TAMING PRECONDITIONER DRIFT: UNLOCKING THE POTENTIAL OF SECOND-ORDER OPTIMIZERS FOR FEDERATED LEARNING ON NON-IID DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Second-order optimizers can significantly accelerate large-scale training, yet their naive federated variants are often unstable or even diverge on non-IID data. We show that a key culprit is *preconditioner drift*: client-side second-order training induces heterogeneous *curvature-defined geometries* (i.e., preconditioner coordinate systems), and server-side model averaging updates computed under incompatible metrics, corrupting the global descent direction. To address this geometric mismatch, we propose `FedPAC`, a *preconditioner alignment and correction* framework for reliable federated second-order optimization. `FedPAC` explicitly decouples parameter aggregation from geometry synchronization by: (i) **Alignment** (i.e.,aggregating local preconditioners into a global reference and warm-starting clients via global preconditioner); and (ii) **Correction** (i.e., steering local preconditioned updates using a global preconditioned direction to suppress long-term drift). We establish a drift-dependent non-convex convergence bound for an idealized preconditioned-gradient formulation of `FedPAC`, quantifying the interaction between preconditioner drift and data heterogeneity. Empirically, `FedPAC` consistently improves stability and accuracy across vision and language tasks.

1 INTRODUCTION

In recent years, the training paradigm of large-scale models, especially large language models (LLMs) Devlin et al. (2019), has shifted significantly. It is increasingly recognized that the optimizer itself is part of the computational budget: under a fixed compute budget, better optimization methods can directly yield shorter training time and higher model quality. Second-order optimizers, such as Sophia Liu et al. (2024), SOAP Vyas et al. (2025), and Muon Jordan et al., leverage richer curvature information and have demonstrated substantial acceleration in centralized large-scale training, achieving 1.5–2× faster convergence over AdamW and SGD Abreu et al. (2025); Loshchilov et al. (2017). This evidence suggests that second-order optimization is no longer merely a theoretical luxury, but is rapidly becoming a practical backbone of large-scale deep learning.

In contrast, another highly practical and increasingly important training paradigm, Federated Learning (FL) McMahan et al. (2017), has remained largely confined to the first-order regime in terms of optimization. Most widely adopted federated algorithms, including FedAvg McMahan et al. (2017), and various variance-reduced variants, rely on first-order SGD updates Bottou (2010). Although these methods partially mitigate data heterogeneity and communication constraints, they continue to suffer from evident limitations in convergence speed, scalability to large models. This discrepancy naturally raises the following question:

If second-order optimizers have already demonstrated their effectiveness in centralized large-scale training, why can we not achieve similar acceleration benefits in FL?

A natural baseline is to employ Sophia/SOAP/Muon as the local optimizer on each client and aggregate parameters as in FedAvg named Local Sophia/SOAP/Muon, as shown in Figure 2. Yet, on non-IID data, such a naive federated adaptation exhibits a critical failure mode: it fails to deliver the expected acceleration and often results in slower convergence or even divergence as shown in Figure 2. Our experiments and mechanistic analysis indicate that this behavior is not mainly due to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the computational cost of second-order information or insufficient client-side compute. Instead, it is driven by a previously underexamined phenomenon, which we refer to as *preconditioner drift*.

In centralized training, second-order preconditioners (e.g., the matrix preconditioners implicit curvature structures exploited by SOAP and Muon) are continuously updated under a single data distribution. In federated learning, however, each client is exposed to its own non-IID data distribution. As clients perform multiple local updates, their preconditioners adapt toward the local geometry associated with their respective data distributions, which leads to the following effects as in Figure 1: (*i*) The preconditioners across different clients gradually drift apart in both scale and orientation; (*ii*) The server aggregates only the model parameters via simple averaging, implicitly assuming that all clients employ similar preconditioners; (*iii*) The resulting global update direction effectively combines inconsistent local curvature estimates to update a shared model, which severely distorts global optimization.

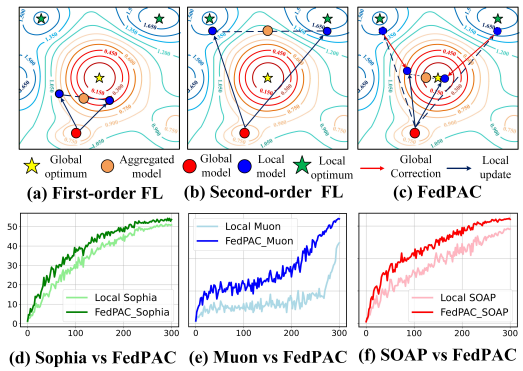


Figure 1: Illustration of preconditioner drift in non-IID FL. (a) First-order methods converge slowly and induce limited client drift. (b) Second-order methods drift toward local optima, causing aggregated global deviation. (c) `FedPAC` corrects this drift through local-global alignment. (d–f) `FedPAC` consistently improves Sophia, Muon, and SOAP on ViT-Tiny over CIFAR-100.

From an intuitive perspective, each client makes effective progress under its own curvature-adapted coordinate system; however, when these trajectories are naively aggregated, the resulting global optimization can become highly inefficient as in Figure 1. Consistent with this intuition, we observe that in non-IID data settings, directly applying second-order optimizers not only fails to replicate the acceleration achieved in centralized training, but may even perform worse than simple first-order methods as in Figure 2. This observation gives rise to an urgent and natural question:

Do second-order optimizers still hold genuine potential in data heterogeneous federated learning?

We answer this question affirmatively. We demonstrate that second-order optimizers can achieve substantial acceleration in FL when equipped with a dedicated framework to explicitly mitigate *preconditioner drift* as in Figure 1. Accordingly, we propose `FedPAC` (**Fe**derate**d** **P**reconditioner **A**lignment and **C**orrection), a principled correction and acceleration framework for second-order federated optimization.

In summary, **main contributions** are as follows:

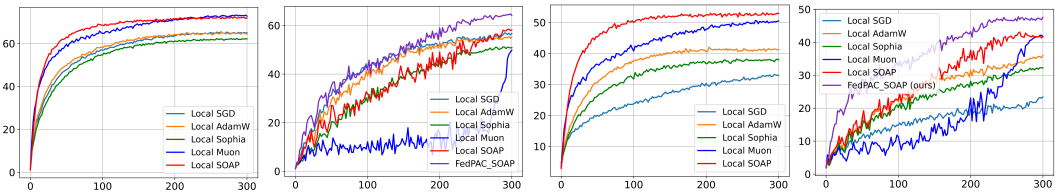
• We identify a previously underexplored failure mode of second-order federated learning: *preconditioner drift*, the mismatch of client-side curvature-induced geometries. We formalize it with an explicit drift metric and empirically show it strongly correlates with degraded convergence.

• We develop a unified framework, `FedPAC`, that targets this geometric mismatch by preconditioner synchronization: clients perform efficient local second-order updates, while preconditioners are periodically *aligned* and local steps are *corrected* using lightweight global preconditioner.

• We establish non-convex convergence guarantees that quantify the coupling between preconditioner drift and data heterogeneity through $\bar{D}_P \sigma_g^2$. Our analysis shows that `FedPAC` achieves a vanishing stationarity bound with suitable stepsizes when the average preconditioner drift vanishes, and an $O(R^{-1/2})$ rate when this drift decays as $O(R^{-1/2})$.

2 Related Work

• **First-order FL under data heterogeneity** In heterogeneous federated learning, most improvements of first-order methods target mitigating client drift and convergence instability caused by non-IID data. FedProx Li et al. (2020) stabilizes local updates via a proximal regularization term, while SCAFFOLD Karimireddy et al. (2020) theoretically analyzes client drift in FedAvg and corrects it using server–client control variates. FedCM Xu et al. (2021) leverages client momentum to stabilize updates. Nevertheless, these methods remain fundamentally first-order, operating solely at the gradient level without explicitly leveraging second-order geometry such as the Hessian or



(a) ResNet-18, IID (b) ResNet-18, non-IID (c) ViT-Tiny, IID (d) ViT-Tiny, non-IID

Figure 2: The x-axis denotes communication rounds, and the y-axis is test accuracy. (a, c): In FL on IID data, second-order optimizers converge significantly faster than SGD and AdamW for training ResNet and Transformer. (b),(d): However, on non-IID data, second-order optimizer (e.g., Local Muon) converges much more slowly and can even underperform first-order methods such as Local SGD.

Gauss–Newton information, leaving considerable room to explore the potential of second-order optimization in heterogeneous federated learning. Our alignment operates on the *preconditioner operators* that define the local metric, not on first-order states such as momentum or control variates.

• **Second-order optimization methods** Traditional second-order methods such as Newton's method are often prohibitively expensive. Recently, several practical second-order or quasi-second-order optimizers have been shown to scale to large models in centralized deep learning, including Shampoo Gupta et al. (2018), SOAP Vyas et al. (2025), Muon Jordan et al., and Sophia Liu et al. (2024). Shampoo Gupta et al. (2018) performs structured gradient preconditioning via Kronecker-factored approximations of the Hessian. SOAP Vyas et al. (2025) stabilizes such preconditioning by running AdamW in the induced feature basis. Muon Jordan et al. orthogonalizes gradient momentum using Newton–Schulz iterations. Sophia Liu et al. (2024) leverages lightweight diagonal Hessian estimates.

• **Second-order optimization in FL** The utilization of second-order information in FL is still in its early stages but is developing rapidly. Fed-Sophia Elbakary et al. (2024) adapts the Sophia optimizer to the federated setting and proposes a communication-efficient second-order federated algorithm. FedPM Ishii et al. (2026) systematically studies the empirical behavior of the second-order classical Newton method in FL, which is computational overhead and not suitable for large-scale models.

3 The Proposed Unified Algorithm

3.1 Problem Setup

FL seeks to learn a global model collaboratively over clients by minimizing the population risk:

$$f_i(\boldsymbol{x}) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i}\big[F_i(\boldsymbol{x}; \xi_i)\big], \qquad f(\boldsymbol{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{x}). \tag{1}$$

The function F_i is the loss function on client i. $\mathbb{E}_{\xi_i \sim \mathcal{D}_i}[\cdot]$ denotes conditional expectation with respect to the sample ξ_i. N is the number of clients, and $\boldsymbol{x}$ is global model.

3.2 Unified Abstraction of Second-Order FL

To investigate the mechanisms of second-order optimizers in FL, we first propose a generalized *Federated Second-Order Algorithm* (FedSOA), as shown in Algorithm 1.

At the k-th local step of round r, client i samples a mini-batch $\xi_i^{r,k} \sim \mathcal{D}_i$ and computes a stochastic gradient $\boldsymbol{g}_i^{r,k} = \nabla F_i(\boldsymbol{x}_i^{r,k}; \xi_i^{r,k})$. Second-order optimizers maintain an internal *preconditioner* $P_i^{r,k}$, which parameterizes a preconditioning operator that maps the gradient to an update direction. To unify different optimizers, we define a generic operator $P(\cdot)$ and write the local update as follows:

$$\boldsymbol{x}_i^{r,k+1} = \boldsymbol{x}_i^{r,k} - \eta_l \, P_i^{r,k}\big(\boldsymbol{g}_i^{r,k}\big), \tag{2}$$

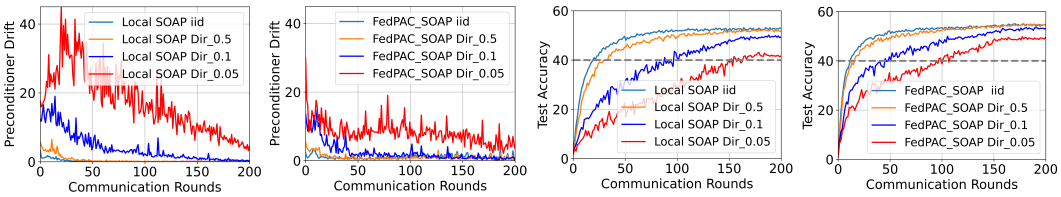
where η_l denotes the local learning rate. The specific form of P depends on the underlying optimizer. For example, SOAP applies structured preconditioning in a Kronecker-factored basis, Sophia rescales gradients using diagonal Hessian estimates, and Muon updates gradients in an orthogonalized rule. After each step, the preconditioner state is updated according to the optimizer rule:

$$P_i^{r,k+1} = \mathrm{Update_Preconditioner}\big(P_i^{r,k}\big). \tag{3}$$

Algorithm 1 Federated Second-Order Algorithm (FedSOA)

Require: Learning rate η_l, communication rounds R, local update steps K, number of clients N

Initialize global model $\boldsymbol{x}^0$
for $r = 0, \ldots, R-1$ **do**
  Server samples a subset of clients $\mathcal{S}_r$
  for each client $i \in \mathcal{S}_r$ in parallel **do**
    $\boldsymbol{x}_i^{r,0} \leftarrow \boldsymbol{x}^r$;
    for $k = 0, \ldots, K-1$ **do**
      Sample mini-batch $B_i^{r,k}$; $\boldsymbol{g}_i^{r,k} \leftarrow \nabla F_i(\boldsymbol{x}_i^{r,k}; B_i^{r,k})$
      $P_i^{r,k+1} \leftarrow \text{Update_Preconditioner}\big(P_i^{r,k}\big)$
      $\tilde{\boldsymbol{g}}_i^{r,k} \leftarrow P_i^{r,k}\big(\boldsymbol{g}_i^{r,k}\big)$; $\boldsymbol{x}_i^{r,k+1} \leftarrow \boldsymbol{x}_i^{r,k} - \eta_l \tilde{\boldsymbol{g}}_i^{r,k}$
    end for
    Client i sends $\Delta \boldsymbol{x}_i^r \leftarrow \boldsymbol{x}_i^{r,K} - \boldsymbol{x}^r$ to the server
  end for
  $\boldsymbol{x}^{r+1} \leftarrow \boldsymbol{x}^r + \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} \Delta \boldsymbol{x}_i^r$
end for



(a) Local SOAP (b) FedPAC_SOAP (c) Local SOAP (d) FedPAC_SOAP

Figure 3: Performance and Preconditioner Drift of Local SOAP, `FedPAC_SOAP` on ViT-Tiny with CIFAR-100. For SOAP, we compute the preconditioner drift in a layer-wise manner by measuring the spectral norm of the difference between each client's left/right preconditioner and the aggregated global preconditioner. Our `FedPAC_SOAP` substantially reduces the preconditioner drift of Local SOAP and accelerates convergence in (a,b), reaching 40% test accuracy in fewer rounds, as shown in (c) and (d).

After K local steps, client i computes the model difference $\Delta \boldsymbol{x}_i^r := \boldsymbol{x}_i^{r,K} - \boldsymbol{x}^r$. A naive second-order federated baseline aggregates these updates in the same manner as FedAvg: $\boldsymbol{x}^{r+1} = \boldsymbol{x}^r + \sum_{i \in \mathcal{S}_r} \frac{1}{|\mathcal{S}_r|} \Delta \boldsymbol{x}_i^r$. However, under strong statistical heterogeneity and multiple local steps, this naive aggregation leads to *preconditioner drift*, which significantly degrades global convergence.

3.3 Local Preconditioner Drift

To characterize the geometric deviation of second-order optimization in federated environments, we consider an idealized centralized second-order optimizer that constructs a global preconditioner P^r on the joint data distribution $\mathcal{D} = \frac{1}{N} \sum_{i=1}^N \mathcal{D}_i$ and performs the update:

$$\boldsymbol{x}^{r+1} = \boldsymbol{x}^r - \eta\, P^r\big(\nabla F(\boldsymbol{x}^r)\big). \tag{4}$$

In this idealized setting, the model always evolves under a unified second-order geometry characterized by P^r. In FL setting, each client i locally updates its preconditioner state $P_i^{r,k}$ based on its own data distribution $\mathcal{D}_i$, which in turn induces a local preconditioner $P_i^{r,k}$. As k and r increase, these local preconditioners progressively *drift* across clients. We formally define the **preconditioner drift** $\boldsymbol{\Delta}_{\mathcal{D}}$, as the difference gap of the preconditioner drift between the global and local preconditioners.

Definition 1 ***(Preconditioner drift).*** *The round-wise $\boldsymbol{\Delta}_{\mathcal{D}}$ of the global preconditioner P^r and the local preconditioners $\left\{P_i^{r,k}\right\}_{i=1,k=1}^{|\mathcal{S}_r|,K}$ is defined as: $\boldsymbol{\Delta}_{\mathcal{D}} = \frac{1}{|\mathcal{S}_r|K} \sum_{i \in \mathcal{S}_r} \sum_{k=1}^K \|P_i^{r,k} - P^r\|^2$.*

Preconditioner drift is a client-level phenomenon: different clients' preconditioners $P_i^{r,k}$ gradually adapt toward their respective local geometries. As shown in Figure 3 (a,b), we observe that as data heterogeneity increases, the preconditioner drift $\boldsymbol{\Delta}_{\mathcal{D}}$ becomes more severe, and the algorithm con-

Algorithm 2 Federated Preconditioner Alignment and Correction Framework (FedPAC)

Require: Local stepsize η_l, server stepsize γ, communication rounds R, local update steps K, number of clients N, mixing coefficient β

1: Initialize global model $\boldsymbol{x}^0$, global preconditioner P^0, and global direction $\boldsymbol{g}_G^0$
2: **for** $r = 0, \ldots, R-1$ **do**
3: Server samples a subset of clients $\mathcal{S}_r$
4: **for** each client $i \in \mathcal{S}_r$ in parallel **do**
5: $\boldsymbol{x}_i^{r,0} \leftarrow \boldsymbol{x}^r$; $P_i^{r,0} \leftarrow P^r$
6: **for** $k = 0, \ldots, K-1$ **do**
7: Sample mini-batch $B_i^{r,k}$; $\boldsymbol{g}_i^{r,k} \leftarrow \nabla F_i(\boldsymbol{x}_i^{r,k}; B_i^{r,k})$
8: $P_i^{r,k+1} \leftarrow \text{Update_Preconditioner}\big(P_i^{r,k}\big)$; $\tilde{\boldsymbol{g}}_i^{r,k} \leftarrow P_i^{r,k}\big(\boldsymbol{g}_i^{r,k}\big)$
9: $\boldsymbol{x}_i^{r,k+1} \leftarrow \boldsymbol{x}_i^{r,k} - \eta_l\big[(1-\beta)\tilde{\boldsymbol{g}}_i^{r,k} + \beta \boldsymbol{g}_G^r\big]$
10: **end for**
11: Client i sends $(\Delta \boldsymbol{x}_i^r \leftarrow \boldsymbol{x}_i^{r,K} - \boldsymbol{x}^r, P_i^{r,K})$ to the server
12: **end for**
13: $\boldsymbol{g}_G^{r+1} \leftarrow -\frac{1}{|\mathcal{S}_r| K \eta_l} \sum_{i \in \mathcal{S}_r} \big(\boldsymbol{x}_i^{r,K} - \boldsymbol{x}_i^{r,0}\big)$
14: $\boldsymbol{x}^{r+1} \leftarrow \boldsymbol{x}^r - \gamma \boldsymbol{g}_G^{r+1}$; $P^{r+1} \leftarrow \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} P_i^{r,K}$
15: **end for**

verges increasingly slowly. Even at a shared model x, heterogeneous preconditioning generally yields $\frac{1}{N}\sum_{i=1}^N P_i \nabla f_i(x) \neq \left(\frac{1}{N}\sum_{i=1}^N P_i\right) \nabla f(x)$, and this discrepancy can reappear after alignment.

4 A Federated Preconditioner Alignment and Correction Framework

To address these issues, we propose a federated preconditioner alignment and correction framework (`FedPAC`) for general second-order optimizers, consisting of two stages.

(1) Aggregation of Preconditioner States (Alignment) At round r, client i uploads $\Delta \boldsymbol{x}_i^r$ and its local preconditioner $P_i^{r,K}$; server then aggregates the preconditioners in Algorithm 2 (i.e., Lines 14):

$$P^{r+1} \;=\; \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} P_i^{r,K}. \tag{5}$$

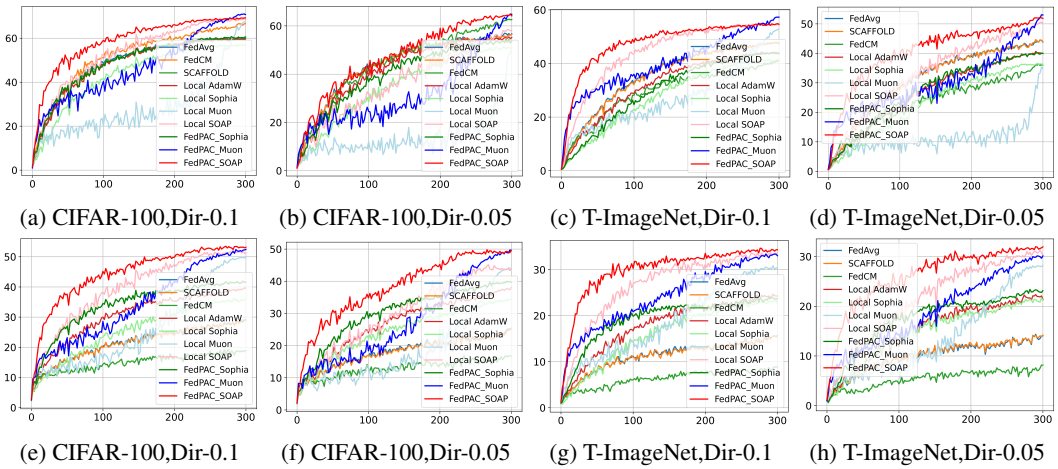
At the beginning of round $r+1$, the server broadcasts $(\boldsymbol{x}^{r+1}, P^{r+1})$ to the participating clients. Upon receiving them, client i aligns its local preconditioner state with the global reference P^r to obtain the aligned preconditioner used for the new round: $P_i^{r,0} \leftarrow P^r$ (i.e., Lines 5). The client then uses $P_i^{r,0}$ as the initial preconditioner state for round $r+1$ and performs the local second-order updates according to equation 2. This mechanism preserves local second-order adaptivity while periodically pulling the optimization geometry back toward a shared reference.

