pp, but is still 3.5pp worse than adapter personalization.

FedAlt outperforms FedSim by a small but consistent margin. Table 4 shows that FedAlt almost always outper-

Table 4. FedAlt vs. FedSim for partial personalization. "FT (part.)" means finetuning the personal parameters v_i while fixing the shared parameters u from FedAvg. The numbers are averaged over 5 random runs and the subscript denotes the standard deviation.

	StackOverflow			GLDv2			EMNIST		
	FT (part.)	FedAlt	FedSim	FT (part.)	FedAlt	FedSim	FT (part.)	FedAlt	FedSim
Input Layer	$\mathbf{24.96}_{0.01}$	$24.44_{0.01}$	$24.81_{0.01}$	$51.97_{0.02}$	$\mathbf{53.94}_{0.06}$	$53.64_{0.08}$	$93.29_{0.00}$	$\mathbf{93.62}_{0.03}$	$93.55_{0.05}$
Output Layer	$24.93_{0.01}$	$\mathbf{25.05}_{0.01}$	$25.02_{0.01}$	$53.21_{0.01}$	$\mathbf{56.64}_{0.05}$	$56.24_{0.04}$	$93.37_{0.01}$	$\mathbf{93.57}_{0.04}$	$\mathbf{93.55}_{0.05}$
Adapter	$24.71_{0.00}$	$\mathbf{24.82}_{0.01}$	$24.74_{0.01}$	$63.86_{0.06}$	$\mathbf{66.41}_{0.05}$	$66.35_{0.03}$	$93.66_{0.00}$	$\mathbf{94.13}_{0.03}$	$94.07_{0.03}$

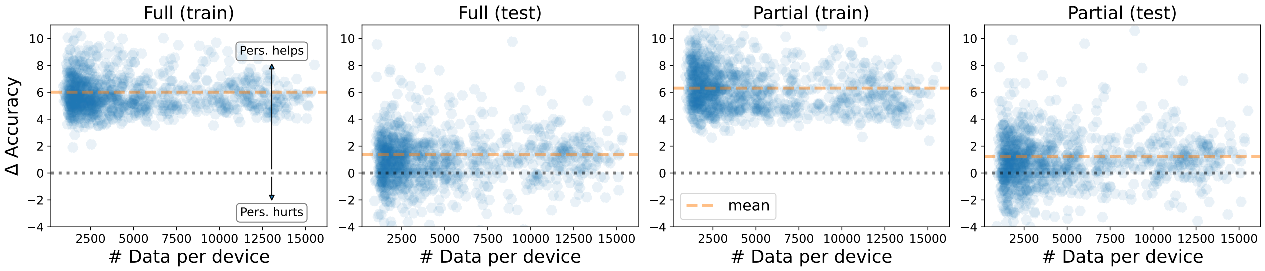


Figure 5. StackOverflow task: Scatter plot of change in training and test accuracy (pp) per-device versus the number of training samples on the device for (a) **Left**: full personalization with finetuning, and, (b) **Right**: partial personalization with the output layer.

forms FedSim by a small margin, e.g., 0.08pp for StackOverflow/Adapter and 0.3pp for GLDv2/Input Layer. FedSim in turn yields a higher accuracy than simply finetuning the personal part of the model by a margin of 0.12pp for StackOverflow/Output Layer and 2.55pp for GLDv2/Adapter. Furthermore, we observe that the difference between FedAlt and FedSim is much larger than the standard deviation across runs. For instance, under output layer personalization for GLDv2, this difference is 0.4pp ($= 8\times$ std).

As a practical recommendation, *we recommend using FedAlt as a default, but it does not hurt much to use FedSim.*

4.2. Effects of Personalization on Generalization

Personalization hurts the test accuracy on some devices. Figure 5 shows the change in training and test accuracy of each device, over a non-personalized model baseline. We see that personalization leads to an improvement in training accuracy across all devices, but *a reduction in test accuracy* on some of the devices. Devices whose testing performance is hurt by personalization are mostly on the left side of the plot, meaning that they have relatively small number of training samples. On the other hand, many devices with the most improved test accuracy also appear on the left side, signaling the benefit of personalization. Therefore, there is a large variation of results for devices with few samples.

Additional results in Appendix C show that using ℓ_2 regularization as in (2), or weight decay does not mitigate this issue. Increasing regularization strength (less personalization) can reduce the spread of per-device accuracy, but degrades the average accuracy. Dropout does not fix this issue either.

An ideal personalized method would boost performance on most of the devices without causing a reduction in (test) accuracy on any device. Realizing this goal calls for a sound statistical analysis for personalized FL and may require sophisticated methods for local performance diagnosis and structured regularization.

5. Discussion

In addition to a much smaller memory footprint than full model personalization and being less susceptible to catastrophic forgetting, partial model personalization has other advantages. For example, it reduces the amount of communication between the server and the devices because only the shared parameters are transmitted. While the communication savings may not be significant (especially when the personal parameters are only a small fraction of the full model), communicating only the shared parameters may have significant implications for privacy. Intuitively, it can be harder to infer private information from partial model information. This is especially the case if the more sensitive features of the data are processed through personal components of the model that are kept local at the devices. For example, we speculate that less noise needs to be added to the communicated parameters in order to satisfy differential privacy requirements (Abadi et al., 2016).

References

Abadi, M., Chu, A., Goodfellow, I. J., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep Learning with Differential Privacy. In *Proc. of ACM SIGSAC*, pp. 308–318. ACM, 2016.

Agarwal, A., Langford, J., and Wei, C. Federated Residual Learning. *arXiv Preprint*, 2020.

Arivazhagan, M. G., Aggarwal, V., Singh, A. K., and Choudhary, S. Federated Learning with Personalization Layers. *arXiv Preprint*, 2019.

Baxter, J. A Model of Inductive Bias Learning. *J. Artif. Intell. Res.*, 12:149–198, 2000.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization Methods for Large-Scale Machine Learning. *SIAM Review*, 60 (2):223–311, 2018.

Bui, D., Malik, K., Goetz, J., Liu, H., Moon, S., Kumar, A., and Shin, K. G. Federated User Representation Learning. *arXiv Preprint*, 2019.

Caruana, R. Multitask learning. *Mach. Learn.*, 28(1):41–75, 1997.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. EMNIST: an extension of MNIST to handwritten letters. *arXiv Preprint*, 2017.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. Exploiting Shared Representations for Personalized Federated Learning. In *Proc. of ICML*, volume 139, pp. 2089–2099, 2021.

Collobert, R. and Weston, J. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *ICML*, volume 307, pp. 160–167, 2008.

Demmel, J. W. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. ImageNet: A large-scale hierarchical image database. In *Proc. of CVPR*, pp. 248–255, 2009.

Deng, Y., Kamani, M. M., and Mahdavi, M. Adaptive Personalized Federated Learning. *arXiv Preprint*, 2020a.

Deng, Y., Kamani, M. M., and Mahdavi, M. Distributionally Robust Federated Averaging. In *NeurIPS*, 2020b.

Dinh, C. T., Tran, N., and Nguyen, J. Personalized Federated Learning with Moreau Envelopes. In *Proc. of NeurIPS*, volume 33, pp. 21394–21405, 2020.

Evgeniou, T. and Pontil, M. Regularized Multi–Task Learning. In *KDD*, pp. 109–117, 2004.

Fallah, A., Mokhtari, A., and Ozdaglar, A. E. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. In *Proc. of NeurIPS*, 2020.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *ICML*, pp. 369–376, 2006.

Hanzely, F., Zhao, B., and Kolar, M. Personalized Federated Learning: A Unified Framework and Universal Optimization Techniques. *arXiv Preprint*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *CVPR*, pp. 770–778, 2016.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-Efficient Transfer Learning for NLP. In *Proc. of ICML*, volume 97, pp. 2790–2799, 2019.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. B. The Non-IID Data Quagmire of Decentralized Machine Learning. In *Proc. of ICML*, volume 119, pp. 4387–4398. PMLR, 2020.

Hsu, T. H., Qi, H., and Brown, M. Federated Visual Classification with Real-World Data Distribution. In *Proc. of ECCV*, volume 12355, pp. 76–92, 2020.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K. A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proc. of ICML*, 2020.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. A Unified Theory of Decentralized SGD with Changing Topology and Local Updates. In *Proc. of ICML*, 2020.

Li, T., Hu, S., Beirami, A., and Smith, V. Ditto: Fair and Robust Federated Learning Through Personalization. In *Proc. of ICML*, volume 139, pp. 6357–6368, 2021.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the Convergence of FedAvg on Non-IID Data. In *ICLR*, 2020.

Liang, P. P., Liu, T., Liu, Z., Salakhutdinov, R., and Morency, L. Think Locally, Act Globally: Federated Learning with Local and Global Representations. In *NeurIPS Workshop on Federated Learning*, 2019.

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. Three Approaches for Personalization with Applications to Federated Learning. *arXiv Preprint*, 2020.

McCloskey, M. and Cohen, N. J. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. volume 24 of *Psychology of Learning and Motivation*, pp. 109–165. Academic Press, 1989.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proc. of AISTATS*, pp. 1273–1282, 2017.

Mikolov, T. and Zweig, G. Context dependent recurrent neural network language model. In *IEEE SLT*, pp. 234–239, 2012.

Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. Cross-stitch Networks for Multi-Task Learning. In *CVPR*, pp. 3994–4003, 2016.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. LibriSpeech: an ASR Corpus based on Public Domain Audio Books. In *ICASSP*, pp. 5206–5210. IEEE, 2015.

Pillutla, K., Laguel, Y., Malick, J., and Harchaoui, Z. Federated Learning with Heterogeneous Data: A Superquantile Optimization Approach. *arXiv Preprint*, 2021.

Rebuffi, S., Bilen, H., and Vedaldi, A. Learning multiple visual domains with residual adapters. In *NeurIPS*, pp. 506–516, 2017.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive Federated Optimization. In *Proc. of ICLR*, 2021.

Singhal, K., Sidahmed, H., Garrett, Z., Wu, S., Rush, K., and Prakash, S. Federated reconstruction: Partially local federated learning. In *Proc. of NeurIPS*, 2021.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated Multi-Task Learning. In *Proc. of NeurIPS*, pp. 4424–4434, 2017.

Synnaeve, G., Xu, Q., Kahn, J., Likhomanenko, T., Grave, E., Pratap, V., Sriram, A., Liptchinsky, V., and Collobert, R. End-to-end ASR: from Supervised to Semi-Supervised Learning with Modern Architectures. *arXiv preprint*, 2019.

TensorFlow Federated. `https://www.tensorflow.org/federated`.

Turc, I., Chang, M.-W., Lee, K., and Toutanova, K. Well-read students learn better: On the importance of pre-training compact models. *arXiv Preprint*, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is All you Need. In *Proc. of NeurIPS*, pp. 5998–6008, 2017.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., et al. A Field Guide to Federated Optimization. *arXiv Preprint*, 2021.

Weyand, T., Araujo, A., Cao, B., and Sim, J. Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *Proc. of CVPR*, pp. 2572–2581, 2020.

Yuan, K., Ling, Q., and Yin, W. On the Convergence of Decentralized Gradient Descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Appendix

Outline

A. Convergence Analysis: Full Proofs

We give the full convergence proofs here. The outline of this section is:

- §A.1: Review of setup and assumptions;
- §A.2: Virtual Full Participation: Background and Details
- §A.3: Convergence analysis of FedAlt and the full proof of Theorem 1 (see Theorem 3 and Corollary 4);
- §A.4: Convergence analysis of FedSim and the full proof of Theorem 2 (see Theorem 11 and Corollary 12);
- §A.5: Technical lemmas used in the analysis.

A.1. Review of Setup and Assumptions

We consider a federated learning system with n devices. Let the loss function on device i be $F_i(u, v_i)$, where $u \in \mathbb{R}^{d_0}$ denotes the shared parameters across all devices and $v_i \in \mathbb{R}^{d_i}$ denotes the personal parameters at device i. We aim to minimize the function

$$F(u, V) := \frac{1}{n}\sum_{i=1}^{n} F_i(u, v_i)\,, \tag{8}$$

where $V = (v_1, \cdots, v_n)$ is a concatenation of all the personalized parameters. This is a special case of (3) with the equal per-device weights, i.e., $\alpha_i = 1/n$. Recall that we assume that F is bounded from below by $F^\star$.

For convenience, we reiterate Assumptions 1, 2 and 3 from the main paper as Assumptions 1′, 2′ and 3′ below respectively, with some additional comments and discussion.

Assumption 1′ (Smoothness)**.** *For each device $i = 1, \ldots, n$, the objective F_i is smooth, i.e., it is continuously differentiable and,*

(a) $u \mapsto \nabla_u F_i(u, v_i)$ is L_u-Lipschitz for all v_i,
(b) $v_i \mapsto \nabla_v F_i(u, v_i)$ is L_v-Lipschitz for all u,
(c) $v_i \mapsto \nabla_u F_i(u, v_i)$ is L_{uv}-Lipschitz for all u, and,
(d) $u \mapsto \nabla_v F_i(u, v_i)$ is L_{vu}-Lipschitz for all v_i.

Further, we assume for some $\chi > 0$ that

$$\max\{L_{uv}, L_{vu}\} \le \chi\sqrt{L_u L_v}\,.$$

The smoothness assumption is a standard one. We can assume without loss of generality that the cross-Lipschitz coefficients L_{uv}, L_{vu} are equal. Indeed, if F_i is twice continuously differentiable, we can show that L_{uv}, L_{vu} are both equal to the operator norm $\|\nabla^2_{uv} F_i(u, v_i)\|_{\mathrm{op}}$ of the mixed second derivative matrix. Further, χ denotes the extent to which u impacts the gradient of v_i and vice-versa.

For concreteness, consider the full personalization setting of Eq. (2), where each F_i is L-smooth; this is a special case of the formulation (8), as we argue in §2. In this case, a simple calculation shows that

$$\chi^2 = \frac{\lambda}{\lambda + L} \le 1\,.$$

Our next assumption is about the variance of the stochastic gradients, and is standard in literature. Compared to the main paper, we adopt a more precise notation about stochastic gradients.

Assumption 2′ (Bounded Variance)**.** *Let $\mathcal{D}_i$ denote a probability distribution over the data space $\mathcal{Z}$ on device i. There exist functions $G_{i,u}$ and $G_{i,v}$ which are unbiased estimates of $\nabla_u F_i$ and $\nabla_v F_i$ respectively. That is, for all u, v_i:*

$$\mathbf{E}_{z\sim\mathcal{D}_i}\left[G_{i,u}(u, v, z)\right] = \nabla_u F_i(u, v_i), \quad \textit{and} \quad \mathbf{E}_{z\sim\mathcal{D}_i}\left[G_{i,v}(u, v, z)\right] = \nabla_v F_i(u, v_i)\,.$$

Furthermore, the variance of these estimators is at most σ_u^2 and σ_v^2 respectively. That is,

$$\mathbf{E}_{z\sim\mathcal{D}_i}\|G_{i,u}(u, v, z) - \nabla_u F_i(u, v_i)\|^2 \le \sigma_u^2\,,$$
$$\mathbf{E}_{z\sim\mathcal{D}_i}\|G_{i,v}(u, v, z) - \nabla_v F_i(u, v_i)\|^2 \le \sigma_v^2\,.$$

In practice, one usually has $G_{i,u}(u, v_i, z) = \nabla_u f_i((u, v_i), z)$, which is the gradient of the loss on datapoint $z \sim \mathcal{D}_i$ under the model (u, v_i), and similarly for $G_{i,v}$.

Finally, we make a gradient diversity assumption.

Assumption 3′ (Partial Gradient Diversity)**.** *There exist $\delta \geq 0$ and $\rho \geq 0$ such that for all u and V,*

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla_u F_i(u, v_i) - \nabla_u F(u, V)\|^2 \leq \delta^2 + \rho^2\|\nabla_u F(u, V)\|^2 \,. \tag{9}$$

This is a generalization of Assumption 3′ used in the main paper, which is a special case of Assumption 3 with $\rho = 0$. We allow the partial gradient diversity to grow with the squared norm of the gradient with a factor of ρ^2. This assumption is analogous to the bounded variance assumption (Assumption 2′), but with the stochasticity coming from the sampling of devices. It characterizes how much local steps on one device help or hurt convergence globally.

Similar gradient diversity assumptions are often used for analyzing non-personalized federated learning (Koloskova et al., 2020; Karimireddy et al., 2020). Finally, it suffices for the partial gradient diversity assumption to only hold at the iterates $(u^{(t)}, V^{(t)})$ generated by either FedSim or FedAlt.

A.2. Virtual Full Participation: Background and Details

We recap the challenge of dependent random variables with FedAlt, and explain the technique of virtual full participation in some more detail. For this section, we assume full gradients on each device ($\sigma_u^2 = 0 = \sigma_v^2$) and a single local update per device ($\tau_u = 1 = \tau_v$). The only stochasticity in the algorithm comes from partial device participation, i.e., sampling m devices in each round.

Background: Stochastic Gradient Convergence Analysis. Consider the minimization problem

$$\min_{w \in \mathbb{R}^d} f(w) \,,$$

where the function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth. Starting from some fixed $w^{(0)} \in \mathbb{R}^d$, consider the stochastic gradient iterations $w^{(t+1)} = w^{(t)} - \gamma g^{(t)}$, where γ is a fixed learning rate, and $g^{(t)}$ is an unbiased estimate of $\nabla f(w^{(t)})$, i.e., $\mathbf{E}[g^{(t)}|w^{(t)}] = \nabla f(w^{(t)})$.

Typical proofs of convergence proceed in the general nonconvex case with the smoothness bound

$$\begin{aligned} f(w^{(t+1)}) - f(w^{(t)}) &\leq \langle \nabla f(w^{(t)}), w^{(t+1)} - w^{(t)} \rangle + \frac{L}{2}\|w^{(t+1)} - w^{(t)}\|^2 \\ &= -\gamma\langle \nabla f(w^{(t)}), g^{(t)} \rangle + \frac{\gamma^2 L}{2}\|g^{(t)}\|^2 \,. \end{aligned} \tag{10}$$

Since the stochastic gradient $g^{(t)}$ is *unbiased*, we get (under typical assumptions) an inequality

$$\mathbf{E}_t\left[f(w^{(t+1)})\right] - f(w^{(t)}) \leq -c\gamma\,\|\nabla f(w^{(t)})\|^2 + O(\gamma^2) \,, \tag{11}$$

where $c > 0$ is some absolute constant and $\mathbf{E}_t[\cdot] = \mathbf{E}[\,\cdot\,|w^{(t)}]$ takes an expectation only over the randomness in step t. The second term is a noise term that can be made small by choosing an appropriately small learning rate γ. Telescoping the inequality over t and rearranging gives a convergence bound.

