DISTRIBUTED LEARNING WITH SELECTIVE STATE SPACE MODELS: ARCHITECTURE-AWARE CONVERGENCE ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern state space models (SSMs), such as Mamba2, provide a compelling alternative to transformers by combining linear-time sequence modeling with recurrent state-space dynamics. However, the behavior of SSMs in distributed learning settings remains poorly understood. In particular, the existing standard federated learning methods are largely architecture-agnostic, and do not account for the stability, selectivity, and state-space parameterization that characterize modern selective SSMs. To address this, we derive architecture-aware gradient and smoothness bounds for single- and multi-layer selective SSMs, and convergence bounds for FedAvg and FedProx, characterizing how recurrent stability, input-dependent discretization, and state projection norms affect federated optimization. We then numerically validate the single-layer bounds on sequences generated by a teacher SSM, using a learner that follows the analyzed recurrence. We use this analysis to formulate expectations about the effects of local training and client heterogeneity, and examine these expectations by comparing nine federated learning algorithms on Mamba2 language modeling across six text domains. These experiments illustrate how SSM-specific bounds can provide a basis for interpreting the behavior of practical federated learning algorithms.

1 INTRODUCTION

The modern generative AI landscape is largely dominated by transformer architecture (Vaswani et al., 2017), with the majority of the state-of-the-art large language models (LLMs) employing them (Agarwal et al., 2025; Team et al., 2026a; Qiu et al., 2026; Team et al., 2026b). Despite this widespread success, transformers face critical limitations in certain scenarios, often driven by the core efficiency bottleneck of their self-attention mechanism, scaling quadratically as $\mathcal{O}(L^2)$ with sequence length L in both time and memory complexity (Tay et al., 2022). To address this inefficiency, a variety of alternative mechanisms and architectures have been proposed, including sparse attention (Child et al., 2019; Zaheer et al., 2020), linear attention approximations (Katharopoulos et al., 2020; Choromanski et al., 2021), and modernized recurrent frameworks (Peng et al., 2023). Among these innovations, state space models (SSMs) (Gu et al., 2022a) have emerged as one of the most promising new paradigms. These modern SSMs, such as Mamba (Gu and Dao, 2024), overcome the quadratic bottleneck by operating with a strictly linear-time complexity, $\mathcal{O}(L)$, during training, and a constant-sized latent state, $\mathcal{O}(1)$.

Driven by these computational advantages, the SSM architecture has undergone rapid development, achieving competitive performance at a reduced computational cost compared to transformers. Architectural innovations, such as the Structured State Space (S4) sequence model, HiPPO initialization (Gu et al., 2022a), and selective mechanisms (Gu and Dao, 2024) have enabled SSMs to dynamically filter information and capture and retain long-range dependencies. Most recently, SSMs have demonstrated capacity for achieving state-of-the-art language modeling performance at scale (Lahoti et al., 2026; Zuo et al., 2024; Blakeman et al., 2025; Team et al., 2024).

Research Gap. Despite these advancements, the deployment of SSMs in *distributed* learning environments, where training is conducted across a set of agents, remains largely unexplored. While some studies have considered the intersection of SSMs with federated learning (FL) (McMahan et al., 2017),

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.

most of these frameworks are *architecture-agnostic*. Conversely, existing SSM-specific distributed learning studies either focus on classical, non-selective SSM settings (Keçeci et al., 2025), or use SSMs for specialized personalized-FL applications (Zhang et al., 2026), leaving the convergence behavior of modern selective SSMs largely uncharacterized. Developing a foundational understanding of selective SSM convergence behavior in distributed settings can enable architecture-aware parameter tuning, protocol selection, and other control tasks for training performance optimization, uncovering another advantage of SSMs over transformer LLM approaches that rely on architecture-agnostic optimization tools (Fang et al., 2026; Soro et al., 2026). We thus pose the following question:

How can we leverage the structural properties of modern selective SSMs to derive architecture-aware convergence bounds that rigorously characterize training behavior in terms of core state space parameters for distributed and federated learning techniques?

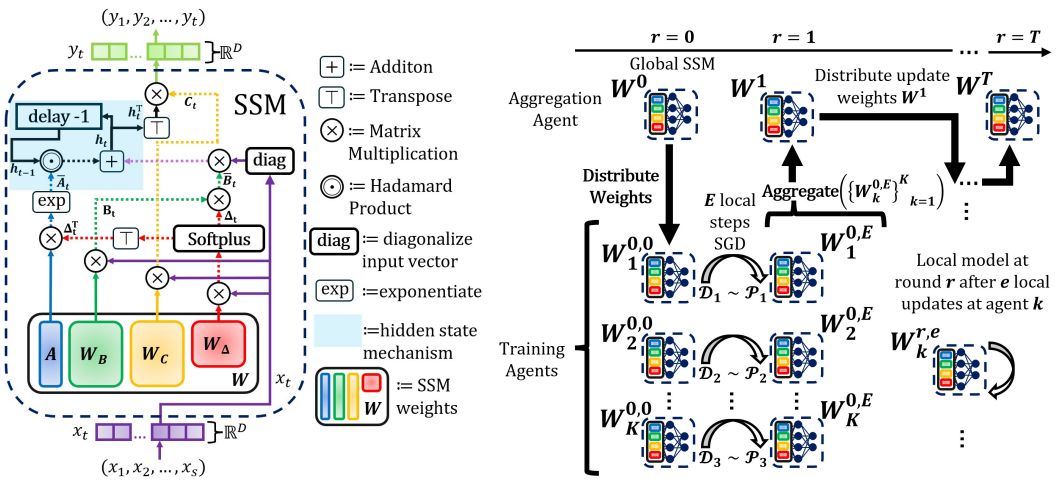
Contributions. In answering this question, our main contributions are summarized as follows:

- **Convergence analysis for distributed selective SSMs:** We provide an architecture-aware gradient norm bound for selective SSMs, revealing the impact of SSM parameters like core state-space parameter blocks. Then, we use this bound to establish the convergence rate of popular FL algorithms training selective SSMs, showing dependence on standard distributed learning factors like smoothness as well as architecture-specific parameters like block-wise spectral-norm envelopes. To the best of our knowledge, our work is the first to provide an architecture-aware convergence rate of FL algorithms for selective SSMs.
- **SSM theory validation and theory-motivated algorithm comparison**: We validate our theoretical results using teacher-model derived sequences and single-layer selective SSMs, observing no gradient-bound violations, increased client disagreement as ρ approaches one, and increased drift with more local steps. We use the theoretical takeaways to rank federated algorithms on heterogeneous-domain datasets. Our results show that these rankings highly correlate with validation performance, supporting the practical relevance of the SSM-specific quantities identified by our analysis.

2 Related Work

State Space Models (SSMs). SSMs originate from classical control theory concepts, where they model dynamic systems through the evolution of latent states over time (Kalman, 1960). The modern SSMs were introduced with the pivotal Structured State Space (S4) (Gu et al., 2022a) and its variants (Gu et al., 2022b; Hasani et al., 2023; Gu et al., 2020), and leverage structured parameterizations to efficiently capture long-range dependencies. Selective SSMs introduced input-dependent state-space parameters to dynamically filter information (Gu and Dao, 2024) while the Structured State Space Duality (SSD) established connections with linear attention and enabled efficient implementations (Dao and Gu, 2024), enabling better expressivity, training stability, and hardware-aware optimizations (Hwang et al., 2024; Kim et al., 2026; Lahoti et al., 2026). These advances have driven growing adoption across language modeling (Waleffe et al., 2024; Blakeman et al., 2025; He et al., 2025; Glorioso et al., 2024; Ren et al., 2025), computer vision (Li et al., 2024; Zhu et al., 2024; Xie et al., 2025), and other sequence modeling applications (Ma et al., 2025).

Distributed Learning with SSMs. Distributed learning enables model training across computational nodes, with federated learning (FL) addressing settings where data remain distributed among clients. Since the introduction of a foundational federated learning algorithm - FedAvg (McMahan et al., 2017) - numerous methods have been proposed to address statistical heterogeneity, communication costs, and client drift (Li et al., 2020; Karimireddy et al., 2020; Wang et al., 2020; Reddi et al., 2021; Acar et al., 2021; Li et al., 2021). However, despite the growing adoption of SSMs, theoretical understanding of their behavior in these settings remains limited. The existing SSM-related approaches address specific challenges: FedAlign (Keçeci et al., 2025) tackles the problem of models converging to different parameter basins in federated system identification, while FedSSM (Zhang et al., 2026) uses SSMs as a meta-learning tool within personalized FL. Related works also explore merging operations on hidden states, contexts, or representations (Liu et al., 2025; Pióro et al., 2024), rather than weight-space aggregation of independently trained selective SSMs. As a result, there remains an open question of how selective SSM dynamics affect optimization and aggregation guarantees in distributed learning settings. This motivates our study of connecting the architectural properties of selective SSMs to their behavior in distributed training.



(a) Selective SSM layer (b) Distributed SSM training

Figure 1: (a) A selective SSM layer generates input-dependent discretization parameters from x_t and uses them to update the hidden state h_t and output y_t. (b) Federated SSM training proceeds over communication rounds: the server broadcasts W^r, clients perform E local updates on local datasets $\mathcal{D}_k$, and the resulting models are aggregated into W^{r+1}.

3 System Model

State Space Model. We consider a decentralized learning environment operating over a global selective state space model (SSM) parameterized by $W = (W_B, W_C, W_\Delta, A)$ where $W_\Delta \in \mathbb{R}^{D\times D}$, $W_B \in \mathbb{R}^{N\times D}$, $W_C \in \mathbb{R}^{N\times D}$, and $A \in \mathbb{R}^{N\times 1}$, with A representing diagonal entries of a structured matrix $\mathcal{A} = \text{diag}(A)$. The SSM architecture is inspired by classical control theory concepts, in particular modeling dynamic systems through the evolution of latent states over time. These systems map a continuous input signal $x(t) \in \mathbb{R}$ to an output sequence $y(t) \in \mathbb{R}$ through an evolving, compressed latent state $h(t) \in \mathbb{R}^N$ using linear time-invariant (LTI) differential equations:

$$h'(t) = \mathcal{A}h(t) + Bx(t) \qquad \text{(1a)} \qquad\qquad y(t) = Ch(t) \qquad \text{(1b)}$$

However, since neural networks process discrete sequences, these continuous-time SSM dynamics are mapped to a discrete recurrence using a discretization rule. Under this discrete formulation, the model maps an input sequence $x = (x_1, \ldots, x_S)$ where $x_t \in \mathbb{R}^{D\times 1}$ to a sequence of outputs $y_t \in \mathbb{R}^{D\times 1}$ through a sequence of compressed latent states $h_t \in \mathbb{R}^{N\times D}$. While earlier foundational models utilized various discretization rules (Gu et al., 2022a), we adopt the first-order exponential-Euler discretization used in the more recent architectures (Lahoti et al., 2026). This discretization scheme retains the exponential state transition while applying a first-order approximation to the input contribution. We analyze the discrete model defined by this recurrence.

In addition to this discrete foundation, modern SSMs introduce a selective mechanism that enables dynamic filtering of relevant information across the input sequence context. The selective SSMs achieve this by making the state-space and time-scale parameters depend on the current input. Specifically, given the model weights W, the input-dependent SSM quantities corresponding to an input token x_t at the sequence position t are defined as:

- **Time-scale parameter:** $\Delta_t = \text{Softplus}(W_\Delta x_t)$, where $\Delta_t \in \mathbb{R}^{D\times 1}$, acts as a gating mechanism, controlling the discretization step size to determine importance of the current token.
- **State transition matrix:** $\bar{A}_t = \exp(A\Delta_t^\top)$, where $\bar{A}_t \in \mathbb{R}^{N\times D}$, determines the retention and decay of the historical context, managing the system's memory across the sequence.
- **Input matrix:** $\bar{B}_t := B_t\Delta_t^\top = (W_B x_t)(\text{Softplus}(W_\Delta x_t))^\top$, where $\bar{B}_t \in \mathbb{R}^{N\times D}$, and $B_t := W_B x_t$. Under the adopted exponential-Euler discretization, this matrix modulates how strongly the current input vector influences the newly updated latent state.
- **Output matrix:** $C_t = W_C x_t$, where $C_t \in \mathbb{R}^{N\times 1}$, linearly projects the compressed latent state back into the target output space.

Given these dynamically generated matrices, the discrete hidden state sequence h_t and the corresponding output y_t at each sequence step are computed according to the following recurrence:

$$h_t = \bar{A}_t \odot h_{t-1} + \bar{B}_t \text{diag}(x_t) \quad \text{(2a)} \qquad y_t = h_t^\top C_t, \quad \text{(2b)}$$

where $h_0 = \mathbf{0}_{N\times D}$, the exponential and Softplus functions are applied element-wise, and $\odot$ denotes the Hadamard product.

With the architecture specified above, we next define the local optimization objective. Let $\xi = (x, y^*)$ denote an input-target sequence pair, where $x = (x_1, \ldots, x_S)$ is the input sequence and $y^* = (y_1^*, \ldots, y_S^*)$ is the corresponding target sequence. The sequence loss is then the average token loss, defined as:

$$\mathcal{L}^S(W; \xi) := \frac{1}{S} \sum_{t=1}^{S} \ell_t\left(y_t(W; x), y_t^*\right) \tag{3}$$

where $y_t(W; x)$ is the model prediction at position t parameterized by W, and ℓ_t is some differentiable loss function. When the sample ξ is fixed, we write $\mathcal{L}^S(W)$ and $\ell_t(y_t)$ for notational simplicity. Figure 1a illustrates the resulting architecture, the parameter transformations, and state recurrence.

Distributed Optimization Environment. Given the selective SSM setup established above, we consider the deployment of this architecture in a decentralized learning setting. In this framework, a central server initializes a global selective SSM model parameterized by weights W^0, which is collaboratively trained across a network of K clients. Each client $k \in \{1, \ldots, K\}$ possesses a local dataset $\mathcal{D}_k = \{\xi_{k,i} \sim \mathcal{P}_k\}_{i=1}^{|\mathcal{D}_k|}$, where $\mathcal{P}_k$ denotes the local data distribution of client k and each sample $\xi_{k,i} = (x_{k,i}, y_{k,i}^*)$ represents a discrete input sequence and its corresponding target sequence. For autoregressive next-token prediction, the target at each sequence position satisfies $y_t^* = x_{t+1}$.

Given this setup, the framework aims to construct a global model that incorporates the combined knowledge of these diverse local distributions. Formally, the global objective is to find the optimal global model parameters W^* that minimize the aggregated expected sequence modeling loss:

$$W^* = \arg\min_W f(W), \quad \text{where} \quad f(W) = \sum_{k=1}^{K} p_k \mathcal{L}_k(W) \tag{4}$$

where p_k are positive weights satisfying $\sum_{k=1}^{K} p_k = 1$, and $\mathcal{L}_k(W) = \mathbb{E}_{\xi\sim\mathcal{P}_k}\left[\mathcal{L}^S(W; \xi)\right]$ is the expected sequence loss for client k over its local distribution.

To solve this optimization problem, the framework employs a decentralized training procedure initialized with the global model W^0. The optimization then proceeds over discrete communication rounds $r \in \mathcal{T} := \{0, 1, \ldots, T-1\}$. At each round r, the server shares the current global parameters W^r with the participating clients. Each client k initializes its local model as $W_k^{r,0} = W^r$ and performs E local steps of stochastic gradient descent (SGD) with step size η to minimize a local objective $F_k(W)$. The specific formulation of $F_k(W)$ depends on the distributed learning algorithm employed. For instance, in the classical FedAvg algorithm (McMahan et al., 2017), the local objective of client k is the expected sequence loss, i.e., $F_k(W) := \mathcal{L}_k(W)$. In contrast, FedProx (Li et al., 2020) modifies the local objective as $F_k^r(W) := \mathcal{L}_k(W) + \frac{\mu}{2}\|W - W^r\|_F^2$, with $\mu > 0$, to mitigate local divergence caused by data heterogeneity.

Through this local optimization step, the features of different distributions $\mathcal{P}_k$ are implicitly embedded into the updated local weights, resulting in a set of specialized models $W_k^{r,E}$. Following the completion of the E local training steps, the server combines these localized models into a unified global representation using a generalized aggregation function: $W^{r+1} = \text{Aggregate}\left(\left\{W_k^{r,E}\right\}_{k=1}^{K}; W^r\right)$. In standard distributed baselines such as FedAvg and FedProx, this aggregation is implemented as a simple data-weighted linear average: $W^{r+1} = \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{\sum_{j=1}^{K}|\mathcal{D}_j|} W_k^{r,E}$. By repeating this process over T communication rounds, the framework iteratively refines the global model toward an optimal parameter set W^* by aggregating information learned across the participating clients. Figure 1b illustrates this iterative training process.

4 Theoretical Analysis

Given this decentralized SSM setup, we consider distributed training procedures that update a global model through repeated local adaptation and server-side aggregation. These procedures induce a global optimization trajectory $\{W^r\}_{r=0}^{T}$ over the objective defined in (4). Since f is generally

nonconvex, we characterize convergence through the standard expected gradient stationarity criterion $\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\left\|\nabla f(W^r)\right\|_F^2$.

The central problem is to derive convergence bounds that explicitly capture the structure of selective SSMs, rather than relying solely on architecture-agnostic smoothness, variance, and heterogeneity constants. In particular, the recurrent state update induce dependencies between the local optimization trajectory and SSM-specific quantities including the stability of the discretized transition matrices, the input-dependent time scale parameter $\Delta_t = \mathrm{Softplus}(W_\Delta x_t)$, and the norms of the state-space projection parameters. Our convergence analysis characterizes how these SSM-specific quantities control gradient growth, client drift, and aggregation error.

As such, in the remainder of this section, we establish theoretical guarantees for federated training of selective SSMs. We first derive an explicit gradient norm bound for a single-layer selective SSM and use it to characterize federated optimization in terms of SSM-specific parameters. We provide convergence guarantees for FedAvg and FedProx, and relate additional federated baselines to the resulting stability, parameter-sensitivity, and distribution-shift quantities. The main text highlights the key results and their implications, while full proofs, supplementary corollaries, and additional baseline analyses are provided in Appendices A, B, C, D, E.

4.1 Gradient Norm Bound for Selective SSMs

The main technical challenge lies in controlling the parameter sensitivity induced by the input-dependent SSM recurrence. The loss gradient depends on the model parameters both directly through the output projection and indirectly through their influence on the recurrent hidden states. By bounding these two contributions separately and subsequently combining them, we derive a sequence-wise gradient bound that is independent of the sequence length.

We first introduce the notation used in our analysis. An input-target pair ξ is admissible if $\xi \in \mathcal{Z}$, where $\mathcal{Z}$ is a fixed, parameter-independent measurable set of input-target sequences satisfying $\Pr_{\xi\sim\mathcal{P}_k}(\xi \in \mathcal{Z}) = 1$ for all clients k. All conditions below hold uniformly for admissible sequences, finite lengths $S \geq 1$, and token indices $t \leq S$, with finite constants independent of S, ξ, and t. Next, we state the necessary definitions and assumptions on the loss and the input sequence.

Definition 4.1 (SSM Stability)**.** For $\gamma, \delta > 0$, define $\mathcal{S}_{\gamma,\delta}$ as the set of parameter tuples W satisfying $A_n \leq -\gamma$ for every $n \in \{1, \ldots, N\}$, where A_n is the n-th element of the transition parameter A, and $[\mathrm{Softplus}(W_\Delta x_t)]_d \geq \delta$ for every feature coordinate $d \in \{1, \ldots, D\}$ and admissible input token. Consequently, every entry of $\bar{A}_t$ satisfies: $\left(\bar{A}_t\right)_{n,d} \leq \rho := \exp(-\gamma\delta) < 1$.

Assumption 4.2 (Output loss regularity)**.** The token loss $\ell_t(\cdot, y_t^*)$ is twice continuously differentiable with respect to its prediction argument. For the parameter tuple W under consideration, there exist finite constants $L_\ell, H_\ell > 0$ such that, for every admissible ξ, the prediction $y = y_t(W; x)$ satisfies

$$\|\nabla_y \ell_t(y, y_t^*)\|_2 \leq L_\ell, \qquad \|\nabla_y^2 \ell_t(y, y_t^*)\|_{\mathrm{op}} \leq H_\ell. \tag{5}$$

Note that for softmax cross-entropy over logits, the Hessian bound holds globally.

Assumption 4.3 (Bounded Inputs)**.** There exists $L_x > 0$ such that $\forall t$, $\|x_t\|_2 \leq L_x$ $\forall$ admissible ξ.

Theorem 4.4 (SSM Gradient Bound)**.** *Suppose Assumptions 4.2 and 4.3 hold for a parameter tuple $W \in \mathcal{S}_{\gamma,\delta}$. For a single-layer SSM defined in Section 3, the following bound holds for every finite integer $S \geq 1$ and every admissible length-S sequence pair $\xi \in \mathcal{Z}$:*

$$\left\|\nabla_W \mathcal{L}^S(W; \xi)\right\|_F \leq \mathcal{G}_{SSM}(W) := L_\ell L_x \left(K_B K_\Delta \tfrac{\sqrt{ND}}{1-\rho} + \|W_C\|_2 \tfrac{K_W}{1-\rho}\right) \tag{6}$$

with

$$K_\Delta := \|W_\Delta\|_2 L_x + \sqrt{D}\log(2), \quad K_B := \|W_B\|_2 L_x^2, \tag{7}$$

$$K_W := NDK_BK_\Delta \left(K_\Delta + \sqrt{D}\|A\|_2 L_x\right)\left(\tfrac{\rho}{1-\rho}\right) + L_x^2\sqrt{N}K_\Delta + L_x\sqrt{D}K_B \tag{8}$$

where $\|\cdot\|_F$ is the Euclidean norm of the block Frobenius norms.

Discussion. The key technical step is to bound both the hidden states and their parameter Jacobians under input-dependent transitions. The stability condition $\rho = \exp(-\gamma\delta) < 1$ allows us to reduce the recurrent contributions to geometric sums, yielding an explicit parameter-gradient bound rather than requiring such a bound as an assumption. The resulting architecture-aware bound is independent of

the sequence length while retaining explicit dependence on ρ and the core selective SSM parameter blocks W_Δ, W_B, W_C, and A. The complete proof is provided in Appendix A. Appendix C extends the analysis to the multi-layer setting by combining stability with layerwise input-output sensitivities, resulting in the bound $\mathcal{G}^{(Q)}_{\mathrm{SSM}}(W)$ stated in Theorem C.3. Together, these results provide architecture-specific control of the gradient norm required for the subsequent federated convergence analysis.

4.2 Convergence Analysis

To characterize federated optimization with selective SSMs, we introduce a local parameter neighborhood and impose the stochastic-gradient conditions required for the convergence analysis.

Definition 4.5 (R-Local Neighborhood)**.** For a reference model $\widetilde{W}$ and radius $R > 0$, define $\mathcal{N}_R(\widetilde{W}) := \{W : \|W - \widetilde{W}\|_F \le R\}$, where the $\|\cdot\|_F$ is defined for a parameter tuple W as $\|W\|_F := \left(\|W_\Delta\|_F^2 + \|W_B\|_F^2 + \|W_C\|_F^2 + \|A\|_F^2\right)^{1/2}$.

In the following convergence analysis, the regularity conditions of Assumption 4.2 are assumed to hold uniformly over each neighborhood $\mathcal{N}_R(W^r)$. In particular, the same finite constants L_ℓ and H_ℓ apply to every $W \in \mathcal{N}_R(W^r)$, every admissible input-target pair, and every token position. The stability margins γ, δ and input bound L_x are likewise fixed throughout the neighborhood.

Assumption 4.6 (Unbiased stochastic gradients)**.** For every client k and every stable local model neighborhood $\mathcal{N}_R(W^r) \subseteq \mathcal{S}_{\gamma,\delta}$, define

$$g_k(W;\xi) := \nabla_W \mathcal{L}^S(W;\xi) \tag{9}$$

as the stochastic sequence-level gradient for a sampled sequence $\xi \sim \mathcal{P}_k$. Then, $\forall\, W \in \mathcal{N}_R(W^r)$

$$\mathbb{E}_{\xi\sim\mathcal{P}_k}\left[g_k(W;\xi)\right] = \nabla_W \mathcal{L}_k(W) \tag{10}$$

holds. At each local update, conditional on all preceding training randomness, the mini-batch consists of Ξ independent sequences drawn from $\mathcal{P}_k$. The mini-batch gradient is their average. Batches drawn by different clients at the same local step are conditionally independent.