(2) Local Preconditioner Correction (Correction) On each client, we correct the preconditioner drift, as follows:

$$\boldsymbol{x}_i^{r,k+1} = \boldsymbol{x}_i^{r,k} - \eta_l[(1-\beta)\tilde{\boldsymbol{g}}_i^{r,k} + \beta \boldsymbol{g}_G^r], \tag{6}$$

where $\boldsymbol{g}_G^r = -\frac{1}{|\mathcal{S}_{r-1}| K \eta_l} \sum_{i \in \mathcal{S}_{r-1}} \big(\boldsymbol{x}_i^{r-1,K} - \boldsymbol{x}_i^{r-1,0}\big)$ is the global correction direction estimated from the *previous* communication round. In particular, since $\boldsymbol{x}_i^{r-1,K} - \boldsymbol{x}_i^{r-1,0}$ is the cumulative local model displacement over round $r-1$, $\boldsymbol{g}_G^r$ can be viewed as the averaged update direction induced by the participating clients in the previous round. The mixing coefficient $\beta \in [0, 1)$ controls the trade-off between the local preconditioned direction $\tilde{\boldsymbol{g}}_i^{r,k}$ and the lagged global correction direction $\boldsymbol{g}_G^r$in Figure 1 and Algorithm 2 (Lines 9 & 13).

Overall, the above steps constitute a unified correction and acceleration framework for second-order federated optimization. The underlying local optimizer can be any structured second-order method, such as Sophia, Muon, SOAP. In this work, we instantiate our framework with three representative second-order optimizers, yielding `FedPAC_Sophia`, `FedPAC_Muon`, and `FedPAC_SOAP`.



(a) CIFAR-100,Dir-0.1 (b) CIFAR-100,Dir-0.05 (c) T-ImageNet,Dir-0.1 (d) T-ImageNet,Dir-0.05

(e) CIFAR-100,Dir-0.1 (f) CIFAR-100,Dir-0.05 (g) T-ImageNet,Dir-0.1 (h) T-ImageNet,Dir-0.05

Figure 4: Test accuracy vs. communication rounds. The x-axis shows communication rounds and the y-axis shows test accuracy. Panels (a–d) use **ResNet-18**, and panels (e–h) use **ViT-Tiny**.

Table 1: Test accuracy (%) on CIFAR-100 and Tiny-ImageNet using **ResNet-18** and **ViT-Tiny** over 300 communication rounds under Dir-0.1 and Dir-0.05. Experiments are conducted with 100 clients, 10% participation, batch size 50, and $K = 50$. Best results are highlighted in **bold**, and second-best results are underlined.

Method	ResNet-18				ViT-Tiny			
	CIFAR-100		Tiny-ImageNet		CIFAR-100		Tiny-ImageNet	
	Dir-0.1	Dir-0.05	Dir-0.1	Dir-0.05	Dir-0.1	Dir-0.05	Dir-0.1	Dir-0.05
FedAvg	60.17	56.75	47.48	43.80	27.24	23.42	15.68	14.05
SCAFFOLD	60.69	56.43	47.76	43.92	26.86	23.23	15.70	14.21
FedCM	66.61	62.65	41.16	36.00	28.23	25.74	18.88	18.15
Local AdamW	59.23	55.24	44.01	40.00	37.57	36.06	24.31	21.35
Local Sophia	56.65	50.89	41.23	36.15	34.05	32.25	22.49	21.14
FedPAC_Sophia	59.96	53.66	43.81	36.37	39.79	32.71	23.01	22.37
Local Muon	67.26	49.86	52.83	34.76	44.00	39.68	30.51	28.25
FedPAC_Muon	**71.85**	**65.56**	**57.95**	**54.00**	47.81	41.76	31.45	30.25
Local SOAP	68.44	58.16	54.42	50.02	49.41	41.68	33.30	30.36
FedPAC_SOAP	69.25	64.16	55.62	51.81	**51.16**	**47.55**	**34.32**	**31.33**

5 Theoretical Analysis

In this section, we give the convergence theoretical analysis of our proposed FedPAC algorithm. Firstly, we state some standard assumptions for the non-convex function f. These assumptions are standard in FL and Second-Order optimization (Karimireddy et al., 2020; Ishii et al., 2026).

Assumption 5.1 (Smoothness (Karimireddy et al., 2020)). *The non-convex f_i is a L-smooth function for all $i \in [N]$, i.e., $\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$.*

Assumption 5.2 (Bounded Stochastic Gradient (Karimireddy et al., 2020)). *$\boldsymbol{g}_i^r = \nabla f_i(\boldsymbol{x}_i^r, \xi_i^r)$ computed by using a sampled mini-batch data ξ_i^r in the local client i is an unbiased estimator of ∇f_i with bounded variance, i.e., $\mathbb{E}_{\xi_i^r}[\boldsymbol{g}_i^r] = \nabla f_i(\boldsymbol{x}_i^r)$ and $\mathbb{E}_{\xi_i^r}\|\boldsymbol{g}_i^r - \nabla f_i(\boldsymbol{x}_i^r)\|^2 \leq \sigma^2$, for all $\boldsymbol{x}_i^r \in \mathbb{R}^d$.*

Assumption 5.3 (Bounded Heterogeneity (Karimireddy et al., 2020)). *The dissimilarity between local clients is bounded on the gradients, i.e., $\frac{1}{N}\sum_{i=1}^N \|\nabla f_i(x) - \nabla f(x)\|^2 \leq \sigma_g^2$, for all $\boldsymbol{x} \in \mathbb{R}^d$.*

Assumption 5.4 (Bounded Preconditioners (Ishii et al., 2026)). There exist constants $0 < m \leq M$ such that $mI \preceq P_i^{r,k} \preceq MI, \forall i, r, k$. Moreover, the global preconditioner satisfies $mI \preceq P^r \preceq MI, \forall r \geq 0$.

Assumption 5.5 (preconditioner-drift bound Ishii et al. (2026)). For each round r, there exists a deterministic quantity $\bar{D}_P \geq 0$ such that almost surely, $\frac{1}{|\mathcal{S}_r|K}\sum_{i\in\mathcal{S}_r}\sum_{k=1}^K \|P_i^{r,k} - P^r\|^2 \leq \bar{D}_P$.

Notation. Let $u = \eta_l K$, $\Delta = f(x^0) - f_*$, and $\mathcal{G}_R := \frac{1}{R}\sum_{r=0}^{R-1}\mathbb{E}\|\nabla f(x^r)\|^2, \overline{D}_R := \frac{1}{R}\sum_{r=0}^{R-1} D_r$. Define $V := M^2\left(\frac{\sigma^2}{SK} + \rho_S\sigma_g^2\right), B := \sigma^2 + \sigma_g^2, B_\chi := \sigma_g^2 + \chi\sigma^2$, where $\rho_S = (N-S)/(S(N-1))$ for $N > 1$ and $\rho_S = 0$ for $N = 1$. Here $\chi = 0$ for predictable preconditioners and $\chi = 1$ otherwise. The notation $\lesssim$ hides only universal numerical constants.

Theorem 5.6 (Convergence of FedSOA). *Under Assumptions 5.1–5.5, if* $0 < u \le m/(24LM^2)$,

$$\mathcal{G}_R \lesssim \frac{\Delta}{muR} + \frac{Lu}{m}V + \frac{M^4L^2u^2}{m^2}B + \boxed{\frac{\overline{D}_R}{m^2}B_\chi}. \tag{7}$$

Theorem 5.7 (Convergence of `FedPAC`). *Under Assumptions 5.1–5.5, initialize* $g_G^0 = 0$ *and let* $\beta \in [0,1)$. *If* $0 < u \le \frac{m}{96LM^2}, 0 < \gamma \le \frac{m}{6LM^2}, \frac{\gamma\beta}{1-\beta} \le \frac{m}{16LM^2}$, *then*

$$\mathcal{G}_R \lesssim \frac{\Delta}{m\gamma R} + \frac{L\gamma}{m}V + \frac{M^2L^2\gamma^2}{m^2}\left(\frac{\beta}{1-\beta}\right)^2 V + \frac{M^4L^2u^2}{m^2}B + \frac{\overline{D}_R}{m^2}B_\chi.$$

Convergence rate and scope. Our analysis concerns an idealized preconditioned stochastic-gradient model in which the applied preconditioning operators are linear, symmetric positive definite, and uniformly bounded as $mI \preceq P_i^{r,k} \preceq MI$. For fixed problem parameters and $\beta < 1$, choose $u = \Theta(R^{-1/2})$ for FedSOA and $u, \gamma = \Theta(R^{-1/2})$ for `FedPAC`, subject to the respective stepsize conditions. Both methods satisfy $\mathcal{G}_R = O(R^{-1/2} + R^{-1}) + O\big(\overline{D}_R B_\chi/m^2\big)$. Thus, the additional condition $\overline{D}_R = O(R^{-1/2})$ yields an $O(R^{-1/2})$ expected stationarity guarantee. For predictable preconditioners, the drift contribution is proportional to $\overline{D}_R\sigma_g^2$. Alignment can tighten the bound insofar as it reduces $\overline{D}_R$ in Figure 3, but bounded drift alone does not guarantee a vanishing stationarity error. These results characterize drift-dependent convergence within the idealized model; they do not automatically cover the full SOAP, Sophia, or Muon implementations, whose momentum, clipping, or nonlinear transformations require additional analysis. `FedPAC`'s round-wise state synchronization is designed to reduce drift relative to FedSOA, as supported by the lower SOAP covariance-state drift observed in Figure 3, although this measurement does not establish a smaller operator-level $\overline{D}_R$.

6 Experiments

Datasets. We evaluate `FedPAC` on both vision and language tasks. (*i*) For image classification, we use CIFAR-100 Krizhevsky et al. (2009), and Tiny-ImageNet Le & Yang (2015). (*ii*) For NLP tasks, we adopt C4 Raffel et al. (2020) dataset. To simulate data heterogeneity across clients, we follow the Dirichlet partitioning scheme Hsu et al. (2019). For Dir-α partitioning, smaller α indicates more severe data heterogeneity.

Model Architectures. We explore a variety of model types: (*i*) ResNet-18 He et al. (2016) as a representative convolutional neural network (CNN), (*ii*) Vision Transformer (ViT-Base) and ViT-Tiny Dosovitskiy et al. (2020) for Vision Transformers, and (*iii*) LLaMA Touvron et al. (2023) for large-scale language model.

Baselines. We compare our method against state-of-the-art FL algorithms: `FedAvg` (`Local SGD`) McMahan et al. (2017), `SCAFFOLD` Karimireddy et al. (2020), `FedCM` Xu et al. (2021), `Local AdamW`, `Local Sophia`, `Local Muon` and `Local SOAP`. Under our `FedPAC` framework, we instantiate three variants: `FedPAC_Sophia`, `FedPAC_Muon`, and `FedPAC_SOAP`. In the Appendix (Table), we compare additional FL algorithms designed to address data heterogeneity.

Hyperparameter Settings. For `FedAvg`, `SCAFFOLD`, `FedCM`, the lr is selected from $\{10^{-2},\ 3\times10^{-2},\ 5\times10^{-2},\ 10^{-1},\ 3\times10^{-1}\}$, and choose the best value $lr = 10^{-1}$. with a weight decay of 0.001. For `Local AdamW`, the lr is selected from $\{10^{-4},\ 3\times10^{-4},\ 5\times10^{-4},\ 8\times10^{-4},\ 10^{-3}\}$, and choose the best value $lr = 3\times10^{-4}$. We set the learning rates for `Local Sophia`, `Local Muon`, and `Local SOAP` to 3×10^{-4}, 3×10^{-2}, and 3×10^{-3}, respectively, $\gamma = \eta K$. Their weight decay is 0.01. We apply cosine learning rate decay, and set `FedPAC` to $\boldsymbol{\beta}=0.5$, weight decay 0.01. We set the learning rate of `FedPAC` variants to be same with `Local Sophia`, `Local Muon` and `Local SOAP`. Additional hyperparameter configurations are detailed in the **Appendix**. We release all code to ensure full reproducibility upon acceptance. All results are averaged over 5 runs with seeds 42, 43, 44, 45, 46. All experiments were performed on NVIDIA RTX 4090 GPU.

Table 2: Comparison of different methods on vision and language tasks. On **ViT-Base**, results are reported under the Dir-0.1 partition over 100 communication rounds with 50 clients, 10% participation, batch size 16, and $K = 50$ local steps. On **LLaMA**, we report validation loss and validation perplexity (**Val PPL**) over 100 communication rounds with 20 clients, 20% participation, batch size 16, and $K = 50$ local steps.

Method	ViT-Base				LLaMA					
	CIFAR-100		Tiny-ImageNet		LLaMA 60M		LLaMA 130M		LLaMA 350M	
	Acc. ↑	Loss ↓	Acc. ↑	Loss ↓	Loss ↓	Val PPL ↓	Loss ↓	Val PPL ↓	Loss ↓	Val PPL ↓
FedAvg	89.28	0.332	86.97	0.421	5.156	173.47	5.258	192.10	5.354	211.45
SCAFFOLD	89.35	0.328	86.85	0.432	5.028	152.63	5.238	188.29	5.365	213.79
FedCM	87.44	0.592	84.24	0.708	5.231	186.98	5.356	211.88	5.426	227.24
Local AdamW	90.10	0.281	86.65	0.252	4.164	64.33	4.056	57.74	3.798	44.61
Local Sophia	90.21	0.275	86.16	0.188	4.012	55.26	3.986	53.84	3.521	33.82
FedPAC_Sophia	90.38	0.262	86.55	0.175	3.864	47.66	3.752	42.61	3.485	32.62
Local Muon	90.32	0.189	87.22	0.184	3.556	35.02	3.456	31.69	3.365	28.93
FedPAC_Muon	90.46	0.165	87.63	0.171	3.445	31.34	3.329	27.91	3.302	27.17
Local SOAP	90.44	0.151	87.72	0.212	3.408	30.20	3.345	28.36	3.312	27.44
FedPAC_SOAP	**90.65**	**0.129**	**87.92**	**0.156**	**3.314**	**27.49**	**3.306**	**27.28**	**3.228**	**25.23**

Table 3: Impact of β on FedPAC_SOAP using **ResNet-18** and **ViT-Tiny** on CIFAR-100 under the Dir-0.05 partition. Best results are highlighted in **bold**, and second-best results are underlined.

Model / β	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ResNet-18	62.05	62.35	63.10	63.57	63.75	**64.16**	63.77	61.75	61.70	58.86
ViT-Tiny	44.56	45.28	45.63	46.22	46.84	**47.55**	46.86	46.59	46.05	45.66
D_P of ViT-Tiny ↓	17.29	15.65	13.56	11.87	10.03	8.20	7.01	5.86	4.21	2.10

6.1 Results on Convolutional Neural Networks

Training on CIFAR-100 with ResNet-18. **Table 1** and **Figure 4** present the test accuracy on CIFAR-100 and Tiny-ImageNet using ResNet-18. **Accuracy under heterogeneity.** Across both datasets, FedPAC consistently improves the corresponding local second-order optimizer, with the largest gains in highly non-IID regimes where *preconditioner drift* is severe. For example on CIFAR-100, Local Muon drops from 67.26% (Dir-0.1) to 49.86% (Dir-0.05), whereas FedPAC_Muon achieves 71.85% and 65.56%, respectively. FedPAC_SOAP also strengthens SOAP on CIFAR-100, improving from 68.44%/58.16% (Local SOAP) to 69.25%/64.16% under Dir-0.1/Dir-0.05. Similar trends hold on Tiny-ImageNet: Local Muon attains 52.83%/34.76% (Dir-0.1/Dir-0.05), while FedPAC_Muon reaches 57.95%/54.00%. Improvements for Sophia are also consistent (e.g., CIFAR-100: 56.65% → 59.96% at Dir-0.1 and 50.89% → 53.66% at Dir-0.05).

Stability and drift mitigation. As illustrated in Figure 3, FedPAC reduces preconditioner drift and yields faster, more stable convergence under strong heterogeneity, explaining why aggressive second-order methods (e.g., Muon/SOAP) remain effective in federated training.

6.2 Results on Vision Transformer

Training ViT-Tiny from scratch. Results are reported in Table 1. FedPAC yields the strongest performance among second-order variants, especially under non-IID data. On CIFAR-100, FedPAC_SOAP improves over Local_SOAP from 49.41%/41.68% to 51.16%/47.55% under Dir-0.1/0.05, respectively. On Tiny-ImageNet, FedPAC_SOAP also achieves the best overall accuracy (e.g. 31.33% at Dir-0.05), while FedPAC_Muon shows clear robustness gains under heterogeneity (Dir-0.1/0.05: 31.45%/30.25% vs. 30.51%/28.25% for Local_Muon). Overall, these results indicate that FedPAC effectively stabilizes federated ViT optimization and preserves accuracy as heterogeneity increases.

Pretrained ViT-Base. Since small-scale datasets can limit ViT performance when trained from scratch, we additionally evaluate a *pretrained ViT-Base* under Dir-0.1 (Table 2). FedPAC continues to provide consistent improvements: FedPAC_SOAP reaches 90.65% on CIFAR-100 and 87.92%

Table 4: Ablation and communication-efficient preconditioner aggregation results for `FedPAC_SOAP` on CIFAR-100 under the Dir-0.05. **Left:** test accuracy (%) after 300 communication rounds. **Right:** communication cost per round (**Comm.**, MB), computation time per round (**Comp.**, s), and test accuracy (%) on ViT-Tiny.

Ablation of `FedPAC_SOAP`			**Communication-Efficient Aggregation**			
Variant	**ResNet-18**	**ViT-Tiny**	**Aggregation**	**Acc.**	**Comm.**	**Comp.**
w/o $\bar{P}$	61.12	43.67	`Local SOAP`	41.68	22.8 (1×)	5.56
w/o Δ_G	62.05	44.56	`FedPAC_SOAP`	47.55	68.4 (3×)	5.85
`FedPAC_SOAP` (full)	64.16	47.55	`FedPAC_SOAP_light`	46.75	25.1 (1.1×)	5.91

on Tiny-ImageNet, surpassing the corresponding local second-order baselines (e.g., `Local_SOAP`: 90.44%/87.72%). These results confirm the effectiveness of `FedPAC` on federated vision Transformers across both from-scratch and pretrained settings.

6.3 C4 FEDERATED PRE-TRAINING WITH LLAMA.

Table 2 summarizes C4 federated pre-training results after 100 rounds (20 clients, 20% participation) for LLaMA 60M/130M/350M. Standard FL optimizers struggle in this heterogeneous, partial-participation regime (e.g., FedAvg: 5.156/5.258/5.354), whereas locally using modern adaptive/second-order optimizers already yields substantial improvements (e.g., Local SOAP: 3.408/3.345/3.312). `FedPAC` consistently matches or improves the strongest local baselines across all scales, with the best results achieved by `FedPAC_SOAP` on 60M/130M/350M (3.314/3.306/3.228). Compared to FedAvg, the best configuration reduces validation loss by over 1.8 absolute on 60M (5.156→3.314), and similarly large margins on larger models, highlighting the practical benefit of stabilizing second-order methods in federated settings.

6.4 ABLATION STUDY

A1: Sensitivity to β (correction strength). Table 3 reveals a trade-off between state consistency and local adaptation. As β increases, the measured SOAP state drift D_P on ViT-Tiny decreases monotonically from 17.29 to 2.10, whereas accuracy peaks at $\beta = 0.5$ for both backbones (ResNet-18: 64.16%; ViT-Tiny: 47.55%). In particular, increasing β from 0.5 to 0.9 further reduces D_P from 8.20 to 2.10, but lowers ViT-Tiny accuracy from 47.55% to 45.66%. Thus, minimizing state drift alone does not maximize predictive performance. This pattern is consistent with the update rule: larger β places more weight on the previous global direction and less on the current local direction, potentially limiting adaptation to new local information. We use $\beta = 0.5$ in all experiments.

A2: Component-wise ablation (Alignment vs. Correction). Table 4 shows that both modules are necessary. On Dir-0.05 CIFAR-100, Alignment-only and Correction-only each lower than `FedPAC_SOAP` achieves the best accuracy, indicating alignment and correction are complementary.

A3: Communication-efficient preconditioner aggregation. Table 4 studies a compressed variant, `FedPAC_SOAP_light`, which uploads the SOAP preconditioner using SVD compression. While full `FedPAC_SOAP` improves accuracy over `Local SOAP` (47.55 vs. 41.68) at the cost of 3× communication (68.4 MB/round), `FedPAC_SOAP_light` largely preserves the gain (46.75) with near-local bandwidth (25.1 MB/round, 1.1×). Notably, the computation overhead remains small across variants (5.56–5.91 s/round), indicating that SVD-based compression offers a favorable accuracy–communication trade-off.

7 CONCLUSION

We study how modern second-order optimizers behave in FL under non-IID data. We identify *preconditioner drift* as a key source of instability: after multiple local steps, clients adapt preconditioners to different local geometries, and naive aggregation across mismatched geometries distorts global updates. To mitigate this issue, we propose `FedPAC`, a *preconditioner alignment and correction* framework that aggregates and broadcasts a global reference preconditioner to align local geometry and suppress drift accumulation. We provide convergence guarantees for non-convex setting.