The **key intuition** behind this proof is that the update is unbiased in linear term of the smoothness upper bound (10). The same intuition holds for most smooth nonconvex stochastic gradient convergence analyses (Bottou et al., 2018). In particular, this takes the following form in this case

$$\mathbf{E}_t\left[\langle \nabla f(w^{(t)}), w^{(t+1)} - w^{(t)} \rangle\right] = \left\langle \nabla f(w^{(t)}), \mathbf{E}_t[w^{(t+1)} - w^{(t)}] \right\rangle . \tag{12}$$

This ensures that the contribution of the stochasticity occurs in a lower order $O(\gamma^2)$ term. As we shall see next, such an equality does not hold for FedAlt in the partial participation case due to dependent random variables.

The Challenge in FedAlt with Partial Participation. Consider the iterates $(u^{(t)}, V^{(t)})$ generated by FedAlt. The progress in one round is the combined progress of the v-step (call it $\mathcal{T}_v$) and the u-step (call it $\mathcal{T}_u$) so that

$$F\left(u^{(t+1)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t)}\right) = \underbrace{F\left(u^{(t)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t)}\right)}_{=:\mathcal{T}_v} + \underbrace{F\left(u^{(t+1)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t+1)}\right)}_{=:\mathcal{T}_u}.$$

The analysis of the v-step is easy because the unbiasedness condition similar to (12) holds:

$$\mathbf{E}_t\left\langle \nabla_V F\left(u^{(t)}, V^{(t)}\right), V^{(t+1)} - V^{(t)}\right\rangle = \left\langle \nabla_V F\left(u^{(t)}, V^{(t)}\right), \mathbf{E}_t\left[V^{(t+1)} - V^{(t)}\right]\right\rangle,$$

since $\mathbf{E}_t[\cdot]$ takes an expectation w.r.t. the client sampling $S^{(t)}$. The recipe laid out earlier gives a descent condition similar to (11).

For the u-step, an unbiasedness condition similar to (12) does not hold:

$$\mathbf{E}_t\left\langle \nabla_u F\left(u^{(t)}, V^{(t+1)}\right), u^{(t+1)} - u^{(t)}\right\rangle \neq \left\langle \mathbf{E}_t\left[\nabla_u F\left(u^{(t)}, V^{(t+1)}\right)\right], \mathbf{E}_t\left[u^{(t+1)} - u^{(t)}\right]\right\rangle.$$

The expectation cannot pass into the inner product because $V^{(t+1)}$ and $u^{(t+1)}$ are dependent random variables. Both are dependent on the device sampling $S^{(t)}$, as shown Figure 3 (left).

Virtual Full Participation. We decouple these random variables by using virtual full participation. Define a virtual iterate $\widetilde{V}^{(t+1)}$ as the result of local v-updates as if *every* device had participated. Specifically, we introduce $\widetilde{V}^{(t+1)}$ on the right hand side of the smoothness bound applied on $\mathcal{T}_u$ to get

$$F\big(u^{(t+1)}, V^{(t+1)}\big) - F\big(u^{(t)}, V^{(t+1)}\big) \le E^{(t)} + \big\langle \nabla_u F(u^{(t)}, \widetilde{V}^{(t+1)}),\, u^{(t+1)} - u^{(t)}\big\rangle + \frac{L_u}{2}\big\|u^{(t+1)} - u^{(t)}\big\|^2,$$

where $E^{(t)}$ is the error term from replacing $V^{(t+1)}$ with $\widetilde{V}^{(t+1)}$ Since $\widetilde{V}^{(t+1)}$ is independent of the client sampling $S^{(t)}$, we can now take an expectation $\mathbf{E}_t[\cdot]$ over $u^{(t+1)}$ only, leading us to a situation similar to (12); cf. Figure 3 (right).

We bound the error term $E^{(t)}$ using Young's inequality and smoothness (Assumption 1′) respectively as

$$\begin{aligned} E^{(t)} &= \langle \nabla_u F(u^{(t)}, V^{(t+1)}) - \nabla_u F(u^{(t)}, \widetilde{V}^{(t+1)}), u^{(t+1)} - u^{(t)}\rangle \\ &\le \frac{L_u}{2}\|u^{(t+1)} - u^{(t)}\|^2 + \frac{1}{2L_u}\|\nabla_u F(u^{(t)}, V^{(t+1)}) - \nabla_u F(u^{(t)}, \widetilde{V}^{(t+1)})\|^2 \\ &\le \frac{L_u}{2}\|u^{(t+1)} - u^{(t)}\|^2 + \frac{\chi^2 L_v}{2n}\sum_{i=1}^n \|\tilde{v}_i^{(t+1)} - v_i^{(t+1)}\|^2. \end{aligned}$$

These two terms are similar to the quadratic terms we get from the smoothness upper bound. We can similarly show $\mathbf{E}_t[E^{(t)}] = O(L_u\gamma_u^2 + \chi^2 L_v \gamma_v^2)$, so the error term from virtual full participation is also a lower order $O(\gamma^2)$ term.

Virutal Iterates in Related Work. Virtual or shadow iterates have long been used in decentralized optimization (Yuan et al., 2016), and have since been adopted in the analysis of federated optimization algorithms in the non-personalized setting (Li et al., 2020; Koloskova et al., 2020; Wang et al., 2021).

In our notation, the shadow iterates used in (Koloskova et al., 2020; Wang et al., 2021) take the form

$$\bar{u}_k^{(t)} = \frac{1}{n}\sum_{i=1}^n u_{i,k}^{(t)},$$

which is an average of the local versions of the shared parameters. This only makes sense for the case of full participation since $u_{i,k}^{(t)}$ is only defined for selected devices $i \in S^{(t)}$. In partial participation case, Li et al. (2020) define the virtual sequence $(\tilde{u}_{i,k}^{(t)})_{k=0}^{\tau_u}$ as the local SGD updates on all devices i irrespective of whether they were selected. Then, they define the average

$$\bar{u}_k^{(t)} = \frac{1}{n}\sum_{i=1}^n \tilde{u}_{i,k}^{(t)}.$$

Their proof relies on the fact that $\mathbf{E}_{S^{(t)}}[u^{(t+1)}] = \bar{u}^{(t)}_{\tau_u}$ due to the properties of the sampling.

In contrast, we consider personalized federated learning — the problem of dependent random variables only shows up in the analysis of FedAlt with partial participation, a setting not considered in prior works. We employ virtual *personal* parameters $\tilde{v}^{(t)}_{i,k}$ to overcome this problem. We believe that this technique of decoupling dependent random variables can be of independent interest for (distributed) stochastic optimization, including personalized extensions of nonsmooth federated learning objectives (Deng et al., 2020b; Pillutla et al., 2021) or more general multi-task learning formulations (Misra et al., 2016).

A.3. Convergence Analysis of FedAlt

We give the full form of FedAlt in Algorithms 4 for the general case of unequal α_i's but focus on $\alpha_i = 1/n$ for the analysis. Theorem 1 of the main paper is a simplification of Corollary 4 below, which in turn is proved based on Theorem 3.

Throughout this section, we use the constants

$$\sigma^2_{\mathrm{alt},1} = \frac{\delta^2}{L_u}\left(1 - \frac{m}{n}\right) + \frac{\sigma_u^2}{L_u} + \frac{\sigma_v^2(m + \chi^2(n-m))}{L_v n}, \qquad \sigma^2_{\mathrm{alt},2} = \frac{\sigma_u^2 + \delta^2}{L_u}(1 - \tau_u^{-1}) + \frac{\sigma_v^2 m}{L_v n}(1 - \tau_v^{-1}) + \frac{\chi^2\sigma_v^2}{L_v}.$$

We also recall the definitions

$$\Delta_u^{(t)} = \left\| \nabla_u F\left(u^{(t)}, V^{(t+1)}\right)\right\|^2, \quad \text{and,} \quad \Delta_v^{(t)} = \frac{1}{n}\sum_{i=1}^n \left\| \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2.$$

Theorem 3 (Convergence of FedAlt). *Suppose Assumptions 1′, 2′ and 3′ hold and the learning rates in FedAlt are chosen as $\gamma_u = \eta/(L_u\tau_u)$ and $\gamma_v = \eta/(L_v\tau_v)$, with*

$$\eta \le \min\left\{ \frac{1}{24(1+\rho^2)}, \frac{m}{128\chi^2(n-m)}, \sqrt{\frac{m}{\chi^2 n}} \right\}.$$

Then, ignoring absolute constants, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{1}{L_u}\mathbf{E}\big[\Delta_u^{(t)}\big] + \frac{m}{nL_v}\mathbf{E}\big[\Delta_v^{(t)}\big]\right) \le \frac{\Delta F_0}{\eta T} + \eta\,\sigma^2_{\mathrm{alt},1} + \eta^2\,\sigma^2_{\mathrm{alt},2}.$$

Before proving the theorem, we have the corollary with optimized learning rates.

Corollary 4 (Final Rate of FedAlt). *Consider the setting of Theorem 3 and let the number of rounds T be known in advance. Suppose we set the learning rates $\gamma_u = \eta/(\tau L_u)$ and $\gamma_v = \eta/(\tau L_v)$, where (ignoring absolute constants),*

$$\eta = \left(\frac{\Delta F_0}{T\sigma^2_{\mathrm{alt},1}}\right)^{1/2} \bigwedge \left(\frac{\Delta F_0^2}{T^2\,\sigma^2_{\mathrm{alt},2}}\right)^{1/3} \bigwedge \frac{1}{1+\rho^2} \bigwedge \frac{m}{\chi^2(n-m)} \bigwedge \sqrt{\frac{m}{\chi^2 n}}.$$

We have, ignoring absolute constants,

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{1}{L_u}\mathbf{E}\left\|\nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 + \frac{m}{L_v n^2}\sum_{i=1}^n \mathbf{E}\left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2\right) \le$$
$$\frac{\left(\Delta F_0\,\sigma^2_{\mathrm{alt},1}\right)^{1/2}}{\sqrt{T}} + \frac{\left(\Delta F_0^2\,\sigma^2_{\mathrm{alt},2}\right)^{1/3}}{T^{2/3}} + \frac{\Delta F_0}{T}\left(1 + \rho^2 + \chi^2\left(\frac{n}{m} - 1\right) + \sqrt{\chi^2\frac{n}{m}}\right).$$

Proof. The proof follows from invoking Lemma 25 on the bound of Theorem 3. □

Remark 5 (Asymptotic Rate). *The asymptotic $1/\sqrt{T}$ rate of Theorem 1 is achieved when the $1/T$ term is dominated by the $1/\sqrt{T}$ term. This happens when (ignoring absolute constants)*

$$T \ge \frac{\Delta F_0}{\sigma^2_{\mathrm{alt},1}}\left(1 + \rho^4 + \chi^4\frac{n^2}{m^2}\right).$$

Algorithm 4 FedAlt: Alternating updates of shared and personalized parameters

```
Input: Initial iterates $u^{(0)}, V^{(0)}$, Number of communication rounds $T$, Number of devices per round $m$, Number of local updates $\tau_u, \tau_v$, Local step sizes $\gamma_u, \gamma_v$,
for $t = 0, 1, \cdots, T-1$ do
    Sample $m$ devices from $[n]$ without replacement in $S^{(t)}$
    for each selected device $i \in S^{(t)}$ in parallel do
        Initialize $v_{i,0}^{(t)} = v_i^{(t)}$
        for $k = 0, \cdots, \tau_v - 1$ do
            // Update personal parameters
            Sample data $z_{i,k}^{(t)} \sim \mathcal{D}_i$
            $v_{i,k+1}^{(t)} = v_{i,k}^{(t)} - \gamma_v G_{i,v}(u^{(t)}, v_{i,k}^{(t)}, z_{i,k}^{(t)})$
        Update $v_i^{(t+1)} = v_{k,\tau_v}^{(t)}$
        Initialize $u_{i,0}^{(t)} = u^{(t)}$
        for $k = 0, \cdots, \tau_u - 1$ do
            // Update shared parameters
            $u_{i,k+1}^{(t)} = u_{i,k}^{(t)} - \gamma_u G_{i,u}(u_{i,k}^{(t)}, v_i^{(t+1)}, z_{i,k}^{(t)})$
        Update $u_i^{(t+1)} = u_{i,\tau_u}^{(t)}$
    Update $u^{(t+1)} = \sum_{i \in S^{(t)}} \alpha_i u_i^{(t+1)} / \sum_{i \in S^{(t)}} \alpha_i$ at the server with secure aggregation
return $u^{(T)}, v_1^{(T)}, \cdots, v_n^{(T)}$
```

We now prove Theorem 3.

Proof of Theorem 3. The proof mainly applies the smoothness upper bound to write out a descent condition with suitably small noise terms. We start with some notation.

We introduce the notation $\widetilde{\Delta}_u^{(t)}$ as the analogue of $\Delta_u^{(t)}$ with the virtual variable $\widetilde{V}^{(t+1)}$:

$$\widetilde{\Delta}_u^{(t)} = \left\| \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right) \right\|^2 .$$

Notation. Let $\mathcal{F}^{(t)}$ denote the σ-algebra generated by $\left(u^{(t)}, V^{(t)}\right)$ and denote $\mathbf{E}_t[\,\cdot\,] = \mathbf{E}[\,\cdot\,|\mathcal{F}^{(t)}]$. For all devices, including those not selected in each round, we define virtual sequences $\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}$ as the SGD updates in Algorithm 4 for all devices regardless of whether they are selected. For the selected devices $i \in S^{(t)}$, we have $v_{i,k}^{(t)} = \tilde{v}_{i,k}^{(t)}$ and $u_{i,k}^{(t)} = \tilde{u}_{i,k}^{(t)}$. Note now that the random variables $\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}$ are independent of the device selection $S^{(t)}$. Finally, we have that the updates for the selected devices $i \in S^{(t)}$ are given by

$$v_i^{(t+1)} = v_i^{(t)} - \gamma_v \sum_{k=0}^{\tau_v - 1} G_{i,v}\left(u^{(t)}, \tilde{v}_{i,k}^{(t)}, z_{i,k}^{(t)}\right) ,$$

and the server update is given by

$$u^{(t+1)} = u^{(t)} - \frac{\gamma_u}{m} \sum_{i \in S^{(t)}} \sum_{k=0}^{\tau_u - 1} G_{i,u}\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,\tau_v}^{(t)}, z_{i,k}^{(t)}\right) .$$

Proof Outline and the Challenge of Dependent Random Variables. We start with

$$\begin{aligned} F\left(u^{(t+1)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t)}\right) = & F\left(u^{(t)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t)}\right) \\ & + F\left(u^{(t+1)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t+1)}\right) . \end{aligned} \tag{13}$$

The first line corresponds to the effect of the v-step and the second line to the u-step. The former is easy to handle with standard techniques that rely on the smoothness of $F\left(u^{(t)}, \cdot\right)$. The latter is more challenging. In particular, the smoothness bound for the u-step gives us

$$F\left(u^{(t+1)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t+1)}\right) \le \left\langle \nabla_u F\left(u^{(t)}, V^{(t+1)}\right), u^{(t+1)} - u^{(t)} \right\rangle + \frac{L_u}{2}\left\| u^{(t+1)} - u^{(t)} \right\|^2 .$$

The standard proofs of convergence of stochastic gradient methods rely on the fact that we can take an expectation w.r.t. the sampling $S^{(t)}$ of devices for the first order term. However, both $V^{(t+1)}$ and $u^{(t+1)}$ depend on the sampling $S^{(t)}$ of devices. Therefore, we cannot directly take an expectation with respect to the sampling of devices in $S^{(t)}$.

Virtual Full Participation to Circumvent Dependent Random Variables. The crux of the proof lies in replacing $V^{(t+1)}$ in the analysis of the u-step with the virtual iterate $\widetilde{V}^{(t+1)}$ so as to move all the dependence of the u-step on $S^{(t)}$ to the $u^{(t+1)}$ term. This allows us to take an expectation; it remains to carefully bound the resulting error terms.

Finally, we will arrive at a bound of the form

$$\frac{1}{T}\sum_{t=0}^{T-1}\Big(\frac{\gamma_u \tau_u}{8}\mathbf{E}[\widetilde{\Delta}_u^{(t)}] + \frac{\gamma_v \tau_v m}{16 n}\mathbf{E}[\Delta_v^{(t)}]\Big) \le \frac{\Delta F_0}{T} + O(\gamma_u^2 + \gamma_v^2) \, .$$

Next, we translate this bound from gradient $\mathbf{E}[\widetilde{\Delta}_u^{(t)}]$ of the virtual $\widetilde{V}^{(t+1)}$ to $\mathbf{E}[\Delta_u^{(t)}]$, which is the gradient computed at the actual iterate $V^{(t)}$. A careful analysis shows that we only incur a lower order term of $O(\gamma_u \gamma_v^2)$ in this translation. Choosing γ_u and γ_v small enough will give us the final result.

Analysis of the u-Step with Virtual Full Participation. We introduce the virtual iterates $\widetilde{V}^{(t+1)}$ into the analysis of the u-step as follows:

$$\begin{aligned}
&F\left(u^{(t+1)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t+1)}\right) \\
&\le \left\langle \nabla_u F\left(u^{(t)}, V^{(t+1)}\right), u^{(t+1)} - u^{(t)} \right\rangle + \frac{L_u}{2}\left\| u^{(t+1)} - u^{(t)} \right\|^2 \\
&= \left\langle \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right), u^{(t+1)} - u^{(t)} \right\rangle + \frac{L_u}{2}\left\| u^{(t+1)} - u^{(t)} \right\|^2 \\
&\qquad + \left\langle \nabla_u F\left(u^{(t)}, V^{(t+1)}\right) - \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right), u^{(t+1)} - u^{(t)} \right\rangle \\
&\le \left\langle \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right), u^{(t+1)} - u^{(t)} \right\rangle + L_u\left\| u^{(t+1)} - u^{(t)} \right\|^2 \\
&\qquad + \frac{1}{2L_u}\left\| \nabla_u F\left(u^{(t)}, V^{(t+1)}\right) - \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right) \right\|^2 \\
&\le \underbrace{\left\langle \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right), u^{(t+1)} - u^{(t)} \right\rangle}_{\mathcal{T}_{1,u}} + \underbrace{L_u\left\| u^{(t+1)} - u^{(t)} \right\|^2}_{\mathcal{T}_{2,u}} + \underbrace{\frac{\chi^2 L_v}{2n}\sum_{i=1}^n \left\| \tilde{v}_i^{(t+1)} - v_i^{(t+1)} \right\|^2}_{\mathcal{T}_{3,u}} .
\end{aligned}$$

The last two inequalities follow from Young's inequality and Lipschitzness of $V \mapsto \nabla_u F(u, V)$ respectively.