Lemma 4.7 (Local SSM smoothness)**.** *Let W^r be a fixed reference parameter tuple and let $0 < R < \infty$ satisfy $\mathcal{N}_R(W^r) \subseteq \mathcal{S}_{\gamma,\delta}$. Suppose Assumptions 4.2, 4.3, and 4.6 hold, with finite and differentiable expected local objectives $\mathcal{L}_k$ on this neighborhood. Furthermore, let the bounds in Assumption 4.2 hold with the same finite constants L_ℓ and H_ℓ $\forall\, W \in \mathcal{N}_R(W^r)$, every admissible input-target sequence pair $\xi \in \mathcal{Z}$, and every token position, independently of sequence length.*

Then, for every client k, $\mathcal{L}_k$ is $L_{\mathrm{SSM},R}$-smooth on $\mathcal{N}_R(W^r)$. Specifically, for all $W, V \in \mathcal{N}_R(W^r)$,

$$\|\nabla_W \mathcal{L}_k(W) - \nabla_W \mathcal{L}_k(V)\|_F \le L_{\mathrm{SSM},R}\|W - V\|_F, \tag{11}$$

where $L_{\mathrm{SSM},R}$ is defined in (129)*.*

With these conditions in place, we characterize the convergence of FedAvg for selective SSMs. The corresponding analysis for FedProx and supplementary corollaries are available in Appendix B, D.

Corollary 4.8 (Single-layer FedAvg convergence rate)**.** *Under Assumptions 4.2- 4.6, the conditions specified in Definitions 4.1- 4.5, and the conditions of Theorem B.5, let M_B, M_C, M_Δ, and M_A denote uniform upper bounds on $\|W_B\|_2$, $\|W_C\|_2$, $\|W_\Delta\|_2$, and $\|A\|_2$, respectively, over the R-local neighborhoods $\mathcal{N}_R(W^r)$, and define*

$$M_R := (1 + M_B)(1 + M_C)(1 + M_\Delta)(1 + M_\Delta + M_A).$$

We further define $\bar{\mathcal{P}} := \sum_{k=1}^K p_k \mathcal{P}_k$, $\mathcal{H}_{\mathrm{TV}} := \sum_{k=1}^K p_k\, \mathrm{TV}(\mathcal{P}_k, \bar{\mathcal{P}})^2$, where TV *denotes the total variation distance. Under these conditions, for FedAvg applied to the single-layer selective SSM defined in Section 3, the global optimization trajectory $\{W^r\}_{r=0}^T$ satisfies*

$$\frac{1}{T}\sum_{r=0}^{T-1} \mathbb{E}\,\|\nabla f(W^r)\|_F^2 = \mathcal{O}\left(\frac{f\left(W^0\right) - f\left(W^*\right)}{\eta E T} + \frac{\eta M_R^4}{\Xi(1-\rho)^8} + \frac{\eta^2 M_R^6 (E-1)}{(1-\rho)^{12}}\left(\frac{1}{\Xi} + E\mathcal{H}_{\mathrm{TV}}\right)\right).$$

The quantities N, D, L_x, L_ℓ, and H_ℓ affect only the implicit constant and do not alter the displayed scaling with the optimization parameters, parameter envelopes, stability, or sequence length.

Proof sketch and discussion. Standard FedAvg convergence bounds assume smoothness and bounds on stochastic gradient variance and client-gradient disagreement. In contrast, our analysis derives these quantities from the selective SSM recurrence. We first bound the hidden states and their parameter derivatives to obtain the gradient bound. The second-order analysis then establishes local smoothness, while the remaining lemmas translate the gradient bound into bounds on stochastic

variance and client disagreement. Together, these results yield a convergence rate whose model dependence is explicit in the parameter envelopes M_B, M_C, M_Δ, M_A and the stability factor ρ, with dimensions and input and loss regularity constants fixed. As a result, the corollary identifies which SSM parameters enter the convergence guarantee without requiring a separately assumed smoothness or gradient-dissimilarity constant for the model.

The derived bound distinguishes sequence length from recurrent memory: the gradient and smoothness estimates contain no explicit sequence-length factor, but increase as the recurrent stability factor ρ approaches one, allowing past state information to persist for longer. This dependence is stronger in the additional local-drift terms, which contain $M_R^6/(1-\rho)^{12}$, as opposed to the baseline stochastic term, which contains $M_R^4/(1-\rho)^8$. Consequently, the bound assigns a larger penalty to performing multiple local updates when ρ is close to one. However, ρ alone does not determine this penalty: even for the same ρ and client distributions, larger parameter envelopes increase the coefficient multiplying $\mathcal{H}_{\mathrm{TV}}$. The analysis therefore separates the amount of distribution mismatch from the influence of the SSM parameters on its contribution to the convergence bound. This heterogeneity contribution vanishes at $E = 1$, but does not decrease with batch size when $E > 1$. Our synthetic experiments observe increasing client disagreement as ρ approaches one, supporting this qualitative connection without establishing the worst-case exponents.

Furthermore, the radius R determines how the broadcast model's weight norms are enlarged to obtain bounds valid throughout local training: each block satisfies $\|W_X\|_2 \le \|W_X^r\|_2 + R$. As such, allowing greater client displacement increases the parameter envelopes entering the convergence bound, even when ρ remains unchanged. Additional local steps can therefore affect the guarantee through both the explicit dependence on E and the larger envelopes needed to cover the resulting trajectories. These trajectories must also remain within a neighborhood satisfying $\mathcal{N}_R(W^r) \subseteq \mathcal{S}_{\gamma,\delta}$. The FedProx extension in Appendix B demonstrates how restricting client movement changes the guarantee: its effective radius $R_\mu \le R$ permits smaller envelopes $M_X^\mu = M_X - R + R_\mu$, reducing the SSM-dependent factors multiplying noise and heterogeneity. However, the proximal term also increases local smoothness by μ. The resulting bound therefore exposes a trade-off between tighter control of the SSM parameters and the additional smoothness cost of proximal regularization.

5 Experiments

In this section, we empirically evaluate the derived bounds and their implications for distributed SSM training in two parts: (i) direct evaluation of gradient and smoothness bounds using sequences generated by frozen SSM teachers, and (ii) Mamba2 language modeling experiments with nine federated learning algorithms. In the latter, we compare theory-motivated expectations with observed performance illustrating the practical utility of the derived bounds. Details on our experimental setup are included in Appendix G, with supplementary experimental results available in Appendix H.

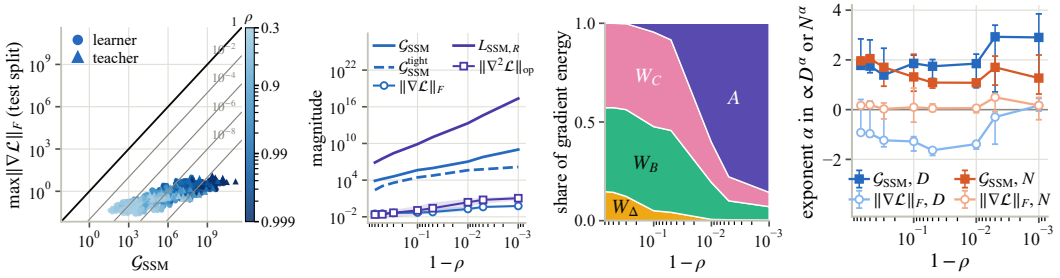
Validation of Derived Bounds. We consider autoregressive token sequences generated by a frozen one-layer selective SSM parameterized by $W = (W_\Delta, W_B, W_C, A)$ analyzed in Theorem 4.4. This setup follows the formulation of Section 3 exactly, with the teacher SSM generating one-hot token ($L_x = 1$) sequences of length $S = 64$ for different combinations of D, N and stability limits ρ, yielding 72 distinct datasets. For each such dataset, we train a learner SSM on 4096 sequences using projected SGD with softmax cross-entropy loss ($L_\ell = \sqrt{2}$), with the model updates inside $\mathcal{S}_{\gamma,\delta}$. We check test-sequence gradients at five learner and one teacher checkpoints against the theorem's $\mathcal{G}_{\mathrm{SSM}}$. We also use $\mathcal{G}_{\mathrm{SSM}}^{\mathrm{tight}}$, tightened via realized stability, one-hot input bounds, and finite-history sums. We also evaluate the smoothness bound and examine the drift and disagreement quantities entering the FedAvg and FedProx analysis. Appendix H.1 provides additional validation results.

Results and Discussion. No violations were detected in 1,411,776 gradient, smoothness and bound-variation checks (Figure 2a, Table 1), or in 103,200 federated checks across 1,200 FedAvg/FedProx runs on 72 teacher datasets. The federated runs explore step sizes beyond the theorems' range. The results also reveal informative patterns.

Memory amplifies gradient and curvature sensitivity. Across all generated datasets, the measured gradient and estimated Hessian norms grow approximately as $(1-\rho)^{-0.5}$ and $(1-\rho)^{-1.0}$, respectively, while their bounds grow as $(1-\rho)^{-1.8}$ and $(1-\rho)^{-3.6}$, as seen in Figure 2b. Although the bounds grow more steeply, the curvature exponent is twice the gradient exponent for both the measured

Table 1: **Gradient and smoothness bounds on the synthetic bank: 1,411,776 evaluations, no violations detected.** Maximum utilization is the largest measured-to-derived ratio, with a/b denoting configured/tightened constants. Evaluations count test sequences at each checkpoint (first row), sequence-checkpoint pairs (curvature rows), or consecutive-checkpoint pairs (trajectory rows).

Statement	Inequality checked	Evaluations	Violations	Max. utilization
Thm. 4.4	$\|\nabla\mathcal{L}\|_F \le \mathcal{G}_{\mathrm{SSM}}$, per sequence	1,327,104	0 / 0	1.0×10^{-3} / 8.3×10^{-3}
Lemma 4.7	$\|\widehat{\nabla^2\mathcal{L}}\|_2 \le L_{\mathrm{SSM},0}$	41,472	0	7.0×10^{-5}
Lemma 4.7	finite-difference ratio $\le L_{\mathrm{SSM},0.001}$	41,472	0	8.2×10^{-5}
Lemma 4.7	$\|\nabla\mathcal{L}(W_a) - \nabla\mathcal{L}(W_b)\|_F \le L_{\mathrm{SSM},R}\|W_a - W_b\|_F$	864	0	1.0×10^{-5}
Lemma B.1	$\|\mathcal{G}_{\mathrm{SSM}}(W_a) - \mathcal{G}_{\mathrm{SSM}}(W_b)\| \le L_{\mathcal{G}_{\mathrm{SSM}},R}\|W_a - W_b\|_F$	864	0	0.42



(a) Gradient versus bound (b) Growth with memory (c) Grad. energy by W_X (d) Exponents in D and N

Figure 2: **Single-layer bounds on the SSM teacher-generated datasets.** (a) Across 1,296 learner and teacher parameter points, every measured per-sequence gradient norm lies below the configured $\mathcal{G}_{\mathrm{SSM}}$. (b) At training end, $\mathcal{G}_{\mathrm{SSM}}$ and $L_{\mathrm{SSM},R}$ scale as $(1-\rho)^{-1.8}$ and $(1-\rho)^{-3.6}$, versus $(1-\rho)^{-0.5}$ and $(1-\rho)^{-1.0}$ for measured gradient and estimated Hessian norms. (c) As $\rho \to 1$, gradient energy shifts from W_B, W_C toward the transition block A. (d) One-factor fits estimate the power-law exponent α for the derived (filled) and measured (open) gradient norms as vocabulary size D (blue) or state size N (orange) varies, with bootstrap 95% intervals.

quantities and the certified constants. These trends are consistent with the recurrent-sensitivity structure of Theorem 4.4 and Lemma 4.7. The shared sensitivity factor $L_x(H_R + M_C J_R)$ enters the gradient bound linearly and the first term of the smoothness bound quadratically in (129). The approximately doubled fitted curvature exponent mirrors this linear-versus-quadratic dependence, although the observed growth is substantially milder than that of the bounds.

Recurrent sensitivity and block-wise gradients. The recurrent-Jacobian term, which propagates parameter perturbations through the transition $\bar{A}_t$, carries at least 99% of $\mathcal{G}_{\mathrm{SSM}}$ at every ρ. The measured gradient energy (the share of $\|\nabla_W\mathcal{L}\|_F^2$ carried by each block) shifts from W_B, W_C (86% at $\rho = 0.3$) to A (85% at $\rho = 0.999$, Figure 2c). Dimension fits (Figure 2d) both vary with ρ and give bound exponents of 1.4-2.9 in D and 1.1-2.0 in N, reflecting the $N\,D^2$-type factors in K_W.

Appendix H.1 examines the quantities entering the convergence analysis. The observed $E^{0.87}$ growth of the $\|W_k^{r,E} - W^r\|_F$ drift is consistent with the linear accumulation bound $\eta E \mathcal{G}_{\mathrm{SSM},R}$ used in the FedAvg analysis. Increasing client disagreement as $\rho \to 1$ likewise agrees qualitatively with the SSM-sensitivity factor $\mathcal{G}^2_{\mathrm{SSM},R}$ in Lemma B.4, which bounds disagreement by $4\mathcal{G}^2_{\mathrm{SSM},R}\mathcal{H}_{\mathrm{TV}}$. Across the bank, variation in log-utilization $\log_{10}(\|\nabla\mathcal{L}\|_F/\mathcal{G}_{\mathrm{SSM}})$ is dominated by the memory factor $\rho/(1-\rho)^2$. Its signed covariance shares are 93% for log-bound variance and 102% for log-utilization variance, the latter partly offset by measured gradient growth. Dimension and norm factors contribute less (Figure 9). This identifies the main source of variation in tightness, not absolute looseness.

Limitations and outlook. The experiments did demonstrate some limitations of our derived bounds, mostly how conservative their values are in absolute terms, and how parameter-envelope growth with radius R over T rounds and saturation of the sequence-level total-variation metric affect the looseness. Nevertheless, the observations agree qualitatively with the analysis's stability, sensitivity, and drift dependencies: no violations were detected, gradients and curvature grow with memory, blockwise gradients shift toward A, and drift grows with local steps. The recurrent factor $\rho/(1-\rho)^2$ dominates variation in tightness across the bank, with smaller contributions from projection and input factors. As such, they point to concrete future improvements: data-dependent constants that use the realized

Method	Theory-linked mechanism	Expected		ΔPPL (%)	
		$\beta = 0$	$\beta = 1$	$\beta = 0$	$\beta = 1$
FedProx (Li et al., 2020)	radius, $R_\mu \leq R$	$-$	$-$	-17.0	-34.4
SCAFFOLD (Karimireddy et al., 2020)	corrected disagreement	$+$	$-$	$+4.0$	-19.4
FedDyn (Acar et al., 2021)	heterogeneity; radius (proximal α)	$-$	$-$	-2.2	-35.4
FedAlign (Keçeci et al., 2025)	state-coordinate alignment	$=$	$=$	-0.3	-1.2
FedSAM (Qu et al., 2022)	expanded evaluation neighborhood	$=$	$=$	-0.4	-1.7
FedAvg (McMahan et al., 2017)	reference			0.0	0.0
FedExP (Jhunjhunwala et al., 2023)	adaptive extrapolation, $\eta_g \geq 1$	$=$	$+$	-0.4	$+14.6$
DiLoCo-style (Douillard et al., 2024)	server step, outer momentum	$+$	$+$	$+2.0$	$+1.5$
FedAdam (Reddi et al., 2021)	server step α_s/τ_s	$+$	$+$	$+17.2$	$+0.6$

Table 2: **Theorem-motivated expectations versus observed behavior on Mamba2.** $-$/=/$+$ denote expected lower/similar/higher validation perplexity relative to FedAvg, motivated by the listed mechanisms. ΔPPL is the mean percentage change across the 24 configurations at each β.

stability margin and the finite sequence length, envelopes that track the actual local trajectories rather than the worst case over the whole radius R, and a token-level dissimilarity that would make the heterogeneity term sensitive to the data partition. The Mamba2 experiments that follow examine whether the same drift and heterogeneity mechanisms help interpret practical federated training.

Practical Federated Optimization. We train randomly initialized Mamba2 (Dao and Gu, 2024) models of depth 4 and 16 across $K = 4$ clients for $T = 50$ rounds with $E \in \{64, 128, 256\}$ local steps and heterogeneity $\beta \in \{0, 0.7, 1\}$ on six RedPajama (Together Computer, 2023) domains under four domain assignments ($\beta = 0$ gives every client the same domain mix, $\beta = 1$ one domain each). Nine algorithms are compared over 72 configurations and three seeds using mean last-ten-round validation perplexity, which also selects their hyperparameters (see Appendix G for details).

Theory-Guided Expectations. An update rule affects radius R (in $\mathcal{G}_{\mathrm{SSM},R}$, $L_{\mathrm{SSM},R}$), a heterogeneity term $\propto \eta^2 L^2_{\mathrm{SSM},R} \mathcal{G}^2_{\mathrm{SSM},R} E(E-1)\mathcal{H}_{\mathrm{TV}}$, and the server step to W^{r+1}. For SGD, FedProx bounds drift by $R_\mu \leq \eta E \mathcal{G}_{\mathrm{SSM},R}$ (Theorem B.6 in Appendix B), at the cost of μ in L_{prox,R_μ}, motivating benefits from tuned proximal regularization. SCAFFOLD and FedDyn can reduce corrected client disagreement when corrections track client-gradient differences, though benefits may be smaller when disagreement is small. FedExP, FedAdam and DiLoCo rescale server updates and may enlarge neighborhoods, while the norm-preserving FedAlign and small-perturbation FedSAM may behave like FedAvg, without a shared convergence guarantee (Appendix E). Table 2 summarizes these expectations for the AdamW-based Mamba2 experiments motivated by the insights provided by our analysis.

Results and Discussion. Across all 72 configurations and all three random seeds, the Pearson correlation between the expected performance direction (encoded as -1, 0, and $+1$ for lower, similar, and higher perplexity, respectively) and the observed log-perplexity change relative to FedAvg is 0.71, increasing to 0.81 when restricted to $\beta = 1.0$. For the two heterogeneity regimes $\beta \in \{0, 1\}$, the percentage changes in validation perplexity relative to FedAvg reported in Table 2, averaged over the 24 configurations at each β and all three seeds, agree with 13 of the 16 expected lower/similar/higher outcomes. FedProx's improvement over FedAvg is consistent with the potential benefit of tighter SSM envelopes identified in Theorem B.6 in Appendix B. SCAFFOLD's improvement in $23/24$ configurations at $\beta = 1$, compared with deterioration in $22/24$ at $\beta = 0$, aligns with the disagreement-correction interpretation of the FedAvg bound developed in Appendix E. Together, these results support using the theorem's displacement and disagreement terms to interpret the observed dependence of algorithm performance on client heterogeneity.

6 Conclusion

In this work, we studied distributed learning with selective SSMs through architecture-aware theory, deriving bounds that connect federated optimization to recurrent stability, input-dependent discretization, projection norms, client drift, and heterogeneity. We then validated the derived bounds and provided experiments that demonstrated their relevance and practical applications.

AI USE STATEMENT

In this work, we used generative AI tools for: assisting in the writing of proofs, formulating mathematical claims (specifically, refining author-derived mathematical statements in response to feedback on proof correctness), proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, supporting qualitative and thematic data analysis and interpreting results.

We have not used generative AI tools for: helping develop theoretical models or conceptual frameworks, generating synthetic data sets, providing critical ingredients for proving mathematical claims, assisting with translation, or cleaning and reformatting a dataset. Additionally, we used generative AI tools for: creating or modifying scientific figures or images (limited to suggestions for figure titles and labels), editing a research paper to improve readability, and suggesting a structure for a research paper.

Our use of generative AI was three-fold. First, generative AI tools were used to provide feedback on theoretical arguments, hypotheses, methodological choices, and interpretations of experimental results. For all proofs, theorems, corollaries, and other mathematical statements in this work, the underlying claims and arguments were first derived and formalized. The resulting statements and proofs were subsequently provided to an AI agent for feedback on their correctness, logical consistency, and the validity of the stated claims. The agent's feedback was independently evaluated, and any resulting revisions were made manually. Similarly, generative AI tools were used to check the consistency of our experimental analyses and interpretations with the reported results. All final analyses, interpretations, and claims were determined by the authors.

Second, AI agents assisted with the implementation of the code used in our experiments. While some experiments use synthetic data generated by deterministic code, AI agents were only utilized to assist with writing the code, not generate the synthetic data. All AI-assisted code was manually inspected for correctness and subsequently evaluated using validation tests before being used in the reported experiments.

Third, we used large language models to improve the presentation of the manuscript, including wording, readability, organization, section flow, and textual elements such as figure titles. The scientific figures and images themselves were not generated or modified using generative AI (though their titles and labels were adjusted with suggestions from AI models). All suggested edits were reviewed by the authors and incorporated manually where appropriate.

We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide complete proofs for the theoretical results presented in this work. For the empirical results, Appendix G provides detailed descriptions of the experimental setups, including the datasets and algorithms used, hyperparameter tuning procedures, and hardware used to conduct the experiments. We additionally provide our codebase containing implementations of the methods used to obtain the reported results, together with instructions for running the experiments as supplementary material.

REFERENCES

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026a.

Zihan Qiu, Zekun Wang, Xiao Li, Yanpeng Li, Yang Xu, Yixuan Wang, Huaqing Zhang, Rui Men, Bochao Mao, Chengruidong Zhang, et al. On the design of qwen3. 8-next architecture: Evaluation, efficiency, and training stability. *arXiv preprint arXiv:2608.30320*, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026b.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14048–14077, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.936. URL https://aclanthology.org/2023.findings-emnlp.936/.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Aakash Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, J Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HwCvaJOiCj.

Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. Falcon mamba: The first competitive attention-free 7b language model. *arXiv preprint arXiv:2410.05355*, 2024.

Aaron Blakeman, Aarti Basant, Abhinav Khattar, Adithya Renduchintala, Akhiad Bercovich, Aleksander Ficek, Alexis Bjorlin, Ali Taghibakhshi, Amala Sanjay Deshmukh, Ameya Sunil Mahabaleshwarkar, et al. Nemotron-h: A family of accurate and efficient hybrid mamba-transformer models. *arXiv preprint arXiv:2504.03624*, 2025.

Jamba Team, Barak Lenz, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, et al. Jamba-1.5: Hybrid transformer-mamba models at scale. *arXiv preprint arXiv:2408.12570*, 2024.

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*, pages 1273–1282. PMLR, 2017.

Ertuğrul Keçeci, Müjde Güzelkaya, and Tufan Kumbasar. Fedalign: Federated learning with state alignment for system identification. In *2025 International Conference on Control, Automation and Diagnosis (ICCAD)*, pages 1–6, 2025. doi: 10.1109/ICCAD64771.2025.11099331.

Weidong Zhang, Dongshang Deng, Xuangou Wu, Tao Zhang, Xiao Zheng, Dusit Niyato, and Dong In Kim. Mitigating catastrophic forgetting in personalized federated learning for edge devices using state-space models. *IEEE Transactions on Mobile Computing*, 25(3):3568–3582, 2026. doi: 10.1109/TMC.2025.3618886.

Wenzhi Fang, Dong-Jun Han, Liangqi Yuan, Seyyedali Hosseinalipour, and Christopher G Brinton. Federated sketching lora: A flexible framework for heterogeneous collaborative fine-tuning of llms. *International Conference on Machine Learning (ICML)*, 2026.

Bedionita Soro, Aoxuan Silvia Zhang, Bruno Andreis, Jaehyeong Jo, Song Chong, and Sung Ju Hwang. Ls-merge: Merging language models in latent space. In *The Fourteenth International Conference on Learning Representations*, 2026.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. ISSN 0021-9223.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 35971–35983. Curran Associates, Inc., 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/e9a32fade47b906de908431991440f7c-Paper-Conference.pdf.

Ramin Hasani, Mathias Lechner, Tsun-Hsuan Wang, Makram Chahine, Alexander Amini, and Daniela Rus. Liquid structural state-space models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=g4OTKRKfS7R.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1474–1487. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/102f0bb6efb3a6128a3c750dd16729be-Paper.pdf.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=ztn8FCR1td.

Sukjun Hwang, Aakash Lahoti, Ratish Puduppully, Tri Dao, and Albert Gu. Hydra: Bidirectional state space models through generalized matrix mixers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 110876–110908. Curran Associates, Inc., 2024. doi: 10.52202/079017-3520. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c7f795dc3b4eb6ae630695d90001a2f8-Paper-Conference.pdf.

Hyunwoo Kim, Byoungchan Ko, Minseok Kang, Minwoo Kim, Dongjin Lee, Jaehoon Lee, Sungroh Yoon, and Dahuin Jung. SSDi8: Accurate and efficient 8-bit quantization for state space duality. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=pjMDZJd4rT.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An empirical study of mamba-based language models. *CoRR*, abs/2406.07887, 2024. URL https://doi.org/10.48550/arXiv.2406.07887.

Wei He, Kai Han, Yehui Tang, Chengcheng Wang, Yujie Yang, Tianyu Guo, and Yunhe Wang. DenseSSM: State space models with dense hidden connection for efficient large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9243–9254, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.467. URL https://aclanthology.org/2025.naacl-long.467/.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. Zamba: A compact 7b ssm hybrid model. *arXiv preprint arXiv:2405.16712*, 2024.

Liliang Ren, Yang Liu, Yadong Lu, yelong shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bIlnpVM4bc.

Shufan Li, Harkanwar Singh, and Aditya Grover. Mamba-nd: Selective state space modeling for multi-dimensional data. In *European Conference on Computer Vision*, pages 75–92. Springer, 2024.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=YbHCqn4qF4.

Fei Xie, Zhongdao Wang, Weijia Zhang, and Chao Ma. Pvmamba: Parallelizing vision mamba via dynamic state aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10218–10228, 2025.