REFERENCES

Natalie Abreu, Nikhil Vyas, Sham Kakade, and Depen Morwani. The potential of second-order optimization for llms: A study with full gauss-newton. *arXiv preprint arXiv:2510.09378*, 2025.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pp. 177–186. Springer, 2010.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Ahmed Elbakary, Chaouki Ben Issaid, Mohammad Shehab, Karim Seddik, Tamer ElBatt, and Mehdi Bennis. Fed-sophia: A communication-efficient second-order federated learning algorithm. In *ICC 2024-IEEE International Conference on Communications*, pp. 950–955. IEEE, 2024.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Hiro Ishii, Kenta Niwa, Hiroshi Sawada, Akinori Fujino, Noboru Harada, and Rio Yokota. Fedpm: Federated learning using second-order optimization with preconditioned mixing of local parameters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22128–22136, 2026.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cecista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, 2020.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *International Conference on Learning Representations*, volume 2024, pp. 1621–1650, 2024.

Junkang Liu, Fanhua Shang, Junchao Zhou, Hongying Liu, Yuanyuan Liu, and Jin Liu. Fedmuon: Accelerating federated learning with matrix orthogonalization. *arXiv preprint arXiv:2510.27403*, 2025.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Yuki Takezawa, Anastasia Koloskova, Xiaowen Jiang, and Sebastian U. Stich. FedMuon: Federated learning with bias-corrected LMO-based optimization. *arXiv preprint arXiv:2509.26337*, 2025. doi: 10.48550/arXiv.2509.26337. URL https://arxiv.org/abs/2509.26337.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam for language modeling. In *International Conference on Learning Representations*, volume 2025, pp. 93423–93444, 2025.

Jing Xu, Sen Wang, Liwei Wang, and Andrew Chi-Chih Yao. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*, 2021.

ETHICS STATEMENT

This work studies federated second-order optimization for vision and language models using existing benchmarks, including CIFAR-100, Tiny ImageNet, and C4. Our experiments simulate federated clients through dataset partitioning rather than collecting data from real users. Although FedPAC keeps raw training data local, communicated model updates and preconditioner information may reveal information about local data. Neither federated training nor preconditioner alignment and correction alone provides a formal privacy guarantee. FedPAC does not establish differential privacy or protection against inference attacks. Models trained with this framework may inherit biases or harmful content from the underlying datasets and pretrained models. Deployment with sensitive data therefore requires additional privacy safeguards and application-specific safety evaluation.

REPRODUCIBILITY STATEMENT

The paper describes the FedPAC framework, including preconditioner aggregation and synchronization, client initialization with the global preconditioner, and local update correction using a global preconditioned direction. It states the assumptions underlying the non-convex convergence analysis and provides detailed proofs in the appendix. The main text and appendix report the benchmark datasets, model architectures, federated training settings, optimizer configurations, and learning-rate choices, including instantiations with Sophia, Muon, and SOAP. Results are averaged over five runs using seeds 42, 43, 44, 45, and 46. Experiments were conducted on NVIDIA RTX 4090 GPUs. .

AI USE STATEMENT

Large language models were used to assist with manuscript preparation, including sentence rephrasing, grammar correction, and readability improvements. The authors reviewed and revised the AI-assisted content and take full responsibility for the final manuscript, including the correctness of its theoretical arguments, experimental results, and conclusions.

APPENDIX CONTENTS

A Preliminaries for the Convergence Analysis 14
A Foundational lemmas 14
B Convergence of FedSOA 23
C Convergence of FedPAC 25
D Necessity of the Bias Term and Consequences for the Main Text 30
E Experimental Setup 33
E.1 Datasets 33
E.2 Federated Learning Configuration 33
E.3 ResNet-18 Architecture 33
E.4 ViT-Tiny Architecture 33
E.5 ViT-Base Fine-tuning 34
E.6 C4 Language-Model Pre-training 34
F Additional Experimental Results 36
F.1 Communication and Computation Cost Analysis 36
F.2 Comparisons with Additional Baselines 37
F.3 IID and Heterogeneity Comparisons 38
G Comparison with Related Federated Methods 39
H Optimizer Instantiations 40
H.1 SOAP and FedPAC_SOAP 40
H.2 Muon and FedPAC_Muon 43

A PRELIMINARIES FOR THE CONVERGENCE ANALYSIS

We analyze the preconditioned stochastic-gradient formulations of FedSOA and FedPAC. The proof is organized into elementary sampling identities, a decomposition of the update direction, local-trajectory estimates, a descent lemma, and a final summation argument. All numerical constants are explicit, and all expectations, boundary terms, and absorption steps are detailed below. In particular, systematic preconditioning bias is kept separate from zero-mean stochastic noise.

Notation and algorithmic conventions. Let $f(x) = N^{-1}\sum_{i=1}^{N} f_i(x)$, let $R \geq 1$ be the number of rounds, and let $K \geq 1$ and $1 \leq S \leq N$ be the local-step and participation counts. Vector norms are Euclidean, and matrix norms are induced operator norms. Write $\eta > 0$ for the local stepsize, $\gamma > 0$ for the server stepsize, and $\beta \in [0, 1)$ for the weight on the previous global direction. We set $\eta = \eta_l$, following the supplied FedPAC algorithm; the local-gradient weight is $1 - \beta$. Stepsizes are constant within one run, but may depend on its horizon R. Define

$$\rho_S = \begin{cases} (N-S)/(S(N-1)), & N > 1, \\ 0, & N = 1, \end{cases} \qquad u = \eta K, \qquad a = \frac{\beta}{1-\beta}. \tag{8}$$

In particular, $\rho_S = 0$ under full participation.

For both algorithms, $x_i^{r,0} = x^r$ and $g_i^{r,k} = \nabla f_i(x_i^{r,k}) + \varepsilon_i^{r,k}$. The matrix $P_i^{r,k}$ always denotes the operator *actually applied* to $g_i^{r,k}$ at step k, irrespective of an implementation's indexing. The FedSOA updates are

$$x_i^{r,k+1} = x_i^{r,k} - \eta P_i^{r,k} g_i^{r,k}, \qquad x^{r+1} = \frac{1}{S}\sum_{i\in\mathcal{S}_r} x_i^{r,K} = x^r - uq^r, \tag{9}$$

whereas FedPAC uses $h^0 = 0$ and

$$x_i^{r,k+1} = x_i^{r,k} - \eta\big[(1-\beta)P_i^{r,k} g_i^{r,k} + \beta h^r\big], \tag{10}$$

$$q^r = \frac{1}{SK}\sum_{i\in\mathcal{S}_r}\sum_{k=0}^{K-1} P_i^{r,k} g_i^{r,k}, \tag{11}$$

$$h^{r+1} = \frac{1}{SK\eta}\sum_{i\in\mathcal{S}_r}(x^r - x_i^{r,K}) = \beta h^r + (1-\beta)q^r, \tag{12}$$

$$x^{r+1} = x^r - \gamma h^{r+1}. \tag{13}$$

Thus h^r is precisely the direction denoted by g_G^r in the supplied FedPAC algorithm. If the server averages the returned models, its stepsize is $\gamma = \eta K = u$; a separately specified γ gives equation 13 instead. FedPAC initializes the local preconditioner state from P^r and sets

$$P^{r+1} = \frac{1}{S}\sum_{i\in\mathcal{S}_r} P_i^{r,K}. \tag{14}$$

Here $P_i^{r,K}$ is the final state to be transmitted. The last applied operator is $P_i^{r,K-1}$ if lagged operators are used. These two quantities must not be interchanged. No estimate of $P^{r+1} - P^r$ is needed below. For a FedSOA implementation without state aggregation, P^r below can instead be any reference matrix measurable before round-r sampling and satisfying the stated bounds; the FedSOA proof does not add a state-upload step.

Assumption A.1 (Smoothness and lower boundedness)**.** Each f_i is differentiable and L-smooth, with $L > 0$: $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$ for all x, y. The objective satisfies $f(x) \geq f_* > -\infty$ for all x. The initial model is deterministic and $\Delta = f(x^0) - f_* < \infty$.

Assumption A.2 (Sampling and stochastic gradients)**.** Let $\mathcal{F}_r$ contain all information available before sampling $\mathcal{S}_r$, including x^r, P^r, h^r. Conditional on $\mathcal{F}_r$, $\mathcal{S}_r$ is uniform among size-S subsets of $[N]$. Reveal the fresh gradient samples within the round in an order consistent with each client's local trajectory. If $\mathcal{H}_{r,i,k}$ denotes the information just before revealing the sample for (i, k), it contains $\mathcal{F}_r$, $\mathcal{S}_r$, and $x_i^{r,k}$, and

$$\mathbb{E}[\varepsilon_i^{r,k} \mid \mathcal{H}_{r,i,k}] = 0, \qquad \mathbb{E}[\|\varepsilon_i^{r,k}\|^2 \mid \mathcal{H}_{r,i,k}] \leq \sigma^2. \tag{15}$$

Fresh independent mini-batches across clients and steps are a sufficient construction. The assumption is on conditional moments, not independence of the entire sequence of trajectory-dependent noise variables.

Assumption A.3 (Gradient heterogeneity)**.** For every x, $N^{-1}\sum_i \|\nabla f_i(x) - \nabla f(x)\|^2 \le \sigma_g^2$.

Assumption A.4 (Bounded positive-definite operators)**.** All applied preconditioners and reference matrices are symmetric and satisfy, almost surely,

$$mI \preceq P_i^{r,k} \preceq MI, \qquad mI \preceq P^r \preceq MI, \qquad 0 < m \le M < \infty. \tag{16}$$

The reference P^r is $\mathcal{F}_r$-measurable. Applied operators may use the current sample. When explicitly stated, we impose the stronger *predictability condition*: $P_i^{r,k}$ is $\mathcal{H}_{r,i,k}$-measurable.

Assumption A.5 (Pathwise drift at indices $1, \dots, K$)**.** There is a deterministic constant $\bar{D}_P \ge 0$ such that, almost surely,

$$d_{r,+} := \frac{1}{SK}\sum_{i\in\mathcal{S}_r}\sum_{k=1}^{K} \|P_i^{r,k} - P^r\|^2 \le \bar{D}_P. \tag{17}$$

This is the indexing in Assumption 5.5 of the main text. For sharper, round-dependent conclusions, let $\widehat{D}_r$ be any deterministic bound satisfying $d_{r,+} \le \widehat{D}_r$ almost surely; the uniform choice $\widehat{D}_r = \bar{D}_P$ is always permitted. A decay condition on $\widehat{D}_r$ is additional and does not follow from this assumption. All such bounds are pathwise; expectation-only bounds cannot replace them in products with gradient heterogeneity or noise.

Lemma A.6 (Transfer to the indices of the applied operators). Define

$$d_{r,-} := \frac{1}{SK}\sum_{i\in\mathcal{S}_r}\sum_{k=0}^{K-1} \|P_i^{r,k} - P^r\|^2,$$

$$J_{r,0} := \frac{1}{S}\sum_{i\in\mathcal{S}_r} \|P_i^{r,0} - P^r\|^2, \qquad J_{r,K} := \frac{1}{S}\sum_{i\in\mathcal{S}_r} \|P_i^{r,K} - P^r\|^2. \tag{18}$$

Then, pathwise,

$$d_{r,-} = d_{r,+} + \frac{J_{r,0} - J_{r,K}}{K} \le \widehat{D}_r + \frac{J_{r,0}}{K}. \tag{19}$$

For FedPAC, round-start alignment gives $J_{r,0} = 0$, so one may use $d_{r,-} \le D_r := \widehat{D}_r$. For FedSOA, let C_r be a deterministic bound on $J_{r,0}$. Then one may use

$$d_{r,-} \le D_r := \widehat{D}_r + C_r/K, \qquad C_r = (M-m)^2 \text{ is always sufficient.} \tag{20}$$

Here the spectral bounds include the actually applied initial operator $P_i^{r,0}$. No reset is imposed on FedSOA.

Proof. The identity follows by canceling the common terms with indices $1, \dots, K-1$. The terminal term $J_{r,K}$ is nonnegative, giving the inequality. For FedPAC, $P_i^{r,0} = P^r$ makes every summand in $J_{r,0}$ zero. For FedSOA, bounded symmetric operators satisfy

$$-(M-m)I \preceq P_i^{r,0} - P^r \preceq (M-m)I,$$

so $\|P_i^{r,0} - P^r\| \le M - m$ and $J_{r,0} \le (M-m)^2$. All statements also hold for $K = 1$, when the common sum is empty. □

Effective drift used in the proof. Throughout the remaining proof, D_r denotes a deterministic bound on $d_{r,-}$, obtained from Lemma A.6 or another explicit pathwise estimate. Define

$$\overline{D}_R = \frac{1}{R}\sum_{r=0}^{R-1} D_r, \qquad \overline{\widehat{D}}_R = \frac{1}{R}\sum_{r=0}^{R-1} \widehat{D}_r, \qquad \overline{C}_R = \frac{1}{R}\sum_{r=0}^{R-1} C_r.$$

The default choices give $\overline{D}_R = \overline{\widehat{D}}_R$ for FedPAC and $\overline{D}_R = \overline{\widehat{D}}_R + \overline{C}_R/K$ for FedSOA. With only the uniform Assumption 5.5, these become $\bar{D}_P$ and $\bar{D}_P + (M-m)^2/K$, respectively. Every sum defining an actual update remains over $k = 0, \dots, K-1$; changing the assumption has not changed the algorithm or which operator multiplies each gradient.

Two versions of the bounds. Set $\chi = 1$ under the general assumptions above. If the applied operators are predictable, one may instead set $\chi = 0$. Throughout, abbreviate

$$B = \sigma^2 + \sigma_g^2, \qquad B_\chi = \sigma_g^2 + \chi\sigma^2, \qquad V = M^2\left(\frac{\sigma^2}{SK} + \rho_S\sigma_g^2\right). \tag{21}$$

The extra σ^2 in B_1 accounts for possible dependence of an applied preconditioner on its current gradient noise.

Scope. The results concern the updates equation 9 and equation 10–equation 13 under the assumptions above. They permit dense symmetric positive-definite matrices and do not require diagonal operators. An implementation using momentum inside the local optimizer, clipping, orthogonalization, decoupled weight decay, or rotations of optimizer states is covered only after its *actual direction* is shown to satisfy this model and these assumptions. In particular, the SOAP, Muon, and Sophia implementation variants do not automatically inherit the theorem merely from their optimizer names. Aggregating curvature states is also not generally the same as averaging their induced operators.

Relation to the supplied algorithm and initialization. The pseudocode updates the state $P_i^{r,k+1}$ but applies $P_i^{r,k}$ to the current gradient. If this applied matrix is fixed before the current mini-batch is drawn, the sharper choice $\chi = 0$ is valid. Merely changing the index in a current-sample-dependent implementation does not establish predictability. The theorems below explicitly initialize $g_G^0 = h^0 = 0$; this specifies the initialization left open in the pseudocode. For FedSOA, the local preconditioner histories may persist across rounds. No reset or preconditioner communication is imposed on that baseline.

Deriving the server direction without an approximation. Summing the local FedPAC updates over k gives, for every sampled client,

$$\begin{aligned} x^r - x_i^{r,K} &= \eta(1-\beta)\sum_{k=0}^{K-1} P_i^{r,k} g_i^{r,k} + \eta K\beta h^r, \\ \frac{1}{SK\eta}\sum_{i\in\mathcal{S}_r}(x^r - x_i^{r,K}) &= (1-\beta)q^r + \beta h^r. \end{aligned} \tag{22}$$

The term βh^r is repeated SK times before normalization, since it is common to all local steps and participating clients. In particular, the server receives an exponentially averaged direction:

$$h^{r+1} = \beta^{r+1}h^0 + (1-\beta)\sum_{t=0}^{r}\beta^{r-t}q^t. \tag{23}$$

This identity follows by induction from equation 12; it does not say that the possibly biased directions q^t become unbiased.

Conditional expectations. Write $\mathbb{E}_r[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_r]$ and $\mathbb{E}_{r,S}[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_r, \mathcal{S}_r]$. For the gradient samples of round r, fix a step-major ordering $j = 1, \dots, SK$ of the pairs (i,k), and write $\mathcal{H}_{r,j}$ for the corresponding pre-sample information. The sigma-fields are nested in this ordering; in particular, all samples already revealed and their applied operators are measurable before the next sample. All operator updates are causal. If two clients are computed in parallel, this ordering only reveals their randomness for the analysis and does not modify the algorithm. Conditional moment assumption equation 15 is understood relative to this joint history, not only relative to an individual iterate.

Finite moments. All quantities manipulated below are finite for every finite number of rounds and local steps. One can check this by induction before performing any absorption. Indeed,

$$\|\nabla f_i(x)\|^2 \le 2\|\nabla f_i(0)\|^2 + 2L^2\|x\|^2, \qquad \mathbb{E}[\|g_i^{r,k}\|^2 \mid \mathcal{H}_{r,i,k}] \le \|\nabla f_i(x_i^{r,k})\|^2 + \sigma^2.$$

The matrices are bounded, x^0 is fixed, and $h^0 = 0$. A local update is a finite linear combination of finite-second-moment vectors, and so are the server updates. This proves the induction. Smoothness also gives $|f(x)| \le |f(0)| + \|\nabla f(0)\|\|x\| + (L/2)\|x\|^2$, ensuring integrability of all objective values used below.

A.1 FOUNDATIONAL LEMMAS

Define the following deterministic second-moment quantities:

$$G_r = \mathbb{E}\|\nabla f(x^r)\|^2, \qquad U_r = \mathbb{E}\left[\frac{1}{SK}\sum_{i\in\mathcal{S}_r}\sum_{k=0}^{K-1}\|x_i^{r,k} - x^r\|^2\right], \tag{24}$$

$$H_r = \mathbb{E}\|h^r\|^2, \qquad Q_r = \mathbb{E}\|q^r\|^2. \tag{25}$$

All sums over the sampled set remain *inside* the expectation.

Elementary inequalities used in the proof. For vectors $v_1, \dots, v_J$ and $\lambda > 0$,

$$\left\|\frac{1}{J}\sum_{j=1}^{J} v_j\right\|^2 \le \frac{1}{J}\sum_{j=1}^{J}\|v_j\|^2, \tag{26}$$

$$\langle v, w\rangle \le \frac{\lambda}{2}\|v\|^2 + \frac{1}{2\lambda}\|w\|^2, \tag{27}$$

$$\left\|\sum_{j=1}^{J} v_j\right\|^2 \le J\sum_{j=1}^{J}\|v_j\|^2. \tag{28}$$

The first and third are consequences of Cauchy–Schwarz; the second follows by expanding $\|\sqrt{\lambda}v - w/\sqrt{\lambda}\|^2 \ge 0$. For matrices B_j and vectors v_j, a second useful consequence is

$$\left\|\frac{1}{J}\sum_{j=1}^{J} B_j v_j\right\|^2 \le \left(\frac{1}{J}\sum_{j=1}^{J}\|B_j\|^2\right)\left(\frac{1}{J}\sum_{j=1}^{J}\|v_j\|^2\right). \tag{29}$$

Indeed, first bound the norm of the sum by $J^{-1}\sum_j \|B_j\|\|v_j\|$, then apply scalar Cauchy–Schwarz. These are pathwise inequalities and require no independence.

Lemma A.7 (Heterogeneity and sampling identities). For every x,

$$\frac{1}{N}\sum_i \|\nabla f_i(x)\|^2 = \|\nabla f(x)\|^2 + \frac{1}{N}\sum_i \|\nabla f_i(x) - \nabla f(x)\|^2. \tag{30}$$

If $v_1, \dots, v_N$ are fixed, $\bar{v} = N^{-1}\sum_i v_i$, and $\mathcal{S}$ is a uniform size-S subset, then

$$\mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}} v_i - \bar{v}\right\|^2 = \rho_S \frac{1}{N}\sum_i \|v_i - \bar{v}\|^2. \tag{31}$$

Proof. Step 1: The gradient second-moment identity. Let $d_i(x) = \nabla f_i(x) - \nabla f(x)$. By the definition of f, $N^{-1}\sum_i d_i(x) = 0$. Expanding the square therefore yields

$$\begin{aligned}\frac{1}{N}\sum_i \|\nabla f_i(x)\|^2 &= \frac{1}{N}\sum_i \big(\|\nabla f(x)\|^2 + 2\langle \nabla f(x), d_i(x)\rangle + \|d_i(x)\|^2\big) \\ &= \|\nabla f(x)\|^2 + \frac{1}{N}\sum_i \|d_i(x)\|^2 \le \|\nabla f(x)\|^2 + \sigma_g^2.\end{aligned} \tag{32}$$

Step 2: First and second inclusion probabilities. For $N > 1$, put $w_i = v_i - \bar{v}$ and $I_i = \mathbf{1}\{i \in \mathcal{S}\}$. Uniform sampling without replacement gives

$$\mathbb{E}I_i = \frac{S}{N}, \qquad \mathbb{E}(I_i I_j) = \frac{S(S-1)}{N(N-1)} \quad (i \ne j).$$

The second expression is zero for $S = 1$. Also $\mathbb{E}[S^{-1}\sum_{i\in\mathcal{S}} v_i] = \bar{v}$. Since $\sum_i w_i = 0$,

$$0 = \left\|\sum_i w_i\right\|^2 = \sum_i \|w_i\|^2 + \sum_{i\ne j}\langle w_i, w_j\rangle.$$

Step 3: Expand the sampling variance. Using $I_i^2 = I_i$, we obtain

$$\mathbb{E}\left\|\frac{1}{S}\sum_i I_i w_i\right\|^2 = \frac{1}{S^2}\left[\sum_i \mathbb{E}I_i \|w_i\|^2 + \sum_{i\neq j}\mathbb{E}(I_i I_j)\langle w_i, w_j\rangle\right]$$
$$= \left(\frac{1}{SN} - \frac{S-1}{SN(N-1)}\right)\sum_i \|w_i\|^2$$
$$= \rho_S \frac{1}{N}\sum_i \|w_i\|^2. \tag{33}$$

For $N = 1$, necessarily $S = 1$, and the sampling error is identically zero. This proves equation 31 in all cases.