We have now successfully eliminated the dependence of the first-order term $\mathcal{T}_{1,u}$ on $V^{(t+1)}$. The virtual iterates $\widetilde{V}^{(t+1)}$ are now independent of $S^{(t)}$. This allows us to take an expectation w.r.t. the sampling $S^{(t)}$ of the devices.

We bound each of these terms in Claims 6 to 8 below to get

$$\begin{aligned}
&\mathbf{E}_t\left[F\left(u^{(t+1)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t+1)}\right) \right] \\
&\qquad \le -\frac{\gamma_u \tau_u}{4}\mathbf{E}_t[\widetilde{\Delta}_u^{(t)}] + \underbrace{\frac{2\gamma_u L_u^2}{n}\sum_{i=1}^n \sum_{k=0}^{\tau_u - 1}\mathbf{E}_t\left\| \tilde{u}_{i,k}^{(t)} - u^{(t)} \right\|^2}_{=:\mathcal{T}'_{2,u}} + 4\gamma_v^2 \tau_v^2 L_v \sigma_v^2 \chi^2 (1 - m/n) \\
&\qquad\quad + \frac{L_u \gamma_u^2 \tau_u^2}{m}\left(\sigma_u^2 + 3\delta^2\left(1 - \frac{m}{n}\right)\right) + 8\gamma_v^2 \tau_v^2 L_v \chi^2 (1 - m/n)\Delta_v^{(t)} \, .
\end{aligned}$$

Note that we used the fact that $24 L_u \gamma_u \tau_u (1+\rho^2) \le 1$ to simply the coefficients of some of the terms above. The second term has also been referred to as client drift in the literature; we bound it with Lemma 22 and invoke the assumption on gradient diversity (Assumption 3′) to get

$$
\begin{aligned}
\mathcal{T}'_{2,u} &\le \frac{16\gamma_u^3 L_u^2 \tau_u(\tau_u-1)}{n} \sum_{i=1}^n \mathbf{E}_t \left\| \nabla_u F_i \left(u^{(t)}, \tilde{v}_i^{(t+1)}\right) \right\|^2 + 8\gamma_u^3 L_u^2 \tau_u^2 (\tau_u - 1)\sigma_u^2 \\
&\le \frac{16\gamma_u^3 L_u^2 \tau_u(\tau_u-1)}{n} \left(\delta^2 + \rho^2 \mathbf{E}_t \left\| \nabla_u F \left(u^{(t)}, \widetilde{V}^{(t+1)}\right) \right\|^2 \right) + 8\gamma_u^3 L_u^2 \tau_u^2 (\tau_u - 1)\sigma_u^2 \,.
\end{aligned}
$$

Plugging this back in, we get,

$$
\begin{aligned}
&\mathbf{E}_t \left[F\left(u^{(t+1)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t+1)}\right) \right] \\
&\qquad \le -\frac{\gamma_u \tau_u}{8} \mathbf{E}_t[\widetilde{\Delta}_u^{(t)}] + \frac{L_u \gamma_u^2 \tau_u^2}{m} \left(\sigma_u^2 + 2\delta^2(1 - m/n)\right) + 4\gamma_v^2 \tau_v^2 L_v \sigma_v^2 \chi^2 (1 - m/n) \\
&\qquad\quad + 8\gamma_v^2 \tau_v^2 L_v \chi^2 (1 - m/n) \Delta_v^{(t)} + 8\gamma_u^2 L_u^3 \tau_u^2 (\tau_u - 1)(\sigma_u^2 + 2\delta_u^2) \,.
\end{aligned}
$$

Note that we used $128\gamma_u^2 L_u^2 \tau_u(\tau_u - 1)\rho^2 \le 1$, which is implied by $24 L_u \gamma_u \tau_u (1+\rho^2) \le 1$.

Bound with the Virual Iterates. We plug this analysis of the u-step and Claim 9 for the v-step into (13) next. We also simplify some coefficients using $128 \gamma_v \tau_v L_v \chi^2 (n/m - 1) \le 1$. This gives us

$$
\begin{aligned}
&\mathbf{E}_t \left[F\left(u^{(t+1)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t)}\right) \right] \\
&\le -\frac{\gamma_u \tau_u}{8} \mathbf{E}_t[\widetilde{\Delta}_u^{(t)}] - \frac{\gamma_v \tau_v m}{16 n} \mathbf{E}_t[\Delta_v^{(t)}] + 4\gamma_v^2 L_v \tau_v^2 \sigma_v^2 \left(\frac{m}{n} + \chi^2 (1 - m/n) \right) \\
&\quad + \frac{\gamma_u^2 L_u \tau_u^2}{m} \left(\sigma_u^2 + 2\delta^2 (1 - m/n)\right) + 8\gamma_u^3 L_u^2 \tau_u^2 (\tau_u - 1)(\sigma_u^2 + 2\delta^2) + \frac{4\gamma_v^3 L_v^2 \tau_v^2 (\tau_v - 1)\sigma_v^2 m}{n} \,.
\end{aligned}
$$

Taking an unconditional expectation, summing it over $t = 0$ to $T - 1$ and rearranging this gives

$$
\begin{aligned}
&\frac{1}{T} \sum_{t=0}^{T-1} \left(\frac{\gamma_u \tau_u}{8} \mathbf{E}[\widetilde{\Delta}_u^{(t)}] + \frac{\gamma_v \tau_v m}{16 n} \mathbf{E}[\Delta_v^{(t)}] \right) \qquad (14) \\
&\qquad \le \frac{\Delta F_0}{T} + 4\gamma_v^2 L_v \tau_v^2 \sigma_v^2 \left(\frac{m}{n} + \chi^2 (1 - m/n) \right) + \frac{\gamma_u^2 L_u \tau_u^2}{m} \left(\sigma_u^2 + 2\delta^2 (1 - m/n)\right) \\
&\qquad\qquad + 8\gamma_u^3 L_u^2 \tau_u^2 (\tau_u - 1)(\sigma_u^2 + 2\delta^2) + \frac{4\gamma_v^3 L_v^2 \tau_v^2 (\tau_v - 1)\sigma_v^2 m}{n} \,.
\end{aligned}
$$

This is a bound in terms of the virtual iterates $\widetilde{V}^{(t+1)}$. However, we wish to show a bound in terms of the actual iterate $V^{(t)}$.

Obtaining the Final Bound. It remains now to relate $\widetilde{\Delta}_u^{(t)}$ with $\Delta_u^{(t)}$. Using the Cauchy-Schwartz inequality and smoothness, we have,

$$
\begin{aligned}
&\mathbf{E}_t \left\| \nabla_u F\left(u^{(t)}, V^{(t)}\right) - \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right) \right\|^2 \\
&\qquad \le \frac{1}{n} \sum_{i=1}^n \mathbf{E}_t \left\| \nabla_u F_i \left(u^{(t)}, v_i^{(t)}\right) - \nabla_u F_i \left(u^{(t)}, \tilde{v}_i^{(t+1)}\right) \right\|^2 \\
&\qquad \le \frac{\chi^2 L_u L_v}{n} \sum_{i=1}^n \mathbf{E}_t \left\| \tilde{v}_i^{(t+1)} - v_i^{(t)} \right\|^2 \\
&\qquad \le \frac{\chi^2 L_u L_v}{n} \sum_{i=1}^n \left(16\gamma_v^2 \tau_v^2 \left\| \nabla_v F_i \left(u^{(t)}, v_i^{(t)}\right) \right\|^2 + 8\gamma_v^2 \tau_v^2 \sigma_v^2 \right) \\
&\qquad = 8\gamma_v^2 \tau_v^2 \sigma_v^2 \chi^2 L_u L_v + 16\gamma_v^2 \tau_v^2 \chi^2 L_u L_v \Delta_v^{(t)} \,,
\end{aligned}
$$

where the last inequality followed from Lemma 23. Using

$$\left\|\nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 \le 2\left\|\nabla_u F\left(u^{(t)}, V^{(t)}\right) - \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right)\right\|^2 + 2\left\|\nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right)\right\|^2,$$

we get,

$$\mathbf{E}[\Delta_u^{(t)}] \le 2\,\mathbf{E}[\widetilde{\Delta}_u^{(t)}] + 16\gamma_v^2\tau_v^2\sigma_v^2\chi^2 L_u L_v + 32\gamma_v^2\tau_v^2\chi^2 L_u L_v\,\mathbf{E}[\Delta_v^{(t)}]\,.$$

Therefore, we get,

$$\begin{aligned}
&\frac{\gamma_u\tau_u}{16}\mathbf{E}[\Delta_u^{(t)}] + \frac{\gamma_v\tau_v m}{32n}\mathbf{E}[\Delta_v^{(t)}] \\
&\quad\le \frac{\gamma_u\tau_u}{8}\mathbf{E}[\widetilde{\Delta}_u^{(t)}] + \frac{\gamma_v\tau_v m}{16n}\left(\frac{1}{2} + \frac{32\eta^2\chi^2 m}{n}\right)\mathbf{E}[\Delta_v^{(t)}] + \gamma_u\tau_u\gamma_v^2\tau_v^2\sigma_v^2\chi^2 L_u L_v \\
&\quad\le \frac{\gamma_u\tau_u}{8}\mathbf{E}[\widetilde{\Delta}_u^{(t)}] + \frac{\gamma_v\tau_v m}{16n}\mathbf{E}[\Delta_v^{(t)}] + \gamma_u\tau_u\gamma_v^2\tau_v^2\sigma_v^2\chi^2 L_u L_v\,,
\end{aligned}$$

where we used $\frac{32\eta^2\chi^2 m}{n} \le 1/2$, which is one of the conditions we assume on η.

Summing this up and plugging in (14) gives

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\Big(\frac{\gamma_u\tau_u}{16}\mathbf{E}[\Delta_u^{(t)}] + \frac{\gamma_v\tau_v m}{32n}\mathbf{E}[\Delta_v^{(t)}]\Big) \\
&\quad\le \frac{1}{T}\sum_{t=0}^{T-1}\Big(\frac{\gamma_u\tau_u}{8}\mathbf{E}[\widetilde{\Delta}_u^{(t)}] + \frac{\gamma_v\tau_v m}{16n}\mathbf{E}[\Delta_v^{(t)}]\Big) + \gamma_u\tau_u\gamma_v^2\tau_v^2\sigma_v^2\chi^2 L_u L_v \\
&\quad\le \frac{\Delta F_0}{T} + 4\gamma_v^2 L_v\tau_v^2\sigma_v^2\left(\frac{m}{n} + \chi^2(1-m/n)\right) + \frac{\gamma_u^2 L_u\tau_u^2}{m}\left(\sigma_u^2 + 2\delta^2(1-m/n)\right) \\
&\qquad\quad + 8\gamma_u^3 L_u^2\tau_u^2(\tau_u-1)(\sigma_u^2+2\delta^2) + \frac{4\gamma_v^3 L_v^2\tau_v^2(\tau_v-1)\sigma_v^2 m}{n} + \gamma_u\tau_u\gamma_v^2\tau_v^2\sigma_v^2\chi^2 L_u L_v\,.
\end{aligned}$$

Plugging in $\gamma_u = \eta/(L_u\tau_u)$ and $\gamma_v = \eta/(L_v\tau_v)$ completes the proof. □

The analysis of each of the terms in the u-step is given in the following claims.

Claim 6 (Bounding $\mathcal{T}_{1,u}$)**.** *We have,*

$$\mathbf{E}_t\left[\mathcal{T}_{1,u}\right] \le -\frac{\gamma_u\tau_u}{2}\mathbf{E}_t\left\|\nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right)\right\|^2 + \frac{\gamma_u L_u^2}{n}\sum_{i=1}^{n}\sum_{k=0}^{\tau_u-1}\mathbf{E}_t\left\|\tilde{u}_{i,k}^{(t)} - u^{(t)}\right\|^2.$$

Proof. For $i \in S^{(t)}$, we have that $\tilde{u}_{i,k}^{(t)} = u_{i,k}^{(t)}$. Therefore, we have,

$$\mathbf{E}_t[\mathcal{T}_{1,u}] = -\gamma_u\mathbf{E}_t\left\langle \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right), \frac{1}{m}\sum_{i\in S^{(t)}}\sum_{k=0}^{\tau_u-1}\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_i^{(t+1)}\right)\right\rangle.$$

Using that $\tilde{u}_{i,k}^{(t)}$ is independent of $S^{(t)}$, we get,

$$\begin{aligned}
\mathbf{E}_t[\mathcal{T}_{1,u}] &= -\gamma_u\mathbf{E}_t\left\langle \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right), \frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{\tau_u-1}\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_i^{(t+1)}\right)\right\rangle \\
&= -\gamma_u\tau_u\mathbf{E}_t\left\|\nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right)\right\|^2 \\
&\quad - \gamma_u\sum_{k=0}^{\tau_u-1}\mathbf{E}_t\left\langle \nabla_u F\left(u^{(t)}, \widetilde{V}^{(t+1)}\right), \frac{1}{n}\sum_{i=1}^{n}\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}^{(t+1)}\right) - \nabla_u F_i\left(u^{(t)}, \tilde{v}^{(t+1)}\right)\right\rangle
\end{aligned}$$

Invoking $\langle x, y\rangle \le \|x\|^2/2 + \|y\|^2/2$ for vectors x, y followed by smoothness completes the proof. □

Claim 7 (Bounding $\mathcal{T}_{2,u}$). *We have,*

$$\mathbf{E}_t\left[\mathcal{T}_{2,u}\right] \le 3L_u\gamma_u^2\tau_u^2\left(1+\frac{2\rho^2}{m}(1-m/n)\right)\mathbf{E}_t\left\|\nabla_u F\left(u^{(t)},\widetilde{V}^{(t+1)}\right)\right\|^2 + \frac{3L_u^2\gamma_u^2\tau_u}{n}\sum_{i=1}^n\sum_{k=0}^{\tau_u-1}\mathbf{E}_t\left\|\tilde{u}_{i,k}^{(t)}-u^{(t)}\right\|^2 + \frac{6L_u\gamma_u^2\tau_u^2\delta^2}{m}(1-m/n)\,.$$

Proof. We use $\mathbf{E}\|z\|^2=\|\mathbf{E}[z]\|^2+\mathbf{E}\|z-\mathbf{E}[z]\|^2$ for a random vector z to get

$$\mathbf{E}_t[\mathcal{T}_{2,u}] \le \frac{L_u\gamma_u^2\tau_u^2\sigma_u^2}{m} + L_u\gamma_u^2\tau_u\sum_{k=0}^{\tau_u-1}\mathbf{E}_t\underbrace{\left\|\frac{1}{m}\sum_{i\in S^{(t)}}\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)},\tilde{v}_i^{(t+1)}\right)\right\|^2}_{=:\mathcal{T}_k'}\,.$$

We break the term $\mathcal{T}_k'$ as

$$\mathcal{T}_k' \le 3\left\|\frac{1}{m}\sum_{i\in S^{(t)}}\left(\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)},\tilde{v}_i^{(t+1)}\right)-\nabla_u F_i\left(u^{(t)},\tilde{v}_i^{(t+1)}\right)\right)\right\|^2 + 3\left\|\frac{1}{m}\sum_{i\in S^{(t)}}\nabla_u F_i\left(u^{(t)},\tilde{v}_i^{(t+1)}\right)-\nabla_u F\left(u^{(t)},\widetilde{V}^{(t+1)}\right)\right\|^2 + 3\left\|\nabla_u F\left(u^{(t)},\widetilde{V}^{(t+1)}\right)\right\|^2.$$

For the first term, we use Jensen's inequality to take the squared norm inside the sum, then use smoothness and take an expectation over the sampling of devices to get

$$\mathbf{E}_t\left\|\frac{1}{m}\sum_{i\in S^{(t)}}\left(\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)},\tilde{v}_i^{(t+1)}\right)-\nabla_u F_i\left(u^{(t)},\tilde{v}_i^{(t+1)}\right)\right)\right\|^2 \le \frac{L_u^2}{n}\sum_{i=1}^n\mathbf{E}_t\left\|\tilde{u}_{i,k}^{(t)}-u^{(t)}\right\|^2.$$

For the second term, we use the fact that $S^{(t)}$ was sampled without replacement (cf. Lemma 21) and invoke the gradient diversity assumption (Assumption 3′) to get,

$$\begin{aligned}&\left\|\frac{1}{m}\sum_{i\in S^{(t)}}\nabla_u F_i\left(u^{(t)},\tilde{v}_i^{(t+1)}\right)-\nabla_u F\left(u^{(t)},\widetilde{V}^{(t+1)}\right)\right\|^2\\ &\qquad\le\left(\frac{n-m}{n-1}\right)\frac{1}{mn}\sum_{i=1}^n\left\|\nabla_u F_i\left(u^{(t)},\tilde{v}_i^{(t+1)}\right)-\nabla_u F\left(u,\widetilde{V}^{(t+1)}\right)\right\|^2\\ &\qquad\le\frac{2}{m}\left(1-\frac{m}{n}\right)\left(\delta^2+\rho^2\mathbf{E}_t\left\|\nabla_u F\left(u^{(t)},\widetilde{V}^{(t+1)}\right)\right\|^2\right)\,.\end{aligned}$$

To complete the proof, we plug these terms back into the definition of $\mathcal{T}_k'$ and $\mathbf{E}_t[\mathcal{T}_{2,u}]$ to complete the proof. □

Claim 8 (Bounding $\mathcal{T}_{3,u}$). *We have,*

$$\mathbf{E}_t\left[\mathcal{T}_{3,u}\right] \le 8\gamma_v^2\tau_v^2L_v\chi^2\left(1-\frac{m}{n}\right)\Delta_v^{(t)} + 4\chi^2\gamma_v^2\tau_v^2L_v\sigma_v^2\left(1-\frac{m}{n}\right)\,.$$

Proof. Since $v_i^{(t+1)}=\tilde{v}_i^{(t+1)}$ for $i\in S^{(t)}$, we have that

$$\mathcal{T}_{3,u}=\frac{\chi^2L_v}{2n}\sum_{i\notin S^{(t)}}\left\|\tilde{v}_i^{(t+1)}-v_i^{(t)}\right\|^2.$$

Since $\left\|\tilde{v}_i^{(t+1)} - v_i^{(t)}\right\|^2$ is independent of $S^{(t)}$, we can take an expectation to get

$$\begin{aligned}
\mathbf{E}_t[\mathcal{T}_{3,u}] &= \frac{\chi^2 L_v}{2n} \sum_{i=1}^n \mathbb{P}(i \notin S^{(t)})\, \mathbf{E}_t \left\|\tilde{v}_i^{(t+1)} - v_i^{(t)}\right\|^2 \\
&= \frac{\chi^2 L_v}{2n} \left(1 - \frac{m}{n}\right) \sum_{i=1}^n \mathbf{E}_t \left\|\tilde{v}_i^{(t+1)} - v_i^{(t)}\right\|^2 .
\end{aligned}$$

Plugging in Lemma 23 completes the proof. □

The analysis of the v-step is given in the next result.