Mingqian Ma, Guoqing Liu, Chuan Cao, Pan Deng, Tri Dao, Albert Gu, Peiran Jin, Zhao Yang, Yingce Xia, Renqian Luo, Pipi Hu, Zun Wang, Yuan-Jyue Chen, Haiguang Liu, and Tao Qin. HybriDNA: A hybrid transformer-mamba2 long-range DNA language model. In *ICLR 2025 Workshop on Machine Learning for Genomics Explorations*, 2025. URL https://openreview.net/forum?id=BJ2uCHIfEW.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020. URL https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/karimireddy20a.html.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=B7v4QMR6Z9w.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

Tian Yu Liu, Alessandro Achille, Matthew Trager, Aditya Golatkar, Luca Zancato, and Stefano Soatto. PICASO: Permutation-invariant context composition with state space models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=88TC1AWV27.

Maciej Pióro, Maciej Wołczyk, Razvan Pascanu, Johannes von Oswald, and João Sacramento. State soup: In-context skill learning, retrieval and mixing. *arXiv preprint arXiv:2406.08423*, 2024.

Together Computer. Redpajama: An open source recipe to reproduce llama training dataset, 2023. URL https://github.com/togethercomputer/RedPajama-Data.

Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18250–18280. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/qu22a.html.

Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=IPrzNbddXV.

Arthur Douillard, Qixuan Feng, Andrei Alex Rusu, Rachita Chhaparia, Yani Donchev, Adhiguna Kuncoro, MarcAurelio Ranzato, Arthur Szlam, and Jiajun Shen. Diloco: Distributed low-communication training of language models. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ICML 2024)*, 2024. URL https://openreview.net/forum?id=pICSfWkJIk.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 7093–7115. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1644c9af28ab7916874f6fd6228a9bcf-Paper-Conference.pdf.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

APPENDIX

A Selective SSM Gradient Bound Proof **17**

B Convergence of Federated SSM Baselines **21**

B.1 Auxiliary Lemmas for SSM Federated Convergence 21

B.2 Theorem B.5: FedAvg SSM Convergence and Proof 31

B.3 Theorem B.6: FedProx SSM Convergence and Proof 33

C Multi-Layer Selective SSM proofs **41**

C.1 Multi-Layer Selective SSM Gradient Bound Proof 41

C.2 Auxiliary Lemmas for Multi-Layer SSM Federated Convergence 52

C.3 Corollary C.8: FedAvg Multi-Layer SSM Convergence and Proof 62

C.4 Corollary C.9: FedProx Multi-Layer SSM Convergence and Proof 65

D Supplementary Corollaries **69**

D.1 Single-layer convergence rate corollaries 69

D.2 Multi-Layer SSM Gradient Scaling Bounds 70

D.3 Multi-layer convergence under common layer bounds 71

D.4 SSM Convergence under Distributed Learning 74

E Discussion on Additional Distributed Learning Frameworks **75**

F Discussion on Model Merging **79**

F.1 Model Merging with SSMs Motivation 79

F.2 Model Merging Methods: Connections to Selective SSM Analysis 80

G Detailed Experimental Setup **81**

G.1 Datasets 82

G.2 Models 86

G.3 Synthetic Distributed Learning Framework 88

G.4 Practical Distributed Learning Framework 90

G.5 Model Merging Framework 92

G.6 Reproducibility 93

H Additional Experimental Results and Ablation Studies **93**

H.1 Single Layer Synthetic Distributed Learning 93

H.2 Practical Distributed Learning 96

H.3 Model Merging 101

A SELECTIVE SSM GRADIENT BOUND PROOF

In this section, we derive an upper bound on the gradient norm of the loss function for some admissible input sequence processed by a single-layer selective state space model (SSM).

Fix a parameter tuple $W \in \mathcal{S}_{\gamma,\delta}$ satisfying the assumptions of Theorem 4.4. Let $S \geq 1$ be an arbitrary integer and $\xi = (x, y^*) \in \mathcal{Z}$ an arbitrary admissible sequence pair of length S. We use the SSM dynamics from Section 3 and the notational conventions from section 4.1.

Given these definitions, we examine the sensitivity of the loss with respect to the learnable model parameter tuple

$$W := (W_\Delta, W_B, W_C, A). \tag{12}$$

Throughout the proof, gradients with respect to W are understood as block gradients, i.e.,

$$\|\nabla_W f\|_F := \left(\|\nabla_{W_\Delta} f\|_F^2 + \|\nabla_{W_B} f\|_F^2 + \|\nabla_{W_C} f\|_F^2 + \|\nabla_A f\|_F^2\right)^{1/2}. \tag{13}$$

Next, let $\theta = \mathrm{vec}(W)$ concatenate the vectorized parameter blocks. Scalar gradients are column vectors after vectorization:

$$\nabla_\theta f = \mathrm{vec}\left(\nabla_W f\right)$$

For a vector- or matrix-valued function F, $\partial F/\partial W$ denotes its flattened block Jacobian, with output entries indexing rows and scalar parameters indexing columns:

$$\left\| \frac{\partial F}{\partial W} \right\|_F^2 := \sum_\alpha \left\| \frac{\partial F}{\partial \theta_\alpha} \right\|_F^2$$

In product-rule expressions, products and transposes involving matrix-valued derivatives are applied separately to each scalar-parameter slice before flattening. Since h_t does not depend on W_C, its W_C-Jacobian block is zero.

For clarity, in this section we omit the client index k and focus on bounding the gradient norm for a single client processing a sequence of length S.

For the fixed sample ξ, using the simplified notation introduced in Section 3, the sequence loss is:

$$\mathcal{L}^S(W) = \frac{1}{S} \sum_{t=1}^{S} \ell_t(y_t)$$

where $\ell_t(y_t) := \ell_t(y_t(W; x), y_t^*)$ represents the instantaneous loss between the predicted and target tokens at the t-th place in the sequence S, and we omit the dependence on a fixed sample ξ.

The goal of this derivation is to bound the norm of the gradient $\nabla_W \mathcal{L}^S(W)$. By the linearity of the gradient operator, it then follows that:

$$\nabla_W \mathcal{L}^S(W) = \frac{1}{S} \sum_{t=1}^{S} \nabla_W \ell_t(y_t) \tag{14}$$

We then apply the chain rule to the token loss $\ell_t(y_t)$, where $\nabla_y \ell_t(y_t)$ is a column vector. Noting that y_t depends on W both directly through the output projection vector C_t and indirectly through the hidden state h_t, we obtain:

$$\begin{aligned}
\nabla_\theta \ell_t(y_t) &= \left(\frac{\partial y_t}{\partial W}\right)^\top \nabla_y \ell_t(y_t) \\
&= \left(\frac{\partial (h_t^\top C_t)}{\partial W}\right)^\top \nabla_y \ell_t(y_t) \\
&= \left(\frac{\partial h_t^\top}{\partial W} C_t + h_t^\top \frac{\partial C_t}{\partial W}\right)^\top \nabla_y \ell_t(y_t).
\end{aligned} \tag{15}$$

Here, the product-rule terms inside the parentheses are formed separately for each scalar-parameter slice and constructed as the columns of the output Jacobian before taking the outer transpose. The

first term contributes only to the W_Δ, W_B, and A blocks, while the second term contributes only to the W_C block.

Under the described block-gradient convention, the schematic chain-rule decomposition gives:

$$\|\nabla_W \ell_t(y_t)\|_F = \|\operatorname{vec}(\nabla_W \ell_t(y_t))\|_2 = \|\nabla_\theta \ell_t(y_t)\|_2 = \left\| \left(\frac{\partial h_t^\top}{\partial W} C_t + h_t^\top \frac{\partial C_t}{\partial W} \right)^\top \nabla_y \ell_t(y_t) \right\|_2 . \tag{16}$$

Next, applying the matrix-vector norm inequality to each scalar-parameter slice and summing the squared bounds gives:

$$\left\| \frac{\partial h_t^\top}{\partial W} C_t \right\|_F \le \left\| \frac{\partial h_t}{\partial W} \right\|_F \|C_t\|_2. \tag{17}$$

Using this bound, the triangle inequality, and the fact that $\|M^\top v\|_2 \le \|M\|_F \|v\|_2$, we get:

$$\begin{aligned} \|\nabla_W \ell_t(y_t)\|_F &\le \left\| \left(\frac{\partial h_t^\top}{\partial W} C_t \right)^\top \nabla_y \ell_t(y_t) \right\|_2 + \left\| \left(\frac{\partial C_t}{\partial W} \right)^\top h_t \nabla_y \ell_t(y_t) \right\|_2 \\ &\le \|\nabla_y \ell_t(y_t)\|_2 \left\| \frac{\partial h_t}{\partial W} \right\|_F \|C_t\|_2 + \left\| \left(\frac{\partial C_t}{\partial W} \right)^\top h_t \nabla_y \ell_t(y_t) \right\|_2 . \end{aligned} \tag{18}$$

Next, utilizing Assumptions 4.2 and 4.3, we bound the two terms of this expression separately.

First, since $C_t = W_C x_t$, we have:

$$\|C_t\|_2 \le \|W_C\|_2 \|x_t\|_2 \le \|W_C\|_2 L_x \tag{19}$$

Therefore,

$$\|\nabla_y \ell_t(y_t)\|_2 \left\| \frac{\partial h_t}{\partial W} \right\|_F \|C_t\|_2 \le L_\ell L_x \|W_C\|_2 \left\| \frac{\partial h_t}{\partial W} \right\|_F \tag{20}$$

For the second term, since h_t does not depend on W_C and x_t is fixed, equation (2b) gives:

$$\nabla_{W_C} \ell_t(y_t) = h_t \nabla_y \ell_t(y_t) x_t^\top . \tag{21}$$

Furthermore, since $C_t = W_C x_t$, the Jacobian $\partial C_t / \partial W$ is zero in the W_Δ, W_B, and A parameter blocks, and nonzero only for the W_C block. As a result, $(\partial C_t / \partial W)^\top h_t \nabla_y \ell_t(y_t)$ contains the vectorized W_C-block gradient and zeros in all other parameter blocks, yielding:

$$\left\| \left(\frac{\partial C_t}{\partial W} \right)^\top h_t \nabla_y \ell_t(y_t) \right\|_2 = \|\nabla_{W_C} \ell_t(y_t)\|_F = \|h_t \nabla_y \ell_t(y_t) x_t^\top\|_F . \tag{22}$$

This can be bounded as:

$$\begin{aligned} \left\| h_t \nabla_y \ell_t(y_t) x_t^\top \right\|_F &= \|h_t \nabla_y \ell_t(y_t)\|_2 \, \|x_t\|_2 \le \|h_t \nabla_y \ell_t(y_t)\|_2 \, L_x && (23) \\ &\le \|h_t\|_F \, \|\nabla_y \ell_t(y_t)\|_2 \, L_x \le L_x L_\ell \, \|h_t\|_F && (24) \end{aligned}$$

By combining this result with (20), we can rewrite inequality (18) as:

$$\|\nabla_W \ell_t(y_t)\|_F \le L_\ell L_x \left(\|h_t\|_F + \|W_C\|_2 \left\| \frac{\partial h_t}{\partial W} \right\|_F \right) \tag{25}$$

Next, for the remaining two terms, we first evaluate the bound on the hidden state matrix $h_t \in \mathbb{R}^{N \times D}$. Given an initialization of $h_0 = \mathbf{0}$ (which is standard when processing independent training sequences), unrolling the recurrence from equation (2a) yields:

$$h_t = \sum_{j=1}^{t} \left(\bigodot_{k=j+1}^{t} \bar{A}_k \right) \odot \left(\bar{B}_j \operatorname{diag}(x_j) \right) \tag{26}$$

where we define the empty cumulative Hadamard product as the all-ones matrix, $\bigodot_{k=t+1}^{t} \bar{A}_k = \mathbf{1}_{N\times D}$. Then, taking the Frobenius norm of both sides and applying the triangle inequality, we get:

$$\|h_t\|_F \leq \sum_{j=1}^{t} \left\| \left(\bigodot_{k=j+1}^{t} \bar{A}_k \right) \odot \left(\bar{B}_j \operatorname{diag}(x_j) \right) \right\|_F = \sum_{j=1}^{t} \left\| \exp\left(A \sum_{k=j+1}^{t} \Delta_k^\top \right) \odot \left(\bar{B}_j \operatorname{diag}(x_j) \right) \right\|_F \tag{27}$$

Utilizing the property of Hadamard product with Frobenius norms, the fact that $\left\|\bar{B}_j \operatorname{diag}(x_j)\right\|_F \leq \left\|\bar{B}_j\right\|_2 \|\operatorname{diag}(x_j)\|_F \leq \left\|\bar{B}_j\right\|_2 L_x$, and the definition of $\bar{B}_t$, it follows that:

$$\left\| \exp\left(A \sum_{k=j+1}^{t} \Delta_k^\top \right) \odot \left(\bar{B}_j \operatorname{diag}(x_j) \right) \right\|_F \leq \left\| \exp\left(A \sum_{k=j+1}^{t} \Delta_k^\top \right) \right\|_F \left\| W_B x_j \Delta_j^\top \right\|_2 L_x \tag{28}$$

Then:

$$\left\| W_B x_j \Delta_j^\top \right\|_2 L_x \leq \|W_B\|_2 \|x_j\|_2 \left\| \Delta_j^\top \right\|_2 L_x \leq \|W_B\|_2 \|\text{Softplus}\left(W_\Delta x_j\right)\|_2 L_x^2 \tag{29}$$

And since $\text{Softplus}(a) \leq |a| + \log(2)$ so $\|\text{Softplus}(\mathbf{x})\|_2 \leq \||\mathbf{x}| + \log(2)\mathbf{1}\|_2 \leq \|\mathbf{x}\|_2 + \sqrt{D}\log(2)$, it follows that $\|\text{Softplus}\left(W_\Delta x_j\right)\|_2 \leq \||W_\Delta x_j| + \log(2)\mathbf{1}\|_2$, and from the triangle inequality, this results in $\||W_\Delta x_j| + \log(2)\mathbf{1}\|_2 \leq \|W_\Delta\|_2 L_x + \sqrt{D}\log(2)$

So the bound becomes:

$$\left\| W_B x_j \Delta_j^\top \right\|_2 L_x \leq \|W_B\|_2 L_x^2 \left(\|W_\Delta\|_2 L_x + \sqrt{D}\log(2) \right) \tag{30}$$

Since these terms are independent of j, it follows that:

$$\|h_t\|_F \leq \|W_B\|_2 L_x^2 \left(\|W_\Delta\|_2 L_x + \sqrt{D}\log(2) \right) \sum_{j=1}^{t} \left\| \exp\left(A \sum_{k=j+1}^{t} \text{Softplus}\left(W_\Delta x_k\right)^\top \right) \right\|_F \tag{31}$$

Next, the term $\frac{\partial h_t}{\partial W}$ can be expanded from equation (2a) as:

$$\frac{\partial h_t}{\partial W} = \frac{\partial \bar{A}_t}{\partial W} \odot h_{t-1} + \bar{A}_t \odot \frac{\partial h_{t-1}}{\partial W} + \frac{\partial \bar{B}_t}{\partial W} \operatorname{diag}(x_t) \tag{32}$$

We bound each term individually, using the fact that:

$$\left\| \frac{\partial h_t}{\partial W} \right\|_F \leq \left\| \frac{\partial h_t}{\partial W_\Delta} \right\|_F + \left\| \frac{\partial h_t}{\partial W_B} \right\|_F + \left\| \frac{\partial h_t}{\partial W_C} \right\|_F + \left\| \frac{\partial h_t}{\partial A} \right\|_F \tag{33}$$

where the W_C-dependent term is zero.

By the triangle inequality:

$$\left\| \frac{\partial h_t}{\partial W} \right\|_F \leq \left\| \frac{\partial \bar{A}_t}{\partial W} \odot h_{t-1} \right\|_F + \left\| \bar{A}_t \odot \frac{\partial h_{t-1}}{\partial W} \right\|_F + \left\| \frac{\partial \bar{B}_t}{\partial W} \operatorname{diag}(x_t) \right\|_F \tag{34}$$

Then, we bound each term. First:

$$\begin{aligned} \left\| \frac{\partial \bar{B}_t}{\partial W} \operatorname{diag}(x_t) \right\|_F &\leq \left\| \frac{\partial \bar{B}_t}{\partial W} \right\|_F \|\operatorname{diag}(x_t)\|_F \leq L_x \left(\left\| \frac{\partial \bar{B}_t}{\partial W_B} \right\|_F + \left\| \frac{\partial \bar{B}_t}{\partial W_\Delta} \right\|_F \right) \\ &\leq L_x \left(\sqrt{N} \|x_t\|_2 \|\Delta_t\|_2 + \|W_B x_t\|_2 \left\| \frac{\partial \Delta_t}{\partial W_\Delta} \right\|_F \right) \\ &\leq L_x \left(\sqrt{N} L_x \left(\|W_\Delta\|_2 L_x + \sqrt{D}\log(2) \right) + \sqrt{D} \|W_B\|_2 L_x^2 \right), \end{aligned} \tag{35}$$

where we used the fact that $\text{Softplus}'(z) = \sigma(z) \leq 1$, so since $\Delta_t = \text{Softplus}(W_\Delta x_t)$, it is true that $\left\| \frac{\partial \Delta_t}{\partial W_\Delta} \right\|_F \leq \sqrt{D}\|x_t\|_2 \leq \sqrt{D} L_x$.

Next:

$$\begin{aligned}\left\|\frac{\partial \bar{A}_t}{\partial W} \odot h_{t-1}\right\|_F &\leq \left\|\frac{\partial \bar{A}_t}{\partial W}\right\|_F \|h_{t-1}\|_F \leq \left(\left\|\frac{\partial \bar{A}_t}{\partial A}\right\|_F + \left\|\frac{\partial \bar{A}_t}{\partial W_\Delta}\right\|_F\right) \|h_{t-1}\|_F \\ &\leq \left(\left\|\bar{A}_t\right\|_F \|\Delta_t\|_2 + \left\|\frac{\partial \bar{A}_t}{\partial \Delta_t}\right\|_F \left\|\frac{\partial \Delta_t}{\partial W_\Delta}\right\|_F\right) \|h_{t-1}\|_F \\ &\leq \left(\left\|\bar{A}_t\right\|_F \|\Delta_t\|_2 + \sqrt{D}\,\|A\|_2 \left\|\bar{A}_t\right\|_F L_x\right) \|h_{t-1}\|_F \\ &\leq \|h_{t-1}\|_F \left\|\bar{A}_t\right\|_F \left(\left(\|W_\Delta\|_2 L_x + \sqrt{D}\log(2)\right) + \sqrt{D}\,\|A\|_2 L_x\right) \end{aligned} \tag{36}$$

Finally, by definition of $\|M\|_{\max} := \max_{i,j} |M_{ij}|$, it follows that:

$$\left\|\bar{A}_t \odot \frac{\partial h_{t-1}}{\partial W}\right\|_F \leq \left\|\bar{A}_t\right\|_{\max} \left\|\frac{\partial h_{t-1}}{\partial W}\right\|_F \tag{37}$$

Combining all the terms together, we get the following recurrence:

$$\begin{aligned}\left\|\frac{\partial h_t}{\partial W}\right\|_F \leq\ & \|W_B\|_2 L_x^2 \left(\|W_\Delta\|_2 L_x + \sqrt{D}\log(2)\right) \sum_{j=1}^{t-1} \left\|\exp\left(A \sum_{k=j+1}^{t-1} \text{Softplus}\left(W_\Delta x_k\right)^\top\right)\right\|_F \\ &\times \left\|\bar{A}_t\right\|_F \left(\left(\|W_\Delta\|_2 L_x + \sqrt{D}\log(2)\right) + \sqrt{D}\,\|A\|_2 L_x\right) \\ &+ L_x \left(\sqrt{N} L_x \left(\|W_\Delta\|_2 L_x + \sqrt{D}\log(2)\right) + \sqrt{D}\,\|W_B\|_2 L_x^2\right) + \left\|\bar{A}_t\right\|_{\max} \left\|\frac{\partial h_{t-1}}{\partial W}\right\|_F \end{aligned} \tag{38}$$

Next, define:

$$K_\Delta := \|W_\Delta\|_2 L_x + \sqrt{D}\log(2) \tag{39}$$

$$K_B := \|W_B\|_2 L_x^2 \tag{40}$$

$$\Phi_{j,t} := \left\|\exp\left(A \sum_{k=j+1}^{t} \text{Softplus}\left(W_\Delta x_k\right)^\top\right)\right\|_F \tag{41}$$

So the recurrence becomes:

$$\begin{aligned}\left\|\frac{\partial h_t}{\partial W}\right\|_F \leq\ & \left\|\bar{A}_t\right\|_F K_B K_\Delta \left(K_\Delta + \sqrt{D}\,\|A\|_2 L_x\right) \sum_{j=1}^{t-1} \Phi_{j,t-1} + L_x^2 \sqrt{N} K_\Delta + L_x \sqrt{D} K_B \\ &+ \left\|\bar{A}_t\right\|_{\max} \left\|\frac{\partial h_{t-1}}{\partial W}\right\|_F \end{aligned} \tag{42}$$

Since $h_0 = \mathbf{0}$ is parameter-independent, $\partial h_0 / \partial W = 0$, iterating the inequality gives:

$$\begin{aligned}\left\|\frac{\partial h_t}{\partial W}\right\|_F \leq \sum_{m=1}^{t} \Bigg[& \left\|\bar{A}_m\right\|_F K_B K_\Delta \left(K_\Delta + \sqrt{D}\,\|A\|_2 L_x\right) \left(\sum_{j=1}^{m-1} \Phi_{j,m-1}\right) \\ & + L_x^2 \sqrt{N} K_\Delta + L_x \sqrt{D} K_B \Bigg] \prod_{n=m+1}^{t} \left\|\bar{A}_n\right\|_{\max}, \end{aligned} \tag{43}$$

where empty sums are calculated as zero and empty scalar products are calculated as one.

Next, by Definition 4.1, each entry of $A\Delta_t^\top$ is at most $-\gamma\delta$. Since $\bar{A}_t = \exp(A\Delta_t^\top)$ element-wise, it follows that:

$$\left\|\bar{A}_t\right\|_{\max} \leq \exp\left(-\gamma\delta\right) := \rho < 1 \tag{44}$$

as well as

$$\left\|\bar{A}_t\right\|_F \leq \sqrt{ND}\rho \tag{45}$$

and in addition, by the definition of $\Phi_{j,t}$, the following inequality holds:

$$\Phi_{j,m-1} \leq \sqrt{ND}\rho^{m-1-j} \tag{46}$$

As a result, the inner summation becomes a simple geometric series:

$$\sum_{j=1}^{m-1} \Phi_{j,m-1} \leq \sqrt{ND} \sum_{j=1}^{m-1} \rho^{m-1-j} = \sqrt{ND} \sum_{r=0}^{m-2} \rho^r \leq \frac{\sqrt{ND}}{1-\rho} \tag{47}$$

As such, we can group all time-independent terms under a new constant:

$$K_W := NDK_BK_\Delta \left(K_\Delta + \sqrt{D}\left\|A\right\|_2 L_x\right)\left(\frac{\rho}{1-\rho}\right) + L_x^2\sqrt{N}K_\Delta + L_x\sqrt{D}K_B \tag{48}$$

reducing the recurrence to:

$$\left\|\frac{\partial h_t}{\partial W}\right\|_F \leq K_W \sum_{m=1}^{t} \rho^{t-m} \leq \frac{K_W}{1-\rho} \tag{49}$$

Finally, combining everything, we get the final bound on the gradient of loss at time t:

$$\left\|\nabla_W \ell_t(y_t)\right\|_F \leq L_\ell L_x \left(K_B K_\Delta \frac{\sqrt{ND}}{1-\rho} + \left\|W_C\right\|_2 \frac{K_W}{1-\rho}\right) \tag{50}$$

Since $\nabla_W \mathcal{L}^S(W) = \frac{1}{S}\sum_{t=1}^{S} \nabla_W \ell_t(y_t)$, by the triangle inequality the same bound holds for the averaged empirical loss:

$$\left\|\nabla_W \mathcal{L}^S(W)\right\|_F \leq L_\ell L_x \left(K_B K_\Delta \frac{\sqrt{ND}}{1-\rho} + \left\|W_C\right\|_2 \frac{K_W}{1-\rho}\right) = \mathcal{G}_{\text{SSM}}(W) \tag{51}$$

Since $S \geq 1$ and the admissible length-S sequence pairs $\xi \in \mathcal{Z}$ were arbitrary, and the right-hand side of the inequality is independent of S and ξ, the bound holds uniformly over all such lengths and admissible sequence pairs for the fixed parameter tuple W, proving Theorem 4.4. □

B Convergence of Federated SSM Baselines

In this section, we first define and prove a set of lemmas required to instantiate standard federated optimization convergence results in terms of selective SSM parameters. We then provide the proof of Theorem B.5, and state the corresponding FedProx theorem.