Step 4: Conditional use for random round-start models. Conditional on $\mathcal{F}_r$, x^r is fixed, so the preceding identities apply to $\nabla f_i(x^r)$. In particular,

$$\mathbb{E}_r\left[\frac{1}{S}\sum_{i\in\mathcal{S}_r}\|\nabla f_i(x^r)\|^2\right] = \frac{1}{N}\sum_i \|\nabla f_i(x^r)\|^2 \leq \|\nabla f(x^r)\|^2 + \sigma_g^2, \tag{34}$$

$$\mathbb{E}_r\left[\frac{1}{S}\sum_{i\in\mathcal{S}_r}\|\nabla f_i(x^r) - \nabla f(x^r)\|^2\right] \leq \sigma_g^2. \tag{35}$$

No identical-distribution assumption is made about the local objectives. □

Lemma A.8 (Orthogonality of predictable gradient-noise increments). For one round, let $J = SK$, let ε_j be the ordered gradient noises, and let W_j be $\mathcal{H}_{r,j}$-measurable with $\|W_j\| \leq M$. Then

$$\mathbb{E}_{r,S}\left[\frac{1}{J}\sum_{j=1}^J W_j\varepsilon_j\right] = 0, \tag{36}$$

$$\mathbb{E}_{r,S}\left\|\frac{1}{J}\sum_{j=1}^J W_j\varepsilon_j\right\|^2 \leq \frac{M^2\sigma^2}{J}. \tag{37}$$

Proof. Let $Y_j = W_j\varepsilon_j$. Predictability and the conditional-gradient assumption imply

$$\mathbb{E}[Y_j \mid \mathcal{H}_{r,j}] = W_j\mathbb{E}[\varepsilon_j \mid \mathcal{H}_{r,j}] = 0, \qquad \mathbb{E}[\|Y_j\|^2 \mid \mathcal{H}_{r,j}] \leq M^2\sigma^2.$$

Since $\sigma(\mathcal{F}_r, \mathcal{S}_r) \subseteq \mathcal{H}_{r,j}$, the tower property gives $\mathbb{E}_{r,S}Y_j = 0$. For $\ell < j$, Y_ℓ is $\mathcal{H}_{r,j}$-measurable, and hence

$$\mathbb{E}_{r,S}\langle Y_\ell, Y_j\rangle = \mathbb{E}_{r,S}\left[\mathbb{E}[\langle Y_\ell, Y_j\rangle \mid \mathcal{H}_{r,j}]\right]$$
$$= \mathbb{E}_{r,S}\left[\langle Y_\ell, \mathbb{E}[Y_j \mid \mathcal{H}_{r,j}]\rangle\right] = 0. \tag{38}$$

Consequently, expanding the square rather than assuming independence gives

$$\mathbb{E}_{r,S}\left\|\frac{1}{J}\sum_j Y_j\right\|^2 = \frac{1}{J^2}\sum_j \mathbb{E}_{r,S}\|Y_j\|^2 + \frac{2}{J^2}\sum_{\ell<j}\mathbb{E}_{r,S}\langle Y_\ell, Y_j\rangle$$
$$\leq \frac{JM^2\sigma^2}{J^2}.$$

This proves both claims. The proof allows each W_j to depend on earlier noises. It does not allow it to depend on its own current noise without an additional orthogonality argument. □

Lemma A.9 (Bias–noise decomposition). For each round there exist vectors e^r, n^r and a symmetric matrix $A^r = (SK)^{-1}\sum_{i\in\mathcal{S}_r,k} P_i^{r,k}$ such that

$$q^r = A^r\nabla f(x^r) + e^r + n^r, \qquad mI \preceq A^r \preceq MI, \tag{39}$$

and, writing $E_r = \mathbb{E}\|e^r\|^2$,

$$\mathbb{E}[n^r \mid \mathcal{F}_r] = 0, \qquad \mathbb{E}\|n^r\|^2 \le V, \tag{40}$$

$$E_r \le 4M^2L^2U_r + 4D_rB_\chi, \qquad Q_r \le 3M^2G_r + 3E_r + 3V. \tag{41}$$

No zero-mean assumption is made on e^r or on A^r.

Proof. Throughout this proof, sums indexed by i, k run over $i \in \mathcal{S}_r$ and $k = 0, \dots, K-1$.

Step 1: Expand one local direction. Set

$$g^r = \nabla f(x^r), \qquad d_i^r = \nabla f_i(x^r) - g^r, \qquad \ell_i^{r,k} = \nabla f_i(x_i^{r,k}) - \nabla f_i(x^r).$$

The identity $g_i^{r,k} = g^r + d_i^r + \ell_i^{r,k} + \varepsilon_i^{r,k}$ gives

$$P_i^{r,k} g_i^{r,k} = P_i^{r,k} g^r + P^r d_i^r + (P_i^{r,k} - P^r) d_i^r + P_i^{r,k} \ell_i^{r,k} + P_i^{r,k} \varepsilon_i^{r,k}. \tag{42}$$

Define

$$b^r = \frac{1}{SK} \sum_{i,k} P_i^{r,k} \ell_i^{r,k}, \qquad c^r = \frac{1}{SK} \sum_{i,k} (P_i^{r,k} - P^r) d_i^r,$$

$$s^r = P^r \frac{1}{S} \sum_{i \in \mathcal{S}_r} d_i^r, \qquad w^r = \frac{1}{SK} \sum_{i,k} P_i^{r,k} \varepsilon_i^{r,k}. \tag{43}$$

Averaging equation 42 yields the exact relation $q^r = A^r g^r + b^r + c^r + s^r + w^r$. The matrix A^r obeys $mI \preceq A^r \preceq MI$, because averaging preserves both Loewner inequalities. It may depend on the round's samples.

Step 2: Separate noise using the correct measurability. If $\chi = 0$, set $e^r = b^r + c^r$ and $n^r = s^r + w^r$. Lemma A.8, with $W_j = P_i^{r,k}$, gives

$$\mathbb{E}_{r,S} w^r = 0, \qquad \mathbb{E}_{r,S}\|w^r\|^2 \le \frac{M^2\sigma^2}{SK}.$$

If $\chi = 1$, write instead

$$w^r = w_0^r + t^r, \qquad w_0^r = P^r \frac{1}{SK} \sum_{i,k} \varepsilon_i^{r,k}, \qquad t^r = \frac{1}{SK} \sum_{i,k} (P_i^{r,k} - P^r) \varepsilon_i^{r,k}. \tag{44}$$

Set $e^r = b^r + c^r + t^r$ and $n^r = s^r + w_0^r$. Since P^r is known before sampling, the same lemma applies to $W_j = P^r$, proving the same bounds for w_0^r. The quantity t^r is retained in the bias term and is not asserted to have conditional mean zero.

Step 3: Combine the sampling and mini-batch variances. Conditional on $\mathcal{F}_r$, Lemma A.7 gives

$$\mathbb{E}_r s^r = P^r \frac{1}{N} \sum_i d_i^r = 0,$$

$$\mathbb{E}_r\|s^r\|^2 = \rho_S \frac{1}{N} \sum_i \|P^r d_i^r\|^2 \le M^2 \rho_S \sigma_g^2. \tag{45}$$

Let $\widehat{w}^r = w^r$ for $\chi = 0$ and $\widehat{w}^r = w_0^r$ for $\chi = 1$. The vector s^r is measurable given $\mathcal{F}_r, \mathcal{S}_r$. Therefore

$$\mathbb{E}_r\langle s^r, \widehat{w}^r\rangle = \mathbb{E}_r\big[\langle s^r, \mathbb{E}_{r,S}\widehat{w}^r\rangle\big] = 0,$$

$$\mathbb{E}_r n^r = 0, \qquad \mathbb{E}_r\|n^r\|^2 = \mathbb{E}_r\|s^r\|^2 + \mathbb{E}_r\|\widehat{w}^r\|^2 \le V. \tag{46}$$

This calculation uses two conditional-expectation levels; it does not assume unconditional independence of client sampling and local trajectories.

Step 4: Bound the trajectory error. Jensen's inequality, the operator bound, and smoothness give successively

$$\|b^r\|^2 \le \frac{1}{SK} \sum_{i,k} \|P_i^{r,k} \ell_i^{r,k}\|^2$$

$$\le \frac{M^2}{SK} \sum_{i,k} \|\ell_i^{r,k}\|^2 \le \frac{M^2L^2}{SK} \sum_{i,k} \|x_i^{r,k} - x^r\|^2. \tag{47}$$

Taking expectation gives $\mathbb{E}\|b^r\|^2 \le M^2L^2U_r$.

Step 5: Bound the heterogeneous preconditioner error. Apply equation 29 pathwise, and then use the deterministic bound $d_{r,-} \le D_r$ established in Lemma A.6:

$$\begin{aligned}\|c^r\|^2 &\le \left(\frac{1}{SK}\sum_{i,k}\|P_i^{r,k} - P^r\|^2\right)\left(\frac{1}{SK}\sum_{i,k}\|d_i^r\|^2\right)\\ &\le D_r\frac{1}{S}\sum_{i\in\mathcal{S}_r}\|d_i^r\|^2. \end{aligned} \tag{48}$$

Now equation 35 yields

$$\mathbb{E}\|c^r\|^2 \le D_r\,\mathbb{E}\left[\mathbb{E}_r\left(\frac{1}{S}\sum_{i\in\mathcal{S}_r}\|d_i^r\|^2\right)\right] \le D_r\sigma_g^2. \tag{49}$$

We did not factor the expectation of two random quantities. In particular, replacing the pathwise drift bound by an expectation-only bound would invalidate this step without another assumption.

Step 6: Bound current-sample preconditioning bias. In the case $\chi = 1$, use $d_{r,-} \le D_r$ from the same transfer lemma. The pathwise argument gives

$$\begin{aligned}\|t^r\|^2 &\le D_r\frac{1}{SK}\sum_{i,k}\|\varepsilon_i^{r,k}\|^2,\\ \mathbb{E}\|t^r\|^2 &\le D_r\,\mathbb{E}\left[\frac{1}{SK}\sum_{i,k}\mathbb{E}[\|\varepsilon_i^{r,k}\|^2 \mid \mathcal{H}_{r,i,k}]\right] \le D_r\sigma^2. \end{aligned} \tag{50}$$

The selected indices are known at the beginning of the round, so the tower property applies to each term in the ordered sum.

Step 7: Assemble the bias and direction second moments. For predictable operators,

$$E_r \le 2\mathbb{E}\|b^r\|^2 + 2\mathbb{E}\|c^r\|^2 \le 2M^2L^2U_r + 2D_r\sigma_g^2.$$

For general operators,

$$E_r \le 3\mathbb{E}\|b^r\|^2 + 3\mathbb{E}\|c^r\|^2 + 3\mathbb{E}\|t^r\|^2 \le 3M^2L^2U_r + 3D_r(\sigma_g^2 + \sigma^2).$$

Both are bounded by $4M^2L^2U_r + 4D_rB_\chi$, which is the common form used subsequently. Finally,

$$\begin{aligned}Q_r &= \mathbb{E}\|A^rg^r + e^r + n^r\|^2\\ &\le 3\mathbb{E}\|A^rg^r\|^2 + 3\mathbb{E}\|e^r\|^2 + 3\mathbb{E}\|n^r\|^2\\ &\le 3M^2G_r + 3E_r + 3V. \end{aligned}$$

This last inequality intentionally does not discard cross terms involving A^r or e^r, since neither need be independent of n^r. □

Remark A.10 (Why no smallness assumption on D_r is needed). The common-gradient term is retained as $A^r\nabla f(x^r)$, whose inner product with $\nabla f(x^r)$ is at least $m\|\nabla f(x^r)\|^2$ pathwise. Only the heterogeneous component d_i^r enters c^r. This avoids an artificial term D_rG_r and an associated drift-absorption restriction. It does not remove the genuine bias term D_rB_χ.

Lemma A.11 (Local trajectories). If $2L^2M^2u^2 \le 1/2$, FedSOA satisfies

$$U_r \le 4u^2M^2(G_r + B). \tag{51}$$

If $4L^2M^2u^2 \le 1/2$, FedPAC satisfies

$$U_r \le 8u^2M^2(G_r + B) + 4u^2H_r. \tag{52}$$

Proof. Step 1: Bound the averaged stochastic-gradient moment. Because $x_i^{r,k}$ is known before its sample is drawn,

$$\mathbb{E}[\|g_i^{r,k}\|^2 \mid \mathcal{H}_{r,i,k}] = \|\nabla f_i(x_i^{r,k})\|^2 + 2\left\langle \nabla f_i(x_i^{r,k}), \mathbb{E}[\varepsilon_i^{r,k} \mid \mathcal{H}_{r,i,k}] \right\rangle + \mathbb{E}[\|\varepsilon_i^{r,k}\|^2 \mid \mathcal{H}_{r,i,k}] \le \|\nabla f_i(x_i^{r,k})\|^2 + \sigma^2. \tag{53}$$

Next, split the local true gradient around x^r:

$$\|\nabla f_i(x_i^{r,k})\|^2 \le 2\|\nabla f_i(x^r)\|^2 + 2\|\nabla f_i(x_i^{r,k}) - \nabla f_i(x^r)\|^2 \le 2\|\nabla f_i(x^r)\|^2 + 2L^2\|x_i^{r,k} - x^r\|^2. \tag{54}$$

Define

$$T_r := \mathbb{E}\left[\frac{1}{SK}\sum_{i\in\mathcal{S}_r}\sum_{k=0}^{K-1}\|g_i^{r,k}\|^2\right]. \tag{55}$$

Averaging equation 53–equation 54, the K copies of $\|\nabla f_i(x^r)\|^2$ cancel the denominator K. Thus

$$T_r \le 2\mathbb{E}\left[\frac{1}{S}\sum_{i\in\mathcal{S}_r}\|\nabla f_i(x^r)\|^2\right] + 2L^2U_r + \sigma^2 \le 2G_r + 2\sigma_g^2 + 2L^2U_r + \sigma^2, \tag{56}$$

where the final step uses equation 34 and the tower property.

Step 2: Unroll the FedSOA local trajectory. For every $k \ge 1$,

$$x_i^{r,k} - x^r = -\eta\sum_{t=0}^{k-1} P_i^{r,t} g_i^{r,t},$$
$$\|x_i^{r,k} - x^r\|^2 \le \eta^2 k\sum_{t=0}^{k-1}\|P_i^{r,t} g_i^{r,t}\|^2 \le \eta^2 kM^2\sum_{t=0}^{k-1}\|g_i^{r,t}\|^2. \tag{57}$$

The term for $k = 0$ is exactly zero. For $K \ge 2$, average the remaining inequalities and interchange the two finite sums:

$$U_r \le \frac{\eta^2M^2}{SK}\,\mathbb{E}\left[\sum_{i\in\mathcal{S}_r}\sum_{k=1}^{K-1} k\sum_{t=0}^{k-1}\|g_i^{r,t}\|^2\right] = \frac{\eta^2M^2}{SK}\,\mathbb{E}\left[\sum_{i\in\mathcal{S}_r}\sum_{t=0}^{K-2}\left(\sum_{k=t+1}^{K-1} k\right)\|g_i^{r,t}\|^2\right] \le \eta^2K^2M^2\,\mathbb{E}\left[\frac{1}{SK}\sum_{i\in\mathcal{S}_r}\sum_{t=0}^{K-1}\|g_i^{r,t}\|^2\right] = u^2M^2T_r. \tag{58}$$

Here $\sum_{k=t+1}^{K-1} k \le K(K-1)/2 \le K^2$, and adding the $t = K-1$ term can only increase the bound. No independence between local steps is used.

Step 3: Close the FedSOA trajectory inequality. Substituting equation 56 into equation 58 gives

$$(1 - 2L^2M^2u^2)U_r \le u^2M^2(2G_r + 2\sigma_g^2 + \sigma^2).$$

The multiplier on the left is at least $1/2$, so

$$U_r \le 2u^2M^2(2G_r + 2\sigma_g^2 + \sigma^2) \le 4u^2M^2(G_r + \sigma_g^2 + \sigma^2) = 4u^2M^2(G_r + B).$$

Step 4: Unroll and close the FedPAC trajectory inequality. The common correction direction h^r does not change inside the round:

$$x_i^{r,k} - x^r = -\eta(1-\beta)\sum_{t=0}^{k-1} P_i^{r,t} g_i^{r,t} - \eta k\beta h^r.$$

Using $\|v+w\|^2 \le 2\|v\|^2 + 2\|w\|^2$, we obtain

$$\|x_i^{r,k} - x^r\|^2 \le 2\eta^2(1-\beta)^2 kM^2 \sum_{t<k} \|g_i^{r,t}\|^2 + 2\eta^2 k^2 \beta^2 \|h^r\|^2. \tag{59}$$

Average the first term exactly as in equation 58. For the second, use $K^{-1}\sum_{k=0}^{K-1} k^2 \le K^2$. This gives

$$\begin{aligned} U_r &\le 2u^2(1-\beta)^2 M^2 T_r + 2u^2\beta^2 H_r \\ &\le 2u^2 M^2 (2G_r + 2\sigma_g^2 + 2L^2 U_r + \sigma^2) + 2u^2 H_r. \end{aligned} \tag{60}$$

The second inequality uses $0 \le \beta < 1$, so that both squared weights are at most one. Moving the U_r term to the left and dividing by $1 - 4L^2M^2u^2 \ge 1/2$ yields

$$\begin{aligned} U_r &\le 4u^2 M^2 (2G_r + 2\sigma_g^2 + \sigma^2) + 4u^2 H_r \\ &\le 8u^2 M^2 (G_r + B) + 4u^2 H_r. \end{aligned}$$

For $K = 1$ the defining sum for U_r consists only of the initial point, so $U_r = 0$ and both claims hold directly. □

Lemma A.12 (Descent at an auxiliary iterate). For FedPAC define

$$z^r = x^r - \gamma a h^r. \tag{61}$$

Then $z^0 = x^0$, $z^{r+1} = z^r - \gamma q^r$, and, if $\gamma \le m/(6LM^2)$,

$$\mathbb{E} f(z^{r+1}) \le \mathbb{E} f(z^r) - \frac{m\gamma}{4} G_r + \frac{3\gamma}{m} E_r + \gamma T H_r + 2L\gamma^2 V, \qquad T = \frac{2M^2L^2\gamma^2 a^2}{m}. \tag{62}$$

For FedSOA, the same inequality holds with $z^r = x^r$, $\gamma = u$, and $T = 0$, provided $u \le m/(6LM^2)$.

Proof. Step 1: Verify the auxiliary-iterate identity. Using $1 + a = 1/(1-\beta)$, we have

$$\begin{aligned} z^{r+1} &= x^{r+1} - \gamma a h^{r+1} = x^r - \gamma(1+a)h^{r+1} \\ &= x^r - \frac{\gamma}{1-\beta}[\beta h^r + (1-\beta)q^r] = x^r - \gamma a h^r - \gamma q^r = z^r - \gamma q^r. \end{aligned} \tag{63}$$

Also $z^0 = x^0$ since $h^0 = 0$. Because x^r, h^r are $\mathcal{F}_r$-measurable, so is z^r.

Step 2: Apply smoothness at z^r. The global objective is L-smooth since

$$\|\nabla f(x) - \nabla f(y)\| \le \frac{1}{N}\sum_i \|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x - y\|.$$

The descent inequality follows, for example, by integrating the gradient along the segment from z^r to $z^r - \gamma q^r$:

$$f(z^{r+1}) \le f(z^r) - \gamma\langle \nabla f(z^r), q^r\rangle + \frac{L\gamma^2}{2}\|q^r\|^2. \tag{64}$$

Step 3: Remove only the valid zero-mean term. By equation 46,

$$\mathbb{E}\langle \nabla f(z^r), n^r\rangle = \mathbb{E}\left[\langle \nabla f(z^r), \mathbb{E}[n^r \mid \mathcal{F}_r]\rangle\right] = 0. \tag{65}$$

This cancellation is valid even though n^r need not be independent of e^r or A^r, since the vector in this inner product is known at the start of the round. No cross term in $\|q^r\|^2$ is discarded.

Step 4: Bound the three remaining products with explicit coefficients. Put $g = \nabla f(x^r)$ and $d = \nabla f(z^r) - \nabla f(x^r)$. Then

$$\begin{aligned} -\langle g + d, A^r g + e^r\rangle &= -\langle g, A^r g\rangle - \langle g, e^r\rangle - \langle d, A^r g\rangle - \langle d, e^r\rangle \\ &\le -m\|g\|^2 + \|g\|\|e^r\| + M\|d\|\|g\| + \|d\|\|e^r\|. \end{aligned} \tag{66}$$

The following three instances of Young's inequality fix all coefficients:

$$\begin{aligned}\|g\|\|e^r\| &\le \frac{m}{4}\|g\|^2 + \frac{1}{m}\|e^r\|^2,\\ M\|d\|\|g\| &\le \frac{m}{4}\|g\|^2 + \frac{M^2}{m}\|d\|^2,\\ \|d\|\|e^r\| &\le \frac{m}{2}\|d\|^2 + \frac{1}{2m}\|e^r\|^2.\end{aligned} \tag{67}$$

Adding them to equation 66 yields

$$\begin{aligned}-\langle g+d, A^r g + e^r\rangle &\le -\frac{m}{2}\|g\|^2 + \frac{3}{2m}\|e^r\|^2 + \left(\frac{M^2}{m} + \frac{m}{2}\right)\|d\|^2\\ &\le -\frac{m}{2}\|g\|^2 + \frac{2}{m}\|e^r\|^2 + \frac{2M^2}{m}\|d\|^2.\end{aligned} \tag{68}$$

The last step uses $m^2 \le M^2$. Furthermore,

$$\|d\|^2 \le L^2\|z^r - x^r\|^2 = L^2\gamma^2 a^2\|h^r\|^2. \tag{69}$$

Step 5: Collect the first- and second-order smoothness terms. Take expectations in equation 64 and use equation 65–equation 69 and $Q_r \le 3M^2 G_r + 3E_r + 3V$. The result is

$$\begin{aligned}\mathbb{E}f(z^{r+1}) \le \mathbb{E}f(z^r) &- \gamma\left(\frac{m}{2} - \frac{3L\gamma M^2}{2}\right)G_r\\ &+ \gamma\left(\frac{2}{m} + \frac{3L\gamma}{2}\right)E_r + \gamma T H_r + \frac{3L\gamma^2}{2}V.\end{aligned} \tag{70}$$

Under $\gamma \le m/(6LM^2)$,

$$\begin{aligned}\frac{3L\gamma M^2}{2} &\le \frac{m}{4},\\ \frac{3L\gamma}{2} &\le \frac{m}{4M^2} \le \frac{1}{4m},\\ \frac{2}{m} + \frac{3L\gamma}{2} \le \frac{9}{4m} \le \frac{3}{m}, &\qquad \frac{3L\gamma^2}{2} \le 2L\gamma^2.\end{aligned}$$

Substituting these inequalities proves equation 62.