Claim 9. *Consider the setting of Theorem 3 and assume that $\gamma_v \tau_v L_v \le 1/8$. We have,*

$$\mathbf{E}_t\left[F\left(u^{(t)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t)}\right)\right] \le -\frac{\gamma_v \tau_v m \Delta_v^{(t)}}{8n} + \frac{\gamma_v^2 \tau_v^2 L_v \sigma_v^2 m}{2n} + \frac{4\gamma_v^3 L_v^2 \tau_v^2 (\tau_v - 1)\sigma_v^2 m}{n} .$$

Proof. From smoothness, we get,

$$F_i\left(u^{(t)}, \tilde{v}_i^{(t+1)}\right) - F_i\left(u^{(t)}, v_i^{(t)}\right) \le \underbrace{\left\langle \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right), \tilde{v}_i^{(t+1)} - v_i^{(t)}\right\rangle}_{\mathcal{T}_{1,v}} + \underbrace{\frac{L_v}{2}\left\|\tilde{v}_i^{(t+1)} - v_i^{(t)}\right\|^2}_{\mathcal{T}_{2,v}} .$$

We bound the first term as

$$\begin{aligned}
\mathbf{E}_t[\mathcal{T}_{1,v}] &= -\gamma_v \mathbf{E}_t\left\langle \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right), \sum_{k=0}^{\tau_v - 1} \nabla_v F_i\left(u^{(t)}, \tilde{v}_{i,k}^{(t)}\right)\right\rangle \\
&= -\gamma_v \tau_v \left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 \\
&\quad - \gamma_v \sum_{k=0}^{\tau_v - 1} \mathbf{E}_t\left\langle \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right), \nabla_v F_i\left(u^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\rangle \\
&\le -\frac{\gamma_v \tau_v}{2}\left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 + \frac{\gamma_v}{2} \sum_{k=0}^{\tau_v - 1} \mathbf{E}_t \left\|\nabla_v F_i\left(u^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 \\
&\le -\frac{\gamma_v \tau_v}{2}\left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 + \frac{\gamma_v L_v^2}{2} \sum_{k=0}^{\tau_v - 1} \left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2 .
\end{aligned}$$

Next, we observe that

$$\mathbf{E}_z\|G_{i,v}(u, v_i, z)\|^2 = \|\nabla_v F_i(u, v_i)\|^2 + \mathbf{E}_z\|G_{i,v}(u, v_i, z) - \nabla_v F_i(u, v_i)\|^2 \le \|\nabla_v F_i(u, v_i)\|^2 + \sigma_v^2 .$$

We invoke this inequality to handle the second term as

$$\begin{aligned}
\mathbf{E}_t[\mathcal{T}_{2,v}] &\le \frac{\gamma_v^2 L_v \tau_v}{2} \sum_{k=0}^{\tau_v - 1} \mathbf{E}_t\left\|G_{i,v}\left(u^{(t)}, \tilde{v}_{i,k}^{(t)}, z_{i,k}^{(t)}\right)\right\|^2 \\
&\le \frac{\gamma_v^2 L_v \tau_v^2 \sigma_v^2}{2} + \frac{\gamma_v^2 L_v \tau_v}{2} \sum_{k=0}^{\tau_v - 1} \mathbf{E}_t\left\|\nabla_v F_i\left(u^{(t)}, \tilde{v}_{i,k}^{(t)}\right)\right\|^2 \\
&\le \frac{\gamma_v^2 L_v \tau_v^2 \sigma_v^2}{2} + \gamma_v^2 L_v \tau_v^2 \left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 \\
&\qquad + \gamma_v^2 L_v \tau_v \sum_{k=0}^{\tau_v - 1} \mathbf{E}_t\left\|\nabla_v F_i\left(u^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 \\
&\le \frac{\gamma_v^2 L_v \tau_v^2 \sigma_v^2}{2} + \gamma_v^2 L_v \tau_v^2 \left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 + \gamma_v^2 L_v^3 \tau_v \sum_{k=0}^{\tau_v - 1} \mathbf{E}_t\left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2 .
\end{aligned}$$

Algorithm 5 FedSim: Simultaneous update of shared and personal parameters

```
Input: Initial iterates $u^{(0)}, V^{(0)}$, Number of communication rounds $T$, Number of devices per round $m$, Number of local updates $\tau$, Local step sizes $\gamma_u, \gamma_v$.
for $t = 0, 1, \cdots, T-1$ do
  Sample $m$ devices from $[n]$ without replacement in $S^{(t)}$
  for each selected device $i \in S^{(t)}$ in parallel do
    Initialize $v_{i,0}^{(t)} = v_i^{(t)}$ and $u_{i,0}^{(t)} = u^{(t)}$
    for $k = 0, \cdots, \tau - 1$ do
      // Update all parameters jointly
      Sample data $z_{i,k}^{(t)} \sim \mathcal{D}_i$
      $v_{i,k+1}^{(t)} = v_{i,k}^{(t)} - \gamma_v G_{i,v}(u_{i,k}^{(t)}, v_{i,k}^{(t)}, z_{i,k}^{(t)})$
      $u_{i,k+1}^{(t)} = u_{i,k}^{(t)} - \gamma_u G_{i,u}(u_{i,k}^{(t)}, v_{i,k}^{(t)}, z_{i,k}^{(t)})$
    Update $v_i^{(t+1)} = v_{i,\tau}^{(t)}$ and $u_i^{(t+1)} = u_{i,\tau}^{(t)}$
  Update $u^{(t+1)} = \sum_{i\in S^{(t)}} \alpha_i u_i^{(t+1)} / \sum_{i \in S^{(t)}} \alpha_i$ at the server with secure aggregation
return $u^{(T)}, v_1^{(T)}, \cdots, v_n^{(T)}$
```

Plugging these bounds for $\mathcal{T}_{1,v}$ and $\mathcal{T}_{2,v}$ into the initial smoothness bound and using $\gamma_v L_v \tau_v \le 1/4$ gives

$$\mathbf{E}_t\Big[F_i\left(u^{(t)}, \tilde v_i^{(t+1)}\right) - F_i\left(u^{(t)}, v_i^{(t)}\right)\Big] \le \\ -\frac{\gamma_v \tau_v}{4}\left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 + \gamma_v L_v^2 \sum_{k=0}^{\tau_v - 1}\left\|\tilde v_{i,k}^{(t)} - v_i^{(t)}\right\|^2 + \frac{\gamma_v^2 L_v \tau_v^2 \sigma_v^2}{2}\,.$$

We invoke Lemma 22 to bound the $\sum_k \mathbf{E}_t\|\tilde v_{i,k}^{(t)} - v_i^{(t)}\|^2$ term, which is also known as client drift. We simplify some coefficients using $8\gamma_v\tau_v L_v \le 1$ to get

$$\mathbf{E}_t\Big[F_i\left(u^{(t)}, \tilde v_i^{(t+1)}\right) - F_i\left(u^{(t)}, v_i^{(t)}\right)\Big] \le \\ -\frac{\gamma_v \tau_v}{8}\left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 + \frac{\gamma_v^2 L_v \tau_v^2 \sigma_v^2}{2} + 4\gamma_v^3 L_v \tau_v^2(\tau_v - 1)\sigma_v^2\,.$$

It remains to invoke that $S^{(t)}$ is a uniformly random sample of m devices from $\{1, \cdots, n\}$ and that $\tilde v_i^{(t+1)}$ is independent of $S^{(t)}$. To this end, note that

$$\begin{aligned}\mathbf{E}_t\left[F\left(u^{(t)}, V^{(t+1)}\right) - F\left(u^{(t)}, V^{(t)}\right)\right] &= \frac{m}{n}\mathbf{E}_t\left[\frac{1}{m}\sum_{i\in S^{(t)}} F_i\left(u^{(t)}, \tilde v_i^{(t+1)}\right) - F_i\left(u^{(t)}, v_i^{(t)}\right)\right] \\ &\le \frac{m}{n^2}\sum_{i=1}^n \mathbf{E}_t\left[F_i\left(u^{(t)}, \tilde v_i^{(t+1)}\right) - F_i\left(u^{(t)}, v_i^{(t)}\right)\right]\,.\end{aligned}$$

Plugging in the previous bound completes the proof. □

Remark 10. *We only invoked the partial gradient diversity assumption (Assumption 3) at (virtual) iterates $(u^{(t)}, \widetilde V^{(t+1)})$; therefore, it suffices if the assumption only holds at iterates $(u^{(t)}, \widetilde V^{(t+1)})$ generated by FedAlt, rather than at all (u, V).*

A.4. Convergence Analysis of FedSim

We give the full form of FedSim in Algorithm 5 for the general case of unequal α_i's but focus on $\alpha_i = 1/n$ for the analysis. Theorem 2 of the main paper is a simplification of Corollary 12 below, which in turn is proved based on Theorem 11.

Throughout this section, we use constants

$$\sigma_{\text{sim},1}^2 = (1+\chi^2)\left(\frac{\delta^2}{L_u}\left(1 - \frac{m}{n}\right) + \frac{\sigma_u^2}{L_u} + \frac{\sigma_v^2 m}{L_v n}\right)\,, \quad \text{and}\,, \quad \sigma_{\text{sim},2}^2 = (1+\chi^2)\left(\frac{\delta^2}{L_u} + \frac{\sigma_u^2}{L_u} + \frac{\sigma_v^2}{L_v}\right)(1 - \tau^{-1})\,.$$

Theorem 11 (**Convergence of FedSim**). *Suppose Assumptions 1′, 2′ and 3′ hold and the learning rates in FedSim are chosen as $\gamma_u = \eta/(L_u\tau)$ and $\gamma_v = \eta/(L_v\tau)$ with*

$$\eta \le \min\left\{\frac{1}{12(1+\chi^2)(1+\rho^2)}, \sqrt{\frac{m/n}{196(1-\tau^{-1})(1+\chi^2)(1+\rho^2)}}\right\}.$$

Then, ignoring absolute constants, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{1}{L_u}\mathbf{E}\big[\Delta_u^{(t)}\big] + \frac{m}{nL_v}\mathbf{E}\big[\Delta_v^{(t)}\big]\right) \le \frac{\Delta F_0}{\eta T} + \eta\,\sigma_{\mathrm{sim},1}^2 + \eta^2\,\sigma_{\mathrm{sim},2}^2.$$

Before proving the theorem, we give the following corollary with optimized learning rates.

Corollary 12 (**Final Rate of FedSim**). *Consider the setting of Theorem 11 and let the total number of rounds T be known in advance. Suppose we set the learning rates $\gamma_u = \eta/(\tau L_u)$ and $\gamma_v = \eta/(\tau L_v)$, where (ignoring absolute constants),*

$$\eta = \left(\frac{\Delta F_0}{T\,\sigma_{\mathrm{sim},1}^2}\right)^{1/2} \bigwedge \left(\frac{\Delta F_0^2}{T^2\,\sigma_{\mathrm{sim},2}^2}\right)^{1/3} \bigwedge \frac{1}{(1+\chi^2)(1+\rho^2)} \bigwedge \sqrt{\frac{m/n}{(1-\tau^{-1})(1+\chi^2)(1+\rho^2)}}.$$

We have, ignoring absolute constants,

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{1}{L_u}\mathbf{E}\left\|\nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 + \frac{m}{L_v n^2}\sum_{i=1}^{n}\mathbf{E}\left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2\right) \le$$
$$\frac{\left(\Delta F_0\,\sigma_{\mathrm{sim},1}^2\right)^{1/2}}{\sqrt{T}} + \frac{\left(\Delta F_0^2\,\sigma_{\mathrm{sim},2}^2\right)^{1/3}}{T^{2/3}} + \frac{\Delta F_0(1+\chi^2)(1+\rho^2)}{T} + \frac{\Delta F_0\sqrt{\frac{n}{m}(1-\tau^{-1})(1+\chi^2)(1+\rho^2)}}{T}.$$

Proof. The proof follows from invoking Lemma 25 on the bound of Theorem 11. □

Remark 13 (**Asymptotic Rate**). *The asymptotic $1/\sqrt{T}$ rate of Theorem 2 is achieved when the $1/T$ term is dominated by the $1/\sqrt{T}$ term. This happens when (ignoring absolute constants)*

$$T \ge \frac{\Delta F_0(1+\chi^2)(1+\rho^2)}{\sigma_{\mathrm{sim},1}^2}\,\max\left\{(1-\tau^{-1})\frac{n}{m},\ (1+\chi^2)(1+\rho^2)\right\}.$$

Note that $T \ge \Omega(n/m)$ is necessary for each device to be seen at least once on average, or the personal parameters of some devices will never be updated.

We now prove Theorem 11.

Proof of Theorem 11. The proof mainly applies the smoothness upper bound to write out a descent condition with suitably small noise terms. We start with some notation.

Notation. Let $\mathcal{F}^{(t)}$ denote the σ-algebra generated by $\left(u^{(t)}, V^{(t)}\right)$ and denote $\mathbf{E}_t[\cdot] = \mathbf{E}[\cdot|\mathcal{F}^{(t)}]$. For all devices, including those not selected in each round, we define virtual sequences $\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}$ as the SGD updates in Algorithm 5 for all devices regardless of whether they are selected. For the selected devices $k \in S^{(t)}$, we have $\left(u_{i,k}^{(t)}, v_{i,k}^{(t)}\right) = \left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right)$. Note now that the random variables $\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}$ are independent of the device selection $S^{(t)}$. The updates for the devices $i \in S^{(t)}$ are given by

$$v_i^{(t+1)} = v_i^{(t)} - \gamma_v \sum_{k=0}^{\tau-1} G_{i,v}\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}, z_{i,k}^{(t)}\right),$$

and the server update is given by

$$u^{(t+1)} = u^{(t)} - \frac{\gamma_u}{m} \sum_{i \in S^{(t)}} \sum_{k=0}^{\tau-1} G_{i,u}\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}, z_{i,k}^{(t)}\right) . \tag{15}$$

Proof Outline. We use the smoothness of F_i, more precisely Lemma 20, to obtain

$$\begin{aligned} & F\big(u^{(t+1)}, V^{(t+1)}\big) - F\big(u^{(t)}, V^{(t)}\big) \\ & \le \underbrace{\left\langle \nabla_u F(u^{(t)}, V^{(t)}), u^{(t+1)} - u^{(t)} \right\rangle}_{\mathcal{T}_{1,u}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left\langle \nabla_v F_i(u^{(t)}, v_i^{(t)}), v_i^{(t+1)} - v_i^{(t)} \right\rangle}_{\mathcal{T}_{1,v}} \\ & \quad + \underbrace{\frac{L_u(1+\chi^2)}{2}\left\| u^{(t+1)} - u^{(t)} \right\|^2}_{\mathcal{T}_{2,u}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n} \frac{L_v(1+\chi^2)}{2}\left\| v_i^{(t+1)} - v_i^{(t)} \right\|^2}_{\mathcal{T}_{2,v}} . \end{aligned} \tag{16}$$

Our goal will be to bound each of these terms to get a descent condition from each step of the form

$$\begin{aligned} & \mathbf{E}_t\left[F\big(u^{(t+1)}, V^{(t+1)}\big) - F\big(u^{(t)}, V^{(t)}\big)\right] \\ & \le -\frac{\gamma_u \tau}{8}\left\|\nabla_u F\big(u^{(t)}, V^{(t)}\big)\right\|^2 - \frac{\gamma_v \tau m}{8n^2}\sum_{i=1}^{n}\left\|\nabla_v F_i\big(u^{(t)}, v_i^{(t)}\big)\right\|^2 + O(\gamma_u^2 + \gamma_v^2) , \end{aligned}$$

where the $O(\gamma_u^2 + \gamma_v^2)$ terms are controlled using the bounded variance and gradient diversity assumptions. Telescoping this descent condition gives the final bound.