B.1 Auxiliary Lemmas for SSM Federated Convergence

Rather than proving the FedAvg or FedProx convergence bounds directly, we first show how the generic smoothness, gradient second-moment, stochastic variance, and client heterogeneity quantities appearing in those bounds can be replaced by SSM-specific constants. In particular, we establish local control of the SSM gradient bound in Lemma B.1, prove the local smoothness constant $L_{\text{SSM},R}$ in Lemma 4.7, and derive the architecture-dependent quantities $G^2_{\text{SSM},R}$, $\sigma^2_{\text{SSM},R,\Xi}$, and $\Gamma^2_{\text{SSM},R}$ in Lemmas B.2, B.3, and B.4, respectively. These lemmas are then used in the remainder of this section to state the final convergence bounds for FedAvg and FedProx distributed learning baselines.

B.1.1 Local Lipschitz Continuity of the SSM Gradient Bound Lemma and Proof

Lemma B.1 (Local Lipschitz Continuity of the SSM Gradient Bound)**.** *Let W^r be some fixed reference parameter tuple and let $0 < R < \infty$ be a finite radius such that $\mathcal{N}_R(W^r) \subseteq \mathcal{S}_{\gamma,\delta}$. Suppose Assumptions 4.2 and 4.3 hold with the neighborhood-uniform constants specified in Section 4.2.*

Let $\mathcal{G}_{\mathrm{SSM}}(W)$ be the bounding function defined in Theorem 4.4. Then $\mathcal{G}_{\mathrm{SSM}}$ is Lipschitz continuous on $\mathcal{N}_R(W^r)$.

Specifically, with $N, D, L_\ell, L_x, \gamma, \delta$ fixed, there exists a finite constant $L_{\mathcal{G}_{\mathrm{SSM},R}} > 0$, depending only on R and the block norms $\|W_\Delta^r\|_2$, $\|W_B^r\|_2$, $\|W_C^r\|_2$, and $\|A^r\|_2$, such that for all $W, V \in \mathcal{N}_R(W^r)$,

$$|\mathcal{G}_{\mathrm{SSM}}(W) - \mathcal{G}_{\mathrm{SSM}}(V)| \leq L_{\mathcal{G}_{\mathrm{SSM},R}} \|W - V\|_F. \tag{52}$$

Proof of Lemma B.1: Fix the reference tuple W^r and the radius R. Throughout this proof, $N, D, L_\ell, L_x, \gamma, \delta$ are constant as W and V vary over the neighborhood $\mathcal{N}_R(W^r)$. In particular, $\rho = \exp(-\gamma\delta)$ is the same for both parameter tuples.

First, by Theorem 4.4, $\mathcal{G}_{\mathrm{SSM}}(W)$ has an explicit definition:

$$\mathcal{G}_{\mathrm{SSM}}(W) = L_\ell L_x \left(K_B K_\Delta \frac{\sqrt{ND}}{1-\rho} + \|W_C\|_2 \frac{K_W}{1-\rho} \right), \tag{53}$$

where W is the tuple of selective SSM weights (W_Δ, W_B, W_C, A).

Since $W = (W_\Delta, W_B, W_C, A)$ is a parameter tuple, given another parameter tuple $V = (V_\Delta, V_B, V_C, V_A)$ we use the block Frobenius norm:

$$\|W - V\|_F^2 := \|W_\Delta - V_\Delta\|_F^2 + \|W_B - V_B\|_F^2 + \|W_C - V_C\|_F^2 + \|A - V_A\|_F^2. \tag{54}$$

For each parameter block $X \in \{\Delta, B, C, A\}$, let W_X denote the corresponding block of W, and let W_X^r denote the corresponding block of the central model W^r (for notational uniformity, we write $W_A := A$ and $W_A^r := A^r$). Given the central model W^r, we evaluate the maximum possible value of a norm of each parameter matrix within the neighborhood $\mathcal{N}_R(W^r)$. Since $W \in \mathcal{N}_R(W^r)$, each block satisfies

$$\|W_X - W_X^r\|_F \leq \|W - W^r\|_F \leq R. \tag{55}$$

Therefore, by the triangle inequality and the fact that $\|\cdot\|_2 \leq \|\cdot\|_F$,

$$\|W_X\|_2 \leq \|W_X^r\|_2 + \|W_X - W_X^r\|_2 \leq \|W_X^r\|_2 + \|W_X - W_X^r\|_F \leq \|W_X^r\|_2 + R. \tag{56}$$

Next, define:

$$M_\Delta := \|W_\Delta^r\|_2 + R \tag{57}$$
$$M_B := \|W_B^r\|_2 + R \tag{58}$$
$$M_C := \|W_C^r\|_2 + R \tag{59}$$
$$M_A := \|A^r\|_2 + R \tag{60}$$

Then, by (56), it is true that for all individual matrices W_B, W_C, W_Δ, A, the following holds:

$$\|W_\Delta\|_2 \leq M_\Delta \tag{61}$$
$$\|W_B\|_2 \leq M_B \tag{62}$$
$$\|W_C\|_2 \leq M_C \tag{63}$$
$$\|A\|_2 \leq M_A \tag{64}$$

The same bounds also hold for the corresponding blocks of any $V \in \mathcal{N}_R(W^r)$.

We further define:

$$M_{K_\Delta} := M_\Delta L_x + \sqrt{D}\log(2) \tag{65}$$
$$M_{K_B} := M_B L_x^2 \tag{66}$$

Next, we evaluate the Lipschitz constants for individual components of the expression for $\mathcal{G}_{\mathrm{SSM}}(W)$. We'll utilize the property that: $|\|X\|_2 - \|Y\|_2| \leq \|X - Y\|_2 \leq \|X - Y\|_F$.

First:

$$|K_\Delta(W) - K_\Delta(V)| \leq L_x \left| \|W_\Delta\|_2 - \|V_\Delta\|_2 \right| \leq L_x \|W_\Delta - V_\Delta\|_F \leq L_x \|W - V\|_F \tag{67}$$
$$|K_B(W) - K_B(V)| \leq L_x^2 \left| \|W_B\|_2 - \|V_B\|_2 \right| \leq L_x^2 \|W_B - V_B\|_F \leq L_x^2 \|W - V\|_F \tag{68}$$

So the Lipschitz constants for the base functions are:

$$L_{K_\Delta} = L_x \tag{69}$$

$$L_{K_B} = L_x^2 \tag{70}$$

$$L_C = 1 \text{ (for the term } \|W_C\|_2 \text{)} \tag{71}$$

$$L_A = 1 \text{ (for the term } \|A\|_2 \text{)} \tag{72}$$

Next, by the product rule for Lipschitz functions, if $|f| \leq M_f$ and $|g| \leq M_g$ and are Lipschitz with constants L_f and L_g on $\mathcal{N}_R(W^r)$, then fg is Lipschitz with constant

$$L_{fg} = M_f L_g + M_g L_f. \tag{73}$$

Given that, let

$$P_1(W) := K_B(W) K_\Delta(W). \tag{74}$$

Then

$$L_{P_1} = M_{K_B} L_{K_\Delta} + M_{K_\Delta} L_{K_B} = M_{K_B} L_x + M_{K_\Delta} L_x^2, \tag{75}$$

and

$$M_{P_1} = M_{K_B} M_{K_\Delta}. \tag{76}$$

Next, define

$$P_2(W) := K_\Delta(W) + \sqrt{D}\|A\|_2 L_x. \tag{77}$$

Then, by the sum rule of Lipschitz functions:

$$L_{P_2} = L_{K_\Delta} + \sqrt{D} L_A L_x = L_x + \sqrt{D} L_x, \tag{78}$$

and

$$M_{P_2} = M_{K_\Delta} + \sqrt{D} M_A L_x. \tag{79}$$

Continuing, define

$$P_3(W) := P_1(W) P_2(W). \tag{80}$$

Then

$$L_{P_3} = M_{P_1} L_{P_2} + M_{P_2} L_{P_1}, \tag{81}$$

and

$$M_{P_3} = M_{P_1} M_{P_2}. \tag{82}$$

Given these definitions, and using the definition of K_W from Theorem 4.4, a valid Lipschitz constant for K_W is:

$$L_{K_W} = ND \frac{\rho}{1-\rho} L_{P_3} + L_x^3 \sqrt{N} + L_x^3 \sqrt{D}, \tag{83}$$

and the upper bound for $K_W(W)$ is:

$$M_{K_W} = ND \frac{\rho}{1-\rho} M_{P_3} + L_x^2 \sqrt{N} M_{K_\Delta} + L_x \sqrt{D} M_{K_B}. \tag{84}$$

Finally, define

$$P_4(W) := \|W_C\|_2 K_W(W). \tag{85}$$

Since $\|W_C\|_2$ is bounded by M_C and has Lipschitz constant $L_C = 1$, the product rule gives

$$L_{P_4} = M_C L_{K_W} + M_{K_W}. \tag{86}$$

Combining everything together and using linearity, we get the final Lipschitz constant for $\mathcal{G}_{\text{SSM}}(W)$:

$$L_{\mathcal{G}_{\text{SSM},R}} = L_\ell L_x \left(\frac{\sqrt{ND}}{1-\rho} L_{P_1} + \frac{1}{1-\rho} L_{P_4} \right). \tag{87}$$

Because $R < \infty$, the block norms of W^r are finite, and $\rho < 1$, all constants M_{K_Δ}, M_{K_B}, M_{P_1}, M_{P_2}, M_{P_3}, M_{K_W}, L_{P_1}, L_{P_2}, L_{P_3}, L_{K_W}, and L_{P_4} are finite. As such, $L_{\mathcal{G}_{\text{SSM},R}} < \infty$, and $\forall W, V \in \mathcal{N}_R(W^r)$,

$$|\mathcal{G}_{\text{SSM}}(W) - \mathcal{G}_{\text{SSM}}(V)| \leq L_{\mathcal{G}_{\text{SSM},R}} \|W - V\|_F. \tag{88}$$

This proves that $\mathcal{G}_{\text{SSM}}(W)$ is locally Lipschitz continuous on $\mathcal{N}_R(W^r)$. □

B.1.2 LOCAL SSM SMOOTHNESS LEMMA PROOF

Proof of Lemma 4.7: Fix the reference tuple W^r and the radius R from the lemma. Let $\xi = (x, y^*) \in \mathcal{Z}$ be an arbitrary admissible sequence pair of finite length $S \geq 1$. The inputs and targets are held fixed when differentiating with respect to the model parameters. For this fixed sequence pair, we use a simplified notation for $h_t(W; x)$, $y_t(W; x)$, and $\ell_t(y_t(W; x), y_t^*)$ as $h_t(W)$, $y_t(W)$, and $\ell_t(y_t(W))$, respectively.

Throughout the proof, $L_\ell, H_\ell, L_x, \gamma, \delta$ are fixed uniformly over $\mathcal{N}_R(W^r)$, with $\rho = \exp(-\gamma\delta) < 1$. The estimates below hold for every $W \in \mathcal{N}_R(W^r)$, every finite sequence length $S \geq 1$, every admissible length-S pair $\xi \in \mathcal{Z}$, and every token position $t \leq S$, with constants independent of S, ξ, and t.

We use the parameter envelopes from (57)-(60) and the induced bounds from (66), (65), and (84).

For convenience, we define:

$$H_R := M_{K_B} M_{K_\Delta} \frac{\sqrt{ND}}{1-\rho} \tag{89}$$

$$J_R := \frac{M_{K_W}}{1-\rho} \tag{90}$$

By the definition of M_X as an upper-bound and from the Proof of Theorem 4.4, we know the following hold:

$$\|h_t(W)\|_F \leq K_B(W) K_\Delta(W) \frac{\sqrt{ND}}{1-\rho} \leq H_R \tag{91}$$

$$\left\| \frac{\partial h_t(W)}{\partial W} \right\|_F \leq \frac{K_W(W)}{1-\rho} \leq J_R \tag{92}$$

Given this setup, we begin by bounding $D^2 h_t$ where D is the Fréchet derivative defined:

$$DF(W)[U] := \frac{d}{d\epsilon} F(W + \epsilon U)\bigg|_{\epsilon=0}$$

$$D^2 F(W)[U, V] := \frac{\partial^2}{\partial\epsilon\partial\zeta} F(W + \epsilon U + \zeta V)\bigg|_{\epsilon=\zeta=0}$$

with U, V being unit perturbations. We also utilize the operator norm defined $\|D\Psi(W)\|_{\text{op}} := \sup_{\|U\|_F=1} \|D\Psi(W)[U]\|_F$. Note that after flattening the input and output tensors, this corresponds to the usual spectral norm of the corresponding Jacobian matrix. Parameter Hessians are represented as matrices using the vectorization of parameter blocks introduced in the single-layer proof. Their operator norms are taken with respect to the corresponding Euclidean norm, which equals the block Frobenius norm before vectorization.

Let $U = (U_\Delta, U_B, U_C, U_A)$ and $V = (V_\Delta, V_B, V_C, V_A)$ be arbitrary block perturbations satisfying

$$\|U\|_F = \|V\|_F = 1. \tag{93}$$

Each perturbation block therefore has Frobenius norm at most one. Since Δ_t depends only on W_Δ, we write $D\Delta_t(W)[U_\Delta]$ as shorthand for $D\Delta_t(W)[(U_\Delta, 0, 0, 0)]$, and use the analogous convention for its second derivative.

By the property of the operator norm (with $\Psi(W) = h_t(W)$),

$$\|Dh_t(W)[U]\|_F \leq \|Dh_t(W)\|_{\text{op}} \|U\|_F = \|Dh_t(W)\|_{\text{op}} \leq \left\| \frac{\partial h_t(W)}{\partial W} \right\|_F \leq J_R. \tag{94}$$

The same argument gives

$$\|Dh_t(W)[V]\|_F \leq J_R. \tag{95}$$

and therefore it follows that

$$\|Dh_t(W)[U]\|_F \leq J_R, \qquad \|Dh_t(W)[V]\|_F \leq J_R. \tag{96}$$

Next, let $b_t(W) := \bar{B}_t(W)\,\mathrm{diag}(x_t)$, then the hidden state recurrence equation becomes:

$$h_t(W) = \bar{A}_t(W) \odot h_{t-1}(W) + b_t(W), \tag{97}$$

and taking the first Fréchet derivative in direction U gives:

$$Dh_t(W)[U] = D\bar{A}_t(W)[U] \odot h_{t-1}(W) + \bar{A}_t(W) \odot Dh_{t-1}(W)[U] + Db_t(W)[U]. \tag{98}$$

Taking a second Fréchet derivative in direction V gives

$$\begin{aligned} D^2h_t(W)[U,V] = {} & D^2\bar{A}_t(W)[U,V] \odot h_{t-1}(W) \\ & + D\bar{A}_t(W)[U] \odot Dh_{t-1}(W)[V] \\ & + D\bar{A}_t(W)[V] \odot Dh_{t-1}(W)[U] \\ & + \bar{A}_t(W) \odot D^2h_{t-1}(W)[U,V] \\ & + D^2b_t(W)[U,V]. \end{aligned} \tag{99}$$

Next, taking the Frobenius norms, and by the triangle inequality, it follows that

$$\begin{aligned} \|D^2h_t(W)[U,V]\|_F \leq {} & \|D^2\bar{A}_t(W)[U,V] \odot h_{t-1}(W)\|_F \\ & + \|D\bar{A}_t(W)[U] \odot Dh_{t-1}(W)[V]\|_F \\ & + \|D\bar{A}_t(W)[V] \odot Dh_{t-1}(W)[U]\|_F \\ & + \|\bar{A}_t(W) \odot D^2h_{t-1}(W)[U,V]\|_F \\ & + \|D^2b_t(W)[U,V]\|_F. \end{aligned} \tag{100}$$

Using $\|X \odot Y\|_F \leq \|X\|_F\|Y\|_F$ and $\|X \odot Y\|_F \leq \|X\|_{\max}\|Y\|_F$, we obtain:

$$\begin{aligned} \|D^2h_t(W)[U,V]\|_F \leq {} & \|D^2\bar{A}_t(W)[U,V]\|_F\|h_{t-1}(W)\|_F \\ & + \|D\bar{A}_t(W)[U]\|_F\|Dh_{t-1}(W)[V]\|_F \\ & + \|D\bar{A}_t(W)[V]\|_F\|Dh_{t-1}(W)[U]\|_F \\ & + \|\bar{A}_t(W)\|_{\max}\|D^2h_{t-1}(W)[U,V]\|_F \\ & + \|D^2b_t(W)[U,V]\|_F. \end{aligned} \tag{101}$$

Then, we already derived the bounds:

$$\|h_{t-1}(W)\|_F \leq H_R \qquad \|Dh_{t-1}(W)[U]\|_F \leq J_R \qquad \|Dh_{t-1}(W)[V]\|_F \leq J_R,$$

and by Definition 4.1:

$$\left\|\bar{A}_t(W)\right\|_{\max} \leq \rho,$$

so the bound becomes:

$$\begin{aligned} \|D^2h_t(W)[U,V]\|_F \leq {} & H_R\|D^2\bar{A}_t(W)[U,V]\|_F \\ & + J_R\|D\bar{A}_t(W)[U]\|_F + J_R\|D\bar{A}_t(W)[V]\|_F \\ & + \rho\|D^2h_{t-1}(W)[U,V]\|_F + \|D^2b_t(W)[U,V]\|_F. \end{aligned} \tag{102}$$

Next, we derive the three missing bounds in (102):

$$\left\|D\bar{A}_t(W)[U]\right\|_F, \quad \left\|D^2\bar{A}_t(W)[U,V]\right\|_F, \quad \left\|D^2b_t(W)[U,V]\right\|_F \tag{103}$$

noting that the same upper bound applies to $\|D\bar{A}_t(W)[U]\|_F$ and $\|D\bar{A}_t(W)[V]\|_F$.

We first derive bounds on the intermediate terms that comprise $\bar{A}_t$, and $\bar{B}_t$, starting with $\Delta_t(W)$.

From the definition of $\Delta_t(W)$:

$$D\Delta_t(W)[U_\Delta] = \mathrm{diag}\left(\mathrm{Softplus}'(W_\Delta x_t)\right) U_\Delta x_t, \tag{104}$$

and since $0 \leq \mathrm{Softplus}'(z) \leq 1$, it follows that:

$$\|D\Delta_t(W)[U_\Delta]\|_2 \leq \|U_\Delta x_t\|_2 \leq \|U_\Delta\|_F\|x_t\|_2 \leq L_x. \tag{105}$$

Similarly,

$$D^2\Delta_t(W)[U_\Delta, V_\Delta] = \text{Softplus}''(W_\Delta x_t) \odot (U_\Delta x_t) \odot (V_\Delta x_t). \tag{106}$$

Since $0 \leq \text{Softplus}''(z) \leq 1/4$, the bound becomes:

$$\left\|D^2\Delta_t(W)[U_\Delta, V_\Delta]\right\|_2 \leq \frac{1}{4}\left\|(U_\Delta x_t) \odot (V_\Delta x_t)\right\|_2 \leq \frac{1}{4}\|U_\Delta x_t\|_2\|V_\Delta x_t\|_2 \leq \frac{1}{4}L_x^2.$$

These bounds allow us to bound $D\bar{A}_t(W)[U]$. We define an intermediate matrix

$$Z_t(W) := A\Delta_t(W)^\top, \tag{107}$$

and calculate its derivative

$$DZ_t(W)[U] = U_A\Delta_t(W)^\top + A\left(D\Delta_t(W)[U_\Delta]\right)^\top. \tag{108}$$

Next, using the property of Frobenius norm for outer product $\|xy^\top\|_F = \|x\|_2\|y\|_2$, it follows that:

$$\|DZ_t(W)[U]\|_F \leq \|U_A\|_2\|\Delta_t(W)\|_2 + \|A\|_2\left\|D\Delta_t(W)[U_\Delta]\right\|_2 \leq M_{K_\Delta} + M_A L_x. \tag{109}$$

This allows us to calculate:

$$\begin{aligned}\left\|D\bar{A}_t(W)[U]\right\|_F &= \left\|\bar{A}_t(W) \odot DZ_t(W)[U]\right\|_F \\ &\leq \left\|\bar{A}_t(W)\right\|_{\max}\|DZ_t(W)[U]\|_F \leq \rho\left(M_{K_\Delta} + M_A L_x\right) := K_{A_{1,R}}\end{aligned} \tag{110}$$

Next, we bound $\left\|D^2\bar{A}_t(W)[U,V]\right\|_F$ by differentiating $D\bar{A}_t(W)[U]$ in direction V:

$$D^2\bar{A}_t(W)[U,V] = \bar{A}_t(W) \odot \left(DZ_t(W)[V] \odot DZ_t(W)[U] + D^2Z_t(W)[U,V]\right). \tag{111}$$

To do that we first bound:

$$D^2Z_t(W)[U,V] = U_A\left(D\Delta_t(W)[V_\Delta]\right)^\top + V_A\left(D\Delta_t(W)[U_\Delta]\right)^\top + A\left(D^2\Delta_t(W)[U_\Delta, V_\Delta]\right)^\top. \tag{112}$$

We then apply previously derived bounds to the individual components:

$$\begin{aligned}\left\|D^2Z_t(W)[U,V]\right\|_F \leq& \|U_A\|_2\left\|D\Delta_t(W)[V_\Delta]\right\|_2 \\ &+ \|V_A\|_2\left\|D\Delta_t(W)[U_\Delta]\right\|_2 + \|A\|_2\left\|D^2\Delta_t(W)[U_\Delta, V_\Delta]\right\|_2 \\ \leq& 2L_x + \frac{1}{4}M_A L_x^2.\end{aligned} \tag{113}$$

Plugging the result back in, we get the second component bound:

$$\begin{aligned}\left\|D^2\bar{A}_t(W)[U,V]\right\|_F &\leq \left\|\bar{A}_t(W)\right\|_{\max}\left(\|DZ_t(W)[V]\|_F\|DZ_t(W)[U]\|_F + \left\|D^2Z_t(W)[U,V]\right\|_F\right) \\ &\leq \rho\left(\left(M_{K_\Delta} + M_A L_x\right)^2 + 2L_x + \frac{1}{4}M_A L_x^2\right) := K_{A_{2,R}}\end{aligned} \tag{114}$$

Finally, we bound the third term $D^2b_t(W)$. From definition, $b_t(W) = \bar{B}_t(W)\,\text{diag}(x_t)$, where $\bar{B}_t(W) = (W_B x_t)\Delta_t(W)^\top$. We first compute the second derivative of $\bar{B}_t(W)$:

$$\begin{aligned}&D^2\bar{B}_t(W)[U,V] \\ &= (U_B x_t)\left(D\Delta_t(W)[V_\Delta]\right)^\top + (V_B x_t)(D\Delta_t(W)[U_\Delta])^\top + (W_B x_t)(D^2\Delta_t(W)[U_\Delta, V_\Delta])^\top,\end{aligned} \tag{115}$$

and using the previously derived bounds, it follows that:

$$\left\|D^2\bar{B}_t(W)[U,V]\right\|_F \leq 2L_x^2 + \frac{1}{4}M_B L_x^3. \tag{116}$$

With that, we get:

$$\begin{aligned}\left\|D^2b_t(W)[U,V]\right\|_F &\leq \left\|D^2\bar{B}_t(W)[U,V]\right\|_F\|\text{diag}(x_t)\|_F \\ &\leq \left(2L_x^2 + \frac{1}{4}M_B L_x^3\right)L_x = 2L_x^3 + \frac{1}{4}M_B L_x^4 := K_{B_{2,R}}\end{aligned} \tag{117}$$

Given these derived bounds, the recurrence becomes:

$$\left\|D^2 h_t(W)[U,V]\right\|_F \leq \rho \left\|D^2 h_{t-1}(W)[U,V]\right\|_F + K_{A_{2,R}} H_R + 2K_{A_{1,R}} J_R + K_{B_{2,R}}. \quad (118)$$

Since, as mentioned in the proof of 4.4, the initial state is initialized as $h_0 = 0$, we have:

$$D^2 h_0(W)[U,V] = 0. \quad (119)$$

As such, unrolling the recurrence we get:

$$\begin{aligned}\left\|D^2 h_t(W)[U,V]\right\|_F &\leq \left(K_{A_{2,R}} H_R + 2K_{A_{1,R}} J_R + K_{B_{2,R}}\right) \sum_{m=0}^{t-1} \rho^m \\ &\leq \frac{K_{A_{2,R}} H_R + 2K_{A_{1,R}} J_R + K_{B_{2,R}}}{1-\rho} := K_{H_{2,R}} \end{aligned} \quad (120)$$

Therefore, for all $W \in \mathcal{N}_R(W^r)$ and all unit block perturbations U, V,

$$\left\|D^2 h_t(W)[U,V]\right\|_F \leq K_{H_{2,R}}. \quad (121)$$

Next, we use this to derive the first and second Fréchet derivatives of the output:

$$y_t(W) = h_t(W)^\top (W_C x_t). \quad (122)$$

The first derivative is:

$$Dy_t(W)[U] = Dh_t(W)[U]^\top C_t(W) + h_t(W)^\top U_C x_t, \quad (123)$$

resulting in a bound

$$\begin{aligned}\|Dy_t(W)[U]\|_2 &\leq \|Dh_t(W)[U]\|_F \|W_C\|_2 \|x_t\|_2 + \|h_t(W)\|_F \|U_C x_t\|_2 \\ &\leq J_R M_C L_x + H_R L_x := K_{Y_{1,R}}. \end{aligned} \quad (124)$$

Taking the Fréchet derivative in direction V, we get:

$$D^2 y_t(W)[U,V] = D^2 h_t(W)[U,V]^\top C_t(W) + Dh_t(W)[U]^\top V_C x_t + Dh_t(W)[V]^\top U_C x_t, \quad (125)$$

where we also use the fact that $D^2(W_C x_t) = 0$. Then:

$$\begin{aligned}\left\|D^2 y_t(W)[U,V]\right\|_2 \leq& \|D^2 h_t(W)[U,V]\|_F \|(W_C x_t)\|_2 \\ &+ \|Dh_t(W)[U]\|_F \|(V_C x_t)\|_2 + \|Dh_t(W)[V]\|_F \|(U_C x_t)\|_2 \\ \leq& K_{H_{2,R}} M_C L_x + 2J_R L_x := K_{Y_{2,R}}. \end{aligned} \quad (126)$$

Finally, consider the token loss $\ell_t(y_t(W))$. By the second-order chain rule:

$$D^2_W \ell_t(y_t(W))[U,V] = D^2_y \ell_t(y_t)[Dy_t(W)[U], Dy_t(W)[V]] + D_y \ell_t(y_t)[D^2 y_t(W)[U,V]], \quad (127)$$

where we use the notation $D^2_y \ell_t(y_t)[a,b] = a^\top \nabla^2_y \ell_t(y_t) b$.