Step 6: Specialize to FedSOA. For equation 9, set $z^r = x^r$ and $\gamma = u$. Then the auxiliary displacement and d are zero, so the TH_r term disappears. All other steps remain valid, and $u \le m/(6LM^2)$ gives the stated result. □

B Convergence of FedSOA

Theorem B.1 (FedSOA under partial participation). Under Assumptions A.1–A.5, choose

$$0 < u = \eta K \le \frac{m}{24LM^2}. \tag{71}$$

Then the iterates of equation 9 satisfy

$$\frac{1}{R}\sum_{r=0}^{R-1} G_r \le \frac{8\Delta}{muR} + \frac{16Lu}{m}V + \frac{384M^4L^2u^2}{m^2}B + \frac{96\overline{D}_R}{m^2}B_\chi. \tag{72}$$

In this theorem, take $\overline{D}_R = \overline{\widehat{D}}_R + \overline{C}_R/K$. Under the uniform Assumption 5.5 alone, the last term is thus $96[\bar{D}_P + (M-m)^2/K]B_\chi/m^2$. The initial-mismatch contribution cannot be dropped without a separate bound or an alignment property for FedSOA. Here $\chi = 1$ always applies; $\chi = 0$ applies if the operators are predictable.

Proof. Step 1: Verify the lemma conditions. The theorem's restriction implies

$$2L^2M^2u^2 \le \frac{2m^2}{24^2M^2} \le \frac{1}{288} \le \frac{1}{2}, \qquad u \le \frac{m}{24LM^2} \le \frac{m}{6LM^2}.$$

Thus both the local-trajectory and descent lemmas apply.

Step 2: Express the bias bound in terms of G_r. Combining equation 41 and equation 51,

$$\begin{aligned} E_r &\le 4M^2L^2[4u^2M^2(G_r + B)] + 4D_rB_\chi \\ &= 16M^4L^2u^2(G_r + B) + 4D_rB_\chi. \end{aligned} \tag{73}$$

Insert this into equation 62 with $\gamma = u$ and $T = 0$:

$$\begin{aligned} \mathbb{E}f(x^{r+1}) \le \mathbb{E}f(x^r) - u\left(\frac{m}{4} - \frac{48M^4L^2u^2}{m}\right)G_r \\ + \frac{48M^4L^2u^3}{m}B + \frac{12uD_r}{m}B_\chi + 2Lu^2V. \end{aligned} \tag{74}$$

Step 3: Absorb the gradient-dependent trajectory term. The coefficient lost to this term satisfies

$$\frac{48M^4L^2u^2}{m} \le \frac{48M^4L^2}{m}\frac{m^2}{24^2L^2M^4} = \frac{m}{12}.$$

The net coefficient is therefore at least $m/4 - m/12 = m/6 \ge m/8$, so equation 74 implies

$$\frac{mu}{8}G_r \le \mathbb{E}f(x^r) - \mathbb{E}f(x^{r+1}) + \frac{48M^4L^2u^3}{m}B + \frac{12uD_r}{m}B_\chi + 2Lu^2V. \tag{75}$$

Step 4: Telescope the objective and retain the correct stepsize powers. The sum of objective differences is exactly $f(x^0) - \mathbb{E}f(x^R) \le \Delta$. Therefore

$$\frac{mu}{8}\sum_{r=0}^{R-1}G_r \le \Delta + \frac{48M^4L^2u^3R}{m}B + \frac{12uB_\chi}{m}\sum_{r=0}^{R-1}D_r + 2Lu^2RV. \tag{76}$$

Divide each term by $muR/8$. In particular,

$$\frac{12uB_\chi\sum_r D_r/m}{muR/8} = \frac{96\overline{D}_RB_\chi}{m^2}, \qquad \frac{2Lu^2RV}{muR/8} = \frac{16Lu}{m}V.$$

The bias term loses its factor u, whereas the variance term retains one. The other two terms give $8\Delta/(muR)$ and $384M^4L^2u^2B/m^2$, respectively. This proves equation 72. □

Corollary B.2 (An explicit horizon-dependent FedSOA rate). Let

$$A_0 = \frac{8\Delta}{m}, \quad A_1 = \frac{16LV}{m}, \quad A_2 = \frac{384M^4L^2B}{m^2}, \quad u_{\max} = \frac{m}{24LM^2}.$$

If $\Delta > 0$, choose

$$u = \min\left\{u_{\max}, \sqrt{\frac{A_0}{A_1R}}, \left(\frac{A_0}{A_2R}\right)^{1/3}\right\}, \tag{77}$$

omitting a candidate when its denominator is zero. Then

$$\frac{1}{R}\sum_r G_r \le \frac{A_0}{u_{\max}R} + 2\sqrt{\frac{A_0A_1}{R}} + 2\left(\frac{A_0^2A_2}{R^2}\right)^{1/3} + \frac{96\overline{D}_RB_\chi}{m^2}. \tag{78}$$

Proof. Write $d_R = 96\overline{D}_RB_\chi/m^2$. Theorem B.1 gives

$$\frac{1}{R}\sum_r G_r \le \frac{A_0}{uR} + A_1u + A_2u^2 + d_R. \tag{79}$$

Assume first $A_1, A_2 > 0$ and set

$$u_1 = \sqrt{\frac{A_0}{A_1 R}}, \qquad u_2 = \left(\frac{A_0}{A_2 R}\right)^{1/3}.$$

Since $u = \min\{u_{\max}, u_1, u_2\}$, we have

$$\begin{aligned}\frac{A_0}{uR} &= \max\left\{\frac{A_0}{u_{\max}R}, \sqrt{\frac{A_0 A_1}{R}}, \left(\frac{A_0^2 A_2}{R^2}\right)^{1/3}\right\} \\ &\le \frac{A_0}{u_{\max}R} + \sqrt{\frac{A_0 A_1}{R}} + \left(\frac{A_0^2 A_2}{R^2}\right)^{1/3}.\end{aligned}$$

Also $u \le u_1$ implies $A_1 u \le \sqrt{A_0 A_1/R}$, and $u \le u_2$ implies $A_2 u^2 \le (A_0^2 A_2/R^2)^{1/3}$. Adding these three estimates in equation 79 proves the claim. If $A_1 = 0$ or $A_2 = 0$, remove its candidate and its zero contribution from the same argument. The assumption $\Delta > 0$ guarantees the retained candidates are positive. For $\Delta = 0$, the finite-step theorem remains valid for every permitted positive u; the balancing formula is not needed. □

In particular, the noise and sampling terms in this bound decrease with the stepsize. With vanishing average effective drift $\overline{\widehat{D}}_R + \overline{C}_R/K$, the bound tends to zero. Vanishing $\overline{\widehat{D}}_R$ alone does not suffice for this FedSOA bound when K is fixed and its initial mismatch is uncontrolled. With fixed nonzero drift, the remaining term is a bias bound; its presence alone is not a lower bound on every instance. A worst-case obstruction is established in Proposition D.1 below.

Effect of preconditioner drift. The final term, $\overline{D}_R B_\chi/m^2$, captures the cost of coupling heterogeneous client gradients with different local preconditioners: unlike the stepsize-dependent terms, it need not vanish as $u \to 0$. For predictable preconditioners, $B_\chi = \sigma_g^2$, so persistent drift under non-IID data can prevent this bound from certifying convergence to a stationary point.

C Convergence of FedPAC

Lemma C.1 (Summed momentum and direction errors). For FedPAC with $h^0 = 0$, if $u \le m/(96LM^2)$, define

$$\begin{aligned}\mathcal{A} &= 160M^4L^2u^2, \\ \mathcal{C}_R &= 64M^4L^2u^2B + 96M^2L^2u^2V + 8\overline{D}_R B_\chi.\end{aligned} \tag{80}$$

Then

$$\sum_{r=0}^{R-1} E_r \le \mathcal{A}\sum_{r=0}^{R-1} G_r + R\mathcal{C}_R, \tag{81}$$

$$\sum_{r=0}^{R-1} H_r \le 3(M^2+\mathcal{A})\sum_{r=0}^{R-1} G_r + 3R(\mathcal{C}_R + V). \tag{82}$$

Proof. For this proof only, abbreviate $\mathsf{G} = \sum_{r=0}^{R-1} G_r$, $\mathsf{E} = \sum_{r=0}^{R-1} E_r$, $\mathsf{H} = \sum_{r=0}^{R-1} H_r$, and $\mathsf{U} = \sum_{r=0}^{R-1} U_r$.

Step 1: Derive the momentum inequality including its boundary terms. For $0 \le \beta < 1$, the exact identity

$$\|\beta v + (1-\beta)w\|^2 = \beta\|v\|^2 + (1-\beta)\|w\|^2 - \beta(1-\beta)\|v-w\|^2$$

implies $H_{r+1} \le \beta H_r + (1-\beta)Q_r$. Sum this inequality over $r = 0, \dots, R-1$. Since $\sum_r H_{r+1} = \mathsf{H} - H_0 + H_R$, we obtain

$$\begin{aligned}(1-\beta)\mathsf{H} &\le (1-\beta)\sum_r Q_r + H_0 - H_R, \\ \mathsf{H} &\le \sum_r Q_r + \frac{H_0 - H_R}{1-\beta} \le \sum_r Q_r \le 3M^2\mathsf{G} + 3\mathsf{E} + 3RV.\end{aligned} \tag{83}$$

The second inequality uses $H_0 = 0$ and $H_R \geq 0$. This step is precisely where zero initialization is used; a nonzero initialization would add $H_0/(1-\beta)$.

Step 2: Eliminate the summed momentum from the trajectory bound. The local-step restriction ensures $4L^2M^2u^2 \leq 4m^2/(96^2M^2) \leq 1/2$. Thus equation 52 gives

$$\begin{aligned}\mathsf{U} &\leq 8u^2M^2\mathsf{G} + 8u^2M^2RB + 4u^2\mathsf{H} \\ &\leq 8u^2M^2\mathsf{G} + 8u^2M^2RB + 4u^2(3M^2\mathsf{G} + 3\mathsf{E} + 3RV) \\ &= 20u^2M^2\mathsf{G} + 12u^2\mathsf{E} + 8u^2M^2RB + 12u^2RV. \end{aligned} \tag{84}$$

Step 3: Close the coupled error inequality. Sum the bias bound equation 41 and substitute equation 84:

$$\begin{aligned}\mathsf{E} &\leq 4M^2L^2\mathsf{U} + 4R\overline{D}_R B_\chi \\ &\leq 80M^4L^2u^2\mathsf{G} + 48M^2L^2u^2\mathsf{E} + 32M^4L^2u^2RB \\ &\quad + 48M^2L^2u^2RV + 4R\overline{D}_R B_\chi. \end{aligned} \tag{85}$$

This is a closed scalar inequality for E; no further independence or relation between $P_i^{r,K}$ and the within-round operators is needed.

Step 4: Verify the denominator before dividing. The stipulated local stepsize gives

$$48M^2L^2u^2 \leq \frac{48m^2}{96^2M^2} \leq \frac{1}{192} \leq \frac{1}{2}.$$

Hence $1 - 48M^2L^2u^2 \geq 1/2$. After moving that multiple of E to the left of equation 85, we obtain

$$\begin{aligned}\mathsf{E} &\leq \frac{80M^4L^2u^2\mathsf{G} + 32M^4L^2u^2RB + 48M^2L^2u^2RV + 4R\overline{D}_R B_\chi}{1 - 48M^2L^2u^2} \\ &\leq 160M^4L^2u^2\mathsf{G} + 64M^4L^2u^2RB + 96M^2L^2u^2RV + 8R\overline{D}_R B_\chi \\ &= \mathcal{A}\mathsf{G} + R\mathcal{C}_R. \end{aligned} \tag{86}$$

All numerators are nonnegative, so replacing the reciprocal denominator by 2 preserves the inequality.

Step 5: Recover the summed momentum bound. Substitute equation 86 back into equation 83:

$$\mathsf{H} \leq 3M^2\mathsf{G} + 3(\mathcal{A}\mathsf{G} + R\mathcal{C}_R) + 3RV = 3(M^2 + \mathcal{A})\mathsf{G} + 3R(\mathcal{C}_R + V).$$

This is equation 82, and equation 86 is equation 81. □

Theorem C.2 (FedPAC under partial participation). Under Assumptions A.1–A.5, let $h^0 = 0$, $\beta \in [0, 1)$, and choose

$$0 < u = \eta K \leq \frac{m}{96LM^2}, \qquad 0 < \gamma \leq \frac{m}{6LM^2}, \qquad \gamma\frac{\beta}{1-\beta} \leq \frac{m}{16LM^2}. \tag{87}$$

The last condition is vacuous when $\beta = 0$. Then

$$\begin{aligned}\frac{1}{R}\sum_{r=0}^{R-1} G_r \leq \frac{8\Delta}{m\gamma R} + \frac{16L\gamma}{m}V + \frac{48M^2L^2\gamma^2}{m^2}\left(\frac{\beta}{1-\beta}\right)^2 V \\ + \frac{2048M^4L^2u^2}{m^2}B + \frac{3072M^2L^2u^2}{m^2}V + \frac{256\overline{D}_R}{m^2}B_\chi. \end{aligned} \tag{88}$$

The general case uses $\chi = 1$; predictable operators allow $\chi = 0$. Here $\overline{D}_R = \overline{\widehat{D}}_R$ by round-start alignment; with only the uniform Assumption 5.5, take $\overline{D}_R = \bar{D}_P$. The terminal index K in that assumption is used only through Lemma A.6; no bound on $P^{r+1} - P^r$ and no smallness condition on $\overline{D}_R$ are required.

Proof. Step 1: Sum the descent bound. The first bound in equation 87 permits Lemma C.1; the second permits Lemma A.12. Sum equation 62 to get

$$\begin{aligned}\frac{m\gamma}{4}\sum_{r=0}^{R-1} G_r &\leq \mathbb{E}f(z^0) - \mathbb{E}f(z^R) + \frac{3\gamma}{m}\sum_r E_r + \gamma T\sum_r H_r + 2L\gamma^2RV \\ &\leq \Delta + \frac{3\gamma}{m}\sum_r E_r + \gamma T\sum_r H_r + 2L\gamma^2RV. \end{aligned} \tag{89}$$

For the second line, $z^0 = x^0$ and the global lower bound $f(z^R) \geq f_*$ were used. The auxiliary point need not be an iterate returned by the server; the lower bound on f holds for every point. Since $\gamma > 0$, division by γ gives

$$\frac{m}{4}\sum_r G_r \leq \frac{\Delta}{\gamma} + \frac{3}{m}\sum_r E_r + T\sum_r H_r + 2L\gamma RV. \tag{90}$$

Step 2: Substitute the two summed moment bounds. Using Lemma C.1, the right-hand side of equation 90 is at most

$$\begin{aligned}&\frac{\Delta}{\gamma} + \frac{3}{m}\left(\mathcal{A}\sum_r G_r + R\mathcal{C}_R\right)\\ &\quad + T\left[3(M^2+\mathcal{A})\sum_r G_r + 3R(\mathcal{C}_R + V)\right] + 2L\gamma RV.\end{aligned}$$

Group the coefficients of $\sum_r G_r$, $R\mathcal{C}_R$, and RV. Moving only the first group to the left yields

$$\begin{aligned}&\left[\frac{m}{4} - \frac{3\mathcal{A}}{m} - 3T(M^2+\mathcal{A})\right]\sum_r G_r\\ &\qquad \leq \frac{\Delta}{\gamma} + \left(\frac{3}{m} + 3T\right)R\mathcal{C}_R + (3T + 2L\gamma)RV.\end{aligned} \tag{91}$$

In particular, the terms proportional to $\overline{D}_R B_\chi$ remain on the right; they are not gradient-dependent terms that could be absorbed.

Step 3: Control the local-error contribution. Using the first bound of equation 87,

$$\begin{aligned}\mathcal{A} = 160M^4L^2u^2 &\leq 160M^4L^2\frac{m^2}{96^2L^2M^4} = \frac{160}{96^2}m^2 = \frac{5}{288}m^2\\ &\leq \frac{m^2}{48} \leq M^2.\end{aligned} \tag{92}$$

Consequently $3\mathcal{A}/m \leq m/16$.

Step 4: Control the auxiliary-displacement contribution. The final bound of equation 87 is $\gamma a \leq m/(16LM^2)$. Therefore

$$\begin{aligned}T = \frac{2M^2L^2}{m}(\gamma a)^2 &\leq \frac{2M^2L^2}{m}\frac{m^2}{16^2L^2M^4} = \frac{m}{128M^2},\\ 3T(M^2+\mathcal{A}) &\leq 6TM^2 \leq \frac{3m}{64} \leq \frac{m}{16}.\end{aligned} \tag{93}$$

For $\beta = 0$, $a = T = 0$, and these inequalities hold automatically. Combining equation 92 and equation 93,

$$\frac{m}{4} - \frac{3\mathcal{A}}{m} - 3T(M^2+\mathcal{A}) \geq \frac{m}{4} - \frac{m}{16} - \frac{m}{16} = \frac{m}{8}.$$

Moreover, since $m \leq M$,

$$\frac{3}{m} + 3T \leq \frac{3}{m} + \frac{3m}{128M^2} \leq \frac{3}{m} + \frac{3}{128m} \leq \frac{4}{m}.$$

Thus all denominators used in the next step are strictly positive.

Step 5: Divide the final scalar inequality. From equation 91,

$$\frac{m}{8}\sum_r G_r \leq \frac{\Delta}{\gamma} + \frac{4}{m}R\mathcal{C}_R + (3T + 2L\gamma)RV.$$

Divide by $mR/8$:

$$\frac{1}{R}\sum_r G_r \leq \frac{8\Delta}{m\gamma R} + \frac{32\mathcal{C}_R}{m^2} + \left(\frac{24T}{m} + \frac{16L\gamma}{m}\right)V. \tag{94}$$

Step 6: Expand each constant separately. By equation 80,

$$\frac{32\mathcal{C}_R}{m^2} = \frac{32 \cdot 64\, M^4L^2u^2}{m^2}B + \frac{32 \cdot 96\, M^2L^2u^2}{m^2}V + \frac{32 \cdot 8\, \overline{D}_R}{m^2}B_\chi$$
$$= \frac{2048M^4L^2u^2}{m^2}B + \frac{3072M^2L^2u^2}{m^2}V + \frac{256\overline{D}_R}{m^2}B_\chi,$$

and

$$\frac{24T}{m}V = \frac{48M^2L^2\gamma^2}{m^2}\left(\frac{\beta}{1-\beta}\right)^2 V.$$

Substituting these two identities into equation 94 proves equation 88. No gradient average remains on the right-hand side, the index transfer is justified by round-start alignment, and no dependence on $\beta/(1-\beta)$ is hidden in a numerical constant. □

Corollary C.3 (Vanishing stationarity error with vanishing drift). Fix m, M, L, S, K, N and $\beta < 1$. Choose positive constants c_u, c_γ such that $c_u \le m/(96LM^2)$, $c_\gamma \le m/(6LM^2)$, and $c_\gamma\beta/(1-\beta) \le m/(16LM^2)$. For a horizon-$R$ run, set $u = c_uR^{-1/2}$ and $\gamma = c_\gamma R^{-1/2}$. Then

$$\frac{1}{R}\sum_r G_r \le \left(\frac{8\Delta}{mc_\gamma} + \frac{16Lc_\gamma V}{m}\right)R^{-1/2} + \frac{48M^2L^2c_\gamma^2a^2V + 2048M^4L^2c_u^2B + 3072M^2L^2c_u^2V}{m^2R} + \frac{256\overline{D}_RB_\chi}{m^2}. \tag{95}$$

In particular, $\overline{D}_R \to 0$ implies average expected stationarity, and $\overline{D}_R = O(R^{-1/2})$ gives an $O(R^{-1/2})$ rate. Alternatively, $B_\chi = 0$ eliminates this bias term without requiring vanishing drift. If $\widehat{r}$ is sampled uniformly from $\{0, \dots, R-1\}$ independently of the run, the left-hand side equals $\mathbb{E}\|\nabla f(x^{\widehat{r}})\|^2$.

Proof. Step 1: Check feasibility for every horizon. Since $R^{-1/2} \le 1$ for $R \ge 1$, the stipulated bounds on c_u, c_γ imply all three inequalities in equation 87 after setting $u = c_uR^{-1/2}$ and $\gamma = c_\gamma R^{-1/2}$. In particular, $\eta = c_u/(K\sqrt{R}) > 0$.