Main Proof. Towards this end, we prove non-asymptotic bounds on each of the terms $\mathcal{T}_{1,v}$, $\mathcal{T}_{1,u}$, $\mathcal{T}_{2,v}$ and $\mathcal{T}_{2,u}$, in Claims 14 to 17 respectively. We then invoke them to get the bound

$$\begin{aligned} \mathbf{E}_t\left[F\big(u^{(t+1)}, V^{(t+1)}\big) - F\big(u^{(t)}, V^{(t)}\big)\right] & \le -\frac{\gamma_u \tau}{4}\Delta_u^{(t)} - \frac{\gamma_v \tau m}{4n}\Delta_v^{(t)} \\ & + \frac{L_u(1+\chi^2)\gamma_u^2\tau^2}{2}\left(\sigma_u^2 + \frac{12\delta^2}{m}(1 - m/n)\right) + \frac{L_v(1+\chi^2)\gamma_v^2\tau^2\sigma_v^2 m}{2n} \\ & + \frac{2}{n}\sum_{i=1}^{n}\sum_{k=0}^{\tau-1}\mathbf{E}_t\left\|u_{i,k}^{(t)} - u^{(t)}\right\|^2\left(L_u^2\gamma_u + \frac{m}{n}\chi^2 L_u L_v \gamma_v\right) \\ & + \frac{2}{n}\sum_{i=1}^{n}\sum_{k=0}^{\tau-1}\mathbf{E}_t\left\|v_{i,k}^{(t)} - v^{(t)}\right\|^2\left(\frac{m}{n}L_v^2\gamma_v + \chi^2 L_u L_v \gamma_u\right) . \end{aligned} \tag{17}$$

Note that we simplified some constants appearing on the gradient norm terms using

$$\gamma_u \le \big(12 L_u (1+\chi^2)(1+\rho^2)\tau\big)^{-1} \quad \text{and} \quad \gamma_v \le \big(6 L_v(1+\chi^2)\tau\big)^{-1}.$$

Our next step is to bound the last two lines of (17) with Lemma 18 and invoke the gradient diversity assumption (Assumption 3′) as

$$\frac{1}{n}\sum_{i=1}^{n}\left\|\nabla_u F_i\big(u^{(t)}, v_i^{(t)}\big)\right\|^2 \le \delta^2 + (1+\rho^2)\left\|\nabla_u F\big(u^{(t)}, V^{(t)}\big)\right\|^2 .$$

This gives, after plugging in the learning rates and further simplifying the constants,

$$\begin{aligned} & \mathbf{E}_t\left[F\big(u^{(t+1)}, V^{(t+1)}\big) - F\big(u^{(t)}, V^{(t)}\big)\right] \\ & \le -\frac{c\Delta_u^{(t)}}{8L_u} - \frac{cm\Delta_v^{(t)}}{8L_v n} + c^2(1+\chi^2)\left(\frac{\sigma_u^2}{2L_u} + \frac{m\sigma_v^2}{nL_v} + \frac{6\delta^2}{L_u m}\left(1 - \frac{m}{n}\right)\right) \\ & \quad + c^3(1+\chi^2)(1-\tau^{-1})\left(\frac{24\delta^2}{L_u} + \frac{4\sigma_u^2}{L_u} + \frac{4\sigma_v^2}{L_u}\right) . \end{aligned}$$

Taking full expectation, telescoping the series over $t = 0, \cdots, T-1$ and rearranging the resulting terms give the desired bound in Theorem 11. □

Claim 14 (Bounding $\mathcal{T}_{1,v}$)**.** *Let $\mathcal{T}_{1,v}$ be defined as in* (16)*. We have,*

$$\begin{aligned}\mathbf{E}_t[\mathcal{T}_{1,v}] \le &-\frac{\gamma_v \tau m}{2n^2}\sum_{i=1}^n \left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 \\ &+ \frac{\gamma_v m}{n}\sum_{i=1}^n\sum_{k=0}^{\tau-1}\mathbf{E}_t\left[\chi^2 L_u L_v \left\|\tilde{u}_{i,k}^{(t)} - u^{(t)}\right\|^2 + L_v^2\left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2\right].\end{aligned}$$

Proof. Define $\mathcal{T}_{1,v,i}$ to be contribution of the ith term to $\mathcal{T}_{1,v}$. For $i \notin S_t$, we have that $\mathcal{T}_{1,v,i} = 0$, since $v_i^{(t+1)} = v_i^{(t)}$. On the other hand, for $i \in S^{(t)}$, we use the unbiasedness of the gradient estimator $G_{i,v}$ and the independence of $z_{i,k}^{(t)}$ from $u_{i,k}^{(t)}, v_{i,k}^{(t)}$ to get

$$\begin{aligned}\mathbf{E}_t\left[\mathcal{T}_{1,v,i}\right] &= -\gamma_v \sum_{k=0}^{\tau-1}\mathbf{E}_t\left\langle \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right), \nabla_v F_i\left(u_{i,k}^{(t)}, v_{i,k}^{(t)}\right)\right\rangle \\ &= -\gamma_v \sum_{k=0}^{\tau-1}\mathbf{E}_t\left\langle \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right), \nabla_v F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right)\right\rangle \\ &= -\gamma_v\tau\left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 \\ &\quad -\gamma_v \sum_{k=0}^{\tau-1}\mathbf{E}_t\left\langle \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right), \nabla_v F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\rangle \\ &\le -\frac{\gamma_v\tau}{2}\left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 + \frac{\gamma_v}{2}\sum_{k=0}^{\tau-1}\mathbf{E}_t\left\|\nabla_v F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2. \end{aligned} \tag{18}$$

For the second term, we add and subtract $\nabla_v F_i\left(u^{(t)}, \tilde{v}_{i,k}^{(t)}\right)$ and use smoothness to get

$$\left\|\nabla_v F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 \le 2\chi^2 L_u L_v\left\|\tilde{u}_{i,k}^{(t)} - u^{(t)}\right\|^2 + 2L_v^2\left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2. \tag{19}$$

Since the right hand side of this bound is independent of S_t, we get,

$$\mathbf{E}_t[\mathcal{T}_{1,v}] = \frac{m}{n}\mathbf{E}_t\left[\frac{1}{m}\sum_{i\in S^{(t)}}\mathcal{T}_{1,v,i}\right] = \frac{m}{n^2}\sum_{i=1}^n \mathbf{E}_t[\mathcal{T}_{1,v,i}],$$

and plugging in (18) and (19) completes the proof. □

Claim 15 (Bounding $\mathcal{T}_{1,u}$)**.** *Consider $\mathcal{T}_{1,u}$ defined in* (16)*. We have the bound,*

$$\begin{aligned}\mathbf{E}_t[\mathcal{T}_{1,u}] \le &-\frac{\gamma_u\tau}{2}\left\|\nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 \\ &+ \frac{\gamma_u}{n}\sum_{i=1}^n\sum_{k=0}^{\tau-1}\mathbf{E}_t\left[L_u^2\left\|\tilde{u}_{i,k}^{(t)} - u^{(t)}\right\|^2 + \chi^2 L_u L_v\left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2\right].\end{aligned}$$

Proof. Due to the independence of $S^{(t)}$ from $\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}$, we have,

$$\begin{aligned}
\mathbf{E}_t\left[u^{(t+1)} - u^{(t)}\right] &= -\gamma_u \mathbf{E}_t\left[\frac{1}{m}\sum_{i\in S^{(t)}}\sum_{k=0}^{\tau-1}\nabla_u F_i\left(u_{i,k}^{(t)}, v_{i,k}^{(t)}\right)\right] \\
&= -\gamma_u \mathbf{E}_t\left[\frac{1}{m}\sum_{i\in S^{(t)}}\sum_{k=0}^{\tau-1}\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right)\right] \\
&= -\frac{\gamma_u}{n}\sum_{i=1}^{n}\sum_{k=0}^{\tau-1}\mathbf{E}_t\left[\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right)\right]\,,
\end{aligned}$$

where the last equality took an expectation over $S^{(t)}$, which is independent of $\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}$. Now, using the same sequence of arguments as Claim 14, we have,

$$\begin{aligned}
&\mathbf{E}_t\left\langle \nabla_u F\big(u^{(t)}, V^{(t)}\big), u^{(t+1)} - u^{(t)}\right\rangle \\
&= -\gamma_u \sum_{k=0}^{\tau-1}\mathbf{E}_t\left\langle \nabla_u F\left(u^{(t)}, V^{(t)}\right), \frac{1}{n}\sum_{i=1}^{n}\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right)\right\rangle \\
&\le -\frac{\gamma_u \tau}{2}\left\|\nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 + \frac{\gamma_u}{2}\sum_{k=0}^{\tau-1}\mathbf{E}_t\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 \\
&\overset{(*)}{\le} -\frac{\gamma_u \tau}{2}\left\|\nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 + \frac{\gamma_u}{2n}\sum_{i=1}^{n}\sum_{k=0}^{\tau-1}\mathbf{E}_t\left\|\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_u F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 \\
&\le -\frac{\gamma_u \tau}{2}\left\|\nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 + \frac{\gamma_u}{n}\sum_{i=1}^{n}\sum_{k=0}^{\tau-1}\mathbf{E}_t\left[L_u^2\left\|\tilde{u}_{i,k}^{(t)} - u^{(t)}\right\|^2 + L_{uv}^2\left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2\right]\,,
\end{aligned}$$

where the inequality $(*)$ follows from Jensen's inequality as

$$\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 \le \frac{1}{n}\sum_{i=1}^{n}\left\|\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right) - \nabla_u F_i\left(u_{i,k}^{(t)}, v^{(t)}\right)\right\|^2.$$

□

Claim 16 (Bounding $\mathcal{T}_{2,v}$)**.** *Consider $\mathcal{T}_{2,v}$ as defined in (16). We have the bound,*

$$\begin{aligned}
\mathbf{E}_t[\mathcal{T}_{2,v}] \le \frac{3L_v(1+\chi^2)\gamma_v^2\tau^2 m}{2n^2}&\sum_{i=1}^{n}\left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 + \frac{L_v(1+\chi^2)\gamma_v^2\tau^2 m\sigma_v^2}{2n} \\
&+ \frac{3L_v(1+\chi^2)\gamma_v^2\tau m}{2n^2}\sum_{i=1}^{n}\sum_{k=0}^{\tau-1}\mathbf{E}_t\left[L_v^2\left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2 + \chi^2 L_u L_v\left\|\tilde{u}_{i,k}^{(t)} - u^{(t)}\right\|^2\right]\,.
\end{aligned}$$

Proof. We start with

$$\begin{aligned}
\mathbf{E}_t\left\|\tilde{v}_{k,\tau}^{(t)} - v^{(t)}\right\|^2 &= \gamma_v^2 \mathbf{E}_t\left\|\sum_{k=0}^{\tau-1} G_{i,v}\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}, z_{i,k}^{(t)}\right)\right\|^2 \\
&\le \gamma_v^2 \tau \sum_{k=0}^{\tau-1} \mathbf{E}_t\left\|G_{i,v}\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}, z_{i,k}^{(t)}\right)\right\|^2 \\
&\le \gamma_v^2 \tau^2 \sigma_v^2 + \gamma_v^2 \tau \sum_{k=0}^{\tau-1} \mathbf{E}_t\left\|\nabla_v F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right)\right\|^2 \\
&\le \gamma_v^2 \tau^2 \sigma_v^2 + 3\gamma_v^2 \tau^2 \left\|\nabla_v F_i\left(u^{(t)}, v_i^{(t)}\right)\right\|^2 \\
&\qquad + 3\gamma_v^2 \tau \sum_{k=0}^{\tau-1} \mathbf{E}_t\left[L_v^2\left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2 + \chi^2 L_u L_v \left\|\tilde{u}_{i,k}^{(t)} - u^{(t)}\right\|^2\right] .
\end{aligned}$$

Using (a) $v_i^{(t+1)} = \tilde{v}_{i,\tau}^{(t)}$ for $i \in S^{(t)}$, and, (b) $S^{(t)}$ is independent from $\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}$, we get,

$$\begin{aligned}
\mathbf{E}_t[\mathcal{T}_{2,v}] &= \frac{L_v(1+\chi^2)m}{2n}\, \mathbf{E}_t\left[\frac{1}{m}\sum_{i\in S^{(t)}}\left\|\tilde{v}_{i,\tau}^{(t)} - v_i^{(t)}\right\|^2\right] \\
&\le \frac{L_v(1+\chi^2)m}{2n^2}\sum_{i=1}^{n} \mathbf{E}_t\left\|\tilde{v}_{i,\tau}^{(t)} - v_i^{(t)}\right\|^2
\end{aligned}$$

Plugging in the bound $\mathbf{E}_t\left\|\tilde{v}_{i,\tau}^{(t)} - v^{(t)}\right\|^2$ completes the proof. □

Claim 17 (Bounding $\mathcal{T}_{2,u}$)**.** *Consider $\mathcal{T}_{2,u}$ as defined in (16). We have,*

$$\begin{aligned}
\mathbf{E}_t[\mathcal{T}_{2,u}] \le\ & \frac{L_u(1+\chi^2)\gamma_u^2\tau^2}{2m}\left(\sigma_u^2 + 12\delta^2\left(1 - \frac{m}{n}\right)\right) \\
& + 3L_u(1+\chi^2)\gamma_u^2\tau^2(1+\rho^2)\left\|\nabla_u F_i\left(u^{(t)}, V^{(t)}\right)\right\|^2 \\
& + \frac{3L_u(1+\chi^2)\gamma_u^2\tau}{2n}\sum_{i=1}^{n}\sum_{k=0}^{\tau-1}\mathbf{E}_t\left[L_u^2\left\|\tilde{u}_{i,k}^{(t)} - u^{(t)}\right\|^2 + \chi^2 L_u L_v\left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2\right] .
\end{aligned}$$

Proof. We proceed with the first two inequalities as in the proof of Claim 16 to get

$$\mathbf{E}_t\left\|u^{(t+1)} - u^{(t)}\right\|^2 \le \frac{\gamma_u^2\tau^2\sigma_u^2}{m} + \gamma_u^2\tau\sum_{k=0}^{\tau-1}\underbrace{\mathbf{E}_t\left\|\frac{1}{m}\sum_{i\in S^{(t)}}\nabla_u F_i\left(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)}\right)\right\|^2}_{=:\mathcal{T}_{3,j}} .$$

For $\mathcal{T}_{3,j}$, (a) we add and subtract $\nabla_u F(u^{(t)}, V^{(t)})$ and $\nabla_u F_i(u^{(t)}, \tilde{v}_{i,k}^{(t)})$, (b) invoke the squared triangle inequality, and, (c) use smoothness to get

$$\begin{aligned}
\mathcal{T}_{3,j} =\ & 6\,\mathbf{E}_t\left\|\frac{1}{m}\sum_{i\in S^{(t)}}\nabla_u F_i\left(u^{(t)}, v_i^{(t)}\right) - \nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 + 6\left\|\nabla_u F\left(u^{(t)}, V^{(t)}\right)\right\|^2 \\
& + 3\mathbf{E}_t\left[\frac{1}{m}\sum_{i\in S^{(t)}}\left(L_u^2\left\|\tilde{u}_{i,k}^{(t)} - u^{(t)}\right\|^2 + \chi^2 L_u L_v\left\|\tilde{v}_{i,k}^{(t)} - v_i^{(t)}\right\|^2\right)\right]
\end{aligned}$$

For the first term, we use the fact that $S^{(t)}$ is obtained by sampling without replacement to apply Lemma 21 together with the gradient diversity assumption to get

$$\begin{aligned}
&\mathbf{E}_t \left\| \frac{1}{m} \sum_{i \in S^{(t)}} \nabla_u F_i \left(u^{(t)}, v_i^{(t)} \right) - \nabla_u F \left(u^{(t)}, V^{(t)} \right) \right\|^2 \\
&\quad \le \frac{1}{m} \left(\frac{n-m}{n-1} \right) \frac{1}{n} \sum_{i=1}^n \left\| \nabla_u F_i \left(u^{(t)}, v_i^{(t)} \right) - \nabla_u F \left(u^{(t)}, V^{(t)} \right) \right\|^2 \\
&\quad \le \frac{1}{m} \left(\frac{n-m}{n-1} \right) \left(\delta^2 + \rho^2 \left\| \nabla_u F \left(u^{(t)}, V^{(t)} \right) \right\|^2 \right) .
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathcal{T}_{3,j} = &\frac{12\delta^2}{m} \left(1 - \frac{m}{n} \right) + 6(1+\rho^2) \left\| \nabla_u F \left(u^{(t)}, V^{(t)} \right) \right\|^2 \\
&+ \frac{3}{n} \sum_{i=1}^n \mathbf{E}_t \left[L_u^2 \left\| \tilde{u}_{i,k}^{(t)} - u^{(t)} \right\|^2 + \chi^2 L_u L_v \left\| \tilde{v}_{i,k}^{(t)} - v_i^{(t)} \right\|^2 \right] ,
\end{aligned}$$

where we also used the independence between $S^{(t)}$ and $(\tilde{u}_{i,k}^{(t)}, \tilde{v}_{i,k}^{(t)})$. Plugging this into the expression for $\mathbf{E}_t \|u^{(t+1)} - u^{(t)}\|^2$ completes the proof. □

Lemma 18. *Let F_i satisfy Assumptions 1′-3′, and consider the iterates*

$$u_{k+1} = u_k - \gamma_u G_{i,u}(u_k, v_k, z_k) , \quad \textit{and}, \quad v_{k+1} = v_k - \gamma_v G_{i,v}(u_k, v_k, z_k) ,$$

for $k = 0, \cdots, \tau - 1$, where $z_k \sim \mathcal{D}_i$. Suppose the learning rates satisfy $\gamma_u = c_u/(\tau L_u)$ and $\gamma_v = c_v/(\tau L_v)$ with $c_u, c_v \le 1/\sqrt{6} \max\{1, \chi^{-2}\}$. Further, define,

$$A = \gamma_u L_u^2 + f \chi^2 \gamma_v L_u L_v , \quad \textit{and}, \quad B = f \gamma_v L_v^2 + \chi^2 \gamma_u L_u L_v ,$$

where $f \in (0, 1]$ is given. Then, we have the bound,

$$\begin{aligned}
\sum_{k=0}^{\tau-1} \mathbf{E}\big[A \|u_k - u_0\|^2 + B \|v_k - u_0\|^2 \big] \le\ &4\tau^2(\tau-1) \left(\gamma_u^2 \sigma_u^2 A + \gamma_v^2 \sigma_v^2 B \right) \\
&+ 12\tau^2(\tau-1) \left(\gamma_u^2 A \|\nabla_u F_i(u_0, v_0)\|^2 + \gamma_v^2 B \|\nabla_v F_i(u_0, v_0)\|^2 \right) .
\end{aligned}$$

Proof. If $\tau = 1$, there is nothing to prove, so we assume $\tau > 1$. Let $\Delta_k := A\|u_k - u_0\|^2 + B\|v_k - v_0\|^2$ and denote by $\mathcal{F}_k$ the sigma-algebra generated by (w_k, v_k). Further, let $\mathbf{E}_k[\cdot] = \mathbf{E}[\cdot | \mathcal{F}_k]$. We use the inequality $2\alpha\beta \le \alpha^2/\delta^2 + \delta^2\beta^2$ for reals α, β, δ to get,

$$\begin{aligned}
\mathbf{E}_k \|u_{k+1} - u_0\|^2 &\le \left(1 + \frac{1}{\tau - 1} \right) \|u_k - u_0\|^2 + \tau \gamma_u^2 \mathbf{E}_k \|G_{i,u}(u_k, v_k, z_k)\|^2 \\
&\le \left(1 + \frac{1}{\tau - 1} \right) \|u_k - u_0\|^2 + \tau \gamma_u^2 \sigma_u^2 + \tau \gamma_u^2 \|\nabla_u F_i(u_k, v_k)\|^2 \\
&\le \left(1 + \frac{1}{\tau - 1} \right) \|u_k - u_0\|^2 + \tau \gamma_u^2 \sigma_u^2 + 3\tau \gamma_u^2 \|\nabla_u F_i(u_0, v_0)\|^2 \\
&\qquad + 3\tau \gamma_u^2 L_u^2 \|u_k - u_0\|^2 + 3\tau \gamma_u^2 L_{uv} \|v_k - v_0\|^2 ,
\end{aligned}$$

where the last inequality followed from the squared triangle inequality (from adding and subtracting $\nabla_u F_i(u_0, v_k)$ and $\nabla_u F_i(u_0, v_0)$) followed by smoothness. Together with the analogous inequality for the v-update, we get,

$$\mathbf{E}_k[\Delta_{k+1}] \le \left(1 + \frac{1}{\tau - 1} \right) \Delta_k + A' \|u_k - u_0\|^2 + B' \|v_k - v_0\|^2 + C ,$$

where we have

$$A' = 3\tau(\gamma_u^2 L_u^2 A + \gamma_v^2 \chi^2 L_u L_v B), \quad \text{and,} \quad B' = 3\tau(\gamma_v^2 L_v^2 B + \gamma_u^2 \chi^2 L_u L_v A) \quad \text{and,}$$
$$C' = \tau\gamma_u^2\sigma_u^2 A + \tau\gamma_v^2\sigma_v^2 B + 3\tau\gamma_u^2 A\|\nabla_u F_i(u_0, v_0)\|^2 + 3\tau\gamma_v^2 B\|\nabla_v F_i(u_0, v_0)\|^2 \,.$$

Next, we apply Lemma 24 to get that $A' \le A/\tau$ and $B' \le B/\tau$ under the assumed conditions on the learning rates; this allows us to write the right hand side completely in terms of Δ_k and unroll the recurrence. The intuition behind Lemma 24 is as follows. Ignoring the dependence on τ, L_u, L_v, χ for a moment, if γ_u and γ_v are both $O(\eta)$, then A', B' are both $O(\eta^3)$, while A and B are $O(\eta)$. Thus, making η small enough should suffice to get $A' \le O(A)$ and $B' \le O(B)$.