By Assumption 4.2, and utilizing all derived bounds, we get:

$$\begin{aligned}\left|D^2_W \ell_t(y_t(W))[U,V]\right| &\leq \left|D^2_y \ell_t(y_t)[Dy_t(W)[U], Dy_t(W)[V]]\right| + \left|D_y \ell_t(y_t)[D^2 y_t(W)[U,V]]\right| \\ &\leq \|\nabla^2_y \ell_t(y_t)\|_{\mathrm{op}} \|Dy_t(W)[U]\|_2 \|Dy_t(W)[V]\|_2 + \|\nabla_y \ell_t(y_t)\|_2 \|D^2 y_t(W)[U,V]\|_2 \\ &\leq H_\ell K^2_{Y_{1,R}} + L_\ell K_{Y_{2,R}} := L_{\mathrm{SSM},R}. \end{aligned} \quad (128)$$

Substituting the definitions, we get:

$$\begin{aligned}L_{\mathrm{SSM},R} =& H_\ell L_x^2 \left(H_R + M_C J_R\right)^2 + L_\ell L_x \left(M_C K_{H_{2,R}} + 2J_R\right) \\ =& H_\ell L_x^2 \left(\frac{M_{K_B} M_{K_\Delta} \sqrt{ND} + M_C M_{K_W}}{1-\rho}\right)^2 \\ &+ L_\ell L_x \left[\frac{M_C}{1-\rho}\left(\rho\left((M_{K_\Delta} + M_A L_x)^2 + 2L_x + \frac{1}{4} M_A L_x^2\right) \frac{M_{K_B} M_{K_\Delta} \sqrt{ND}}{1-\rho}\right.\right.\end{aligned}$$

$$+ 2\rho(M_{K_\Delta} + M_A L_x)\frac{M_{K_W}}{1-\rho} + 2L_x^3 + \frac{1}{4}M_B L_x^4\Bigg) + \frac{2M_{K_W}}{1-\rho}\Bigg]. \tag{129}$$

Since U, V were arbitrary unit block perturbations, the previous bound implies

$$\left\|\nabla_W^2 \ell_t(y_t(W))\right\|_{\mathrm{op}} \le L_{\mathrm{SSM},R}. \tag{130}$$

Finally, the sequence loss was defined in (3) as

$$\mathcal{L}^S(W) = \frac{1}{S}\sum_{t=1}^{S} \ell_t(y_t(W)),$$

using the simplified notation, where $\mathcal{L}^S(W) := \mathcal{L}^S(W;\xi)$ and $\ell_t(y_t(W)) = \ell_t(y_t(W;x), y_t^*)$, and we further incorporate W to emphasize y_t's dependence on the current weights.

As a result,

$$\begin{aligned}
\|\nabla_W^2 \mathcal{L}^S(W)\|_{\mathrm{op}} &= \left\|\frac{1}{S}\sum_{t=1}^{S} \nabla_W^2 \ell_t(y_t(W))\right\|_{\mathrm{op}} \\
&\le \frac{1}{S}\sum_{t=1}^{S} \|\nabla_W^2 \ell_t(y_t(W))\|_{\mathrm{op}} \\
&\le L_{\mathrm{SSM},R}.
\end{aligned} \tag{131}$$

Then, for an client k, the local expected objective is

$$\mathcal{L}_k(W) = \mathbb{E}_{\xi\sim\mathcal{P}_k}\left[\mathcal{L}^S(W;\xi)\right]. \tag{132}$$

For any $W, V \in \mathcal{N}_R(W^r)$, convexity of the neighborhood implies that

$$W_\alpha := V + \alpha(W - V), \qquad \alpha \in [0,1], \tag{133}$$

lies in $\mathcal{N}_R(W^r)$. Applying the fundamental theorem of calculus to the fixed-sequence gradient and using the Hessian bound from (131) gives

$$\begin{aligned}
\left\|\nabla_W \mathcal{L}^S(W;\xi) - \nabla_W \mathcal{L}^S(V;\xi)\right\|_F &= \left\|\mathrm{vec}\left(\nabla_W \mathcal{L}^S(W;\xi) - \nabla_W \mathcal{L}^S(V;\xi)\right)\right\|_2 \\
&= \left\|\int_0^1 \nabla_W^2 \mathcal{L}^S(W_\alpha;\xi)\,\mathrm{vec}(W-V)d\alpha\right\|_2 \\
&\overset{(a)}{\le} \int_0^1 \left\|\nabla_W^2 \mathcal{L}^S(W_\alpha;\xi)\,\mathrm{vec}(W-V)\right\|_2 d\alpha \\
&\overset{(b)}{\le} \int_0^1 \left\|\nabla_W^2 \mathcal{L}^S(W_\alpha;\xi)\right\|_{\mathrm{op}} \|W-V\|_F\, d\alpha \\
&\overset{(c)}{\le} \int_0^1 L_{\mathrm{SSM},R}\|W-V\|_F\, d\alpha = L_{\mathrm{SSM},R}\|W-V\|_F.
\end{aligned} \tag{134}$$

where:

- (a) follows from the triangle inequality for an integral $\left\|\int v\right\| \le \int \|v\|$.
- (b) follows from the operator-norm inequality: $\|Hd\|_2 \le \|H\|_{\mathrm{op}}\|d\|_2$ and the fact that $\|\mathrm{vec}(W-V)\|_2 = \|W-V\|_F$.
- (c) follows from (131).

This inequality holds uniformly over all admissible sequence pairs and finite sequence lengths.

Finally, by Assumption 4.6 and Jensen's inequality,

$$\begin{aligned}
\|\nabla_W \mathcal{L}_k(W) - \nabla_W \mathcal{L}_k(V)\|_F &= \left\|\mathbb{E}_{\xi\sim\mathcal{P}_k}\left[\nabla_W \mathcal{L}^S(W;\xi) - \nabla_W \mathcal{L}^S(V;\xi)\right]\right\|_F \\
&\le \mathbb{E}_{\xi\sim\mathcal{P}_k}\left[\|\nabla_W \mathcal{L}^S(W;\xi) - \nabla_W \mathcal{L}^S(V;\xi)\|_F\right] \\
&\le L_{\mathrm{SSM},R}\|W-V\|_F.
\end{aligned} \tag{135}$$

Therefore, $\mathcal{L}_k$ is $L_{\mathrm{SSM},R}$-smooth on $\mathcal{N}_R(W^r)$. □

B.1.3 SSM LOSS GRADIENT SECOND-MOMENT CONSTANT LEMMA AND PROOF

In the following two lemmas, we consider some arbitrary client data distribution $\mathcal{P}$ and write $\mathcal{L}(W) = \mathbb{E}_{\xi\sim\mathcal{P}}\left[\mathcal{L}^S(W;\xi)\right]$. Expectations written as $\mathbb{E}_\xi$ are taken with respect to the distribution $\mathcal{P}$.

Lemma B.2 (SSM loss gradient second-moment constant)**.** *Suppose the Assumptions 4.2 and 4.3 hold. Let $\mathcal{N}_R(W^r) \subseteq \mathcal{S}_{\gamma,\delta}$ be a stable local model neighborhood around the round-r broadcast model W^r. Let:*

$$g(W;\xi) := \nabla_W \mathcal{L}^S(W;\xi) \tag{136}$$

denote the stochastic sequence-level gradient for a sampled input-target sequence pair $\xi = (x_{1:S}, y^*_{1:S})$. Then, for every $W \in \mathcal{N}_R(W^r)$,

$$\mathbb{E}_\xi \left\|g(W;\xi)\right\|_F^2 \le G^2_{\mathrm{SSM},R}, \tag{137}$$

where

$$G_{\mathrm{SSM},R} := L_\ell L_x \left(M_{K_B} M_{K_\Delta} \frac{\sqrt{ND}}{1-\rho} + M_C \frac{M_{K_W}}{1-\rho} \right), \tag{138}$$

where $M_{K_B}, M_{K_\Delta}, M_C,$ and M_{K_W} are defined in (66)*,* (65)*,* (59)*, and* (84) *respectively.*

Proof of Lemma B.2: By Theorem 4.4, under the assumptions of the lemma, for every $W \in \mathcal{N}_R(W^r)$, every finite integer $S \ge 1$, and every admissible length-S sequence pair $\xi \in \mathcal{Z}$, we have:

$$\left\|\nabla_W \mathcal{L}^S(W;\xi)\right\|_F \le L_\ell L_x \left(K_B K_\Delta \frac{\sqrt{ND}}{1-\rho} + \|W_C\|_2 \frac{K_W}{1-\rho} \right).$$

Since $g(W;\xi) = \nabla_W \mathcal{L}^S(W;\xi)$, it follows that:

$$\|g(W;\xi)\|_F \le L_\ell L_x \left(K_B K_\Delta \frac{\sqrt{ND}}{1-\rho} + \|W_C\|_2 \frac{K_W}{1-\rho} \right). \tag{139}$$

Restricting to $W \in \mathcal{N}_R(W^r)$, by the local neighborhood bounds defined previously ((66), (65), (59), and (84)),

$$K_B \le M_{K_B}, \qquad K_\Delta \le M_{K_\Delta}, \qquad K_W \le M_{K_W}, \qquad \|W_C\|_2 \le M_C. \tag{140}$$

Therefore,

$$\|g(W;\xi)\|_F \le L_\ell L_x \left(M_{K_B} M_{K_\Delta} \frac{\sqrt{ND}}{1-\rho} + M_C \frac{M_{K_W}}{1-\rho} \right) = G_{\mathrm{SSM},R}. \tag{141}$$

Since both values are non-negative, squaring both sides gives:

$$\|g(W;\xi)\|_F^2 \le G^2_{\mathrm{SSM},R} = L_\ell^2 L_x^2 \left(M_{K_B} M_{K_\Delta} \frac{\sqrt{ND}}{1-\rho} + M_C \frac{M_{K_W}}{1-\rho} \right)^2 \tag{142}$$

Finally, taking expectation over the sampled sequence ξ preserves the bound:

$$\mathbb{E}_\xi \left[\|g(W;\xi)\|_F^2 \right] \le G^2_{\mathrm{SSM},R} \tag{143}$$

proving the lemma. □

B.1.4 SSM STOCHASTIC GRADIENT VARIANCE BOUND LEMMA AND PROOF

Lemma B.3 (SSM stochastic gradient variance bound)**.** *Suppose the assumptions of Lemma B.2 and Assumption 4.6 hold. Let*

$$\mathcal{L}(W) := \mathbb{E}_\xi \left[\mathcal{L}^S(W;\xi) \right]$$

denote the expected sequence loss over the local data distribution, and let

$$g(W;\xi) := \nabla_W \mathcal{L}^S(W;\xi)$$

denote the single-sequence stochastic gradient. For a mini-batch $\Xi_\xi = \{\xi_1, \dots, \xi_\Xi\}$ of Ξ independent sampled sequences, define

$$g_\Xi(W; \Xi_\xi) := \frac{1}{\Xi} \sum_{j=1}^{\Xi} g(W; \xi_j). \tag{144}$$

Then, for every $W \in \mathcal{N}_R(W^r)$ it follows that

$$\mathbb{E}_{\Xi_\xi} \left\| g_\Xi(W; \Xi_\xi) - \nabla_W \mathcal{L}(W) \right\|_F^2 \le \sigma^2_{\mathrm{SSM},R,\Xi}, \tag{145}$$

where

$$\sigma^2_{\mathrm{SSM},R,\Xi} := \frac{1}{\Xi} \sup_{W \in \mathcal{N}_R(W^r)} \left(\mathbb{E}_\xi \|g(W;\xi)\|_F^2 - \|\nabla_W \mathcal{L}(W)\|_F^2 \right). \tag{146}$$

Moreover, by Lemma B.2,

$$\sigma^2_{\mathrm{SSM},R,\Xi} \le \frac{G^2_{\mathrm{SSM},R}}{\Xi}. \tag{147}$$

Proof of Lemma B.3: By unbiasedness and independence of the sampled sequences, the standard mini-batch variance identity gives:

$$\mathbb{E}_{\Xi_\xi} \left[\left\| g_\Xi(W; \Xi_\xi) - \nabla_W \mathcal{L}(W) \right\|_F^2 \right] = \frac{1}{\Xi} \mathbb{E}_\xi \left[\left\| g(W;\xi) - \nabla_W \mathcal{L}(W) \right\|_F^2 \right].$$

Using the identity

$$\mathbb{E}_\xi \left[\left\| g(W;\xi) - \nabla_W \mathcal{L}(W) \right\|_F^2 \right] = \mathbb{E}_\xi \|g(W;\xi)\|_F^2 - \|\nabla_W \mathcal{L}(W)\|_F^2, \tag{148}$$

and taking the supremum over $W \in \mathcal{N}_R(W^r)$, we obtain

$$\mathbb{E}_{\Xi_\xi} \left[\left\| g_\Xi(W; \Xi_\xi) - \nabla_W \mathcal{L}(W) \right\|_F^2 \right] \le \sigma^2_{\mathrm{SSM},R,\Xi}. \tag{149}$$

Finally, Lemma B.2 gives

$$\mathbb{E}_\xi \|g(W;\xi)\|_F^2 \le G^2_{\mathrm{SSM},R},$$

and since $\|\nabla_W \mathcal{L}(W)\|_F^2 \ge 0$,

$$\sigma^2_{\mathrm{SSM},R,\Xi} \le \frac{G^2_{\mathrm{SSM},R}}{\Xi}. \tag{150}$$

This proves the claim. □

B.1.5 SSM CLIENT HETEROGENEITY BOUND UNDER TOTAL VARIATION LEMMA AND PROOF

Lemma B.4 (SSM client heterogeneity bound under total variation)**.** *Suppose the assumptions of Lemma B.2 and Assumption 4.6 hold. Let*

$$f(W) := \sum_{k=1}^{K} p_k \mathcal{L}_k(W) \tag{151}$$

be the global objective, where $p_k \ge 0$ and $\sum_{k=1}^K p_k = 1$. Let $\mathcal{P}_k$ denote the local data distribution of client k, and define the global mixture distribution as:

$$\bar{\mathcal{P}} := \sum_{k=1}^{K} p_k \mathcal{P}_k. \tag{152}$$

Define the local client heterogeneity constant

$$\Gamma^2_{\mathrm{SSM},R} := \sup_{W \in \mathcal{N}_R(W^r)} \sum_{k=1}^{K} p_k \left\| \nabla_W \mathcal{L}_k(W) - \nabla_W f(W) \right\|_F^2. \tag{153}$$

Then:

$$\Gamma^2_{\mathrm{SSM},R} \le 4 G^2_{\mathrm{SSM},R} \sum_{k=1}^{K} p_k \, \mathrm{TV}(\mathcal{P}_k, \bar{\mathcal{P}})^2, \tag{154}$$

where

$$\mathrm{TV}(P, Q) := \sup_A |P(A) - Q(A)|. \tag{155}$$

Proof of Lemma B.4: Recall that for each client k, the local objective is:

$$\mathcal{L}_k(W) = \mathbb{E}_{\xi\sim\mathcal{P}_k}\left[\mathcal{L}^S(W;\xi)\right],$$

and, by Assumption 4.6,

$$\nabla_W \mathcal{L}_k(W) = \mathbb{E}_{\xi\sim\mathcal{P}_k}\left[g(W;\xi)\right]. \tag{156}$$

Similarly, since

$$f(W) := \sum_{k=1}^{K} p_k \mathcal{L}_k(W),$$

we have

$$\nabla_W f(W) = \sum_{k=1}^{K} p_k \nabla_W \mathcal{L}_k(W) = \sum_{k=1}^{K} p_k \mathbb{E}_{\xi\sim\mathcal{P}_k}\left[g(W;\xi)\right]. \tag{157}$$

Next, by the definition of the mixture distribution $\bar{\mathcal{P}}$, we can write this as:

$$\nabla_W f(W) = \mathbb{E}_{\xi\sim\bar{\mathcal{P}}}\left[g(W;\xi)\right]. \tag{158}$$

Therefore, it follows that

$$\nabla_W \mathcal{L}_k(W) - \nabla_W f(W) = \mathbb{E}_{\xi\sim\mathcal{P}_k}[g(W;\xi)] - \mathbb{E}_{\xi\sim\bar{\mathcal{P}}}[g(W;\xi)]. \tag{159}$$

By the pointwise bound (141) established in the proof of Lemma B.2, for every $W \in \mathcal{N}_R(W^r)$ and every admissible sequence pair $\xi \in \mathcal{Z}$,

$$\|g(W;\xi)\|_F \le G_{\mathrm{SSM},R}.$$

Since $\mathcal{P}_k(\mathcal{Z}) = \bar{\mathcal{P}}(\mathcal{Z}) = 1$, the total-variation inequality for bounded vector-valued functions gives:

$$\left\|\mathbb{E}_{\xi\sim\mathcal{P}_k}[g(W;\xi)] - \mathbb{E}_{\xi\sim\bar{\mathcal{P}}}[g(W;\xi)]\right\|_F \le 2G_{\mathrm{SSM},R}\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}}). \tag{160}$$

So:

$$\|\nabla_W \mathcal{L}_k(W) - \nabla_W f(W)\|_F^2 \le 4G_{\mathrm{SSM},R}^2\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2. \tag{161}$$

Multiplying both sides by p_k and summing over all clients gives:

$$\sum_{k=1}^{K} p_k \|\nabla_W \mathcal{L}_k(W) - \nabla_W f(W)\|_F^2 \le 4G_{\mathrm{SSM},R}^2 \sum_{k=1}^{K} p_k\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2. \tag{162}$$

Since the right-hand side of this inequality is independent of W, taking the supremum over $W \in \mathcal{N}_R(W^r)$ we get:

$$\Gamma_{\mathrm{SSM},R}^2 := \sup_{W\in\mathcal{N}_R(W^r)} \sum_{k=1}^{K} p_k \|\nabla_W \mathcal{L}_k(W) - \nabla_W f(W)\|_F^2 \le 4G_{\mathrm{SSM},R}^2 \sum_{k=1}^{K} p_k\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2. \tag{163}$$

This proves the SSM client heterogeneity bound. □

B.2 Theorem B.5: FedAvg SSM Convergence and Proof

Theorem B.5 (FedAvg SSM Convergence)**.** *Consider a distributed learning framework established in Section 3, with the objective*

$$f(W) := \sum_{k=1}^{K} p_k \mathcal{L}_k(W), \qquad p_k \ge 0, \qquad \sum_{k=1}^{K} p_k = 1,$$

where W parameterizes a single-layer selective SSM defined in the same section and each $\mathcal{L}_k$ is finite and differentiable on the considered neighborhoods.

Let W^0 be deterministic with $f(W^0) < \infty$ and suppose f is bounded below by a finite constant f^*.

Consider full-participation FedAvg distributed training algorithm applied to $f(W)$ for $T \geq 1$ communication rounds, $E \geq 1$ local steps per round, $\Xi \geq 1$ mini-batch size, and constant deterministic local step size $\eta > 0$, with the global model evolution described as:

$$\begin{aligned} W_k^{r,0} &= W^r, \\ W_k^{r,e+1} &= W_k^{r,e} - \eta g_{k,\Xi}^{r,e} \quad \text{for} \quad e \in \{0, \dots, E-1\}, \\ W^{r+1} &= \sum_{k=1}^{K} p_k W_k^{r,E} \end{aligned}$$

where $g_{k,\Xi}^{r,e}$ is the average of Ξ sequence-level gradients evaluated at $W_k^{r,e}$ with sampling and conditional unbiasedness, and $E, \Xi, T \in \mathbb{Z}_{>0}$.

Suppose this model evolution satisfies, almost surely:

$$W_k^{r,e} \in \mathcal{N}_R(W^r) \subseteq \mathcal{S}_{\gamma,\delta}, \quad \|W_X^r\|_2 + R \leq M_X \ \ \forall \ \ X \in \{\Delta, B, C, A\},$$

for every client k and local step $0 \leq e \leq E$, where $W_A := A$, and $R, \gamma, \delta > 0$ and the finite deterministic envelopes M_X are common across rounds. Let Assumptions 4.2- 4.6 hold uniformly on these neighborhoods.

Define $\rho := \exp(-\gamma\delta)$ and auxiliary SSM-specific quantities as follows:

$$\begin{aligned} M_{K_\Delta} &:= M_\Delta L_x + \sqrt{D}\log 2, \\ M_{K_B} &:= M_B L_x^2, \\ M_{K_W} &:= NDM_{K_B}M_{K_\Delta}\left(M_{K_\Delta} + \sqrt{D}M_A L_x\right)\frac{\rho}{1-\rho} + L_x^2\sqrt{N}M_{K_\Delta} + L_x\sqrt{D}M_{K_B}, \\ H_R &:= \frac{M_{K_B}M_{K_\Delta}\sqrt{ND}}{1-\rho}, \qquad J_R := \frac{M_{K_W}}{1-\rho}, \\ M_{A_{1,R}} &:= \rho(M_{K_\Delta} + M_A L_x), \\ M_{A_{2,R}} &:= \rho\left[(M_{K_\Delta} + M_A L_x)^2 + 2L_x + \frac{1}{4}M_A L_x^2\right], \\ M_{B_{2,R}} &:= 2L_x^3 + \frac{1}{4}M_B L_x^4, \\ M_{H_{2,R}} &:= \frac{M_{A_{2,R}}H_R + 2M_{A_{1,R}}J_R + M_{B_{2,R}}}{1-\rho}. \end{aligned}$$

These quantities determine the common SSM-specific smoothness and gradient bounds:

$$\begin{aligned} L_{\mathrm{SSM},R} &:= H_\ell L_x^2(H_R + M_C J_R)^2 + L_\ell L_x(M_C M_{H_{2,R}} + 2J_R), \\ G_{\mathrm{SSM},R} &:= L_\ell L_x(H_R + M_C J_R). \end{aligned}$$

For the client distributions, further define

$$\bar{\mathcal{P}} := \sum_{k=1}^{K} p_k \mathcal{P}_k, \qquad \mathcal{H}_{\mathrm{TV}} := \sum_{k=1}^{K} p_k \,\mathrm{TV}\left(\mathcal{P}_k, \bar{\mathcal{P}}\right)^2,$$

where $\mathrm{TV}(P,Q) := \sup_{\mathcal{A}} |P(\mathcal{A}) - Q(\mathcal{A})|$ is the total variation distance, with the supremum taken over measurable sets $\mathcal{A}$.

Then, for any constant deterministic step size $\eta > 0$ satisfying

$$\eta E L_{\mathrm{SSM},R} \leq \frac{1}{2}, \qquad 12\eta^2 L_{\mathrm{SSM},R}^2 E(E-1) \leq 1,$$

the following holds:

$$\begin{aligned} \frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\|\nabla f(W^r)\|_F^2 \leq & \frac{4\left(f(W^0) - f^*\right)}{\eta E T} + \frac{4\eta L_{\mathrm{SSM},R} G_{\mathrm{SSM},R}^2}{\Xi}\sum_{k=1}^{K} p_k^2 \\ & + \frac{6\eta^2 L_{\mathrm{SSM},R}^2 (E-1) G_{\mathrm{SSM},R}^2}{\Xi} + 48\eta^2 L_{\mathrm{SSM},R}^2 E(E-1) G_{\mathrm{SSM},R}^2 \mathcal{H}_{\mathrm{TV}}. \end{aligned}$$

Proof of Theorem B.5: We start from a standard nonconvex FedAvg convergence bound applicable to the setting considered here. Previous works establish a general bound (see Supplemental Material C.6, Equation (96) of Wang et al. (2020)) for smooth local objectives, unbiased stochastic gradients with bounded variance, and bounded client gradient dissimilarity.