Step 2: Substitute into each term of the finite-horizon theorem. The first three terms become

$$\frac{8\Delta}{m\gamma R} = \frac{8\Delta}{mc_\gamma}R^{-1/2},$$
$$\frac{16L\gamma}{m}V = \frac{16Lc_\gamma V}{m}R^{-1/2},$$
$$\frac{48M^2L^2\gamma^2a^2}{m^2}V = \frac{48M^2L^2c_\gamma^2a^2V}{m^2}R^{-1}.$$

Similarly,

$$\frac{2048M^4L^2u^2}{m^2}B = \frac{2048M^4L^2c_u^2B}{m^2R}, \qquad \frac{3072M^2L^2u^2}{m^2}V = \frac{3072M^2L^2c_u^2V}{m^2R}.$$

The remaining term is $256\overline{D}_RB_\chi/m^2$ and has no additional factor of $R^{-1/2}$. Adding the six terms proves equation 95. For a family of horizon-dependent runs, the assumptions and their constants $m, M, L, \sigma, \sigma_g$ are understood to hold uniformly over that family.

Step 3: Pass to the stated asymptotic conclusions. With fixed $\beta < 1$, all displayed coefficients are independent of R. If $\overline{D}_R \to 0$, all three contributions $O(R^{-1/2})$, $O(R^{-1})$, and $O(\overline{D}_R)$ tend to zero. If $\overline{D}_R \le C_DR^{-1/2}$ for some constant C_D, its term is at most $(256C_DB_\chi/m^2)R^{-1/2}$, yielding the asserted rate. When $B_\chi = 0$, that term vanishes exactly.

Step 4: Interpret the average as a random-output guarantee. Let $\mathcal{T}_R$ contain the entire training history. An independent uniform index $\widehat{r}$ satisfies

$$\mathbb{E}[\|\nabla f(x^{\widehat{r}})\|^2 \mid \mathcal{T}_R] = \sum_{r=0}^{R-1}\frac{1}{R}\|\nabla f(x^r)\|^2,$$

$$\mathbb{E}\|\nabla f(x^{\widehat{r}})\|^2 = \frac{1}{R}\sum_{r=0}^{R-1}G_r.$$

The theorem is a first-order stationarity guarantee, not a last-iterate, almost-sure, or global-optimality claim. □

Corollary C.4 (A sufficient operator-stability condition). Suppose the applied operators are aligned at $k = 0$ and, in addition,

$$P_i^{r,0} = P^r, \qquad \|P_i^{r,k+1} - P_i^{r,k}\| \le \Lambda\eta \quad \text{almost surely}, \tag{96}$$

for $k = 0, \dots, K-1$, with a deterministic Λ independent of R. Then the positive-index drift in Assumption 5.5 admits the bound

$$\widehat{D}_r = \Lambda^2\eta^2 \frac{(K+1)(2K+1)}{6} \le \Lambda^2 u^2.$$

A sharper direct bound on the applied-index drift, which is the quantity needed in the proof, permits taking

$$D_r = \Lambda^2\eta^2 \frac{(K-1)(2K-1)}{6} \le \frac{\Lambda^2 u^2}{3}. \tag{97}$$

Consequently, the stepsizes of Corollary C.3 yield a vanishing FedPAC bound, with an $O(R^{-1})$ drift term.

Proof. At $k = 0$, the discrepancy is zero by alignment. For $k \ge 1$, telescope the operator increments and apply the triangle inequality:

$$\|P_i^{r,k} - P^r\| = \left\| \sum_{t=0}^{k-1} (P_i^{r,t+1} - P_i^{r,t}) \right\| \le \sum_{t=0}^{k-1} \|P_i^{r,t+1} - P_i^{r,t}\| \le k\Lambda\eta. \tag{98}$$

Applying the preceding estimate for $k = 1, \dots, K$ gives

$$d_{r,+} \le \frac{\Lambda^2\eta^2}{K} \sum_{k=1}^{K} k^2 = \Lambda^2\eta^2 \frac{(K+1)(2K+1)}{6} \le \Lambda^2 u^2.$$

For the sharper applied-index bound, squaring before averaging gives the pathwise inequality

$$\begin{aligned} \frac{1}{SK} \sum_{i\in\mathcal{S}_r} \sum_{k=0}^{K-1} \|P_i^{r,k} - P^r\|^2 &\le \frac{\Lambda^2\eta^2}{SK} \sum_{i\in\mathcal{S}_r} \sum_{k=0}^{K-1} k^2 \\ &= \frac{\Lambda^2\eta^2}{K} \frac{(K-1)K(2K-1)}{6} \\ &= \Lambda^2\eta^2 \frac{(K-1)(2K-1)}{6} \le \frac{\Lambda^2\eta^2 K^2}{3}. \end{aligned}$$

This proves equation 97 for every realized sampled set, which is stronger than a mere bound in expectation. Under $u = c_u/\sqrt{R}$, we may therefore take $\overline{D}_R \le \Lambda^2 c_u^2/(3R)$. Its contribution to equation 88 is bounded by

$$\frac{256\overline{D}_R B_\chi}{m^2} \le \frac{256\Lambda^2 c_u^2 B_\chi}{3m^2 R}.$$

The other terms are controlled by Corollary C.3, which completes the proof. □

Condition equation 96 is additional and must be established for the chosen preconditioner update. It does not follow from boundedness, round-start alignment, or the averaging rule equation 14. For example, holding a shared operator fixed at every index $k = 0, \dots, K$ during each round gives $D_r = \widehat{D}_r = 0$, but constitutes a particular implementation choice.

D NECESSITY OF THE BIAS TERM AND CONSEQUENCES FOR THE MAIN TEXT

Proposition D.1 (Persistent drift can prevent stationarity). For any fixed $\beta \in [0, 1)$ and arbitrarily small positive drift bounds, there exists a one-dimensional, deterministic, full-participation instance of FedPAC satisfying Assumptions A.1–A.5, including predictable operators and round-start alignment, such that $\nabla f(x^r) \neq 0$ for every r. The instance also applies to FedSOA by taking $\beta = 0$ and $\gamma = \eta K$.

Proof. Step 1: Construct objectives satisfying all objective assumptions. Let $N = S = 2$, $K = 2$, and

$$f_1(x) = \tfrac{1}{2}(x-1)^2, \qquad f_2(x) = \tfrac{1}{2}(x+1)^2.$$

Then $f(x) = (x^2+1)/2$, $\nabla f(x) = x$, and each local Hessian equals one. Thus $L = 1$ and $f_* = 1/2$. Use exact gradients, so $\sigma = 0$. Since

$$\nabla f_1(x) - \nabla f(x) = -1, \qquad \nabla f_2(x) - \nabla f(x) = 1,$$

the heterogeneity identity gives $\sigma_g^2 = 1$ uniformly over x.

Step 2: Construct aligned, bounded, predictable preconditioners. Fix $0 < \delta < 1$ and, at every round, set

$$P^r = P_1^{r,0} = P_2^{r,0} = 1,$$
$$P_1^{r,1} = 1+\delta, \qquad P_2^{r,1} = 1-\delta,$$
$$P_1^{r,2} = P_2^{r,2} = 1.$$

They are deterministic scalars, hence predictable positive-definite operators, with $m = 1-\delta$ and $M = 1+\delta$. Their drift at the indices specified by Assumption 5.5 is exactly

$$d_{r,+} = \frac{1}{4}(\delta^2 + 0 + \delta^2 + 0) = \frac{\delta^2}{2}.$$

Thus the uniform assumption holds with $\bar{D}_P = \delta^2/2$. Their applied-index drift is also exactly

$$\frac{1}{SK}\sum_{i=1}^{2}\sum_{k=0}^{1} |P_i^{r,k} - P^r|^2 = \frac{1}{4}(0 + \delta^2 + 0 + \delta^2) = \frac{\delta^2}{2}.$$

Thus $D_r = \delta^2/2$. The uploaded endpoint average equals $(P_1^{r,2} + P_2^{r,2})/2 = 1$, so the server aggregation and the next round's alignment hold. Even averaging the last applied operators would give $(1+\delta+1-\delta)/2 = 1$.

Step 3: Compute both local gradients exactly. Choose $\eta > 0$ with $0 < \eta(1-\beta) < 1$ and put $t = 1-\eta(1-\beta) \in (0,1)$. Let $c_1 = 1, c_2 = -1$. If $h^r = 0$, then

$$g_i^{r,0} = x^r - c_i,$$
$$x_i^{r,1} = x^r - \eta(1-\beta)(x^r - c_i),$$
$$g_i^{r,1} = x_i^{r,1} - c_i = [1-\eta(1-\beta)](x^r - c_i) = t(x^r - c_i).$$

Consequently,

$$\begin{aligned} q^r &= \tfrac{1}{4}\big[(x^r-1) + (x^r+1) + t(1+\delta)(x^r-1) + t(1-\delta)(x^r+1)\big] \\ &= \tfrac{1}{4}[2x^r + 2tx^r - 2t\delta] = \tfrac{1}{2}[(1+t)x^r - t\delta]. \end{aligned} \tag{99}$$

Step 4: Initialize at a nonstationary fixed point. Set

$$h^0 = 0, \qquad x^0 = x_\delta := \frac{t\delta}{1+t} > 0. \tag{100}$$

If $x^r = x_\delta$ and $h^r = 0$, then equation 99 gives $q^r = 0$. The server recurrence therefore gives $h^{r+1} = \beta \cdot 0 + (1-\beta)\cdot 0 = 0$ and $x^{r+1} = x^r - \gamma \cdot 0 = x_\delta$. Induction starting from $r = 0$ proves this for every round and every positive server stepsize. Hence

$$\frac{1}{R}\sum_{r=0}^{R-1} \|\nabla f(x^r)\|^2 = x_\delta^2 = \left(\frac{t\delta}{1+t}\right)^2 > 0. \tag{101}$$

Step 5: Check compatibility with small stepsizes and drift. Choose η, γ small enough to satisfy equation 87; all its upper bounds are positive for every fixed $\beta < 1$ and $\delta < 1$. The drift $\delta^2/2$ can be made arbitrarily small by choosing δ small. If $t \geq 1/2$, then $t/(1+t) \geq 1/3$, so

$$x_\delta^2 \geq \frac{\delta^2}{9} = \frac{2}{9} D_r \sigma_g^2.$$

Thus an actual nonzero gradient floor can have the same order as the drift–heterogeneity term. Setting $\beta = 0$ and $\gamma = \eta K$ gives the corresponding FedSOA instance. □

Proposition D.2 (Fixed initialization and horizon-dependent stepsizes). In the construction of Proposition D.1, fix $x^0 = \delta/2$ and $h^0 = 0$, independently of the horizon. Let the local and server stepsizes be constant within each run, with $0 < \eta(1-\beta) \leq 1/2$ and $0 < \gamma \leq 1$. Then every round satisfies

$$x^r \geq \delta/6, \qquad \frac{1}{R}\sum_{r=0}^{R-1} \|\nabla f(x^r)\|^2 \geq \frac{\delta^2}{36}.$$

In particular, the obstruction also holds for sequences of horizon-dependent stepsizes $\eta_R, \gamma_R \to 0$.

Proof. Step 1: Compute the direction when h^r is nonzero. Write $b = 1-\beta > 0$, $t = 1-\eta b$, $c = (1+t)/2$, and $x_\delta = t\delta/(1+t)$. The first local step now gives

$$g_i^{r,1} = t(x^r - c_i) - \eta\beta h^r.$$

Using $P_1^{r,1} + P_2^{r,1} = 2$ in the average,

$$q^r = \tfrac{1}{2}[(1+t)x^r - t\delta] - \frac{\eta\beta}{2} h^r = c(x^r - x_\delta) - \frac{\eta\beta}{2} h^r. \tag{102}$$

Thus, with $\widetilde{\beta} = \beta(1 - \eta b/2) \in [0, 1)$,

$$h^{r+1} = \widetilde{\beta} h^r + bc(x^r - x_\delta). \tag{103}$$

Step 2: Express the dynamics relative to the fixed point. Set $y^r = x^r - x_\delta$, $v^r = -\gamma h^r$, and $v^0 = 0$. For $r \geq 1$, the server update implies $v^r = x^r - x^{r-1}$. Equation equation 103 is equivalent to

$$v^{r+1} = \widetilde{\beta} v^r - \gamma bc\, y^r, \qquad y^{r+1} = y^r + v^{r+1}. \tag{104}$$

These identities are exact for every round.

Step 3: Construct a nonincreasing quadratic energy. Define

$$\mathcal{L}_r = \frac{c}{2}(y^r)^2 + \frac{\widetilde{\beta}}{2\gamma b}(v^r)^2.$$

This definition is also valid when $\widetilde{\beta} = 0$. Using $cy^r = (\widetilde{\beta} v^r - v^{r+1})/(\gamma b)$,

$$\begin{aligned}
\mathcal{L}_{r+1} - \mathcal{L}_r &= cy^r v^{r+1} + \frac{c}{2}(v^{r+1})^2 + \frac{\widetilde{\beta}}{2\gamma b}[(v^{r+1})^2 - (v^r)^2] \\
&= \frac{\widetilde{\beta}}{\gamma b} v^r v^{r+1} + \left(-\frac{1}{\gamma b} + \frac{c}{2} + \frac{\widetilde{\beta}}{2\gamma b}\right)(v^{r+1})^2 - \frac{\widetilde{\beta}}{2\gamma b}(v^r)^2 \\
&\leq \left(-\frac{1-\widetilde{\beta}}{\gamma b} + \frac{c}{2}\right)(v^{r+1})^2.
\end{aligned} \tag{105}$$

The last line uses $2v^r v^{r+1} \leq (v^r)^2 + (v^{r+1})^2$ multiplied by the nonnegative factor $\widetilde{\beta}/(2\gamma b)$. Since $c \leq 1$, $1 - \widetilde{\beta} \geq 1 - \beta = b$, and $\gamma \leq 1$,

$$-\frac{1-\widetilde{\beta}}{\gamma b} + \frac{c}{2} \leq -\frac{1}{\gamma} + \frac{1}{2} \leq -\frac{1}{2}.$$

Therefore $\mathcal{L}_{r+1} \leq \mathcal{L}_r$ for all r.

Step 4: Convert energy stability into a gradient lower bound. Because $v^0 = 0$ and $c > 0$,

$$\frac{c}{2}(y^r)^2 \le \mathcal{L}_r \le \mathcal{L}_0 = \frac{c}{2}(y^0)^2, \qquad |x^r - x_\delta| \le |x^0 - x_\delta|.$$

Compute the initial displacement exactly:

$$|x^0 - x_\delta| = \frac{\delta}{2} - \frac{(1-\eta b)\delta}{2-\eta b} = \frac{\delta \eta b}{2(2-\eta b)}.$$

When $\eta b \le 1/2$, $t \ge 1/2$ and hence $x_\delta \ge \delta/3$, while the last expression is at most $\delta/6$. It follows that $x^r \ge x_\delta - |x^r - x_\delta| \ge \delta/6$. Since $\nabla f(x) = x$, each squared gradient is at least $\delta^2/36$, as claimed. The proof applies separately to every horizon with the same fixed initialization and objective, so it requires no exchange of limits. □

Why the initial term is necessary for unaligned FedSOA. Take $N = S = 2$, $K = 1$, exact gradients, $f_1(x) = (x-1)^2/2$, $f_2(x) = (x+1)^2/2$, and, at every round,

$$P^r = 1, \qquad P_1^{r,0} = 1 + \delta, \quad P_2^{r,0} = 1 - \delta, \qquad P_1^{r,1} = P_2^{r,1} = 1, \quad 0 < \delta < 1.$$

These deterministic operators obey the spectral bounds with $m = 1 - \delta$, $M = 1 + \delta$. Assumption 5.5 holds with $\bar{D}_P = 0$, since it only sees index 1. Nevertheless,

$$q^r = \tfrac{1}{2}[(1+\delta)(x^r - 1) + (1-\delta)(x^r + 1)] = x^r - \delta.$$

Initializing $x^0 = \delta$ gives $x^r = \delta$ for all rounds and $\|\nabla f(x^r)\|^2 = \delta^2 > 0$, for every positive stepsize. Here $J_{r,0} = \delta^2$, exactly identifying the missing contribution. This example uses the freedom of round-start states in the stated FedSOA model: it is a counterexample to an implication from the listed assumptions, not a claim about every particular optimizer update. If an implementation requires a specified cross-round state-continuity rule, that rule must be stated and used to derive its own bound on C_r.

Correct interpretation of the results. Theorems B.1 and C.2 distinguish sampling variance from systematic preconditioning bias. The variance term V appears with a vanishing stepsize factor; the drift term need not. Alignment can improve the bound by reducing the *actual* drift $\bar{D}_R$. Averaging a previous global direction does not, by itself, cancel a biased mean update. A vanishing error guarantee under drift therefore requires the conditions of Corollary C.3. These upper bounds alone do not establish a uniform convergence-rate advantage of FedPAC over FedSOA.

Proof details. First, $\mathbb{E}[P_i^{r,k}\varepsilon_i^{r,k}] = 0$ requires predictability or another justified orthogonality condition; boundedness of $P_i^{r,k}$ alone is insufficient. Second, Assumption 5.5 controls indices $k = 1, \dots, K$, whereas the updates use $k = 0, \dots, K-1$. Their difference is exactly $(J_{r,0} - J_{r,K})/K$. FedPAC removes $J_{r,0}$ by alignment; FedSOA must retain its contribution, bounded by C_r/K. Third, dividing a descent sum by γR cancels a term's factor γ; it does not turn a systematic $D_r\sigma_g^2$ error into $\gamma D_r \sigma_g^2$. Fourth, constants depending on $\beta/(1-\beta)$ are not universal as $\beta \uparrow 1$. Finally, setting an artificial previous target equal to h^0 does not eliminate the actual initial mismatch to $P^0\nabla f(x^0)$. The virtual-iterate proof above uses the real initialization $h^0 = 0$ and the exact equality $z^0 = x^0$.

Table 5: Official image-dataset statistics. These counts describe the dataset releases, not an additional validation split used for hyperparameter selection.

Dataset	#Classes	Image Size	Train	Val	Test	Total	Train / class
CIFAR-100	100	$3 \times 32 \times 32$	50,000	—	10,000	60,000	500
Tiny ImageNet	200	$3 \times 64 \times 64$	100,000	10,000	10,000	120,000	500

Notes. (1) CIFAR-100 provides official training and test splits but no separate validation split.
(2) CIFAR-100 contains 100 fine-grained classes; 20 coarse superclasses are also defined for hierarchical labeling.
(3) Tiny ImageNet is a subset of ImageNet synsets: per class 500 train, 50 val, and 50 test images (test labels are not publicly released).
(4) Both image datasets contain RGB images; the table lists their native resolutions.

E Experimental Setup

E.1 Datasets

For image classification, we evaluate our methods on **CIFAR-100** and **Tiny ImageNet**.

- **CIFAR-100** (Krizhevsky et al., 2009): Contains 100 classes with 600 color images per class at a resolution of 32×32. It is a standard benchmark for evaluating federated image classification methods.
- **Tiny ImageNet**: A subset of ImageNet with 200 classes and 500 training images per class, providing a more challenging and high-resolution classification task.

For language modeling, we use C4 with LLaMA models of 60M, 130M, and 350M parameters, as reported in the main text.

E.2 Federated Learning Configuration

The from-scratch image-classification experiments use 100 clients, 10% participation per round, 300 communication rounds, mini-batch size 50, and $K = 50$ local steps. We compare IID partitions with Dirichlet partitions of concentration $\alpha \in \{0.5, 0.1, 0.05\}$; smaller α indicates stronger label heterogeneity. The local learning rates and optimizer settings are listed in Table 6. We use cosine learning-rate decay and $\beta = 0.5$ for FedPAC unless varied in the ablation study. Each FedPAC variant uses the same local learning rate as its corresponding local optimizer within each experimental setting. Results are means over five runs with seeds 42, 43, 44, 45, and 46; experiments use NVIDIA RTX 4090 GPUs. The accuracy tables below report means, without standard deviations.

In the hyperparameter tables, β denotes the FedPAC correction coefficient; the FedCM entry is its method-specific coefficient and should not be interpreted as the same mechanism. For Muon, β_1 denotes μ in Algorithms 5 and 6; these algorithms have no second-moment coefficient β_2.

E.3 ResNet-18 Architecture

We adopt **ResNet-18** as the backbone model. To better adapt it to CIFAR-100, we modify its architecture following standard practices (He et al., 2016):

- Replace the original 7×7 convolution with a 3×3 kernel.
- Remove the initial downsampling layers (stride-2 convolution and max pooling).

We use Batch Normalization (BN) in the ResNet-18 experiments, denoted as **ResNet-18-BN**.

E.4 ViT-Tiny Architecture

We construct a lightweight Vision Transformer model, **ViT-Tiny**, specifically tailored for federated learning on the CIFAR-100 dataset. The design is based on the standard ViT architecture (Dosovitskiy et al., 2020), with modifications to accommodate the small input size and limited data per client.

Table 6: Hyperparameter configuration of ResNet-18 and ViT-Tiny (CIFAR-100 and Tiny ImageNet) across different algorithms.

Method	Local Optimizer	Local LR	β	β_1	β_2	Weight Decay
FedAvg (Local SGD)	SGD	0.1	—	—	—	0.001
SCAFFOLD	SGD	0.1	—	—	—	0.001
FedCM	SGD	0.1	0.9	—	—	0.001
Local AdamW	AdamW	3e-4	—	0.9	0.999	0.01
Local Sophia	Sophia	3e-4	—	0.9	0.99	0.01
Local Muon	Muon	3e-2	—	0.9	—	0.01
Local SOAP	SOAP	3e-3	—	0.95	0.95	0.01
FedPAC_Sophia	Sophia	3e-4	0.5	0.9	0.99	0.01
FedPAC_Muon	Muon	3e-2	0.5	0.9	—	0.01
FedPAC_SOAP	SOAP	3e-3	0.5	0.95	0.95	0.01

- **Input resolution:** 32×32
- **Patch size:** 4×4, resulting in 64 image-patch tokens (excluding any class token)
- **Embedding dimension:** 192
- **Number of Transformer layers:** 6
- **Number of attention heads:** 3
- **Normalization:** LayerNorm (applied before attention and MLP blocks)
- **Classification head:** Linear projection to 100 classes (CIFAR-100)
- **Activation:** GELU
- **Initialization:** Xavier/Glorot for linear layers; sinusoidal positional encoding

To regularize training, we apply dropout (0.1) to both attention and MLP layers. The ViT-Tiny models in these experiments are trained from scratch; the ViT-Base experiments below use pretrained initialization.