Concretely, Lemma 24 gives

$$\mathbf{E}_k[\Delta_{k+1}] \le \left(1 + \frac{2}{\tau - 1}\right)\mathbf{E}[\Delta_k] + C \,,$$

and unrolling this recurrence gives for $k \le \tau - 1$

$$\begin{aligned}\mathbf{E}[\Delta_k] &\le \sum_{j=0}^{k-1}\left(1 + \frac{2}{\tau-1}\right)^j C \le \frac{\tau-1}{2}\left(1 + \frac{2}{\tau-1}\right)^k C \\ &\le \frac{\tau-1}{2}\left(1 + \frac{2}{\tau-1}\right)^{\tau-1} C \le \frac{e^2}{2}(\tau-1)C \,,\end{aligned}$$

where we used $(1 + 1/\alpha)^\alpha \le e$ for all $\alpha > 0$. Summing over k and using the numerical bound $e^2 < 8$ completes the proof. □

Remark 19. *We only invoked the partial gradient diversity assumption (Assumption 3) at iterates $(u^{(t)}, V^{(t)})$; therefore, it suffices if the assumption only holds at iterates $(u^{(t)}, V^{(t)})$ generated by FedSim, rather than at all (u, V).*

A.5. Technical Lemmas

The first lemma involves smoothness of two blocks of variables; we use this in the proof of FedSim.

Lemma 20 (Block Smoothness)**.** *Suppose $F_i : \mathbb{R}^d \times \mathbb{R}^{d_i}$ satisfy Assumption 1′. Then, it holds that*

$$\begin{aligned}F_i(w', v_i') - F_i(w, v_i) \le \langle \nabla_w F_i(w, v_i), w' - w\rangle + \langle \nabla_v F_i(w, v_i), v_i' - v_i\rangle \\ + \frac{L_w}{2}(1+\chi^2)\|w' - w\|^2 + \frac{L_v}{2}(1+\chi^2)\|v_i' - v_i\|^2 \,.\end{aligned}$$

Proof. Using the L_w-smoothness of $F(\cdot, v_i')$ and the L_v-smoothness of $F(w, \cdot)$, we have

$$\begin{aligned}F_i(w', v_i') - F_i(w, v_i') &\le \langle \nabla_w F_i(w, v_i'), w' - w\rangle + \frac{L_w}{2}\|w' - w\|^2, \\ F_i(w, v_i') - F_i(w, v_i) &\le \langle \nabla_w F_i(w, v_i), v_i' - v_i\rangle + \frac{L_v}{2}\|v_i' - v_i\|^2.\end{aligned}$$

Summing the above two inequalities together gives

$$\begin{aligned}F_i(w', v_i') - F_i(w, v_i) \le \langle \nabla_w F_i(w, v_i'), w' - w\rangle + \langle \nabla_v F_i(w, v_i), v_i' - v_i\rangle \\ + \frac{L_w}{2}\|w' - w\|^2 + \frac{L_v}{2}\|v_i' - v_i\|^2 \,.\end{aligned} \tag{20}$$

We can bound the first inner product term on the right-hand side of the above inequality as

$$\begin{aligned}\langle \nabla_w F_i(w, v_i'), w' - w\rangle &= \langle \nabla_w F_i(w, v_i), w' - w\rangle + \langle \nabla_w F_i(w, v_i') - \nabla_w F_i(w, v_i), w' - w\rangle \\ &\le \langle \nabla_w F_i(w, v_i), w' - w\rangle + \|\nabla_w F_i(w, v_i') - \nabla_w F_i(w, v_i)\|\|w' - w\| \\ &\le \langle \nabla_w F_i(w, v_i), w' - w\rangle + L_{wv}\|v_i' - v_i\|\|w' - w\| \\ &\le \langle \nabla_w F_i(w, v_i), w' - w\rangle + \chi\sqrt{L_w L_v}\|v_i' - v_i\|\|w' - w\| \\ &\le \langle \nabla_w F_i(w, v_i), w' - w\rangle + \chi^2\frac{L_v}{2}\|v_i' - v_i\|^2 + \chi^2\frac{L_w}{2}\|w' - w\|^2,\end{aligned}$$

where the first inequality is due to Cauchy-Schwarz, the second inequality is due to L_{wv}-Lipschitz property of $\nabla_w F_i(w, \cdot)$, the third inequality is due to the definition of χ in (5), and the last inequality is due to Young's inequality. Substituting the above inequality into (20) yields the desired result. □

Next, we have the variance of sampling without replacement. Note the correction factor of $(n-m)/(n-1)$ over sampling with replacement. We include the elementary proof for completeness.

Lemma 21 (Sampling Without Replacement)**.** *Let $a_1, \cdots, a_n \in \mathbb{R}^d$ be given. Let S be a uniformly random sample of size m from this collection, where the sampling is without replacement. Denoting the mean $\bar{a} = \sum_{i=1}^n a_i/n$, we have,*

$$\mathbf{E}_S \left\| \frac{1}{m} \sum_{i \in S} a_i - \bar{a} \right\|^2 \le \left(\frac{n-m}{n-1} \right) \frac{1}{m} \left(\frac{1}{n} \sum_{i=1}^n \|a_i - \bar{a}\|^2 \right) .$$

Proof. The statement is trivially true if $m = 1$ or $m = n$. Therefore, we assume now that $2 \le m \le n-1$. Further, without loss of generality, we assume that $\bar{a} = 0$. Finally, let $\mathcal{S}$ denote the set of all subsets of $[n]$ of size m. Note that $|\mathcal{S}| = \binom{n}{m}$. We now have,

$$\mathbf{E}_S \left\| \frac{1}{m} \sum_{i \in S} a_i \right\|^2 = \frac{1}{m^2 \binom{n}{m}} \sum_{S \in \mathcal{S}} \left(\sum_{i \in S} \|a_i\|^2 + \sum_{i,j \in S \colon i \ne j} \langle a_i, a_j \rangle \right) .$$

For the first term, we have,

$$\sum_{S \in \mathcal{S}} \sum_{i \in S} \|a_i\|^2 = \sum_{i=1}^n \sum_{S \in \mathcal{S} \,:\, i \in S} \|a_i\|^2 = \binom{n-1}{m-1} \sum_{i=1}^n \|a_i\|^2 .$$

Likewise, for the second term, we use $\sum_{j \ne i} a_j = -a_i$ to get,

$$\sum_{i,j \in S \colon i \ne j} \langle a_i, a_j \rangle = \sum_{i=1}^n \sum_{j \ne i} \sum_{S \in \mathcal{S} \,:\, i,j \in S} \langle a_i, a_j \rangle = \binom{n-2}{m-2} \sum_{i=1}^n \sum_{j \ne i} \langle a_i, a_j \rangle = -\binom{n-2}{m-2} \sum_{i=1}^n \|a_i\|^2 .$$

Therefore, we get,

$$\mathbf{E}_S \left\| \frac{1}{m} \sum_{i \in S} a_i \right\|^2 = \frac{\binom{n-1}{m-1} - \binom{n-2}{m-2}}{m^2 \binom{n}{m}} \sum_{i=1}^n \|a_i\|^2 = \frac{\binom{n-2}{m-1}}{m^2 \binom{n}{m}} \sum_{i=1}^n \|a_i\|^2 = \frac{n-m}{mn(n-1)} \sum_{i=1}^n \|a_i\|^2 .$$

□

The next two lemmas are about the effect of the local updates in the local SGD literature. The first lemma has also appeared in (Karimireddy et al., 2020); we give the proof for completeness.

Lemma 22. *Consider $f : \mathbb{R}^d \to \mathbb{R}$ which is L-smooth and fix a $w^{(0)} \in \mathbb{R}^d$. Define the sequence $(w^{(t)})$ of iterates produced by stochastic gradient descent with a fixed learning rate γ starting from $w^{(0)}$:*

$$w^{(t+1)} = w^{(t)} - \gamma g^{(t)} ,$$

where $g^{(t)}$ is an unbiased (and independent of w) estimator of $\nabla f(w)$ with bounded variance σ^2. Fix a number τ of steps. If $\gamma \le (\sqrt{2}\tau L)^{-1}$, we have the bound

$$\sum_{t=0}^{\tau-1} \|w^{(t)} - w^{(0)}\|^2 \le 8\gamma^2\tau^2(\tau-1)\|\nabla f(w^{(0)})\|^2 + 4\gamma^2\tau^2(\tau-1)\sigma^2 .$$

Proof. If $\tau = 1$, we have nothing to prove. Assume now that $\tau \ge 2$. Let $\mathcal{F}^{(t)}$ be the sigma-algebra generated by $w^{(t)}$ and denote $\mathbf{E}_t[\cdot] = \mathbf{E}[\cdot \,|\mathcal{F}^{(t)}]$. We will use the inequality

$$\mathbf{E}_t \left\| g^{(t)} \right\|^2 = \mathbf{E}_t \left\| g^{(t)} - \nabla f(w^{(t)}) \right\|^2 + \left\| \nabla f(w^{(t)}) \right\|^2 \le \sigma^2 + \left\| \nabla f(w^{(t)}) \right\|^2 . \tag{21}$$

We now successively deduce,

$$\begin{aligned}
\mathbf{E}_t\|w^{(t+1)} - w^{(0)}\|^2 &= \|w^{(t)} - w^{(0)} - \gamma g^{(t)}\|^2 \\
&\overset{(a)}{\le} \left(1 + \frac{1}{\tau - 1}\right) \|w^{(t)} - w^{(0)}\|^2 + \gamma^2 \tau \mathbf{E}_t \|g^{(t)}\|^2 \\
&\overset{(b)}{\le} \left(1 + \frac{1}{\tau - 1}\right) \|w^{(t)} - w^{(0)}\|^2 + 2\gamma^2\tau\|\nabla f(w^{(t)}) - \nabla f(w^{(0)})\|^2 + 2\gamma^2\tau\|\nabla f(w^{(0)})\|^2 + \gamma^2\tau\sigma^2 \\
&\overset{(c)}{\le} \left(1 + \frac{1}{\tau - 1} + 2\gamma^2\tau L^2\right) \|w^{(t)} - w^{(0)}\|^2 + 2\gamma^2\tau\|\nabla f(w^{(0)})\|^2 + \gamma^2\tau\sigma^2 \\
&\overset{(d)}{\le} \left(1 + \frac{2}{\tau - 1}\right) \|w^{(t)} - w^{(0)}\|^2 + 2\gamma^2\tau\|\nabla f(w^{(0)})\|^2 + \gamma^2\tau\sigma^2 \,.
\end{aligned}$$

Above, we used (a) the inequality $2\alpha\beta \le \alpha^2/\delta^2 + \delta^2\beta^2$ for reals α, β, δ, (b) Eq. (21), (c) L-smoothness of f, and, (d) the condition on the learning rate.

Let $C = 2\gamma^2\tau\|\nabla f(w^{(0)})\|^2 + \gamma^2\tau\sigma^2$. Unrolling the inequality and summing up the series gives for all $t \le \tau - 1$

$$\begin{aligned}
\|w^{(t)} - w^{(0)}\|^2 \le C \sum_{j=0}^{t-1} \left(1 + \frac{2}{\tau - 1}\right)^j &\le \frac{C}{2}(\tau - 1)\left(1 + \frac{2}{\tau - 1}\right)^t \\
&\le \frac{C}{2}(\tau - 1)\left(1 + \frac{2}{\tau - 1}\right)^{\tau - 1} \le \frac{C}{2}(\tau - 1)e^2 \,,
\end{aligned}$$

where we used the bound $(1 + 1/\alpha)^\alpha \le e$ for all $\alpha > 0$. Summing over t and using the numerical bound $e^2 < 8$ completes the proof. □

Lemma 23. *Consider the setting of Lemma 22. If $\gamma \le (2\tau L)^{-1}$, we have the bound*

$$\|w^{(\tau)} - w^{(0)}\|^2 \le 16\gamma^2\tau^2\|\nabla f(w^{(0)})\|^2 + 8\gamma^2\tau^2\sigma^2 \,.$$

Proof. Proceeding similar to the last proof (expect using $\delta = \tau$) gives us

$$\mathbf{E}_t\left\|w^{(t+1)} - w^{(0)}\right\|^2 \le \left(1 + \frac{2}{\tau}\right)\left\|w^{(t)} - w^{(0)}\right\|^2 + 4\gamma^2\tau\left\|\nabla f(w^{(0)})\right\|^2 + 2\gamma^2\tau\sigma^2 \,.$$

Unrolling and summing up the sequence completes the proof, similar to that of Lemma 22. □

The next lemma is about bounding constants.

Lemma 24. *Let $\gamma_u, \gamma_v, L_w, L_v, \chi, f \in \mathbb{R}_+$ and a natural number τ be given. Denote*

$$A := \gamma_u L_u^2 + f\gamma_v\chi^2 L_u L_v \,, \quad \textit{and}, \quad B := f\gamma_v L_v^2 + \gamma_u\chi^2 L_u L_v \,.$$

Suppose $\gamma_u = c_u/(\tau L_u)$ and $\gamma_v = c_v/(\tau L_v)$ with $c_u, c_v > 0$ satisfying

$$c_u, c_v \le \frac{1}{\sqrt{6}} \max\{1, \chi^{-2}\} \,.$$

Then, we have that

$$\gamma_v^2\chi^2 L_u L_v B + \gamma_u^2 L_u^2 A \le A/(3\tau^2) \,, \quad \textit{and}, \quad \gamma_u^2\chi^2 L_u L_v A + \gamma_v^2 L_v^2 B \le B/(3\tau^2) \,.$$

Proof. Note that it suffices to show

$$3\tau^2\chi^2\gamma_v^2 L_u L_v B \le A/2 \,, \quad \text{and}, \quad 3\tau^2\chi^2\gamma_u^2 L_u L_v A \le B/2 \,.$$

Plugging in γ_u, γ_v, these are equivalent to

$$6\chi^2 f c_v^3 + 6\chi^4 c_v^2 c_u \le \chi^2 f c_v + c_u \quad \text{and}, \quad 6\chi^2 c_u^3 + 6\chi^4 f c_v c_u^2 \le f c_v + \chi^2 c_u \,.$$

The assumption on c_v implies that $6\chi^2 f c_v^3 \le \chi^2 f c_v$ and $6\chi^4 c_v^2 c_u \le c_u$. Therefore, the first condition holds. Similarly, the second condition holds too. □

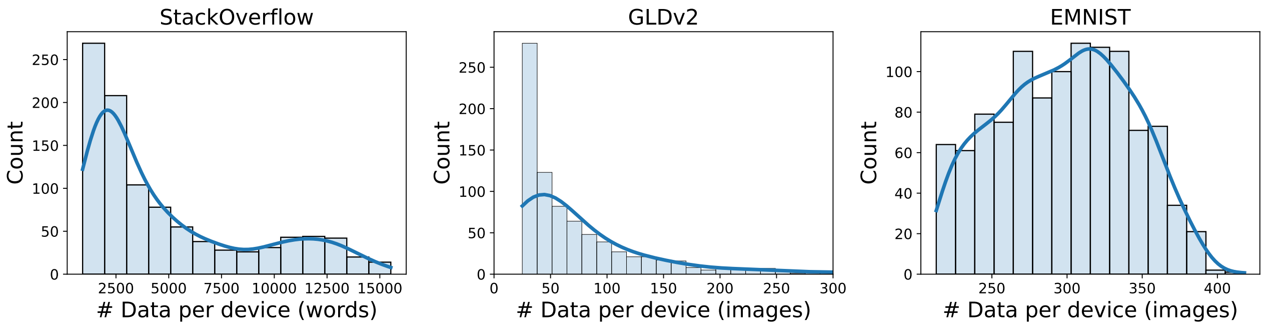


Figure 6. Distribution of number of training samples per device for each of the tasks considered in the experiments. For GLDv2, we do not show the long right tail, where the maximum number of data points per device is 1000 (cf. Table 1).

The final lemma is about tuning the learning rate: the proof is elementary and is omitted.

Lemma 25. *Consider the map $\varphi : (0, \Gamma] \to \mathbb{R}_+$ given by*

$$\varphi(\gamma) = \frac{A}{\gamma T} + B\gamma + C\gamma^2 \,,$$

where $\Gamma, A, B, C > 0$ are given. Then, we have,

$$\varphi(\gamma^\star) \le \frac{A}{\Gamma T} + 2\left(\frac{AB}{T}\right)^{1/2} + 2C^{1/3}\left(\frac{A}{T}\right)^{2/3} \,,$$

where $\gamma^\star$ is given by

$$\gamma^\star = \min\left\{\Gamma, \sqrt{\frac{A}{BT}}, \left(\frac{A}{CT}\right)^{1/3}\right\} \,.$$

B. Experiments: Detailed Setup and Hyperparameters

We conduct our experiments on four datasets from three modalities, namely images, text, and speech. The datasets contain a natural, non-i.i.d. split of data which is reflective of data heterogeneity encountered in federated learning. We describe in detail the experimental setup and hyperparameters. The code to reproduce the experimental results will be publicly released.