For a setting with full client participation, E local SGD steps per client, and objective and aggregation weights p_k, the bound in Wang et al. (2020) takes the following form:

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\|\nabla f(W^r)\|_F^2 \le \frac{4\left(f(W^0)-f^*\right)}{\eta ET}+4\eta L\sigma^2\sum_{k=1}^{K}p_k^2+6\eta^2L^2(E-1)\sigma^2+12\eta^2L^2E(E-1)\Gamma^2, \tag{164}$$

whenever $\eta > 0$ satisfies

$$\eta LE \le \frac{1}{2}, \qquad 12\eta^2L^2E(E-1) \le 1.$$

Here, L is a common smoothness constant, σ^2 bounds the conditional variance of each mini-batch gradient estimator, and Γ^2 bounds the weighted client gradient disagreement:

$$\sum_{k=1}^{K}p_k\|\nabla\mathcal{L}_k(W)-\nabla f(W)\|_F^2 \le \Gamma^2. \tag{165}$$

Note that the inequality (164) specializes the result in Wang et al. (2020) applied to full client participation, E local SGD steps at every client, and aggregation weights equal to the objective weights p_k.

We apply this bound to the selective SSM setting by substituting the constants established in our lemmas. First, Lemma 4.7, evaluated with the common envelopes M_X, gives

$$L = L_{\mathrm{SSM},R}, \tag{166}$$

with $L_{\mathrm{SSM},R}$ defined in Theorem B.5. Since each neighborhood $\mathcal{N}_R(W^r)$ is convex and contains all local iterates in round r, it also contains their weighted averages and the segment joining W^r to W^{r+1}. As such, all smoothness inequalities used in the convergence argument hold with the common constant $L_{\mathrm{SSM},R}$.

Next, Lemma B.2 derives the common stochastic gradient second-moment bound $G^2_{\mathrm{SSM},R}$. Together with the sampling conditions in Assumption 4.6, Lemma B.3 establishes the choice

$$\sigma^2 = \frac{G^2_{\mathrm{SSM},R}}{\Xi}. \tag{167}$$

Finally, Lemma B.4 establishes the common heterogeneity bound

$$\Gamma^2 = 4G^2_{\mathrm{SSM},R}\sum_{k=1}^{K}p_k\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2 = 4G^2_{\mathrm{SSM},R}\mathcal{H}_{\mathrm{TV}}. \tag{168}$$

Substituting these quantities into (164), we obtain the inequality stated in Theorem B.5 whenever $\eta > 0$ satisfies the stated conditions with $L = L_{\mathrm{SSM},R}$. □

B.3 Theorem B.6: FedProx SSM Convergence and Proof

The FedProx algorithm modifies the local client objective by adding a proximal penalty around the round-r broadcast model W^r:

$$F_k^r(W) := \mathcal{L}_k(W) + \frac{\mu}{2}\|W-W^r\|_F^2, \qquad \mu > 0, \tag{169}$$

thus changing the local optimization dynamics.

In this section we introduce a theorem that quantifies the convergence behavior of a distributed learning framework that utilizes the FedProx with selective State Space Models, and provide a corresponding proof.

Theorem B.6 (FedProx convergence for selective SSMs)**.** *Consider a distributed learning framework established in Section 3, with the objective*

$$f(W) := \sum_{k=1}^{K} p_k \mathcal{L}_k(W), \qquad p_k \geq 0, \qquad \sum_{k=1}^{K} p_k = 1,$$

where W parameterizes a single-layer selective SSM defined in the same section and each $\mathcal{L}_k$ is finite and differentiable on the considered neighborhoods.

Let W^0 be deterministic with $f(W^0) < \infty$ and suppose f is bounded below by a finite constant f^*.

Consider full-participation FedProx distributed training algorithm applied to $f(W)$ for $T \geq 1$ communication rounds, $E \geq 1$ local steps per round, $\Xi \geq 1$ mini-batch size, constant deterministic local step size $\eta > 0$, and constant deterministic proximal coefficient $\mu > 0$, with the global model evolution described as:

$$\begin{aligned} W_k^{r,0} &= W^r, \\ W_k^{r,e+1} &= W_k^{r,e} - \eta \left(g_{k,\Xi}^{r,e} + \mu \left(W_k^{r,e} - W^r \right) \right) \quad \text{for} \quad e \in \{0, \dots, E-1\}, \\ W^{r+1} &= \sum_{k=1}^{K} p_k W_k^{r,E} \end{aligned}$$

where $g_{k,\Xi}^{r,e}$ is the average of Ξ sequence-level gradients evaluated at $W_k^{r,e}$ with sampling and conditional unbiasedness, and $E, \Xi, T \in \mathbb{Z}_{>0}$.

Suppose that, almost surely for every round $r < T$:

$$W_k^{r,e} \in \mathcal{N}_R(W^r) \subseteq \mathcal{S}_{\gamma,\delta}, \quad \|W_X^r\|_2 + R \leq M_X \ \ \forall \ X \in \{\Delta, B, C, A\}, \tag{170}$$

for every client k and local step $0 \leq e \leq E$, where $W_A := A$, and $R, \gamma, \delta > 0$ and the finite deterministic envelopes M_X are common across rounds. Let Assumptions 4.2- 4.6 hold uniformly on these neighborhoods.

Define $\rho := \exp(-\gamma\delta)$. Assume $0 < \eta\mu \leq 1$, and define the effective radius R_μ by

$$R_\mu := \min \left\{ R, \eta G_{\mathrm{SSM},R} \sum_{j=0}^{E-1} (1 - \eta\mu)^j \right\} = \min \left\{ R, \frac{G_{\mathrm{SSM},R}}{\mu} \left[1 - (1 - \eta\mu)^E \right] \right\},$$

where $G_{\mathrm{SSM},R}$ is defined in Theorem B.5, with the corresponding FedProx parameter envelopes defined as:

$$M_X^\mu := M_X - R + R_\mu \ \ \forall \ X \in \{\Delta, B, C, A\}.$$

The corresponding auxiliary SSM-specific quantities are then defined as:

$$\begin{aligned} M_{K_\Delta}^\mu &:= M_\Delta^\mu L_x + \sqrt{D} \log 2, \\ M_{K_B}^\mu &:= M_B^\mu L_x^2, \\ M_{K_W}^\mu &:= NDM_{K_B}^\mu M_{K_\Delta}^\mu \left(M_{K_\Delta}^\mu + \sqrt{D} M_A^\mu L_x \right) \frac{\rho}{1-\rho} + L_x^2 \sqrt{N} M_{K_\Delta}^\mu + L_x \sqrt{D} M_{K_B}^\mu, \\ H_{R_\mu} &:= \frac{M_{K_B}^\mu M_{K_\Delta}^\mu \sqrt{ND}}{1-\rho}, \qquad J_{R_\mu} := \frac{M_{K_W}^\mu}{1-\rho}, \\ M_{A_{1,R}}^\mu &:= \rho(M_{K_\Delta}^\mu + M_A^\mu L_x), \\ M_{A_{2,R}}^\mu &:= \rho \left[(M_{K_\Delta}^\mu + M_A^\mu L_x)^2 + 2L_x + \frac{1}{4} M_A^\mu L_x^2 \right], \\ M_{B_{2,R}}^\mu &:= 2L_x^3 + \frac{1}{4} M_B^\mu L_x^4, \\ M_{H_{2,R}}^\mu &:= \frac{M_{A_{2,R}}^\mu H_{R_\mu} + 2M_{A_{1,R}}^\mu J_{R_\mu} + M_{B_{2,R}}^\mu}{1-\rho}. \end{aligned}$$

Let:

$$L_{\mathrm{SSM},R_\mu} := H_\ell L_x^2 (H_{R_\mu} + M_C^\mu J_{R_\mu})^2 + L_\ell L_x (M_C^\mu M_{H_{2,R}}^\mu + 2J_{R_\mu}),$$
$$L_{\mathrm{prox},R_\mu} := L_{\mathrm{SSM},R_\mu} + \mu.$$
$$G_{\mathrm{SSM},R_\mu} := L_\ell L_x (H_{R_\mu} + M_C^\mu J_{R_\mu}),$$

denote common deterministic SSM-specific smoothness and gradient bounds constants evaluated on $\mathcal{N}_{R_\mu}(W^r)$.

For the client distributions, further define

$$\bar{\mathcal{P}} := \sum_{k=1}^K p_k \mathcal{P}_k, \qquad \mathcal{H}_{\mathrm{TV}} := \sum_{k=1}^K p_k \,\mathrm{TV}\left(\mathcal{P}_k, \bar{\mathcal{P}}\right)^2,$$

where $\mathrm{TV}(P,Q) := \sup_{\mathcal{A}} |P(\mathcal{A}) - Q(\mathcal{A})|$ *is the total variation distance, with the supremum taken over measurable sets* $\mathcal{A}$.

Then, for any constant deterministic step size $\eta > 0$ *satisfying*

$$\eta E L_{\mathrm{SSM},R_\mu} \le \frac{1}{2}, \qquad 12\eta^2 L_{\mathrm{prox},R_\mu}^2 E(E-1) \le 1,$$

the following bound holds:

$$\frac{1}{T}\sum_{r=0}^{T-1} \mathbb{E}\,\|\nabla f(W^r)\|_F^2 \le \frac{3\left(f(W^0) - f^*\right)}{\eta E T} + \frac{3\eta L_{\mathrm{SSM},R_\mu} G_{\mathrm{SSM},R_\mu}^2}{\Xi} \sum_{k=1}^K p_k^2 + \frac{3\eta^2 L_{\mathrm{prox},R_\mu}^2 (E-1) G_{\mathrm{SSM},R_\mu}^2}{\Xi} + 24\eta^2 L_{\mathrm{prox},R_\mu}^2 E(E-1) G_{\mathrm{SSM},R_\mu}^2 \mathcal{H}_{\mathrm{TV}}.$$

Moreover, almost surely, for every round $r < T$, *client* k, *and local step* $0 \le e \le E$,

$$\|W_k^{r,e} - W^r\|_F \le \min\left\{R, \frac{G_{\mathrm{SSM},R}}{\mu}\left[1 - (1-\eta\mu)^e\right]\right\} \le R_\mu.$$

Consequently, all local iterates and the aggregated model W^{r+1} *lie in* $\mathcal{N}_{R_\mu}(W^r) \subseteq \mathcal{S}_{\gamma,\delta}$.

Proof of Theorem B.6: The FedProx framework modifies the local update as:

$$W_k^{r,e+1} = W_k^{r,e} - \eta\left(g_{k,\Xi}^{r,e} + \mu(W_k^{r,e} - W^r)\right), \qquad W_k^{r,0} = W^r. \tag{171}$$

Subtracting the broadcast model W^r gives

$$W_k^{r,e+1} - W^r = (1-\eta\mu)(W_k^{r,e} - W^r) - \eta g_{k,\Xi}^{r,e} \tag{172}$$

Taking Frobenius norms, using $0 < \eta\mu \le 1$, and by the triangle inequality,

$$\left\|W_k^{r,e+1} - W^r\right\|_F \le (1-\eta\mu)\left\|W_k^{r,e} - W^r\right\|_F + \eta\left\|g_{k,\Xi}^{r,e}\right\|_F. \tag{173}$$

By the bound (141) established in the proof of Lemma B.2, every admissible sequence gradient satisfies $\|g_k(W;\xi)\|_F \le G_{\mathrm{SSM},R}$ for all $W \in \mathcal{N}_R(W^r)$. As such, the mini-batch gradients satisfy:

$$\left\|g_{k,\Xi}^{r,e}\right\|_F = \left\|\frac{1}{\Xi}\sum_{i=1}^{\Xi} g_k(W_k^{r,e};\xi_{k,i}^{r,e})\right\|_F \le \frac{1}{\Xi}\sum_{i=1}^{\Xi}\left\|g_k(W_k^{r,e};\xi_{k,i}^{r,e})\right\|_F \le G_{\mathrm{SSM},R} \tag{174}$$

Therefore:

$$\left\|W_k^{r,e+1} - W^r\right\|_F \le (1-\eta\mu)\left\|W_k^{r,e} - W^r\right\|_F + \eta G_{\mathrm{SSM},R}. \tag{175}$$

Unrolling the recursion from $W_k^{r,0} = W^r$ for every $0 \le e \le E$, we get:

$$\|W_k^{r,e} - W^r\|_F \le \eta G_{\mathrm{SSM},R}\sum_{e'=0}^{e-1}(1-\eta\mu)^{e'} = \frac{G_{\mathrm{SSM},R}}{\mu}\left[1 - (1-\eta\mu)^e\right] \tag{176}$$

Combining this with the established neighborhood bounds $\|W_k^{r,e} - W^r\|_F \le R$, we get:

$$\|W_k^{r,e} - W^r\|_F \le \min\left(R, \frac{G_{\mathrm{SSM},R}}{\mu}\left[1-(1-\eta\mu)^e\right]\right) \le R_\mu, \tag{177}$$

and since $p_k \ge 0$ and $\sum_k p_k = 1$, it follows that:

$$\left\|W^{r+1} - W^r\right\|_F = \left\|\sum_{k=1}^K p_k\left(W_k^{r,E} - W^r\right)\right\|_F \le \sum_{k=1}^K p_k\left\|W_k^{r,E} - W^r\right\|_F \le R_\mu. \tag{178}$$

This shows that all local iterates as well as W^{r+1} belong to $\mathcal{N}_{R_\mu}(W^r) \subseteq \mathcal{S}_{\gamma,\delta}$. Since this neighborhood is convex, the segments used below in the smoothness arguments also remain inside it.

Next, for every $W \in \mathcal{N}_{R_\mu}(W^r)$ and parameter block $X \in \{\Delta, B, C, A\}$,

$$\|W_X\|_2 \le \|W_X^r\|_2 + R_\mu \le M_X - R + R_\mu =: M_X^\mu. \tag{179}$$

These M_X^μ are uniform across rounds and deterministic. Let L_{SSM,R_μ} and G_{SSM,R_μ} denote the SSM smoothness and stochastic-gradient constants constructed using M_X^μ, with the stability, input, and loss-regularity constants remaining unchanged. In particular:

$$M_{K_\Delta}^\mu := M_\Delta^\mu L_x + \sqrt{D}\log 2, \tag{180}$$

$$M_{K_B}^\mu := M_B^\mu L_x^2, \tag{181}$$

$$M_{K_W}^\mu := NDM_{K_B}^\mu M_{K_\Delta}^\mu\left(M_{K_\Delta}^\mu + \sqrt{D}M_A^\mu L_x\right)\frac{\rho}{1-\rho} + L_x^2\sqrt{N}M_{K_\Delta}^\mu + L_x\sqrt{D}M_{K_B}^\mu, \tag{182}$$

$$H_{R_\mu} := \frac{M_{K_B}^\mu M_{K_\Delta}^\mu\sqrt{ND}}{1-\rho}, \qquad J_{R_\mu} := \frac{M_{K_W}^\mu}{1-\rho}, \tag{183}$$

$$M_{A_{1,R}}^\mu := \rho(M_{K_\Delta}^\mu + M_A^\mu L_x), \tag{184}$$

$$M_{A_{2,R}}^\mu := \rho\left[(M_{K_\Delta}^\mu + M_A^\mu L_x)^2 + 2L_x + \frac{1}{4}M_A^\mu L_x^2\right], \tag{185}$$

$$M_{B_{2,R}}^\mu := 2L_x^3 + \frac{1}{4}M_B^\mu L_x^4, \tag{186}$$

$$M_{H_{2,R}}^\mu := \frac{M_{A_{2,R}}^\mu H_{R_\mu} + 2M_{A_{1,R}}^\mu J_{R_\mu} + M_{B_{2,R}}^\mu}{1-\rho}, \tag{187}$$

$$L_{\mathrm{SSM},R_\mu} := H_\ell L_x^2(H_{R_\mu} + M_C^\mu J_{R_\mu})^2 + L_\ell L_x(M_C^\mu M_{H_{2,R}}^\mu + 2J_{R_\mu}), \tag{188}$$

$$G_{\mathrm{SSM},R_\mu} := L_\ell L_x(H_{R_\mu} + M_C^\mu J_{R_\mu}), \tag{189}$$

and in addition define:

$$L_{\mathrm{prox},R_\mu} = L_{\mathrm{SSM},R_\mu} + \mu \tag{190}$$

On this neighborhood, Lemma 4.7 implies that each $\mathcal{L}_k$ is L_{SSM,R_μ}-smooth.

Then, for any $W, V \in \mathcal{N}_{R_\mu}(W^r)$, the gradients of the local proximal objective satisfy:

$$\begin{aligned}
&\left\|\left(\nabla\mathcal{L}_k(W) + \mu(W - W^r)\right) - \left(\nabla\mathcal{L}_k(V) + \mu(V - W^r)\right)\right\|_F \\
&\quad\le \|\nabla\mathcal{L}_k(W) - \nabla\mathcal{L}_k(V)\|_F + \mu\|W - V\|_F \\
&\quad\le \left(L_{\mathrm{SSM},R_\mu} + \mu\right)\|W - V\|_F \\
&\quad= L_{\mathrm{prox},R_\mu}\|W - V\|_F.
\end{aligned} \tag{191}$$

Therefore, the local proximal objective $\mathcal{L}_k(W) + \frac{\mu}{2}\|W - W^r\|_F^2$ is L_{prox,R_μ}-smooth on $\mathcal{N}_{R_\mu}(W^r)$. Moreover, since $f = \sum_k p_k\mathcal{L}_k$ with $p_k \ge 0$ and $\sum_k p_k = 1$, the global objective f remains L_{SSM,R_μ}-smooth on the same neighborhood.

Next, define the following common deterministic variance constant:

$$\sigma_{\mathrm{SSM},R_\mu}^2 := \frac{G_{\mathrm{SSM},R_\mu}^2}{\Xi}. \tag{192}$$

By Lemma B.4, for every round $r < T$,

$$\sup_{W \in \mathcal{N}_{R_\mu}(W^r)} \sum_k p_k \|\nabla \mathcal{L}_k(W) - \nabla f(W)\|_F^2 \leq 4G_{\mathrm{SSM},R_\mu}^2 \mathcal{H}_{\mathrm{TV}}. \tag{193}$$

The right-hand side provides a common deterministic heterogeneity bound across rounds.

Next, fix a round $r < T$ and let $\mathbb{E}_r[\cdot]$ denote an expectation conditional on the training history at the beginning of that round, with W^r being fixed under this conditioning.

We begin by unrolling the local update and subtracting the starting weights:

$$\begin{aligned}
W_k^{r,e} - W^r &= -\eta \sum_{j=0}^{e-1} \left[g_{k,\Xi}^{r,j} + \mu \left(W_k^{r,j} - W^r \right) \right] \\
&= -\eta \sum_{j=0}^{e-1} \left[g_{k,\Xi}^{r,j} + \mu \left(W_k^{r,j} - W^r \right) \right] \\
&\quad + \eta \sum_{j=0}^{e-1} \left(\nabla \mathcal{L}_k \left(W^r \right) - \nabla \mathcal{L}_k \left(W_k^{r,j} \right) \right) - \eta \sum_{j=0}^{e-1} \left(\nabla \mathcal{L}_k \left(W^r \right) - \nabla \mathcal{L}_k \left(W_k^{r,j} \right) \right) \\
&= -\eta e \nabla \mathcal{L}_k \left(W^r \right) - \eta \sum_{j=0}^{e-1} \left[g_{k,\Xi}^{r,j} - \nabla \mathcal{L}_k \left(W_k^{r,j} \right) \right] \\
&\quad - \eta \sum_{j=0}^{e-1} \left[\mu \left(W_k^{r,j} - W^r \right) - \nabla \mathcal{L}_k \left(W^r \right) + \nabla \mathcal{L}_k \left(W_k^{r,j} \right) \right].
\end{aligned} \tag{194}$$

Next, when evaluating the norms of the individual quantities on the right of (194), by Lemma B.3, the following holds

$$\mathbb{E}_r \left[\left\| \sum_{j=0}^{e-1} \left(g_{k,\Xi}^{r,j} - \nabla \mathcal{L}_k \left(W_k^{r,j} \right) \right) \right\|_F^2 \right] \leq e \sigma_{\mathrm{SSM},R_\mu}^2 \tag{195}$$

for all $0 \leq e \leq E$.

Then, by substituting $W_k^{r,j}$ and W^r into (191), it follows that:

$$\left\| \mu \left(W_k^{r,j} - W^r \right) - \nabla \mathcal{L}_k \left(W^r \right) + \nabla \mathcal{L}_k \left(W_k^{r,j} \right) \right\|_F \leq L_{\mathrm{prox},R_\mu} \left\| W_k^{r,j} - W^r \right\|_F \tag{196}$$

Taking the norm of (194), applying the squared-normed inequality for a sum of three terms as well as the Cauchy-Schwarz inequality, and substituting these results, we get:

$$\begin{aligned}
\mathbb{E}_r \left[\left\| W_k^{r,e} - W^r \right\|_F^2 \right] &\leq 3\eta^2 e^2 \left\| \nabla \mathcal{L}_k \left(W^r \right) \right\|_F^2 + 3\eta^2 e \sigma_{\mathrm{SSM},R_\mu}^2 \\
&\quad + 3\eta^2 e L_{\mathrm{prox},R_\mu}^2 \sum_{j=0}^{e-1} \mathbb{E}_r \left[\left\| W_k^{r,j} - W^r \right\|_F^2 \right].
\end{aligned} \tag{197}$$

By multiplying (197) by p_k and summing over all clients, we get:

$$\begin{aligned}
\sum_{k=1}^K p_k \mathbb{E}_r \left[\left\| W_k^{r,e} - W^r \right\|_F^2 \right] &\leq 3\eta^2 e \sigma_{\mathrm{SSM},R_\mu}^2 + 3\eta^2 e^2 \sum_{k=1}^K p_k \left\| \nabla \mathcal{L}_k \left(W^r \right) \right\|_F^2 \\
&\quad + 3\eta^2 e L_{\mathrm{prox},R_\mu}^2 \sum_{j=0}^{e-1} \sum_{k=1}^K p_k \mathbb{E}_r \left[\left\| W_k^{r,j} - W^r \right\|_F^2 \right].
\end{aligned} \tag{198}$$

Next, note that:

$$\sum_{k=1}^K p_k \left\| \nabla \mathcal{L}_k \left(W^r \right) \right\|_F^2 = \sum_{k=1}^K p_k \left\| \nabla \mathcal{L}_k \left(W^r \right) + \nabla f(W^r) - \nabla f(W^r) \right\|_F^2$$

$$\leq \|\nabla f(W^r)\|_F^2 + \sum_{k=1}^{K} p_k \|\nabla \mathcal{L}_k(W^r) - \nabla f(W^r)\|_F^2$$
$$\leq \|\nabla f(W^r)\|_F^2 + 4G^2_{\mathrm{SSM},R_\mu}\mathcal{H}_{\mathrm{TV}}. \tag{199}$$

where the last inequality follows from (193). Substituting this result into (198), we get:

$$\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\|W_k^{r,e} - W^r\|_F^2\right] \leq 3\eta^2 e\sigma^2_{\mathrm{SSM},R_\mu} + 3\eta^2 e^2\left(\|\nabla f(W^r)\|_F^2 + 4G^2_{\mathrm{SSM},R_\mu}\mathcal{H}_{\mathrm{TV}}\right)$$
$$+ 3\eta^2 e L^2_{\mathrm{prox},R_\mu}\sum_{j=0}^{e-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\left\|W_k^{r,j} - W^r\right\|_F^2\right]. \tag{200}$$

Averaging (200) over $e = 0, \ldots, E-1$, where we utilize $\frac{1}{E}\sum_{e=0}^{E-1} e = \frac{E-1}{2}$ and $\frac{1}{E}\sum_{e=0}^{E-1} e^2 \leq E(E-1)$, yields:

$$\frac{1}{E}\sum_{e=1}^{E-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\|W_k^{r,e} - W^r\|_F^2\right] \leq \frac{3}{2}\eta^2(E-1)\sigma^2_{\mathrm{SSM},R_\mu}$$
$$+ 3\eta^2 E(E-1)\left(\|\nabla f(W^r)\|_F^2 + 4G^2_{\mathrm{SSM},R_\mu}\mathcal{H}_{\mathrm{TV}}\right)$$
$$+ 3\eta^2 E(E-1)L^2_{\mathrm{prox},R_\mu}\left[\frac{1}{E}\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\|W_k^{r,e} - W^r\|_F^2\right]\right], \tag{201}$$

where for the last term, we used:

$$\frac{1}{E}\sum_{e=0}^{E-1} e\sum_{j=0}^{e-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\left\|W_k^{r,j} - W^r\right\|_F^2\right] \leq E(E-1)\left[\frac{1}{E}\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\|W_k^{r,e} - W^r\|_F^2\right]\right] \tag{202}$$

Rearranging the terms in (201) and dividing out by $1 - 3\eta^2 E(E-1)L^2_{\mathrm{prox},R_\mu}$, yields:

$$\frac{1}{E}\sum_{e=1}^{E-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\|W_k^{r,e} - W^r\|_F^2\right] \leq \frac{1}{1 - 3\eta^2 E(E-1)L^2_{\mathrm{prox},R_\mu}}\Bigg(\frac{3}{2}\eta^2(E-1)\sigma^2_{\mathrm{SSM},R_\mu}$$
$$+ 3\eta^2 E(E-1)\left(\|\nabla f(W^r)\|_F^2 + 4G^2_{\mathrm{SSM},R_\mu}\mathcal{H}_{\mathrm{TV}}\right)\Bigg). \tag{203}$$

Next, note that by the step-size condition $12\eta^2 E(E-1)L^2_{\mathrm{prox},R_\mu} \leq 1$, it is true that $3\eta^2 E(E-1)L^2_{\mathrm{prox},R_\mu} \leq \frac{1}{4}$, and therefore $\frac{1}{1-3\eta^2 E(E-1)L^2_{\mathrm{prox},R_\mu}} \leq \frac{4}{3}$. Substituting that result into (203) yields:

$$\frac{1}{E}\sum_{e=1}^{E-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\|W_k^{r,e} - W^r\|_F^2\right] \leq 2\eta^2(E-1)\sigma^2_{\mathrm{SSM},R_\mu}$$
$$+ 4\eta^2 E(E-1)\left(\|\nabla f(W^r)\|_F^2 + 4G^2_{\mathrm{SSM},R_\mu}\mathcal{H}_{\mathrm{TV}}\right). \tag{204}$$

Next, we evaluate $W^{r+1} - W^r$. By the update rule:

$$W^{r+1} - W^r = -\eta\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left[g_{k,\Xi}^{r,e} + \mu\left(W_k^{r,e} - W^r\right)\right]. \tag{205}$$

Since the segment joining W^r and W^{r+1} lies in $\mathcal{N}_{R_\mu}(W^r)$, by the smoothness of f established in Lemma 4.7, it is true that:

$$\begin{aligned}\mathbb{E}_r\left[f(W^{r+1})\right] \leq & f(W^r) \\ & - \eta \left\langle \nabla f\left(W^r\right), \mathbb{E}_r\left[\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(\nabla \mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right)\right)\right]\right\rangle_F \\ & + \frac{\eta^2 L_{\mathrm{SSM},R_\mu}}{2}\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(g_{k,\Xi}^{r,e} + \mu\left(W_k^{r,e} - W^r\right)\right)\right\|_F^2\right]. \end{aligned} \tag{206}$$

Here, the tower property and because of conditional unbiasedness, the stochastic gradients can be replaced by local gradients inside the expected inner product.