E.5 ViT-Base Fine-tuning

To demonstrate the effectiveness of our method on large-scale vision models, we conduct fine-tuning experiments using **ViT-Base** on **Tiny ImageNet** and **CIFAR-100**. For both datasets, we initialize ViT-Base from the same ImageNet-22K pretrained weights across methods (Dosovitskiy et al., 2020).

We fine-tune all layers under the Dir-0.1 partition. Optimizer settings are given in Table 7.

Data Preprocessing. To align with the input resolution required by ViT, we resize images from both datasets to 224×224 using bilinear interpolation. Standard data augmentation techniques such as random cropping, horizontal flipping, and RandAugment are applied locally at the client side.

E.6 C4 Language-Model Pre-training

We train LLaMA models with 60M, 130M, and 350M parameters on C4 for 100 communication rounds, using 20 clients and 20% participation (four clients per round). Performance is evaluated by validation loss; lower values are better.

Table 8 reports the optimizer hyperparameters.

Table 7: Hyperparameter configuration of ViT-Base fine-tuning across different algorithms.

Method	Local Optimizer	Local LR	β	β_1	β_2	Weight Decay
FedAvg (Local SGD)	SGD	0.1	—	—	—	0.001
SCAFFOLD	SGD	0.1	—	—	—	0.001
FedCM	SGD	0.1	0.9	—	—	0.001
Local AdamW	AdamW	1e-4	—	0.9	0.999	0.01
Local Sophia	Sophia	1e-4	—	0.9	0.99	0.01
Local Muon	Muon	1e-2	—	0.9	—	0.01
Local SOAP	SOAP	1e-3	—	0.95	0.95	0.01
FedPAC_Sophia	Sophia	1e-4	0.5	0.9	0.99	0.01
FedPAC_Muon	Muon	1e-2	0.5	0.9	—	0.01
FedPAC_SOAP	SOAP	1e-3	0.5	0.95	0.95	0.01

Table 8: Hyperparameter configuration of LLaMA on C4 across different algorithms.

Method	Local Optimizer	Local LR	β	β_1	β_2	Weight Decay
FedAvg (Local SGD)	SGD	0.1	—	—	—	0.001
SCAFFOLD	SGD	0.1	—	—	—	0.001
FedCM	SGD	0.1	0.9	—	—	0.001
Local AdamW	AdamW	3e-4	—	0.9	0.999	0.01
Local Sophia	Sophia	3e-4	—	0.9	0.99	0.01
Local Muon	Muon	3e-2	—	0.9	—	0.01
Local SOAP	SOAP	3e-3	—	0.95	0.95	0.01
FedPAC_Sophia	Sophia	3e-4	0.5	0.9	0.99	0.01
FedPAC_Muon	Muon	3e-2	0.5	0.9	—	0.01
FedPAC_SOAP	SOAP	3e-3	0.5	0.95	0.95	0.01

Table 9: Reported communication payload and per-round computation time on CIFAR-100 with ViT-Tiny (Dir-0.1, $R = 300$, $K = 50$). Payload ratios are relative to the 22.8 MB model-update baseline; accuracy is in percent.

Method / Strategy	Payload ratio	Payload (MB)	Time (s)	Accuracy (%)
FedAvg	1×	22.8	4.56	27.24
SCAFFOLD	2×	45.6	5.22	26.86
FedCM	1×	22.8	4.68	28.23
Local AdamW	1×	22.8	4.89	37.57
Local Sophia	1×	22.8	4.92	34.05
Local Muon	1×	22.8	5.14	44.00
Local SOAP	1×	22.8	5.56	49.41
FedPAC_Sophia	2×	45.6	5.08	39.79
FedPAC_Muon	2×	45.6	5.25	47.81
FedPAC_SOAP	3×	68.4	5.68	51.16
FedPAC_Sophia_Light	1.05×	23.9	5.11	39.45
FedPAC_Muon_Light	1.05×	23.9	5.28	47.23
FedPAC_SOAP_Light	1.10×	25.1	5.69	50.56

Table 10: Test accuracy of each method on CIFAR-100 and Tiny ImageNet using **ResNet-18** over 300 communication rounds under IID, Dir-0.5, Dir-0.1, and Dir-0.05 (100 clients, 10% participation, batch size 50, $K = 50$).

Method	**CIFAR-100 (ResNet-18)**				**Tiny ImageNet (ResNet-18)**			
	IID	Dir-0.5	Dir-0.1	Dir-0.05	IID	Dir-0.5	Dir-0.1	Dir-0.05
FedAvg	65.04	65.07	60.17	56.75	53.88	52.55	47.48	43.80
SCAFFOLD	66.04	65.51	60.69	56.43	54.49	53.36	47.76	43.92
FedCM	71.12	69.85	66.61	62.65	46.51	44.06	41.16	36.00
Local AdamW	64.37	63.58	59.23	55.24	51.06	49.97	44.01	40.00
Fed-Sophia	62.56	60.62	57.29	51.02	49.86	47.62	41.89	36.65
FedPM	62.86	61.23	57.25	50.98	49.63	48.21	41.53	36.62
Local Sophia	62.17	60.96	56.65	50.89	49.28	47.99	41.23	36.15
Local Muon	72.78	**73.02**	67.26	49.86	**60.50**	**60.40**	52.83	34.76
Local SOAP	71.98	71.19	68.44	58.16	58.01	56.82	54.42	50.02
`FedPAC_Sophia`	64.90	64.71	59.96	53.66	49.30	50.92	43.81	36.37
`FedPAC_Muon`	**72.79**	72.50	**71.85**	**65.56**	58.85	58.31	**57.95**	**54.00**
`FedPAC_SOAP`	69.79	71.05	69.25	64.16	56.57	56.22	55.62	51.81

F Additional Experimental Results

F.1 Communication and Computation Cost Analysis

The payload ratios in Table 9 describe the reported accounting relative to the model-update baseline; they are not universal optimizer-state size formulas. In particular, dense SOAP factors for an $m \times n$ block contain $m^2 + n^2$ entries, whereas the block itself contains mn parameters. Actual communication therefore depends on block shapes, compression, numerical precision, and the messages counted. A full communication budget must also account for server broadcasts, including the global correction direction. The nominal SOAP light payload is $1.10 \times 22.8 = 25.08$ MB, rounded to 25.1 MB. Its accuracy is 0.60 percentage points below full FedPAC_SOAP under Dir-0.1 (50.56 versus 51.16). The main-text communication ablation instead uses Dir-0.05; its accuracy and timing values should not be substituted into this table.

Table 11: Test accuracy (%) on CIFAR-100 and Tiny ImageNet using **ViT-Tiny** over 300 communication rounds under IID, Dir-0.5, Dir-0.1, and Dir-0.05 (100 clients, 10% participation, batch size 50, $K = 50$). Best results are highlighted in bold.

Method	CIFAR-100 (ViT-Tiny)				Tiny ImageNet (ViT-Tiny)			
	IID	Dir-0.5	Dir-0.1	Dir-0.05	IID	Dir-0.5	Dir-0.1	Dir-0.05
FedAvg	32.97	33.66	27.24	23.42	18.61	18.03	15.68	14.05
SCAFFOLD	32.29	33.01	26.86	23.23	18.32	18.36	15.70	14.21
FedCM	21.97	22.02	28.23	25.74	10.23	11.66	18.88	18.15
Local AdamW	41.27	41.21	37.57	36.06	27.94	26.82	24.31	21.35
Fed-Sophia	38.34	38.56	34.23	32.27	23.56	24.46	23.85	22.17
FedPM	38.56	38.75	34.29	32.62	23.86	24.51	23.89	22.85
Local Sophia	37.94	38.48	34.05	32.25	22.90	23.46	22.49	21.14
Local Muon	50.50	51.75	44.00	39.68	34.80	34.57	30.51	28.25
Local SOAP	52.85	51.61	49.41	41.68	34.81	33.91	33.30	30.36
`FedPAC_Sophia`	42.53	42.29	39.79	32.71	23.26	22.16	23.01	22.37
`FedPAC_Muon`	52.70	52.02	47.81	41.76	31.78	33.48	31.45	30.25
`FedPAC_SOAP`	**52.86**	**52.46**	**51.16**	**47.55**	**35.56**	**35.12**	**34.32**	**31.33**

F.2 Comparisons with Additional Baselines

ResNet-18 results. Table 10 reports the final *test accuracy* after 300 communication rounds on CIFAR-100 and Tiny ImageNet with ResNet-18 under increasing data heterogeneity (IID, Dir-0.5, Dir-0.1, Dir-0.05; 100 clients, 10% participation, batch size 50, and K=50 local steps). Most methods lose accuracy under stronger heterogeneity, although individual entries need not vary monotonically with the Dirichlet concentration.

Specialized second-order FL baselines. We additionally compare with **Fed-Sophia** and **FedPM**, beyond the local second-order baselines. As shown in Table 10, both specialized baselines still exhibit noticeable performance drops as heterogeneity increases. For example, on CIFAR-100, Fed-Sophia decreases from 60.62 (Dir-0.5) to 51.02 (Dir-0.05), and FedPM decreases from 61.23 (Dir-0.5) to 50.98 (Dir-0.05), showing that these baselines remain sensitive to heterogeneity. Accuracy comparisons alone do not identify the cause of this degradation.

FedPAC improves robustness under strong heterogeneity. Across both datasets, `FedPAC_Muon` yields the best or near-best accuracy under strong non-IID settings, demonstrating substantial robustness benefits. On CIFAR-100 with Dir-0.05, `FedPAC_Muon` achieves 65.56, improving over FedAvg (56.75) by 8.81 percentage points and over Local Muon (49.86) by 15.70 percentage points. On Tiny ImageNet with Dir-0.05, `FedPAC_Muon` reaches 54.00, outperforming FedAvg (43.80) by 10.20 percentage points and Local Muon (34.76) by 19.24 percentage points. These results align with our analysis that the server-side *alignment* and client-side *correction* in FedPAC explicitly mitigate the mismatch among client preconditioners, which becomes particularly harmful when data heterogeneity is severe.

FedPAC also complements Sophia-style methods. Compared with the specialized Sophia-style FL baselines, `FedPAC_Sophia` provides consistent gains on CIFAR-100 across all heterogeneity levels (e.g., 53.66 vs. 51.02/50.98 under Dir-0.05 for Fed-Sophia/FedPM). On Tiny ImageNet, `FedPAC_Sophia` exceeds both baselines under Dir-0.5 and Dir-0.1, but is lower under IID (49.30 versus 49.86/49.63) and Dir-0.05 (36.37 versus 36.65/36.62). Thus, improvements over these baselines are setting-dependent. These comparisons support the usefulness of alignment and correction in heterogeneous training, without implying uniform superiority across optimizers and data partitions.

Table 12: Controlled ablation of `FedPAC_SOAP` on CIFAR-100 under Dir-0.05. Test accuracy (%) is evaluated after 300 communication rounds with 100 clients, 10% participation, batch size 50, and $K = 50$ local steps. Alignment synchronizes SOAP covariance statistics; correction mixes the local direction with the previous global direction. Dashes indicate results or standard deviations not yet available.

ID	Variant	ResNet-18	ViT-Tiny
A	Local SOAP	58.16	41.68
B	Local SOAP, reduced local LR	56.64	40.23
C	Local SOAP + server momentum	59.65	42.68
D	Alignment only	62.05	44.56
E	Correction only	61.12	43.67
F	Full FedPAC	64.16	47.55

F.3 IID and Heterogeneity Comparisons

To further strengthen the empirical comparisons (especially on **IID** data) and to examine the robustness trend as heterogeneity increases, we report additional results under IID and Dirichlet partitions with varying concentration parameters. Tables 10 and 11 summarize the final *test accuracy* after 300 communication rounds on CIFAR-100 and Tiny ImageNet with **ResNet-18** and **ViT-Tiny**, respectively (100 clients, 10% participation, batch size 50, K=50).

ResNet-18 results. From Table 10, we observe that under IID (and mild heterogeneity such as Dir-0.5), several strong baselines can be competitive (e.g., `Local Muon` on CIFAR-100). However, as heterogeneity becomes severe (Dir-0.1 and Dir-0.05), methods that rely on purely local second-order states (e.g., `Local Muon`) or specialized second-order FL baselines (e.g., **Fed-Sophia** and **FedPM**) degrade markedly, consistent with sensitivity to client heterogeneity. In contrast, `FedPAC_Muon` remains consistently strong and achieves the best accuracy under the most heterogeneous regime on both datasets, demonstrating improved robustness under non-IID distributions.

ViT-Tiny results. Table 11 shows a similar trend for transformer-style backbones: while several baselines perform reasonably under IID/Dir-0.5, severe heterogeneity leads to substantial drops. Notably, `FedPAC_SOAP` achieves the highest reported mean accuracy across all heterogeneity levels on both CIFAR-100 and Tiny ImageNet with ViT-Tiny, suggesting that FedPAC can effectively stabilize curvature-aware training for modern architectures under client heterogeneity.

Disentangling alignment, correction, and momentum. Table 12 examines whether the gains of `FedPAC_SOAP` can be explained by a smaller local learning rate or server momentum. Reducing the local learning rate decreases accuracy from 58.16% to 56.64% on ResNet-18 and from 41.68% to 40.23% on ViT-Tiny. Adding server momentum provides modest gains of 1.49 and 1.00 percentage points, respectively, but remains below the full method by 4.51 and 4.87 points. Both proposed components improve upon Local SOAP: alignment alone yields gains of 3.89/2.88 points on ResNet-18/ViT-Tiny, while correction alone yields 2.96/1.99 points. Combining them achieves the highest accuracy, 64.16% and 47.55%, improving over alignment alone by a further 2.11 and 2.99 points. These results support the complementary benefits of state alignment and local correction and indicate that the observed gains are not reproduced by either learning-rate reduction or server momentum alone under the tested configurations.

Controlled ablation protocol. We compare six variants to distinguish the effects of local step size, server momentum, state alignment, and local correction. Variant A is the Local SOAP baseline. Variant B uses a reduced local learning rate $\eta_{\mathrm{red}} = (1-\beta)\eta$, matching the coefficient of the current local direction in FedPAC, without state synchronization or correction. Variant C retains the Local SOAP client updates and applies momentum only at the server:

$$q^r = -\frac{1}{SK\eta}\sum_{i\in S_r}(x_i^{r,K} - x^r), \qquad h^{r+1} = \beta h^r + (1-\beta)q^r, \qquad x^{r+1} = x^r - \gamma h^{r+1},$$

Family	**What is synchronized (server $\leftrightarrow$ clients)?**	**Extra cost (comm./state)**	**Main issue it targets**
FedAvg / Local-SGD	Model parameters/updates x only	$\approx \lvert x\rvert$ per round	General FL baseline; does not explicitly correct non-IID induced drift
FedOpt (server adaptive; e.g., FedAdam, FedYogi)	x only; optimizer state (moments) kept *on server* (not shared as a common client geometry)	$\approx \lvert x\rvert$ (no extra client-side sync)	Improves global update scaling/adaptivity, but does *not* align heterogeneous client metrics
Control variates / drift correction (e.g., SCAFFOLD)	x + first-order control variates (gradient-level states)	typically larger (e.g., SCAFFOLD uses $\approx 2\lvert x\rvert$ in our comm accounting)	Corrects *first-order* client drift under non-IID via control variates; operates at gradient level, not metric level
Naive second-order FL (Local Sophia, SOAP, or Muon with model averaging)	x only; each client maintains local preconditioner state Θ but *does not synchronize it*	$\approx \lvert x\rvert$ per round	Can suffer from heterogeneous local optimizer states under non-IID data
FedPAC (ours)	x + **optimizer state Θ used to construct update directions**: server aggregates and broadcasts a reference state (Alignment); local steps mix in a previous global direction (Correction)	$\lvert x\rvert + \lvert\Theta\rvert$ for a model update and the complete uploaded state; compressed variants replace $\lvert\Theta\rvert$ with the encoded payload size	**Targets preconditioner drift directly** by synchronizing the *metric* (Alignment) and suppressing long-term drift (Correction)

Table 13: Positioning of FedPAC. The table distinguishes model aggregation, server adaptivity, first-order control variates, and synchronization of local optimizer states. Here $\lvert x\rvert$ and $\lvert\Theta\rvert$ count the model-update and complete auxiliary-state entries in an upload; server broadcasts are additional. The synchronized state depends on the optimizer and is not necessarily an explicit positive-definite preconditioning operator.

where $h^0 = 0$ and $\gamma = K\eta$. Unlike local correction, this variant does not inject h^r into the client updates. Variant D synchronizes the SOAP covariance statistics without local correction. Variant E applies local correction without synchronizing the covariance statistics. Variant F combines both components. Controlled comparisons should use the same data partitions, initializations, client-sampling sequences, and evaluation protocol, with equal tuning budgets where tuning is performed.

G Comparison with Related Federated Methods

H Optimizer Instantiations

The following algorithms specify the matrix-block update cores for SOAP and Muon, together with their FedPAC variants. They retain the state persistence and update rules of the supplied pseudocode. Weight decay, learning-rate scheduling, and updates for parameters outside these matrix blocks are not expanded in the pseudocode. The model-averaging server update corresponds to the main-text choice $\gamma = K\eta$. The direction $\mathbf{\Delta}_G^r$ is a normalized previous-round model update, rather than a new global gradient evaluation. For $\beta = 0$, the correction term vanishes but state synchronization remains; this is an alignment-only variant, not the local-optimizer baseline.

These implementations are distinct from the idealized bounded, symmetric-positive-definite operator model used in the convergence analysis. In particular, finite-step Muon transformations and SOAP momentum updates are not automatically covered by that analysis.

H.1 SOAP and FedPAC_SOAP

SOAP transports its first moment exactly when changing bases. Its second-moment transport retains only a diagonal approximation and omits cross-coordinate second moments. In FedPAC_SOAP, synchronization resets the factors P_L, P_R; the bases are refreshed on the specified frequency schedule. Consequently, factor synchronization does not imply immediate equality of the applied update operators across clients.

Algorithm 3 Local SOAP with diagonal second-moment transport

Require: Global model $\boldsymbol{x}^0$; rounds R, local steps K; total clients N, participating clients $S \in \{1, \dots, N\}$; learning rate $\eta > 0$; EMA coefficients $\beta_1, \beta_2 \in [0, 1)$; $\epsilon > 0$, basis-update frequency $f \geq 1$.

Each client i initializes a persistent counter $\tau_i \leftarrow 0$.
For each $m \times n$ parameter block, initialize $P_{L,i} \leftarrow 0_{m \times m}$, $P_{R,i} \leftarrow 0_{n \times n}$, $M_i, V_i \leftarrow 0_{m \times n}$, $Q_{L,i} \leftarrow I_m$, and $Q_{R,i} \leftarrow I_n$.
Block indices are omitted. States are stored separately for each block.
for $r = 0, \dots, R-1$ **do**
  Server samples $\mathcal{S}_r \subseteq [N]$ uniformly without replacement, with $|\mathcal{S}_r| = S$.
  Server broadcasts $\boldsymbol{x}^r$ to participating clients.
  for each client $i \in \mathcal{S}_r$ in parallel **do**
    $\boldsymbol{x}_i^{r,0} \leftarrow \boldsymbol{x}^r$
    for $k = 0, \dots, K-1$ **do**
      Sample mini-batch $\xi_i^{r,k}$ and compute $\boldsymbol{g}_i^{r,k} \leftarrow \nabla F_i(\boldsymbol{x}_i^{r,k}; \xi_i^{r,k})$.
      $\tau_i \leftarrow \tau_i + 1$
      for each parameter block **do**
        Let G_i be the corresponding block of $\boldsymbol{g}_i^{r,k}$.
        $\widetilde{D}_i \leftarrow \text{SOAPBLOCK}(G_i, i, \tau_i, \beta_1, \beta_2, \epsilon, f)$
      end for
      Assemble the block directions $\widetilde{D}_i$ into $\boldsymbol{d}_i^{r,k}$.
      $\boldsymbol{x}_i^{r,k+1} \leftarrow \boldsymbol{x}_i^{r,k} - \eta \boldsymbol{d}_i^{r,k}$
    end for
    $\boldsymbol{\delta}_i^r \leftarrow \boldsymbol{x}_i^{r,K} - \boldsymbol{x}^r$
    Client i sends $\boldsymbol{\delta}_i^r$ to the server and retains all local optimizer states.
  end for
  $\boldsymbol{x}^{r+1} \leftarrow \boldsymbol{x}^r + \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} \boldsymbol{\delta}_i^r$
end for

function SOAPBLOCK($G_i, i, \tau_i, \beta_1, \beta_2, \epsilon, f$)
The states of the current block are updated in place.
  $P_{L,i} \leftarrow \beta_2 P_{L,i} + (1 - \beta_2) G_i G_i^\top$
  $P_{R,i} \leftarrow \beta_2 P_{R,i} + (1 - \beta_2) G_i^\top G_i$
  if $\tau_i \bmod f = 0$ **then**
    $Q_L^+ \leftarrow \text{EIGENBASIS}(P_{L,i})$, $Q_R^+ \leftarrow \text{EIGENBASIS}(P_{R,i})$
    $A_i \leftarrow (Q_L^+)^\top Q_{L,i}$, $B_i \leftarrow Q_{R,i}^\top Q_R^+$
    $M_i \leftarrow A_i M_i B_i$ ▷ Exact first-moment transport
    $V_i \leftarrow (A_i \odot A_i) V_i (B_i \odot B_i)$ ▷ Diagonal second-moment approximation
    $Q_{L,i} \leftarrow Q_L^+$, $Q_{R,i} \leftarrow Q_R^+$
  end if
  $g_i \leftarrow Q_{L,i}^\top G_i Q_{R,i}$
  $M_i \leftarrow \beta_1 M_i + (1 - \beta_1) g_i$
  $V_i \leftarrow \beta_2 V_i + (1 - \beta_2)(g_i \odot g_i)$
  $D_i \leftarrow M_i \oslash (\sqrt{V_i} + \epsilon)$
  return $Q_{L,i} D_i Q_{R,i}^\top$
end function
EIGENBASIS *returns an orthonormal eigenbasis of a symmetric matrix using the same deterministic ordering and tie-breaking convention in both SOAP variants. Nonparticipating clients retain all local states.*

Algorithm 4 `FedPAC_SOAP` with diagonal second-moment transport

Require: Global model $\boldsymbol{x}^0$; rounds R, local steps K; total clients N, participating clients $S \in \{1, \dots, N\}$; learning rate $\eta > 0$, mixing coefficient $\beta \in [0, 1)$; EMA coefficients $\beta_1, \beta_2 \in [0, 1)$; $\epsilon > 0$, basis-update frequency $f \geq 1$.