The outline of this section is:

- §B.1 describes the tasks and their associated datasets and metrics.
- §B.2 describes the experimental pipeline as well as the baselines we compare to.
- §B.3 presents the hyperparameters of all the algorithms.

As discussed in §1, we take the weight α_k to be proportional to the number of datapoints available on the device.

B.1. Datasets, Tasks and Models

We consider four tasks motivated by real-world applications of federated learning. The tasks are summarized in Table 1 of the main paper and the distribution of data across the clients is visualized in Figure 6.

For each model, we consider three partial personalization architectures:

(a) **Input layer personalization**: Motivated by Liang et al. (2019), this architecture places the first layer on-device to learn a personalized representation per-client, while the rest of the model is shared. For the next-word prediction transformer model, we use the first transformer layer in place of the word embedding layer owing to its large size.

(b) **Output layer personalization**: Motivated by Collins et al. (2021), this architecture learns a shared global representation but personalizes the prediction layer. For the next-word transformer model, we use the last transformer layer in place of the last prediction layer owing to its large size. For the same reason, we use the second fully connected layer within the final transformer block for the speech-to-text transformer.

Table 5. Summary of partial personalization architectures for the transformer model for next word prediction.

Personalization Type	Layer on-device	# Personalized Params.	# Shared Params.
Input Layer	1st transformer block	$0.8M$	$4.9M$
Output Layer	Last transformer block	$0.8M$	$4.9M$
Adapter	Adapter modules	$0.07M$	$5.7M$

(c) **Adapter personalization**: We also consider a novel partial personalization architecture, where the full model is shared among all clients, while each client adds personalized adapter modules, which are lightweight modules added between layers of the shared model. We use the transformer adapters proposed by Houlsby et al. (2019) and residual adapters proposed by Rebuffi et al. (2017).

B.1.1. StackOverflow for Next Word Prediction

Dataset. The StackOverflow dataset comprises of questions and answers from the programming question-answer website stackoverflow.com. The goal of the next word prediction task is to predict the next word given a partial sequence of words in a question or answer. This task is a good open-source benchmark for next word predictions in mobile keyboards. We use the StackOverflow dataset provided by TensorFlow Federated.

Client Distributions. Each client corresponds to one user on Stack Overflow; the data on the client corresponds to the questions and answers posted by this user. We only consider clients with at least 100 training sequences and 10 testing sequences, where a sequence refers to either a question or an answer. We use a fixed subsample of 1000 of them. Following Reddi et al. (2021), we restrict the vocabulary to the top 10000 most frequently occurring words in the dataset. We pad and truncate each sequence of each client to length 20 and consider at most 1000 training sequences on each client.

Model. We use a transformer model (Vaswani et al., 2017) commensurate in size with BERT Mini (Turc et al., 2019). It has with 4 transformer blocks and 4 attention heads in each self-attention layer with a transformer hidden dimension of 256 and a fully-connected hidden dimension of 1024. The output layer is a causal language modeling head, i.e., a fully connected layer which assigns a score for each possible vocabulary item, including the special tokens. The model has 6 million parameters, which require around 23 megabytes of memory.

Partial Personalization Architecture. The partial personalization architectures used are summarized in Table 5.

Loss Function and Evaluation Metric. We train the model with the causal language modeling objective. That is, for each partial sequence, we treat the prediction of the next word as a multiclass classification problem to minimize the multinomial logistic loss, also known as cross entropy loss. For evaluation, we use the top-1 accuracy of predicting words in the proper 10000-word vocabulary (i.e., ignoring special tokens such as padding, out-of-vocabulary, and beginning/end of sequence).

B.1.2. GLDv2 for Visual Landmark Recognition

Dataset. GLDv2 stands for Google Landmarks Dataset v2 (Weyand et al., 2020), which is a large-scale image dataset. It contains images of popular landmarks from around the world taken and uploaded by Wikipedia contributors. While the images vary in size, the most common image size is 800×600 pixels.

The goal of the visual landmark recognition task is to identify the landmark from its image. This task resembles a scenario where smartphone users take photos of natural and architectural landmarks while traveling. We use the federated version of the GLDv2 dataset introduced by Hsu et al. (2020) with 2028 landmarks and provided by TensorFlow Federated.

Client Distributions. Each client corresponds to one Wikipedia user and contains all the images contributed by that user. We only all 823 clients with at least 50 datapoints. We do not use original test set from GLDv2 from evaluation as it comes from different clients. Instead, we take 50% of the data on each client as a testing set.

Model. We use a ResNet-18 (He et al., 2016) model pretrained on ImageNet (Deng et al., 2009), with group normalization instead of batch normalization (Hsieh et al., 2020). We resize all images to 224×224. We use two data augmentations for training: a random crop from 256×256 and a random horizontal flip. The model has 12 million parameters, which require

Table 6. Summary of partial personalization architectures for the ResNet-18 model for visual landmark recognition.

Personalization Type	Layer on-device	# Personalized Params.	# Shared Params.
Input Layer	1st conv. layer	$0.01M$	$12.2M$
Output Layer	Last fully connected layer	$1M$	$11.2M$
Adapter	Residual adapter modules	$1.4M$	$12.2M$

Table 7. Summary of partial personalization architectures for the ResNet-18 model for character recognition.

Personalization Type	Layer on-device	# Personalized Params.	# Shared Params.
Input Layer	1st conv. layer	$0.7K$	$11.2M$
Output Layer	Last fully connected layer	$0.03M$	$11.2M$
Adapter	Residual adapter modules	$1.4M$	$11.2M$

around 49 megabytes of storage.

Partial Personalization Architecture. The partial personalization architectures used are summarized in Table 6.

Loss Function and Evaluation Metric. We use the multinomial logistic loss, also known as cross entropy loss. We evaluate the performance of the model using its classification accuracy.

B.1.3. EMNIST for Character Recognition

Dataset. EMNIST (Cohen et al., 2017) is a character recognition dataset. The goal is to identify images of handwritten digits or letters; there are 62 possible options (a-z,A-Z, 0-9). The images are grey-scaled pictures of $28 \times 28 = 784$ pixels. We use the EMNIST dataset provided by TensorFlow Federated.

Client Distributions. Each client corresponds to one "writer", i.e., the human subject who hand-wrote the digit/letter during the data collection process. We only use those clients with at least 100 training points and 25 testing points: there are 1114 of such clients.

Model. We use a ResNet-18 (He et al., 2016) model with group normalization instead of batch normalization (Hsieh et al., 2020). We make two modifications to handle the smaller image size ($28 \times 28 \times 1$ as opposed to the $224 \times 224 \times 3$ which the original ResNet was designed to accept): (a) we use a convolutional kernel of size 3×3 rather than the original 7×7 in the first convolution layer, and, (b) we drop the first pooling layer. The model has 11 million parameters, which require around 45 megabytes. Note that the number of parameters in this ResNet is smaller than the one for GLDv2 due to the architectural modifications we make for smaller images as well as the smaller number of classes.

Partial Personalization Architecture. The partial personalization architectures used are summarized in Table 7.

Loss Function and Evaluation Metric. We use the multinomial logistic loss, also known as cross entropy loss. We evaluate the performance of the model using its classification accuracy.

B.1.4. LibriSpeech for Automatic Speech Recognition

Dataset. Librispeech is a speech-to-text dataset containing snippets of speech and the associated text from open domain audiobooks (Panayotov et al., 2015). Given an utterance containing read English speech, the goal is output a text transcription. Each device corresponds to the narrator of the utterance, leading to a natural non-identical split of the data with differences in accent, tone, and voice across devices. This task is reflective of voice commands and speech recognition on mobile phones.

We create a federated version of LibriSpeech. We use the "clean" subsets of LibriSpeech (a total of 460h of speech) to pretrain a model in a non-federated manner. We use the "train-other-500" subset (a total of 500h of audio), which typically contains noiser audio, to construct a federated dataset. Real-world federated tasks often contain proxy data used to pretrain a

Table 8. Summary of partial personalization architectures for the transformer model for speech recognition.

Personalization Type	Layer on-device	# Personalized Params.	# Shared Params.
Input Layer	Convolutional subsamplers	$0.8M$	$12.6M$
Output Layer	2nd f.c. in last transformer block	$0.6M$	$12.8M$
Adapter	Adapter modules	$0.15M$	$13.4M$

model prior to federated training, such as ImageNet-pretrained vision models. We emulate this setup by first pretraining all our models on the non-federated clean subset of LibriSpeech.

Client Distributions. We construct the federated dataset from the train-other-500 subset of LibriSpeech and do not use the corresponding dev and test sets. Of the 1166 narrators, we discard those with only one chapter of data.[2] For each narrator, we assign one chapter as the test data and the remaining as the training data. This is done to ensure that each device has between $10 - 50\%$ of the device's total data in terms of length of audio[3] — this leads to approximately 30% of the available audio being used for testing and the remaining 70% for training. Overall, we get a federated dataset with 902 narrators, each of whom corresponds to a device in the federated setting.

Model. We use a transformer model (Vaswani et al., 2017) with convolutional subsamplers, as proposed by Synnaeve et al. (2019). The input audio is represented as a sequence of 40 log-mel filterbank coefficients. The model has two 1D convolutional layers with a stride of 2, followed by 6 transformer blocks and 6 attention heads in each self-attention layer with a transformer hidden dimension of 384 and a fully-connected hidden dimension of 1536. The final output layer produces log probabilities on an output vocabulary of 5000 byte pair encodings of subwords. The model has 15 million parameters, requiring around 60 megabytes of memory.

Partial Personalization Architecture. The partial personalization architectures used are summarized in Table 8.

Loss Function and Evaluation Metric. We train the model with the Connectionist Temporal Classification (CTC) loss (Graves et al., 2006). This is a structured prediction loss that uses dynamic programming to marginalize over all possible alignments between the per-frame subwords and the text transcription. For evaluation, we use the word error rate (WER) obtained from a greedy decoding of the model prediction for a given utterance (or equivalently, beam search with a beam size of 1 with no external language models).

B.2. Experimental Pipeline and Baselines

There are three components in the training pipeline for all experiments:

(a) Non-personalized federated training: The first step involves training a global model w_g using the one-model-fits-all approach of (1) with FedAvg variants.

(b) Personalized federated training: This optional second step involves training the shared parameters w together with the personalized parameters v_k using a personalized federated learning approach. We warm-start w, v_k from the non-personalized model w_g from the previous step.

(c) Final finetuning: The last step involves only finetuning the personalized parameters v_k while the shared parameters w remain unchanged.

For step (b), we initialize v_k for each k to be the appropriate part of w_g for input/output layer personalization. On the other hand, for adapters, we initialize v_k to be equal to the *same* set of randomly initialized weights for each device k.

We consider the following baselines:

- **Non-personalized**: This denotes the performance of step (a) of the pipeline above, i.e., non-personalized federated training with FedAvg variants.
- **Full model personalization**: We consider three baselines of personalization of the full model:

[2] LibriSpeech organizes the data for each narrator into chapters of the source book.

[3] When multiple candidate chapters are available for use as a test set, we use the one closest in size to 20% of the data.

Table 9. Hyperparameters for each dataset/task.

	Hyperparameter	StackOverflow	GLDv2	EMNIST	LibriSpeech
Common	Batch size	64	64	32	32
	Devices per round	50	50	10	50
	Local epochs	1	1	1	1
	Server Optimizer	FedAdam	FedAdam	FedAvg	FedAdam
	Client Optimizer	SGD	SGD	SGD	SGD
	Global Scheduler	Linear	Linear	Exponential	Linear
	Warm up	10% of rounds	10% of rounds	N/A	10% of rounds
	LR decay rounds	N/A	N/A	500	N/A
	Max. grad. norm.	0.1	N/A	N/A	0.25
Non-personalized training (step (a) of the pipeline)	# Rounds	1000	2500	2000	500
	Server learning rate	5×10^{-4}	2×10^{-4}	1.0	10^{-3}
	Client learning rate	1	10^{-2}	0.5	10^{-2}
Personalized training (step (b) of the pipeline)	# Rounds	500	600	500	500
	Server learning rate	5×10^{-5}	2×10^{-5}	1.0	10^{-3}
	Client learning rate	10^{-1}	10^{-3}	10^{-2}	10^{-2}
Local finetuning (step (c) of the pipeline)	#Epochs	5	5	5	5
	Optimizer	SGD	SGD	SGD	SGD
	Client learning rate	10^{-1}	10^{-3}	10^{-2}	10^{-4}

(i) **Finetune**: The non-personalized model from step (a) of the pipeline above is finetuned locally on each client (step (c) of the pipeline). Step (b) is skipped for this baseline.

(ii) **Ditto** (Li et al., 2021): The non-personalized model from step (a) of the pipeline above is finetuned locally on each client (step (c) of the pipeline) with ℓ_2 regularization $\|v - w_g\|^2$. Step (b) is skipped for this baseline.

(iii) **pFedMe** (Dinh et al., 2020): The non-personalized baseline model from step (a) is trained further in step (b) to optimize (2) using the pFedMe algorithm of Dinh et al. (2020). Finally the resulting model w is finetuned locally in step (c).

- **Partial Model Personalization**: We consider partial model personalization with three different architectures, as defined in §B.1. For each personalization approach, we start with the non-personalized model in step (a), continue personalization in step (b) using either FedAlt or FedSim as the algorithm, and finally run step (c) for the local finetuning.

B.3. Hyperparameters and Evaluation Details

All the tuning of hyperparameters was performed on validation data, formed by holding out 20% of the training data on each device. Once the tuning was complete, we reran the experiments on the full training data, including those held out for validation.

Evaluation Metric. Our primary evaluation metric for next-word prediction and image classification is the weighted average of the test accuracy on each client, weighted by the number of test examples (the details of how the accuracy is computed on each dataset is given in §B.1 in the paragraph on "Loss Function and Evaluation Metric"). This corresponds to the unweighted accuracy obtained by pooling all the data locally, similar to the loss as discussed in §1. The same metric is used for hyperparameter tuning and is reported in all the tables and plots, unless explicitly noted otherwise. For speech recognition, we similarly use a weighted average of the word error rate (WER).

The final hyperparameters we use are given in Table 9.

Rounds. We start with the number of communication rounds (i.e., the number of calls to secure aggregation routine for the shared parameters), which is used to measure the progress of each algorithm. For the non-personalized training, we use 1000 rounds for StackOverflow, 2500 rounds for GLDv2 and 2000 rounds for EMNIST. For the personalized training, we warm-start the model from the non-personalized one, and run the training for 500 rounds for StackOverflow and EMNIST and 600 rounds for GLDv2.

Devices per Round. All devices are assumed to be available and selections are made uniformly at random. Following

Table 10. Memory requirements (in megabytes) for training partial model personalization and full model personalization for the experimental settings considered here.

Mode	StackOverflow	GLDv2	EMNIST
No personalization	71	186	142
Input layer personalization	67	186	142
Output layer personalization	67	174	142
Adapter personalization	72	222	159
Full personalization	116	263	232
Memory savings with partial personalization	**42%**	**34%**	**39%**

(Reddi et al., 2021; Weyand et al., 2020), we select 50 devices per round for StackOverflow/GLDv2 and 10 per round for EMNIST, for both the non-personalized as well as the personalized training.

Local Updates and Minibatch Size. Each selected device locally runs 1 epoch of mini-batch stochastic gradient descent locally for non-personalized as well as personalized federated training. The final finetuning at the end of personalized training is performed for 5 epochs. We use a minibatch size of 64 for StackOverflow/GLDv2 and 32 for EMNIST for all settings.

Server and Client Optimizer Details. We use FedAvg for EMNIST and FedAdam (Reddi et al., 2021) for StackOverflow and GLDv2. We also use a global scheduler, which applies a schedule on the client learning rates across rounds, while the client learning rate within each round is held constant. We use either a linear scheduler or an exponential scheduler (also called "stepLR" in PyTorch). A linear scheduler applies a linear warmup, if applicable, until the maximum learning rate followed by a linear decay to 0. An exponential scheduler halves the client learning rate once every fixed number of rounds. Both the client and server learning rates are tuned using the validation set.

Regularization Coefficient for pFedMe and Ditto. We tune the regularization coefficient $\lambda_k = \lambda$ for pFedMe and Ditto using the validation data from the set $\{10^{-4}, 10^{-3}, \cdots, 10^{0}\}$ of possible values. The tuned values are:

- StackOverflow: 10^{-3} for Ditto and 10^{-4} for pFedMe,
- GLDv2: 10^{-1} for both Ditto and pFedMe,
- EMNIST: 10^{-1} for both Ditto and pFedMe.

Random Seed. We report numbers averaged over 5 random seeds for all experiments, with the exception of the speech recognition task.

B.4. Estimated Memory Requirement

We estimate the memory footprint for partial versus full personalization during training below. During deployment, the memory footprint of partial and full model personalization is the same since one full model is deployed.

Estimation Procedure. We assume that the following are needed to be stored on device $i \in S^{(t)}$ during round t of training:

- $u^{(t)}$, the previous broadcast global model, which is needed to calculate the model delta to be sent back to the server,
- current iterate of the shared parameter $u_{i,k}^{(t)}$,
- current iterate of the personal parameter $v_{i,k}^{(t)}$,
- their respective gradients ∇_u and ∇_v, and,
- the internal buffers required for backpropagation.

The total memory consumption is therefore,

$$\text{Memory} = 3 \times \text{size}(u) + 2 \times \text{size}(v) + \text{size(backprop)}\,.$$

Table 11. A comparison of FedAlt and FedSim on the **speech recognition** task in terms of the word error rate (WER) %. Smaller values indicate better predictive performance.

Personalization	FedAlt	FedSim
Finetune	15.55	15.55
Input Layer	**15.13**	15.47
Output Layer	15.53	15.51
Adapter	15.50	15.54

Table 12. The change in accuracy (percentage points) from the final finetuning for FedAlt and FedSim with stateful devices. The subscript denotes the standard deviation over 5 random seeds.