Next, note that:

$$\begin{aligned}\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(g_{k,\Xi}^{r,e} + \mu\left(W_k^{r,e} - W^r\right)\right) &= \sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(g_{k,\Xi}^{r,e} + \mu\left(W_k^{r,e} - W^r\right) + \nabla\mathcal{L}_k\left(W_k^{r,e}\right) - \nabla\mathcal{L}_k\left(W_k^{r,e}\right)\right) \\ &= \sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(\nabla\mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right)\right) \\ &\quad + \sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(g_{k,\Xi}^{r,e} - \nabla\mathcal{L}_k\left(W_k^{r,e}\right)\right). \end{aligned} \tag{207}$$

When taking norms, for a sum of two matrices, the squared norm is at most twice the sum of their squared norms. As such, it follows that:

$$\begin{aligned}\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(g_{k,\Xi}^{r,e} + \mu\left(W_k^{r,e} - W^r\right)\right)\right\|_F^2\right] &\leq 2\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(g_{k,\Xi}^{r,e} - \nabla\mathcal{L}_k\left(W_k^{r,e}\right)\right)\right\|_F^2\right] \\ &\quad + 2\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(\nabla\mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right)\right)\right\|_F^2\right] \end{aligned} \tag{208}$$

Next, the centered gradient-noise differences have conditional mean zero given the complete history before each local step. Conditional independence across clients eliminates the same-step cross terms, while conditional unbiasedness and the tower property eliminate the cross terms between different local steps. Applying Lemma B.3 therefore gives

$$\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(g_{k,\Xi}^{r,e} - \nabla\mathcal{L}_k\left(W_k^{r,e}\right)\right)\right\|_F^2\right] \leq \sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k^2\sigma_{\mathrm{SSM},R_\mu}^2 = E\sigma_{\mathrm{SSM},R_\mu}^2\sum_{k=1}^{K} p_k^2, \tag{209}$$

and as a result:

$$\begin{aligned}\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(g_{k,\Xi}^{r,e} + \mu\left(W_k^{r,e} - W^r\right)\right)\right\|_F^2\right] &\leq 2E\sigma_{\mathrm{SSM},R_\mu}^2\sum_{k=1}^{K} p_k^2 \\ &\quad + 2\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(\nabla\mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right)\right)\right\|_F^2\right] \end{aligned} \tag{210}$$

When it comes to the inner-product term, we note that, since $f(W^r) = \sum_{k=1}^{K} p_k\mathcal{L}_k(W^r)$, it is true that:

$$E\nabla f(W^r) = \sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\nabla\mathcal{L}_k(W^r). \tag{211}$$

Using this result, and the $-2\langle U, V\rangle_F = \|U - V\|_F^2 - \|U\|_F^2 - \|V\|_F^2$ equality, the following holds:

$$-\eta \left\langle \nabla f\left(W^r\right), \mathbb{E}_r\left[\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(\nabla \mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right)\right)\right]\right\rangle_F \tag{212}$$

$$= -\frac{\eta E}{2}\left\|\nabla f(W^r)\right\|_F^2$$

$$- \frac{\eta}{2E}\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(\nabla \mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right)\right)\right\|_F^2\right]$$

$$+ \frac{\eta}{2E}\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(\nabla \mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right) - \nabla \mathcal{L}_k\left(W^r\right)\right)\right\|_F^2\right], \tag{213}$$

where the E-terms appeared from evaluating the inner-product with $E\nabla f(W^r)$ and dividing out by E.

Then, the positive term is bounded as:

$$\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(\nabla \mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right) - \nabla \mathcal{L}_k\left(W^r\right)\right)\right\|_F^2\right]$$

$$\leq E\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\left\|\nabla \mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right) - \nabla \mathcal{L}_k\left(W^r\right)\right\|_F^2\right]$$

$$\leq EL_{\text{prox},R_\mu}^2 \sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\left\|W_k^{r,e} - W^r\right\|_F^2\right] \tag{214}$$

where the first inequality is true by Jensen's inequality with weights p_k/E, and the second inequality follows from (191).

Substituting these results back into (206), we obtain:

$$\mathbb{E}_r\left[f\left(W^{r+1}\right)\right] \leq f\left(W^r\right) - \frac{\eta E}{2}\left\|\nabla f(W^r)\right\|_F^2$$

$$+ \left(\eta^2 L_{\text{SSM},R_\mu} - \frac{\eta}{2E}\right)\mathbb{E}_r\left[\left\|\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k\left(\nabla \mathcal{L}_k\left(W_k^{r,e}\right) + \mu\left(W_k^{r,e} - W^r\right)\right)\right\|_F^2\right]$$

$$+ \frac{\eta L_{\text{prox},R_\mu}^2}{2}\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\left\|W_k^{r,e} - W^r\right\|_F^2\right]$$

$$+ \eta^2 EL_{\text{SSM},R_\mu}\sigma_{\text{SSM},R_\mu}^2 \sum_{k=1}^{K} p_k^2. \tag{215}$$

The step-size condition $\eta E L_{\text{SSM},R_\mu} \leq 1/2$ implies that $\eta^2 L_{\text{SSM},R_\mu} - \frac{\eta}{2E} \leq 0$, and therefore the term multiplied by this factor is never positive. As such, the following holds:

$$\mathbb{E}_r\left[f\left(W^{r+1}\right)\right] \leq f\left(W^r\right) - \frac{\eta E}{2}\left\|\nabla f(W^r)\right\|_F^2$$

$$+ \frac{\eta L_{\text{prox},R_\mu}^2}{2}\sum_{e=0}^{E-1}\sum_{k=1}^{K} p_k \mathbb{E}_r\left[\left\|W_k^{r,e} - W^r\right\|_F^2\right]$$

$$+ \eta^2 EL_{\text{SSM},R_\mu}\sigma_{\text{SSM},R_\mu}^2 \sum_{k=1}^{K} p_k^2. \tag{216}$$

Substituting the result from (204) into this inequality, we get:

$$\mathbb{E}_r\left[f\left(W^{r+1}\right)\right] \leq f\left(W^r\right)$$

$$
\begin{aligned}
&+ \left(2\eta^3 E^2(E-1)L^2_{\text{prox},R_\mu} - \frac{\eta E}{2}\right)\|\nabla f(W^r)\|_F^2 \\
&+ \eta^3 E(E-1)L^2_{\text{prox},R_\mu}\sigma^2_{\text{SSM},R_\mu} \\
&+ 8\eta^3 E^2(E-1)L^2_{\text{prox},R_\mu}G^2_{\text{SSM},R_\mu}\mathcal{H}_{\text{TV}} \\
&+ \eta^2 E L_{\text{SSM},R_\mu}\sigma^2_{\text{SSM},R_\mu}\sum_{k=1}^{K} p_k^2. \qquad (217)
\end{aligned}
$$

Next, by the second step size condition $12\eta^2 L^2_{\text{prox},R_\mu}E(E-1) \leq 1$ it follows that:

$$
\begin{aligned}
2\eta^3 E^2(E-1)L^2_{\text{prox},R_\mu} - \frac{\eta E}{2} &= \eta E\left(2\eta^2 E(E-1)L^2_{\text{prox},R_\mu} - \frac{1}{2}\right) \\
&\leq \eta E\left(\frac{1}{6} - \frac{1}{2}\right) = -\frac{\eta E}{3}. \qquad (218)
\end{aligned}
$$

Substituting in (217), we get:

$$
\begin{aligned}
\mathbb{E}_r\left[f\left(W^{r+1}\right)\right] \leq & f\left(W^r\right) \\
&- \frac{\eta E}{3}\|\nabla f(W^r)\|_F^2 \\
&+ \eta^3 E(E-1)L^2_{\text{prox},R_\mu}\sigma^2_{\text{SSM},R_\mu} \\
&+ 8\eta^3 E^2(E-1)L^2_{\text{prox},R_\mu}G^2_{\text{SSM},R_\mu}\mathcal{H}_{\text{TV}} \\
&+ \eta^2 E L_{\text{SSM},R_\mu}\sigma^2_{\text{SSM},R_\mu}\sum_{k=1}^{K} p_k^2. \qquad (219)
\end{aligned}
$$

Taking total expectations in (219), rearranging, and summing over $r = 0, \ldots, T-1$ makes the objective differences telescope. Using $\mathbb{E}[f(W^T)] \geq f^*$ and dividing by $\eta ET/3$, we obtain:

$$
\begin{aligned}
\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\|\nabla f(W^r)\|_F^2 \leq & \frac{3\left(f(W^0) - f^*\right)}{\eta ET} + 3\eta L_{\text{SSM},R_\mu}\sigma^2_{\text{SSM},R_\mu}\sum_{k=1}^{K} p_k^2 \\
&+ 3\eta^2(E-1)L^2_{\text{prox},R_\mu}\sigma^2_{\text{SSM},R_\mu} + 24\eta^2 E(E-1)L^2_{\text{prox},R_\mu}G^2_{\text{SSM},R_\mu}\mathcal{H}_{\text{TV}}. \qquad (220)
\end{aligned}
$$

Finally, substituting $\sigma^2_{\text{SSM},R_\mu} = G^2_{\text{SSM},R_\mu}/\Xi$ (which follows from Lemma B.3) into (220), gives the stated bound, completing the proof. □

C Multi-Layer Selective SSM proofs

In this section we derive the corresponding Gradient, FedAvg, and FedProx convergence bounds for a Selective SSM with of an arbitrary finite depth Q, extending our derivations in Sections A, and B to a multi-layer selective State Space Model (SSM) case.

C.1 Multi-Layer Selective SSM Gradient Bound Proof

Similar to the single-layer setting, we first derive an upper bound on the gradient norm of the loss function for an arbitrary sequence of length S processed by a selective state space model (SSM) composed of Q sequential selective SSM layers. The analysis once again utilizes definitions from Section 3 and an input sequence $x = (x_1, \ldots, x_S)$ where $x_t \in \mathbb{R}^{D\times 1}$.

The multi-layer selective SSM consists of Q layers arranged sequentially. Each layer $q \in \{1, \ldots, Q\}$ is parameterized by the learnable parameter tuple

$$
W^{(q)} = \left(W_\Delta^{(q)}, W_B^{(q)}, W_C^{(q)}, A^{(q)}\right), \qquad (221)
$$

where $W_\Delta^{(q)} \in \mathbb{R}^{D\times D}, W_B^{(q)} \in \mathbb{R}^{N\times D}, W_C^{(q)} \in \mathbb{R}^{N\times D}, A^{(q)} \in \mathbb{R}^{N\times 1}$. Note that we therefore consider a multi-layer SSM where all of its Q layers have the same input/output dimension D and state dimension N.

The full set of model parameters is then given by:

$$W = \left(W^{(1)}, \ldots, W^{(Q)}\right). \tag{222}$$

In this multi-layer formulation, we denote the input to layer q at time step t by $x_t^{(q)}$, where $x_t^{(1)} = x_t$ and

$$x_t^{(q+1)} = y_t^{(q)} \tag{223}$$

with $y_t^{(q)}$ denoting the output of layer q.

Given this formulation, each layer q of the multi-layer selective SSM operates according to the following dynamics:

$$\Delta_t^{(q)} = \text{Softplus}\left(W_\Delta^{(q)} x_t^{(q)}\right), \tag{224}$$

$$\bar{A}_t^{(q)} = \exp\left(A^{(q)}\left(\Delta_t^{(q)}\right)^\top\right), \tag{225}$$

$$\bar{B}_t^{(q)} = \left(W_B^{(q)} x_t^{(q)}\right)\left(\text{Softplus}\left(W_\Delta^{(q)} x_t^{(q)}\right)\right)^\top, \tag{226}$$

$$h_t^{(q)} = \bar{A}_t^{(q)} \odot h_{t-1}^{(q)} + \bar{B}_t^{(q)} \text{diag}\left(x_t^{(q)}\right), \tag{227}$$

$$y_t^{(q)} = \left(h_t^{(q)}\right)^\top W_C^{(q)} x_t^{(q)} = \left(h_t^{(q)}\right)^\top C_t^{(q)} \tag{228}$$

where the hidden state of each layer q is initialized as $h_0^{(q)} = \mathbf{0}_{N\times D}$, just as in the single-layer setting.

Finally, the empirical loss for processing a sequence of length S for such defined multi-layer selective SSM is:

$$\mathcal{L}^{S,(Q)}(W;\xi) = \frac{1}{S}\sum_{t=1}^{S} \ell_t\left(y_t^{(Q)}\right) = \frac{1}{S}\sum_{t=1}^{S} \ell_t\left(x_t^{(Q+1)}\right). \tag{229}$$

Here, $\xi = (x, y^*)$ is an admissible input-target pair of length $S \geq 1$, and $\ell_t\left(y_t^{(Q)}\right)$ abbreviates $\ell_t(y_t^{(Q)}(W;x), y_t^*)$. For fixed ξ, we write $\mathcal{L}^{S,(Q)}(W)$ in place of $\mathcal{L}^{S,(Q)}(W;\xi)$.

In this multi-layer setting, the full gradient is the block tuple:

$$\nabla_W \mathcal{L}^{S,(Q)}(W) = \left(\nabla_{W^{(1)}}\mathcal{L}^{S,(Q)}(W), \ldots, \nabla_{W^{(Q)}}\mathcal{L}^{S,(Q)}(W)\right). \tag{230}$$

Similar to the single-layer proof, in the multi-layer formulation, gradients with respect to the full parameter tuple $W = \left(W^{(1)}, \ldots, W^{(Q)}\right)$ are understood as block gradients across layers. As such, for any function f,

$$\|\nabla_W f\|_F^2 := \sum_{q=1}^{Q} \|\nabla_{W^{(q)}} f\|_F^2, \tag{231}$$

where, for each layer q, the gradient is further decomposed into parameter blocks as:

$$\|\nabla_{W^{(q)}} f\|_F^2 := \left\|\nabla_{W_\Delta^{(q)}} f\right\|_F^2 + \left\|\nabla_{W_B^{(q)}} f\right\|_F^2 + \left\|\nabla_{W_C^{(q)}} f\right\|_F^2 + \|\nabla_{A^{(q)}} f\|_F^2. \tag{232}$$

As in the single-layer proof, parameter derivatives with respect to $W^{(q)}$ are understood as flattened block Jacobians, with rows representing the output coordinates and columns corresponding to the scalar parameters of layer q. We use the same vectorization and scalar-parameter-slice conventions for products and transposes.

For this multi-layer formulation, we introduce the following new assumption and definition, that extend the single-layer analysis assumptions to the multi-layer setting:

Assumption C.1 (Bounded Layer Inputs)**.** For every layer $q \in \{1, \dots, Q\}$ of a multi-layer selective SSM of depth Q parameterized by a fixed set of weights $W = \left(W^{(1)}, \dots, W^{(Q)}\right)$, there exists a constant $L_{x,q} > 0$ such that:

$$\left\| x_t^{(q)}\left(W; \xi\right) \right\|_2 \leq L_{x,q} \; \forall t \in \{1, \dots, S\}. \tag{233}$$

This holds for every finite integer length $S \geq 1$, every admissible input-target pair $\xi \in \mathcal{Z}$, and every token index $t \in \{1, \dots, S\}$, with finite constants independent of S, ξ, and t.

Definition C.2 (Multi-layer SSM Stability)**.** For each layer $q \in \{1, \dots, Q\}$, let $\gamma_q > 0$ and $\delta_q > 0$. We define $\mathcal{S}^{(Q)}_{\boldsymbol{\gamma},\boldsymbol{\delta}}$ as follows:

$$\begin{aligned} \mathcal{S}^{(Q)}_{\boldsymbol{\gamma},\boldsymbol{\delta}} := \Bigg\{ W \Bigg| & A_n^{(q)} \leq -\gamma_q, \left[\text{Softplus}\left(W_\Delta^{(q)} x_t^{(q)}\left(W; \xi\right)\right)\right]_d \geq \delta_q \\ & \forall q \in \{1, \dots, Q\}, \forall n \in \{1, \dots, N\}, \forall d \in \{1, \dots, D\}, \forall t \in \{1, \dots, S\}, \\ & \text{finite integer } S \geq 1, \text{and every admissible length} - S \text{ pair } \xi \in \mathcal{Z} \Bigg\}, \end{aligned} \tag{234}$$

where $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_Q)$ and $\boldsymbol{\delta} = (\delta_1, \dots, \delta_Q)$

We then say that the full parameter tuple W is stable with these margins if $W \in \mathcal{S}^{(Q)}_{\boldsymbol{\gamma},\boldsymbol{\delta}}$.

For each layer q, we further define:

$$\rho_q := \exp(-\gamma_q \delta_q) < 1. \tag{235}$$

As a result of Definition C.2, for a layer q, it follows that:

$$\left\| \bar{A}_t^{(q)} \right\|_{\max} \leq \rho_q. \tag{236}$$

Given such setup, we state the following theorem:

Theorem C.3 (Multi-Layer SSM Gradient Bound)**.** *Let $W = \left(W^{(1)}, \dots, W^{(Q)}\right)$ be a full parameter tuple satisfying Assumptions 4.2 and C.1, with $W \in \mathcal{S}^{(Q)}_{\boldsymbol{\gamma},\boldsymbol{\delta}}$, that parametrizes a multi-layer selective SSM model of finite integer depth $Q \geq 1$. Then, for such W, for every finite integer $S \geq 1$ and every admissible input-target pair $\xi \in \mathcal{Z}$ of length S, the following bound holds:*

$$\left\| \nabla_W \mathcal{L}^{S,(Q)}(W; \xi) \right\|_F \leq L_\ell \left(\sum_{q=1}^{Q} L_{x,q}^2 \left(H_q + \left\| W_C^{(q)} \right\|_2 J_q \right)^2 \prod_{q'=q+1}^{Q} \kappa_{q'}^2 \right)^{1/2} := \mathcal{G}^{(Q)}_{\textit{SSM}}(W) \tag{237}$$

with

$$K_{\Delta,q} := \left\| W_\Delta^{(q)} \right\|_2 L_{x,q} + \sqrt{D} \log(2), \quad K_{B,q} := \left\| W_B^{(q)} \right\|_2 L_{x,q}^2, \tag{238}$$

$$K_{W,q} := NDK_{B,q}K_{\Delta,q} \left(K_{\Delta,q} + \sqrt{D} \left\| A^{(q)} \right\|_2 L_{x,q} \right) \left(\frac{\rho_q}{1 - \rho_q} \right) + L_{x,q}^2 \sqrt{N} K_{\Delta,q} + L_{x,q} \sqrt{D} K_{B,q}, \tag{239}$$

$$H_q := \frac{K_{B,q}K_{\Delta,q}\sqrt{ND}}{1 - \rho_q}, \quad J_q := \frac{K_{W,q}}{1 - \rho_q}, \quad \kappa_q := \left\| W_C^{(q)} \right\|_2 \left(H_q + \frac{L_{x,q}\Lambda_{x,q}}{1 - \rho_q} \right), \tag{240}$$

$$\Lambda_{x,q} := \rho_q \left\| A^{(q)} \right\|_2 \left\| W_\Delta^{(q)} \right\|_2 H_q + \left\| W_B^{(q)} \right\|_2 L_{x,q} \left(2K_{\Delta,q} + L_{x,q} \left\| W_\Delta^{(q)} \right\|_2 \right). \tag{241}$$

Proof of Theorem C.3: Fix an arbitrary full parameter tuple W satisfying Assumptions 4.2 and C.1 such that $W \in \mathcal{S}^{(Q)}_{\boldsymbol{\gamma},\boldsymbol{\delta}}$. Let $\xi \in \mathcal{Z}$ be some fixed admissible sequence pair of length S ($S \geq 1$ is some fixed finite integer).

By the definition of a gradient with respect to a tuple in equation (231), it follows that:

$$\left\| \nabla_W \mathcal{L}^{S,(Q)}(W) \right\|_F = \left(\sum_{q=1}^{Q} \left\| \nabla_{W^{(q)}} \mathcal{L}^{S,(Q)}(W) \right\|_F^2 \right)^{1/2} \tag{242}$$

As a result, to derive a bound on the gradient norm with respect to the full model parameter tuple W, it is sufficient to derive the bound on:

$$\left\|\nabla_{W^{(q)}}\mathcal{L}^{S,(Q)}(W)\right\|_F^2 \tag{243}$$

for an arbitrary layer $q \in \{1,\dots,Q\}$, and combine the resulting layer-wise bounds.