$\boldsymbol{\Delta}_G^0 \leftarrow \mathbf{0}$
For each $m \times n$ parameter block, initialize $P_{L,G}^0 \leftarrow 0_{m \times m}$ and $P_{R,G}^0 \leftarrow 0_{n \times n}$.
Each client i initializes a persistent counter $\tau_i \leftarrow 0$ and, for each block, $M_i, V_i \leftarrow 0_{m \times n}$, $Q_{L,i} \leftarrow I_m$, and $Q_{R,i} \leftarrow I_n$.
for $r = 0, \dots, R-1$ **do**
  Server samples $\mathcal{S}_r \subseteq [N]$ uniformly without replacement, with $|\mathcal{S}_r| = S$.
  Server broadcasts $(\boldsymbol{x}^r, \boldsymbol{\Delta}_G^r, \{P_{L,G}^r, P_{R,G}^r\})$ to participating clients.
  for each client $i \in \mathcal{S}_r$ in parallel **do**
    $\boldsymbol{x}_i^{r,0} \leftarrow \boldsymbol{x}^r$
    For each block, $P_{L,i} \leftarrow P_{L,G}^r$ and $P_{R,i} \leftarrow P_{R,G}^r$. ▷ Align second-order statistics
    for $k = 0, \dots, K-1$ **do**
      Sample mini-batch $\xi_i^{r,k}$ and compute $\boldsymbol{g}_i^{r,k} \leftarrow \nabla F_i(\boldsymbol{x}_i^{r,k}; \xi_i^{r,k})$.
      $\tau_i \leftarrow \tau_i + 1$
      for each parameter block **do**
        Let G_i be the corresponding block of $\boldsymbol{g}_i^{r,k}$.
        $\widetilde{D}_i \leftarrow \text{SOAPBLOCK}(G_i, i, \tau_i, \beta_1, \beta_2, \epsilon, f)$ ▷ Algorithm 3
      end for
      Assemble the block directions $\widetilde{D}_i$ into $\boldsymbol{d}_i^{r,k}$.
      $\boldsymbol{x}_i^{r,k+1} \leftarrow \boldsymbol{x}_i^{r,k} - \eta\big[(1-\beta)\boldsymbol{d}_i^{r,k} + \beta \boldsymbol{\Delta}_G^r\big]$
    end for
    $\boldsymbol{\delta}_i^r \leftarrow \boldsymbol{x}_i^{r,K} - \boldsymbol{x}^r$
    Client i sends $(\boldsymbol{\delta}_i^r, \{P_{L,i}, P_{R,i}\})$ to the server.
    Retain $\{M_i, V_i, Q_{L,i}, Q_{R,i}\}$ and τ_i for future participation.
  end for
  $\boldsymbol{\Delta}_G^{r+1} \leftarrow -\frac{1}{|\mathcal{S}_r| K \eta} \sum_{i \in \mathcal{S}_r} \boldsymbol{\delta}_i^r$
  $\boldsymbol{x}^{r+1} \leftarrow \boldsymbol{x}^r + \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} \boldsymbol{\delta}_i^r$
  For each block, $P_{L,G}^{r+1} \leftarrow \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} P_{L,i}$ and $P_{R,G}^{r+1} \leftarrow \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} P_{R,i}$.
end for

Algorithm 5 Local Muon under partial participation

Require: Global model $\boldsymbol{x}^0$; rounds R, local steps K; total clients N, participating clients $S \in \{1, \dots, N\}$; learning rate $\eta > 0$, momentum coefficient $\mu \in [0, 1)$; $\epsilon > 0$, Newton–Schulz steps $s \geq 1$; block scaling factor $\gamma(m, n) > 0$.

Each client i initializes a persistent momentum $M_i \leftarrow 0_{m \times n}$ for each $m \times n$ parameter block.
Block indices are omitted. Matrix operations are applied separately to each block.
for $r = 0, \dots, R-1$ **do**
  Server samples $\mathcal{S}_r \subseteq [N]$ uniformly without replacement, with $|\mathcal{S}_r| = S$.
  Server broadcasts $\boldsymbol{x}^r$ to participating clients.
  for each client $i \in \mathcal{S}_r$ in parallel **do**
    $\boldsymbol{x}_i^{r,0} \leftarrow \boldsymbol{x}^r$
    for $k = 0, \dots, K-1$ **do**
      Sample mini-batch $\xi_i^{r,k}$ and compute $\boldsymbol{g}_i^{r,k} \leftarrow \nabla F_i(\boldsymbol{x}_i^{r,k}; \xi_i^{r,k})$.
      for each $m \times n$ parameter block **do**
        Let G_i be the corresponding block of $\boldsymbol{g}_i^{r,k}$.
        $M_i \leftarrow \mu M_i + (1-\mu) G_i$
        $U_i \leftarrow \textsc{NewtonSchulz}(M_i, s, \epsilon)$
        $\widetilde{U}_i \leftarrow \gamma(m, n) U_i$
      end for
      Assemble the block directions $\widetilde{U}_i$ into $\boldsymbol{d}_i^{r,k}$.
      $\boldsymbol{x}_i^{r,k+1} \leftarrow \boldsymbol{x}_i^{r,k} - \eta \boldsymbol{d}_i^{r,k}$
    end for
    $\boldsymbol{\delta}_i^r \leftarrow \boldsymbol{x}_i^{r,K} - \boldsymbol{x}^r$
    Client i sends $\boldsymbol{\delta}_i^r$ to the server and retains $\{M_i\}$.
  end for
  $\boldsymbol{x}^{r+1} \leftarrow \boldsymbol{x}^r + \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} \boldsymbol{\delta}_i^r$
end for

function $\textsc{NewtonSchulz}(G, s, \epsilon)$
  $(m, n) \leftarrow \mathrm{shape}(G)$
  $X \leftarrow G / (\|G\|_F + \epsilon)$
  if $m > n$ **then**
    $X \leftarrow X^\top$
  end if
  for $j = 1, \dots, s$ **do**
    $A \leftarrow X X^\top$
    $X \leftarrow \frac{1}{2}(3I - A)X$ ▷ I matches the size of A
  end for
  if $m > n$ **then**
    $X \leftarrow X^\top$
  end if
  return X
end function
Nonparticipating clients retain their momentum states. Finite Newton–Schulz iterations provide an approximate matrix transformation.

H.2 Muon and FedPAC_Muon

FedPAC_Muon synchronizes the momentum matrices used by the local matrix transformation. These matrices need not be symmetric or positive definite. The finite Newton–Schulz loop below is an approximate transformation; its iteration count and block scaling must match the implementation.

Algorithm 6 `FedPAC_Muon` under partial participation

Require: Global model $\boldsymbol{x}^0$; rounds R, local steps K; total clients N, participating clients $S \in \{1, \dots, N\}$; learning rate $\eta > 0$, momentum coefficient $\mu \in [0, 1)$; mixing coefficient $\beta \in [0, 1)$; $\epsilon > 0$, Newton–Schulz steps $s \geq 1$; block scaling factor $\gamma(m, n) > 0$.

$\boldsymbol{\Delta}_G^0 \leftarrow \mathbf{0}$
For each $m \times n$ parameter block, initialize $M_G^0 \leftarrow 0_{m \times n}$.
Block indices are omitted. Matrix operations are applied separately to each block.
for $r = 0, \dots, R-1$ **do**
  Server samples $\mathcal{S}_r \subseteq [N]$ uniformly without replacement, with $|\mathcal{S}_r| = S$.
  Server broadcasts $(\boldsymbol{x}^r, \boldsymbol{\Delta}_G^r, \{M_G^r\})$ to participating clients.
  for each client $i \in \mathcal{S}_r$ in parallel **do**
    $\boldsymbol{x}_i^{r,0} \leftarrow \boldsymbol{x}^r$
    For each block, $M_i \leftarrow M_G^r$. ▷ Align momentum states
    for $k = 0, \dots, K-1$ **do**
      Sample mini-batch $\xi_i^{r,k}$ and compute $\boldsymbol{g}_i^{r,k} \leftarrow \nabla F_i(\boldsymbol{x}_i^{r,k}; \xi_i^{r,k})$.
      for each $m \times n$ parameter block **do**
        Let G_i be the corresponding block of $\boldsymbol{g}_i^{r,k}$.
        $M_i \leftarrow \mu M_i + (1-\mu) G_i$
        $U_i \leftarrow \textsc{NewtonSchulz}(M_i, s, \epsilon)$ ▷ Algorithm 5
        $\widetilde{U}_i \leftarrow \gamma(m, n) U_i$
      end for
      Assemble the block directions $\widetilde{U}_i$ into $\boldsymbol{d}_i^{r,k}$.
      $\boldsymbol{x}_i^{r,k+1} \leftarrow \boldsymbol{x}_i^{r,k} - \eta\big[(1-\beta)\boldsymbol{d}_i^{r,k} + \beta \boldsymbol{\Delta}_G^r\big]$
    end for
    $\boldsymbol{\delta}_i^r \leftarrow \boldsymbol{x}_i^{r,K} - \boldsymbol{x}^r$
    Client i sends $(\boldsymbol{\delta}_i^r, \{M_i\})$ to the server.
  end for
  $\boldsymbol{\Delta}_G^{r+1} \leftarrow -\frac{1}{|\mathcal{S}_r| K \eta} \sum_{i \in \mathcal{S}_r} \boldsymbol{\delta}_i^r$
  $\boldsymbol{x}^{r+1} \leftarrow \boldsymbol{x}^r + \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} \boldsymbol{\delta}_i^r$
  For each block, $M_G^{r+1} \leftarrow \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} M_i$.
end for

H.3 Comparison with Muon-Specific Federated Optimizers

Scope of existing methods. Two closely related studies investigate federated optimization specifically through Muon. Liu et al. (Liu et al., 2025) combine momentum aggregation with local–global update alignment to address data heterogeneity. Takezawa et al. (Takezawa et al., 2025) investigate the bias introduced by local linear minimization oracle (LMO) updates and develop a bias-corrected federated method, including an analysis of approximate LMO computation. Both methods are developed around Muon or its underlying matrix-transformation mechanism.

A framework spanning multiple optimizer families. FedPAC studies a broader problem: coordinating heterogeneous client-side optimization geometries across different families of preconditioned optimizers. Its framework combines synchronization of optimizer-specific states with local update correction using a previous global direction. We instantiate and evaluate this framework with three representative optimizers: Sophia, SOAP, and Muon. These optimizers employ different state representations and update transformations, ranging from diagonal curvature estimates to structured covariance statistics and momentum-based matrix orthogonalization. Thus, the empirical scope of FedPAC extends beyond Muon-specific federated optimization.

Relationship and contribution boundary. We acknowledge that the Muon instantiation shares the momentum-synchronization and local–global correction structure of Liu et al. (Liu et al., 2025). Our contribution therefore lies in the cross-optimizer formulation, optimizer-specific instantiations, and systematic evaluation of this framework, together with a drift-dependent analysis of heterogeneous preconditioning. For example, FedPAC_SOAP synchronizes left and right covariance statistics, whereas FedPAC_Muon synchronizes momentum matrices. These distinctions require specifying which states are communicated and how they affect subsequent local updates; the synchronized objects are not interchangeable across optimizers.

Scope of theoretical guarantees. Our empirical results support the framework across the three evaluated optimizers, while our convergence analysis concerns an idealized model with bounded, symmetric positive-definite preconditioning operators. Synchronization of optimizer states does not necessarily imply equality of the resulting update operators. Consequently, extending the guarantees to the full Sophia, SOAP, and Muon implementations requires additional optimizer-specific analysis.

Optimizer states, preconditioning operators, and update directions. We distinguish three objects that play different roles in federated optimization: (i) the internal optimizer state $\Theta_i^{r,k}$; (ii) a preconditioning operator $P_i^{r,k}$, when such an operator representation is applicable; and (iii) the actual update direction $d_i^{r,k}$. The optimizer state may contain momentum, curvature estimates, covariance factors, coordinate bases, and auxiliary statistics. It is therefore generally neither a matrix of a fixed shape nor a positive-definite operator.

At local step k of round r, client i computes

$$g_i^{r,k} = \nabla F_i(x_i^{r,k}; \xi_i^{r,k}), \tag{106}$$

updates its internal state, and constructs its update direction:

$$\Theta_i^{r,k+1} = \mathcal{U}\big(\Theta_i^{r,k}, g_i^{r,k}\big), \tag{107}$$

$$d_i^{r,k} = \mathcal{T}\big(\Theta_i^{r,k+1}, g_i^{r,k}\big). \tag{108}$$

Here $\mathcal{U}$ and $\mathcal{T}$ are optimizer-specific mappings. This notation describes implementations that update their states before constructing the current direction. A lagged-state implementation can instead use $\Theta_i^{r,k}$ in Eq. equation 108; the distinction must follow the actual implementation.

The local baseline applies

$$x_i^{r,k+1} = x_i^{r,k} - \eta_l d_i^{r,k}, \tag{109}$$

whereas FedPAC applies

$$x_i^{r,k+1} = x_i^{r,k} - \eta_l \left[(1-\beta) d_i^{r,k} + \beta h^r\right], \qquad \beta \in [0,1), \tag{110}$$

where h^r denotes the previous global update direction, corresponding to g_G^r in Algorithm 2.

State synchronization. FedPAC synchronizes a selected component of the optimizer state rather than necessarily communicating an explicit preconditioning operator. Write

$$\Theta_i^{r,k} = \big(A_i^{r,k}, B_i^{r,k}\big), \tag{111}$$

where $A_i^{r,k}$ contains the states selected for synchronization and $B_i^{r,k}$ contains the remaining client-local states. For the arithmetic state-aggregation rule considered here, the server computes

$$A^{r+1} = \frac{1}{|S_r|} \sum_{i \in S_r} A_i^{r,K}. \tag{112}$$

At the beginning of the next round, each participating client sets $A_i^{r+1,0} \leftarrow A^{r+1}$. The initialization or persistence of $B_i^{r+1,0}$ follows the corresponding optimizer-specific implementation.

We use *alignment* to refer to this synchronization of selected optimizer states. It provides a shared reference for subsequent local optimization, but does not generally imply identical preconditioning operators or update directions across clients.

H.4 Distinguishing Optimizer States, Operators, and Update Directions

Why the distinction matters. An optimizer state stores the information used to construct future updates. A preconditioning operator specifies a transformation applied to a vector. An update direction is the vector actually used to change the model parameters. These objects are related, but are not interchangeable. In particular, averaging optimizer states need not be equivalent to averaging their induced operators.

Suppose, for illustration, that an operator can be constructed from a state through a mapping $\mathcal{P}$. For a nonlinear $\mathcal{P}$, one generally has

$$\mathcal{P}\left(\frac{1}{|S_r|}\sum_{i\in S_r}\Theta_i\right) \neq \frac{1}{|S_r|}\sum_{i\in S_r}\mathcal{P}(\Theta_i). \tag{113}$$

Moreover, if only part of the state is synchronized, the resulting operators can remain different because they also depend on client-local states. FedPAC should therefore be understood as an optimizer-state synchronization and update-correction framework, with operator-level alignment requiring additional conditions.

SOAP: covariance statistics versus effective operators. For a matrix parameter block, SOAP maintains left and right covariance statistics, orthogonal bases, and first- and second-moment estimates in the corresponding basis. To distinguish covariance statistics from applied operators, denote the left and right statistics by $C_{L,i}$ and $C_{R,i}$. These are the quantities denoted by $P_{L,i}$ and $P_{R,i}$ in Algorithms 3 and 4.

In FedPAC_SOAP, the synchronized state consists of $(C_{L,i}, C_{R,i})$. The local first and second moments, bases, and update counter are retained as specified in Algorithm 4. Consequently, synchronizing the covariance statistics does not immediately synchronize the complete SOAP update.

This distinction can be seen from the block direction

$$D_i = Q_{L,i}\left[M_i \oslash \left(\sqrt{V_i}+\epsilon\right)\right] Q_{R,i}^\top, \tag{114}$$

where M_i and V_i are expressed in the local basis, and the square root and division are elementwise. Define

$$Q_i = Q_{R,i}\otimes Q_{L,i}, \qquad m_i = \mathrm{vec}\left(Q_{L,i}M_iQ_{R,i}^\top\right). \tag{115}$$

Conditioned on the current bases and second moments, Eq. equation 114 can be written as

$$\mathrm{vec}(D_i) = P_{\mathrm{eff},i}m_i, \tag{116}$$

$$P_{\mathrm{eff},i} = Q_i\,\mathrm{Diag}\left(\mathrm{vec}\left[\left(\sqrt{V_i}+\epsilon\right)^{-1}\right]\right)Q_i^\top. \tag{117}$$

Here the inverse inside vec is elementwise. The effective operator depends on both the bases and the second-moment estimates, not solely on the synchronized covariance statistics. Furthermore, its input is the momentum-derived vector m_i, rather than necessarily the current stochastic gradient. This representation therefore does not by itself establish that the full SOAP implementation satisfies the preconditioned-gradient model used in our analysis.

Muon: momentum states versus matrix transformations. For an $m\times n$ parameter block, the Muon instantiation maintains a momentum matrix

$$M_i^{r,k+1} = \mu M_i^{r,k} + (1-\mu)G_i^{r,k}, \tag{118}$$

and constructs the block direction

$$D_i^{r,k} = c_{m,n}\,\mathrm{NS}_s\left(M_i^{r,k+1}\right), \tag{119}$$

where NS_s denotes the finite-step Newton–Schulz transformation and $c_{m,n}$ is the block-scaling factor.

FedPAC_Muon synchronizes the momentum matrices used to initialize these local transformations. A momentum matrix can be rectangular and need not be symmetric or positive definite. It is therefore an optimizer state, rather than the positive-definite operator assumed in the theoretical model. In addition, the mapping from momentum to the final Muon direction is nonlinear. Accordingly, averaging momentum matrices generally differs from averaging their transformed directions.

Sophia: curvature estimates versus clipped updates. Sophia constructs updates using diagonal curvature estimates, momentum, and coordinatewise clipping. Its internal curvature estimate is distinct from the diagonal rescaling operator derived from that estimate, and both are distinct from the final clipped update. Synchronization of a curvature-related state therefore does not automatically synchronize the complete update: local momentum and the coordinates affected by clipping can still differ. The synchronized state components, estimator update schedule, and clipping rule must be specified for the particular Sophia implementation.

Interpretation of the empirical drift measurement. The SOAP drift measurement in our experiments compares client-side left and right covariance statistics with their aggregated reference statistics. It should therefore be interpreted as a *covariance-state drift* measurement. This diagnostic is distinct from the applied-operator drift used in the convergence analysis. A reduction in covariance-state drift indicates improved consistency of the synchronized statistics, but does not alone establish an equivalent reduction in effective-operator drift or update-direction disagreement.

Relating the two quantitatively requires additional regularity conditions. For example, if an induced operator map depends only on a state Θ and satisfies

$$\|\mathcal{P}(\Theta_i) - \mathcal{P}(\Theta)\|_2 \le L_{\mathcal{P}} \|\Theta_i - \Theta\|_{\text{state}}, \tag{120}$$

then squared state discrepancy bounds squared operator discrepancy up to $L_{\mathcal{P}}^2$. Such a condition is not automatic. When additional local states affect the operator, their discrepancies must also be controlled.

Relationship to the convergence analysis. Our theoretical analysis studies the idealized update

$$d_i^{r,k} = P_i^{r,k} g_i^{r,k}, \qquad mI \preceq P_i^{r,k} \preceq MI. \tag{121}$$

In this model, $P_i^{r,k}$ denotes the operator actually applied to the current gradient, irrespective of the indexing of an implementation's internal states. Operator-level round-start alignment, $P_i^{r,0} = P^r$, is a property of this theoretical model; it does not follow automatically from synchronization of only selected optimizer states.

The analysis characterizes how applied-operator drift interacts with gradient heterogeneity and stochastic noise. Its conclusions apply to an implementation only when the implementation's actual update direction and state evolution satisfy the stated assumptions. Momentum, clipping, basis changes, and nonlinear matrix transformations require additional analysis where applicable. Thus, the framework's empirical scope across Sophia, SOAP, and Muon is broader than the scope of the idealized convergence guarantees.