	StackOverflow		GLDv2		EMNIST	
	FedAlt	FedSim	FedAlt	FedSim	FedAlt	FedSim
Input Layer	$-0.06_{0.01}$	$0.04_{0.02}$	$0.12_{0.02}$	$0.17_{0.03}$	$0.12_{0.01}$	$0.12_{0.03}$
Output Layer	$0.00_{0.01}$	$0.25_{0.02}$	$0.49_{0.02}$	$0.57_{0.03}$	$0.09_{0.01}$	$0.09_{0.03}$
Adapter	$0.01_{0.01}$	$0.40_{0.08}$	$0.14_{0.02}$	$0.17_{0.01}$	$0.27_{0.02}$	$0.33_{0.03}$

We estimate the size of the backpropagation buffers for a batch size of 1.

Training Memory Requirement. For full model personalization size(v) = size(u), whereas size$(v) \ll$ size(u) for the partial personalization architectures we have considered. Therefore, the total memory requirement of training partial model personalization will be smaller than full model model personalization.

From Table 10, we see that partial personalization can result in a 34% to 42% reduction in the memory consumption across the models and datasets considered in the experiments.

C. Experiments: Additional Results

We now present the detailed experimental results.

C.1. Speech Recognition: FedAlt vs. FedSim

We compare FedAlt and FedSim for speech recognition in Table 11. We find that input layer personalization with FedAlt has the smallest word error rate of all the models considered.

C.2. Ablation: Final Finetuning for FedAlt and FedSim

We now study the effect of the final finetuning (step (c) of the experimental pipeline; cf. §B.2) for FedAlt and FedSim.

The final finetuning has a minimal impact on partial personalization. We see from Table 12 that the effect of the final finetuning is much smaller than the improvements from personalization. For instance, the improvements from finetuning are close to 0 for FedAlt on the StackOverflow dataset. For GLDv2, the finetuning accounts for < 0.5pp of improvement, whereas personalization overall accounts for 5 to 15pp.

The final finetuning is more important to FedSim than FedAlt. Table 12 also shows that the final finetuning helps FedSim more than FedAlt. However, FedAlt still outperforms FedSim, as we saw in Table 4. Overall, this shows that FedAlt is a better algorithm than FedSim. The final finetuning helps FedSim make up some percentage points in accuracy, but not enough to make up its gap with FedAlt.

C.3. Effect of Personalization on Per-Device Generalization

Summary of all scatter plots. All the scatter plots shown in the main paper are summarized in the violin plot of Figure 7. We see from the leftmost figure that the training accuracies on all devices improve with personalization. From the second

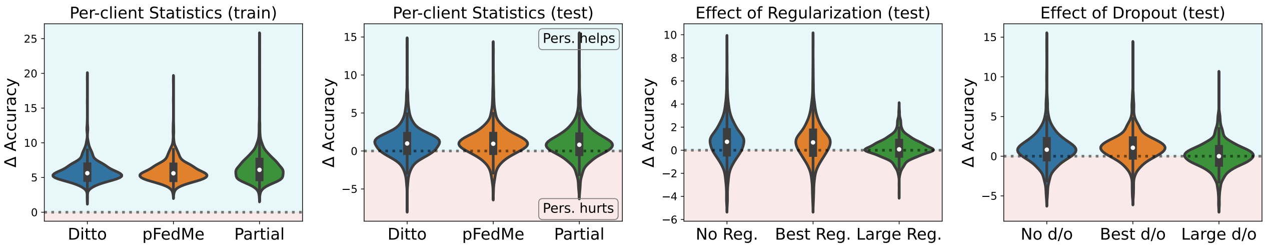


Figure 7. **Left two**: Distribution of change in the per-device train (left most) and test (center left) accuracy due to personalization on the StackOverflow dataset. **Right two**: Distribution of change in the per-device test accuracy of partial personalization under regularization on the StackOverflow dataset: (a) center right: adapter personalization under ℓ_2 regularization, and, (b) rightmost: output layer personalization under dropout. Note that the "No Reg." and "No d/o" plots on the right two are different because they personalize different model parts. **Interpretation**: The white dot in inside the violin denotes the median, while the black box enclosing this white dot marks the interquartile range (i.e., 25^{th} and 75^{th} percentiles). The body of the violin is a kernel density estimate of the distribution of accuracies. The lines extend out to the minimum and maximum accuracy in each case.

figure, we see that the test accuracy of some of the devices reduces with personalization; this is true for both partial and full personalization.

From the third plot of Figure 7, we see that regularization does not mitigate this overfitting. In fact, the regularization tuned for best average accuracy leads to a nearly identical distribution of test accuracies. A larger regularization reduces the spread of accuracies, but does so at the expense of a smaller median (white dot). The fourth plot of Figure 7 shows that the effect of dropout is similar. The best dropout improves the median accuracy, but it does not mitigate the issue of some devices being hurt by personalization.

Train Accuracy plots for devices. From Figure 8, we see that personalization leads to *a reduction in test accuracy* on some of the devices beyond the initial non-personalized model. The corresponding train accuracy plot is given in Figure 8. We observe that the personalization always leads to an improvement in the training accuracy but not in the test accuracy. The analogous plots for GLDv2 are in Figure 9, where the trends are similar.

Whether personalization helps a device or not depends on the random seed. We see in Figure 11 that the shaded region for some of the devices intersects the dotted line at 0. In other words, personalization sometimes helps this device and sometimes hurts it, depending on the random seed. This indicates that the best fix in practice is to use A/B testing on the *deployed model* to choose whether to use the personalized model or the non-personalized one.

Regularization and dropout do not mitigate this issue. From the first row of Figure 10, we see that the weight decay with best mean accuracy exactly matches the unreguarlized case in terms of per-device statistics. Increasing the regularization weight can reduce the spread of per-device accuracy. However, this only leads to a worse mean accuracy and does not mitigate the issue of personalization hurting individual devices.

From the second row of Figure 10, we see that the best dropout (0.3 in this case) leads to slight increase in average accuracy (0.18 pp). It also reduces the number of devices hurt by personalization from 256 out of 1000 to 193, but it does not fix this issue. Increasing dropout further only leads to a degradation of per-device statistics.

C.4. Partial Personalization for Stateless Devices

The algorithms we considered in this paper, namely FedAlt and FedSim, require the devices to maintain the personalized parameters v_i's as state across rounds. In cross-device federated learning settings, it is also interesting to consider *stateless* devices, which are not allowed to maintain state between training rounds.

We give preliminary experiments in this setting. We modify the FedAlt and FedSim algorithms from the main paper so that the personalized parameters v_i are reinitialized each time device i is chosen for participation. We warm-start v_i from the appropriate part of the non-personalized model trained in step (a) of the pipeline. For adapters, we fix a random initialization once, and reuse it.

FedAlt is better than FedSim for stateless devices, although the improvement is smaller. We see from Table 13 that all

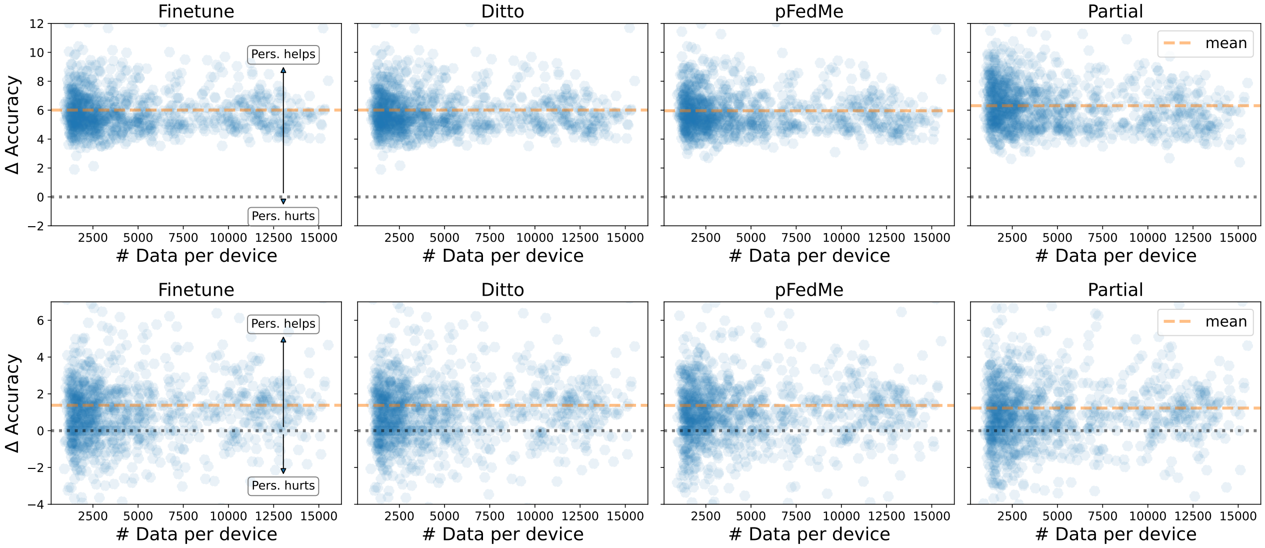


Figure 8. Scatter plot of change in accuracy (pp) per-device versus the number of training samples on the device for StackOverflow. **Top**: Training accuracy. **Bottom**: Test accuracy. This is the full version of Figure 5 from the main paper.

Table 13. This is the counterpart of Table 4 to stateless devices. We compare FedAlt and FedSim for partial model personalization with stateless devices. "FT (part.)" corresponds to finetuning the personal parameters v_i locally while fixing the shared parameters u from a non-personalized training. The numbers are averaged over 5 random seeds; the boldfaced numbers denote the highest accuracy in each row.

	StackOverflow			GLDv2			EMNIST		
	FT (part.)	FedAlt	FedSim	FT (part.)	FedAlt	FedSim	FT (part.)	FedAlt	FedSim
Input Layer	$\mathbf{24.96}_{0.01}$	$24.84_{0.01}$	$24.89_{0.01}$	$51.97_{0.02}$	$\mathbf{52.76}_{0.06}$	$52.74_{0.02}$	$93.29_{0.00}$	$\mathbf{93.51}_{0.03}$	$93.48_{0.04}$
Output Layer	$24.93_{0.01}$	$\mathbf{24.94}_{0.01}$	$24.94_{0.01}$	$53.21_{0.01}$	$\mathbf{53.30}_{0.06}$	$53.30_{0.08}$	$93.37_{0.01}$	$\mathbf{93.53}_{0.03}$	$93.51_{0.04}$
Adapter	$\mathbf{24.71}_{0.00}$	$24.69_{0.01}$	$24.71_{0.01}$	$63.86_{0.06}$	$\mathbf{64.10}_{0.14}$	$63.19_{0.04}$	$93.66_{0.00}$	$\mathbf{93.97}_{0.04}$	$93.89_{0.02}$

algorithms perform similarly for the stateless setting. Nevertheless, we see that FedAlt obtains mild improvements over both FedSim and finetuning for GLDv2, e.g., 0.24pp with adapters.

The final finetuning is crucial for stateless devices. We see from Table 14 that the final finetuning accounts for most of improvements in the stateless case. For instance, for GLDv2, the final finetuning accounts for 11.68 and 10.42pp out of a total of 12.67 and 11.76pp for FedAlt and FedSim respectively. However, the personalized federated training (step (b) of the pipeline; cf. §B.2) still leads to an increase in accuracy of 1 to 1.34pp.

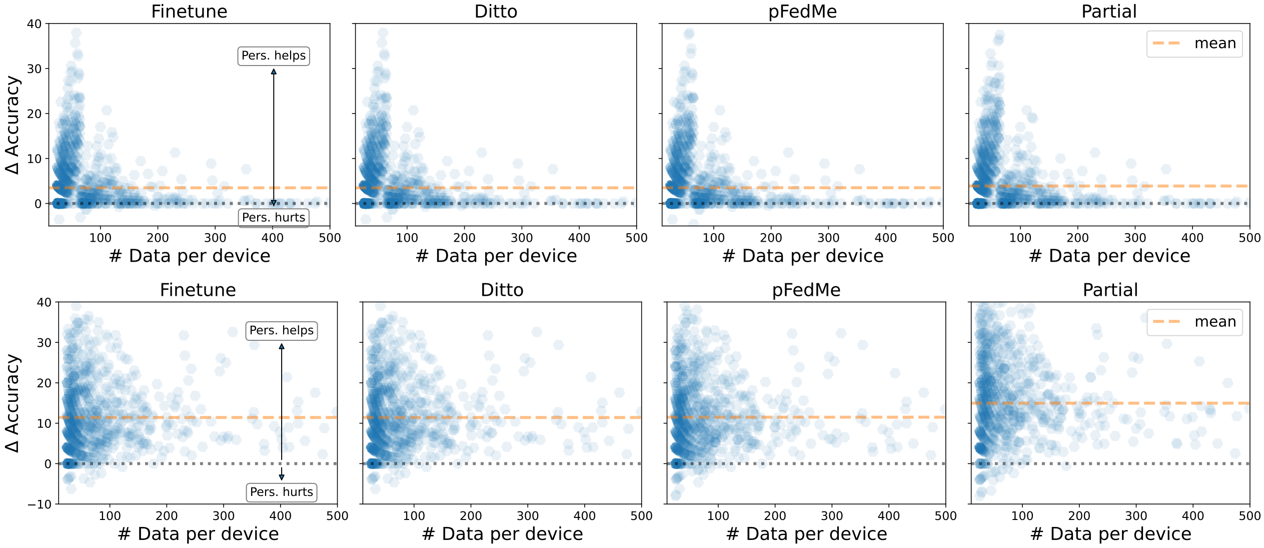


Figure 9. Scatter plot of change in accuracy (pp) per-device versus the number of training samples on the device for GLDv2. **Top**: Training accuracy. **Bottom**: Test accuracy.

Table 14. The change in accuracy (percentage points) from the final finetuning for FedAlt and FedSim with stateless devices. The subscript denotes the standard deviation over 5 random seeds.

	StackOverflow		GLDv2		EMNIST	
	FedAlt	FedSim	FedAlt	FedSim	FedAlt	FedSim
Input Layer	$0.86_{0.03}$	$1.00_{0.02}$	$0.44_{0.03}$	$0.42_{0.03}$	$0.11_{0.02}$	$0.10_{0.04}$
Output Layer	$1.08_{0.03}$	$1.10_{0.02}$	$1.47_{0.04}$	$1.46_{0.05}$	$0.15_{0.02}$	$0.11_{0.02}$
Adapter	$0.84_{0.04}$	$0.88_{0.02}$	$11.68_{0.20}$	$10.42_{0.09}$	$0.46_{0.02}$	$0.42_{0.04}$

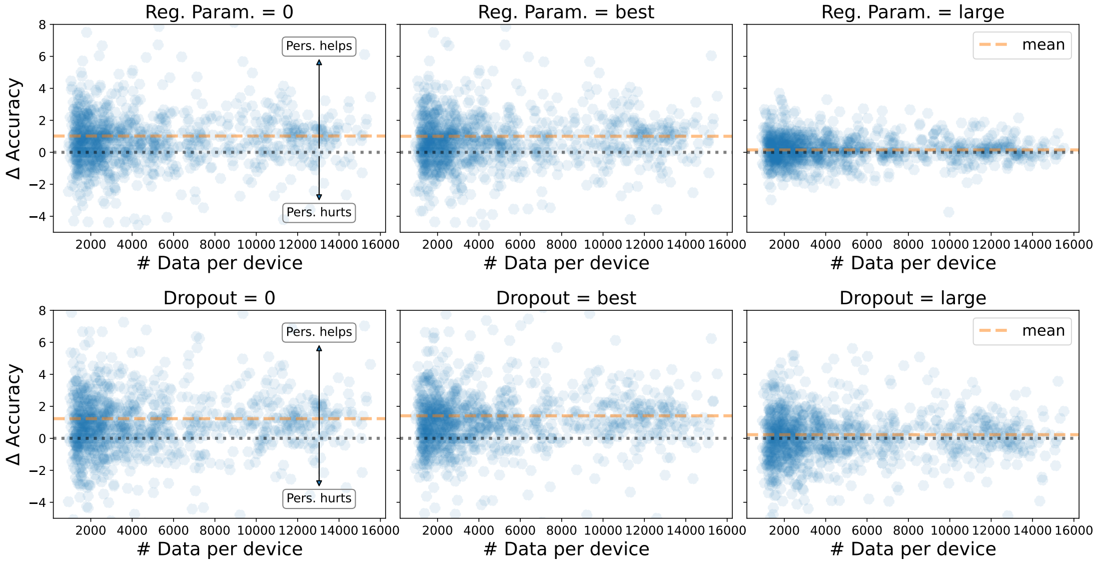


Figure 10. Scatter plot of change in accuracy (pp) per-device versus the number of training samples on the device with the effect of regularization. **Top**: ℓ_2 regularization a.k.a. weight decay. **Bottom**: dropout. The “best” values of the ℓ_2 regularization parameter and dropout are chosen to maximize the average test accuracy across all devices.

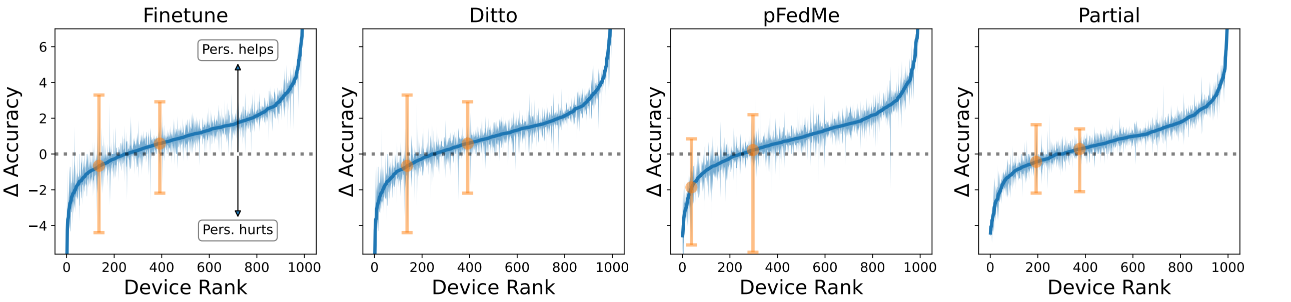


Figure 11. Change in per-device accuracy (pp) due to personalization. The solid line is the mean over 5 random runs and the shaded area denotes the max/min across these runs. The devices are sorted in ascending order of accuracy change. The points in orange depict two example devices who might either be helped or harmed by personalization depending on the random seed.