This gradient can be expanded as:

$$\nabla_{W^{(q)}}\mathcal{L}^{S,(Q)}(W) = \nabla_{W^{(q)}}\left(\frac{1}{S}\sum_{t=1}^{S}\ell_t\left(y_t^{(Q)}\right)\right) = \frac{1}{S}\sum_{t=1}^{S}\nabla_{W^{(q)}}\ell_t\left(y_t^{(Q)}\right) \tag{244}$$

Applying the chain rule again, we get:

$$\operatorname{vec}\left(\nabla_{W^{(q)}}\mathcal{L}^{S,(Q)}(W)\right) = \frac{1}{S}\sum_{t=1}^{S}\left(\frac{\partial y_t^{(Q)}}{\partial W^{(q)}}\right)^{\top}\nabla_y\ell_t\left(y_t^{(Q)}\right). \tag{245}$$

For $Q \geq 2$, we first consider the case $q = Q-1$ and focus on $\partial y_t^{(Q)}/\partial W^{(q)}$, since $\nabla_y\ell_t\left(y_t^{(Q)}\right)$ is independent of q. By the product rule and using the SSM dynamics, it follows that:

$$\frac{\partial y_t^{(Q)}}{\partial W^{(Q-1)}} = \frac{\partial\left(h_t^{(Q)}\right)^{\top}}{\partial W^{(Q-1)}}C_t^{(Q)} + \left(h_t^{(Q)}\right)^{\top}\frac{\partial C_t^{(Q)}}{\partial W^{(Q-1)}} \tag{246}$$

Next, we evaluate (246) coordinate-wise, as this avoids any ambiguity associated with calculating the matrix-valued derivatives with respect to the parameter tuples. Let $w^{(Q-1)}$ denote an arbitrary scalar coordinate of $W^{(Q-1)}$, and let $d \in \{1,\dots,D\}$ denote an arbitrary coordinate of $y_t^{(Q)}$. Then:

$$y_{t,d}^{(Q)} = \sum_{n=1}^{N}h_{t,n,d}^{(Q)}C_{t,n}^{(Q)}, \tag{247}$$

and by the product rule:

$$\frac{\partial y_{t,d}^{(Q)}}{\partial w^{(Q-1)}} = \sum_{n=1}^{N}\left[C_{t,n}^{(Q)}\frac{\partial h_{t,n,d}^{(Q)}}{\partial w^{(Q-1)}} + h_{t,n,d}^{(Q)}\frac{\partial C_{t,n}^{(Q)}}{\partial w^{(Q-1)}}\right] \tag{248}$$

Next, since the weight $w^{(Q-1)}$ is not a parameter of layer Q, it influences $h_t^{(Q)}$ at any coordinate only through the inputs to layer Q (i.e. $h_t^{(Q)}$ is a function of all $x_1^{(Q)},\dots,x_j^{(Q)},\dots,x_t^{(Q)}$, which depend on the parameters of the previous layer through $x_j^{(Q)} = y_j^{(Q-1)}$). As a result:

$$\frac{\partial h_{t,n,d}^{(Q)}}{\partial w^{(Q-1)}} = \sum_{j=1}^{t}\sum_{d'=1}^{D}\frac{\partial h_{t,n,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}}\frac{\partial x_{j,d'}^{(Q)}}{\partial w^{(Q-1)}}. \tag{249}$$

Furthermore, since $C_{t,n}^{(Q)} = \sum_{d'=1}^{D}\left(W_C^{(Q)}\right)_{n,d'}x_{t,d'}^{(Q)}$, and $W_C^{(Q)}$ is independent of any parameter of $W^{(Q-1)}$, it follows that:

$$\frac{\partial C_{t,n}^{(Q)}}{\partial w^{(Q-1)}} = \sum_{d'=1}^{D}\left(W_C^{(Q)}\right)_{n,d'}\frac{\partial x_{t,d'}^{(Q)}}{\partial w^{(Q-1)}} \tag{250}$$

Substituting these expressions into (248) results in:

$$\frac{\partial y_{t,d}^{(Q)}}{\partial w^{(Q-1)}} = \sum_{n=1}^{N}\left[C_{t,n}^{(Q)}\sum_{j=1}^{t}\sum_{d'=1}^{D}\frac{\partial h_{t,n,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}}\frac{\partial x_{j,d'}^{(Q)}}{\partial w^{(Q-1)}} + h_{t,n,d}^{(Q)}\sum_{d'=1}^{D}\left(W_C^{(Q)}\right)_{n,d'}\frac{\partial x_{t,d'}^{(Q)}}{\partial w^{(Q-1)}}\right]$$

$$= \sum_{n=1}^{N} \left[C_{t,n}^{(Q)} \sum_{j=1}^{t} \sum_{d'=1}^{D} \frac{\partial h_{t,n,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} \frac{\partial x_{j,d'}^{(Q)}}{\partial w^{(Q-1)}} + h_{t,n,d}^{(Q)} \sum_{j=1}^{t} \sum_{d'=1}^{D} \mathbb{1}_{j=t} \left(W_C^{(Q)} \right)_{n,d'} \frac{\partial x_{j,d'}^{(Q)}}{\partial w^{(Q-1)}} \right]$$

$$= \sum_{j=1}^{t} \sum_{d'=1}^{D} \left[\sum_{n=1}^{N} \left(C_{t,n}^{(Q)} \frac{\partial h_{t,n,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} + \mathbb{1}_{j=t} h_{t,n,d}^{(Q)} \left(W_C^{(Q)} \right)_{n,d'} \right) \right] \frac{\partial x_{j,d'}^{(Q)}}{\partial w^{(Q-1)}} \tag{251}$$

where $\mathbb{1}_{j=t}$ is 1 if $j = t$ and 0 otherwise.

Next, we note that:

$$\frac{\partial y_{t,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} = \frac{\partial \left(\sum_{n=1}^{N} h_{t,n,d}^{(Q)} C_{t,n}^{(Q)} \right)}{\partial x_{j,d'}^{(Q)}} = \sum_{n=1}^{N} \frac{\partial \left(h_{t,n,d}^{(Q)} C_{t,n}^{(Q)} \right)}{\partial x_{j,d'}^{(Q)}} = \sum_{n=1}^{N} \left[C_{t,n}^{(Q)} \frac{\partial h_{t,n,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} + h_{t,n,d}^{(Q)} \frac{\partial C_{t,n}^{(Q)}}{\partial x_{j,d'}^{(Q)}} \right]$$

$$= \sum_{n=1}^{N} \left[C_{t,n}^{(Q)} \frac{\partial h_{t,n,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} + h_{t,n,d}^{(Q)} \frac{\partial \left(\sum_{d''=1}^{D} \left(W_C^{(Q)} \right)_{n,d''} x_{t,d''}^{(Q)} \right)}{\partial x_{j,d'}^{(Q)}} \right]$$

$$= \sum_{n=1}^{N} \left(C_{t,n}^{(Q)} \frac{\partial h_{t,n,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} + \mathbb{1}_{j=t} h_{t,n,d}^{(Q)} \left(W_C^{(Q)} \right)_{n,d'} \right) \tag{252}$$

where we use the fact that if $j \neq t$,

$$\frac{\partial \left(\sum_{d''=1}^{D} \left(W_C^{(Q)} \right)_{n,d''} x_{t,d''}^{(Q)} \right)}{\partial x_{j,d'}^{(Q)}} = 0 \tag{253}$$

and if $j = t$,

$$\frac{\partial \left(\sum_{d''=1}^{D} \left(W_C^{(Q)} \right)_{n,d''} x_{j,d''}^{(Q)} \right)}{\partial x_{j,d'}^{(Q)}} = \left(W_C^{(Q)} \right)_{n,d'} \tag{254}$$

As a result, it follows that:

$$\frac{\partial y_{t,d}^{(Q)}}{\partial w^{(Q-1)}} = \sum_{j=1}^{t} \sum_{d'=1}^{D} \frac{\partial y_{t,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} \frac{\partial x_{j,d'}^{(Q)}}{\partial w^{(Q-1)}} \tag{255}$$

and finally, since $x_{j,d'}^{(Q)} = y_{j,d'}^{(Q-1)}$,

$$\frac{\partial y_{t,d}^{(Q)}}{\partial w^{(Q-1)}} = \sum_{j=1}^{t} \sum_{d'=1}^{D} \frac{\partial y_{t,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} \frac{\partial y_{j,d'}^{(Q-1)}}{\partial w^{(Q-1)}} \tag{256}$$

For $Q \geq 3$, the same coordinate-wise chain rule argument applies to $w^{(Q-2)}$. That is, since $w^{(Q-2)}$ is not a parameter of layer Q, it influences $y_t^{(Q)}$ only through the inputs sequence $x_1^{(Q)}, \ldots, x_t^{(Q)}$. As such, it follows that:

$$\frac{\partial y_{t,d}^{(Q)}}{\partial w^{(Q-2)}} = \sum_{j=1}^{t} \sum_{d'=1}^{D} \frac{\partial y_{t,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} \frac{\partial y_{j,d'}^{(Q-1)}}{\partial w^{(Q-2)}} \tag{257}$$

Furthermore, using the same argument applied to layer $Q - 1$, we can say that:

$$\frac{\partial y_{j,d'}^{(Q-1)}}{\partial w^{(Q-2)}} = \sum_{j'=1}^{j} \sum_{d''=1}^{D} \frac{\partial y_{j,d'}^{(Q-1)}}{\partial x_{j',d''}^{(Q-1)}} \frac{\partial y_{j',d''}^{(Q-2)}}{\partial w^{(Q-2)}}. \tag{258}$$

Substituting this result into (257), we obtain:

$$\frac{\partial y_{t,d}^{(Q)}}{\partial w^{(Q-2)}} = \sum_{j=1}^{t}\sum_{d'=1}^{D} \frac{\partial y_{t,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} \sum_{j'=1}^{j}\sum_{d''=1}^{D} \frac{\partial y_{j,d'}^{(Q-1)}}{\partial x_{j',d''}^{(Q-1)}} \frac{\partial y_{j',d''}^{(Q-2)}}{\partial w^{(Q-2)}} = \sum_{j=1}^{t}\sum_{j'=1}^{j}\sum_{d'=1}^{D}\sum_{d''=1}^{D} \frac{\partial y_{t,d}^{(Q)}}{\partial x_{j,d'}^{(Q)}} \frac{\partial y_{j,d'}^{(Q-1)}}{\partial x_{j',d''}^{(Q-1)}} \frac{\partial y_{j',d''}^{(Q-2)}}{\partial w^{(Q-2)}} \tag{259}$$

This result means that, compared to the case of $w^{(Q-1)}$, the derivative with respect to $w^{(Q-2)}$ contains one additional input-output derivative, which corresponds to its propagation through layer $Q-1$. Applying this argument recursively, we derive the final derivative of the d-th coordinate of the final output with respect to some parameter $w^{(q)}$ of the layer $q \in \{1, \dots, Q-1\}$ as:

$$\frac{\partial y_{t,d}^{(Q)}}{\partial w^{(q)}} = \sum_{j_{Q-1}=1}^{j_Q} \cdots \sum_{j_q=1}^{j_{q+1}} \sum_{d_{Q-1}=1}^{D} \cdots \sum_{d_q=1}^{D} \left[\prod_{q'=q+1}^{Q} \frac{\partial y_{j_{q'},d_{q'}}^{(q')}}{\partial x_{j_{q'-1},d_{q'-1}}^{(q')}} \right] \frac{\partial y_{j_q,d_q}^{(q)}}{\partial w^{(q)}}, \tag{260}$$

where we also have $j_Q = t$, $d_Q = d$, and $j_q \le j_{q+1} \le \cdots \le j_{Q-1} \le j_Q = t$.

We next collect the coordinate derivatives in (260) into matrices, which transforms sums over feature coordinates into matrix products, while the sums over time indices remain explicit.

For each layer q, define the input-output Jacobian matrix as:

$$\mathbf{K}_{t,j}^{(q)} := \frac{\partial y_t^{(q)}}{\partial x_j^{(q)}} = \begin{pmatrix} \frac{\partial y_{t,1}^{(q)}}{\partial x_{j,1}^{(q)}} & \cdots & \frac{\partial y_{t,1}^{(q)}}{\partial x_{j,D}^{(q)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_{t,D}^{(q)}}{\partial x_{j,1}^{(q)}} & \cdots & \frac{\partial y_{t,D}^{(q)}}{\partial x_{j,D}^{(q)}} \end{pmatrix} \in \mathbb{R}^{D\times D}. \tag{261}$$

The Jacobian is computed with the parameters of layer q and all input tokens other than $x_j^{(q)}$ held fixed, and evaluated at the input sequence received by that layer. Additionally, by causality, $\mathbf{K}_{t,j}^{(q)} = \mathbf{0}_{D\times D}$ for $j > t$.

Then, for any layer $q \in \{1, \dots, Q-1\}$, for any subsequent layer $q' \in \{q+1, \dots, Q\}$, the parameters $W^{(q)}$ affect $y^{(q')}$ only through the input sequence to layer q'. Since $x_j^{(q')} = y_j^{(q'-1)}$, the chain rule can be written in the matrix form as:

$$\frac{\partial y_t^{(q')}}{\partial W^{(q)}} = \sum_{j=1}^{t} \frac{\partial y_t^{(q')}}{\partial x_j^{(q')}} \frac{\partial x_j^{(q')}}{\partial W^{(q)}} = \sum_{j=1}^{t} \mathbf{K}_{t,j}^{(q')} \frac{\partial y_j^{(q'-1)}}{\partial W^{(q)}}. \tag{262}$$

Applying this recursively, we get:

$$\frac{\partial y_t^{(Q)}}{\partial W^{(q)}} = \sum_{j_{Q-1}=1}^{j_Q} \cdots \sum_{j_q=1}^{j_{q+1}} \left[\mathbf{K}_{j_Q,j_{Q-1}}^{(Q)} \times \cdots \times \mathbf{K}_{j_{q+1},j_q}^{(q+1)} \right] \frac{\partial y_{j_q}^{(q)}}{\partial W^{(q)}} \tag{263}$$

for any $q \in \{1, \dots, Q-1\}$, the matrix product contains only $\mathbf{K}_{t,j_{Q-1}}^{(Q)}$ for the case $q = Q-1$, the $\times$ operator denotes matrix multiplication, and $j_Q = t$.

Note that equation (263) provides an explicit expression for the parameter Jacobian $\frac{\partial y_t^{(Q)}}{\partial W^{(q)}}$ appearing in (245), in terms of the own-layer parameter Jacobians $\frac{\partial y_{j_q}^{(q)}}{\partial W^{(q)}}$ and the input-output Jacobians of the subsequent layers.

We next use this decomposition to bound the norm of the parameter Jacobian. We use the Frobenius norm $\|\cdot\|_F$ for the flattened parameter Jacobians and the spectral norm $\|\cdot\|_2$ (where for $M \in \mathbb{R}^{D\times D}$, $\|M\|_2 := \sup_{\substack{v\in\mathbb{R}^D \\ \|v\|_2=1}} \|Mv\|_2$) for the input-output matrices. Applying the triangle inequality and the fact that $\|AB\|_F \le \|A\|_2 \|B\|_F$, it follows that:

$$\left\| \frac{\partial y_t^{(Q)}}{\partial W^{(q)}} \right\|_F = \left\| \sum_{j_{Q-1}=1}^{j_Q} \cdots \sum_{j_q=1}^{j_{q+1}} \left[\mathbf{K}_{j_Q,j_{Q-1}}^{(Q)} \times \cdots \times \mathbf{K}_{j_{q+1},j_q}^{(q+1)} \right] \frac{\partial y_{j_q}^{(q)}}{\partial W^{(q)}} \right\|_F$$

$$\leq \sum_{j_{Q-1}=1}^{j_Q} \cdots \sum_{j_q=1}^{j_{q+1}} \left\| \left[\mathbf{K}^{(Q)}_{j_Q,j_{Q-1}} \times \cdots \times \mathbf{K}^{(q+1)}_{j_{q+1},j_q} \right] \frac{\partial y^{(q)}_{j_q}}{\partial W^{(q)}} \right\|_F$$

$$\leq \sum_{j_{Q-1}=1}^{j_Q} \cdots \sum_{j_q=1}^{j_{q+1}} \left[\prod_{q'=q+1}^{Q} \left\| \mathbf{K}^{(q')}_{j_{q'},j_{q'-1}} \right\|_2 \right] \left\| \frac{\partial y^{(q)}_{j_q}}{\partial W^{(q)}} \right\|_F, \tag{264}$$

where $j_Q = t$ and $q \in \{1, \ldots, Q-1\}$.

Next, we bound the terms of this expression separately. Thanks to Assumption C.1 and Definition C.2, we can apply the single-layer gradient proof argument from Appendix A to each layer q, bounding the term $\left\| \frac{\partial y^{(q)}_{j_q}}{\partial W^{(q)}} \right\|_F$.

First, for each $q \in \{1, \ldots, Q\}$, let:

$$K_{\Delta,q} := \left\| W^{(q)}_{\Delta} \right\|_2 L_{x,q} + \sqrt{D}\log(2), \quad K_{B,q} := \left\| W^{(q)}_{B} \right\|_2 L^2_{x,q}, \tag{265}$$

$$K_{W,q} := NDK_{B,q}K_{\Delta,q}\left(K_{\Delta,q} + \sqrt{D}\left\|A^{(q)}\right\|_2 L_{x,q}\right)\left(\frac{\rho_q}{1-\rho_q}\right) + L^2_{x,q}\sqrt{N}K_{\Delta,q} + L_{x,q}\sqrt{D}K_{B,q}. \tag{266}$$

Given these definitions, we apply the single-layer argument from Appendix A for every layer $q \in \{1, \ldots, Q\}$. Since the input to any layer q depends only on the parameters of the preceding layers, it remains fixed when differentiating with respect to $W^{(q)}$. It therefore follows that:

$$\left\| h^{(q)}_j \right\|_F \leq \frac{K_{B,q}K_{\Delta,q}\sqrt{ND}}{1-\rho_q} := H_q, \tag{267}$$

$$\left\| \frac{\partial h^{(q)}_j}{\partial W^{(q)}} \right\|_F \leq \frac{K_{W,q}}{1-\rho_q} := J_q, \tag{268}$$

for every $q \in \{1, \ldots, Q\}$ and $j \in \{1, \ldots, S\}$. Here, the state Jacobian is flattened as in the single-layer analysis.

Then, by definition of $C^{(q)}_j$, Assumption C.1 gives:

$$\left\| C^{(q)}_j \right\|_2 = \left\| W^{(q)}_C x^{(q)}_j \right\|_2 \leq \left\| W^{(q)}_C \right\|_2 L_{x,q}. \tag{269}$$

Finally, by applying the same product-rule argument to $y^{(q)}_j = \left(h^{(q)}_j\right)^\top C^{(q)}_j$ as in Appendix A, and using (267) and (268), we arrive at:

$$\left\| \frac{\partial y^{(q)}_j}{\partial W^{(q)}} \right\|_F \leq \left\| C^{(q)}_j \right\|_2 \left\| \frac{\partial h^{(q)}_j}{\partial W^{(q)}} \right\|_F + \left\| h^{(q)}_j \right\|_F \left\| x^{(q)}_j \right\|_2$$

$$\leq \left\| W^{(q)}_C \right\|_2 L_{x,q}J_q + H_qL_{x,q} = L_{x,q}\left(H_q + \left\| W^{(q)}_C \right\|_2 J_q\right). \tag{270}$$

This bound holds for every $q \in \{1, \ldots, Q\}$ and $j \in \{1, \ldots, S\}$ uniformly over all admissible input sequences.

Substituting (270) into (264), we get:

$$\left\| \frac{\partial y^{(Q)}_t}{\partial W^{(q)}} \right\|_F \leq L_{x,q}\left(H_q + \left\| W^{(q)}_C \right\|_2 J_q\right)\left(\sum_{j_{Q-1}=1}^{j_Q} \cdots \sum_{j_q=1}^{j_{q+1}} \left[\prod_{q'=q+1}^{Q} \left\| \mathbf{K}^{(q')}_{j_{q'},j_{q'-1}} \right\|_2 \right] \right) \tag{271}$$

Next, we derive a bound on the term:

$$\sum_{j_{Q-1}=1}^{j_Q} \cdots \sum_{j_q=1}^{j_{q+1}} \left[\prod_{q'=q+1}^{Q} \left\| \mathbf{K}^{(q')}_{j_{q'},j_{q'-1}} \right\|_2 \right] \tag{272}$$

To derive the bound, for an arbitrary layer $q' \in \{q+1, \dots, Q\}$, we begin by expressing $\mathbf{K}_{t,j}^{(q')}$ in terms of the layer's hidden-state sensitivity. Then, by definition:

$$\left(\mathbf{K}_{t,j}^{(q')}\right)_{d,d'} = \frac{\partial y_{t,d}^{(q')}}{\partial x_{j,d'}^{(q')}} = \sum_{n=1}^{N} C_{t,n}^{(q')} \frac{\partial h_{t,n,d}^{(q')}}{\partial x_{j,d'}^{(q')}} + \mathbb{1}_{j=t} \sum_{n=1}^{N} h_{t,n,d}^{(q')} \left(W_C^{(q')}\right)_{n,d'}, \tag{273}$$

for $1 \le j \le t \le S$ and $d, d' \in \{1, \dots, D\}$.

Next, by definition of the spectral norm, we know that:

$$\left\|\mathbf{K}_{t,j}^{(q')}\right\|_2 = \sup_{\substack{u \in \mathbb{R}^D \\ \|u\|_2 = 1}} \left\|\mathbf{K}_{t,j}^{(q')} u\right\|_2. \tag{274}$$

As such, we first evaluate the value of $\mathbf{K}_{t,j}^{(q')} u$ for some arbitrary unit vector $u \in \mathbb{R}^D$. Multiplying (273) by $u_{d'}$, summing over d', and collecting the output coordinates, we get:

$$\mathbf{K}_{t,j}^{(q')} u = \left(\sum_{d'=1}^{D} u_{d'} \frac{\partial h_t^{(q')}}{\partial x_{j,d'}^{(q')}}\right)^\top C_t^{(q')} + \mathbb{1}_{j=t} \left(h_t^{(q')}\right)^\top W_C^{(q')} u. \tag{275}$$

Note that, since u is a unit vector, the sum in the first term is, by definition, the directional derivative of the hidden state with respect to $x_j^{(q')}$ in direction u. We denote it as:

$$D_{x_j^{(q')}} h_t^{(q')}[u] := \sum_{d'=1}^{D} u_{d'} \frac{\partial h_t^{(q')}}{\partial x_{j,d'}^{(q')}} \in \mathbb{R}^{N \times D}. \tag{276}$$

As such, it follows that:

$$\mathbf{K}_{t,j}^{(q')} u = \left(D_{x_j^{(q')}} h_t^{(q')}[u]\right)^\top C_t^{(q')} + \mathbb{1}_{j=t} \left(h_t^{(q')}\right)^\top W_C^{(q')} u \tag{277}$$

We then bound both terms of the sum in (277) separately.

We evaluate the term $\left(D_{x_j^{(q')}} h_t^{(q')}[u]\right)^\top C_t^{(q')}$ first. Recall that by (227):

$$h_t^{(q')} = \bar{A}_t^{(q')} \odot h_{t-1}^{(q')} + \bar{B}_t^{(q')} \operatorname{diag}\left(x_t^{(q')}\right).$$

Since, for fixed layer q' weights, $\bar{A}_t^{(q')}$ and $\bar{B}_t^{(q')}$ depend only on the input $x_t^{(q')}$, their derivatives with respect to $x_j^{(q')}$ vanish when $t \neq j$. Furthermore, $D_{x_j^{(q')}} x_t^{(q')}[u] = \mathbb{1}_{j=t} u$. As such, applying these results and the product rule, we get:

$$\begin{aligned} D_{x_j^{(q')}} h_t^{(q')}[u] = \bar{A}_t^{(q')} \odot D_{x_j^{(q')}} h_{t-1}^{(q')}[u] + \mathbb{1}_{j=t} \Bigg[\left(D_{x_t^{(q')}} \bar{A}_t^{(q')}[u]\right) \odot h_{t-1}^{(q')} + \\ \left(D_{x_t^{(q')}} \bar{B}_t^{(q')}[u]\right) \operatorname{diag}\left(x_t^{(q')}\right) + \bar{B}_t^{(q')} \operatorname{diag}(u) \Bigg], \end{aligned} \tag{278}$$

where we also used the fact that $D_{x_j^{(q')}} \operatorname{diag}\left(x_t^{(q')}\right)[u] = \mathbb{1}_{j=t} \operatorname{diag}(u)$.

Next, we consider three cases. When $0 \le t < j$, because the hidden state is initialized to 0 and its evolution is causal, then:

$$D_{x_j^{(q')}} h_t^{(q')}[u] = \mathbf{0}, \qquad \forall t \in \{0, \dots, j-1\}. \tag{279}$$

When, $t = j$, $D_{x_j^{(q')}} h_{t-1}^{(q')}[u] = D_{x_t^{(q')}} h_{t-1}^{(q')}[u] = \mathbf{0}$, so:

$$D_{x_t^{(q')}} h_t^{(q')}[u] = \left(D_{x_t^{(q')}} \bar{A}_t^{(q')}[u]\right) \odot h_{t-1}^{(q')} + \left(D_{x_t^{(q')}} \bar{B}_t^{(q')}[u]\right) \operatorname{diag}\left(x_t^{(q')}\right) + \bar{B}_t^{(q')} \operatorname{diag}(u). \tag{280}$$

Finally, for $t > j$, the recurrence becomes:

$$D_{x_j^{(q')}} h_t^{(q')}[u] = \bar{A}_t^{(q')} \odot D_{x_j^{(q')}} h_{t-1}^{(q')}[u]. \tag{281}$$

Next, by equation (236), which follows from definition C.2, we obtain:

$$\left\| D_{x_j^{(q')}} h_t^{(q')}[u] \right\|_F \leq \rho_{q'} \left\| D_{x_j^{(q')}} h_{t-1}^{(q')}[u] \right\|_F \leq \rho_{q'}^{t-j} \left\| D_{x_j^{(q')}} h_j^{(q')}[u] \right\|_F, \qquad \forall t > j. \tag{282}$$

As a result, we just need to bound $\left\| D_{x_j^{(q')}} h_t^{(q')}[u] \right\|_F$ for the case when $t = j$, and apply the recursive relationship. By the triangle inequality, when $t = j$, it follows that:

$$\begin{aligned} \left\| D_{x_j^{(q')}} h_j^{(q')}[u] \right\|_F \leq & \left\| \left(D_{x_j^{(q')}} \bar{A}_j^{(q')}[u] \right) \odot h_{j-1}^{(q')} \right\|_F \\ & + \left\| \left(D_{x_j^{(q')}} \bar{B}_j^{(q')}[u] \right) \operatorname{diag}\left(x_j^{(q')} \right) \right\|_F + \left\| \bar{B}_j^{(q')} \operatorname{diag}(u) \right\|_F \end{aligned} \tag{283}$$

Then, evaluating it term by term

$$\begin{aligned} D_{x_j^{(q')}} \bar{A}_j^{(q')}[u] &= D_{x_j^{(q')}} \left(\exp\left(A^{(q')} \left(\Delta_j^{(q')} \right)^\top \right) \right)[u] \\ &= \exp\left(A^{(q')} \left(\Delta_j^{(q')} \right)^\top \right) \odot \left[A^{(q')} \left(D_{x_j^{(q')}} \Delta_j^{(q')}[u] \right)^\top \right] \\ &= \bar{A}_j^{(q')} \odot \left[A^{(q')} \left(D_{x_j^{(q')}} \Delta_j^{(q')}[u] \right)^\top \right]. \end{aligned} \tag{284}$$

Using the definition:

$$D_{x_j^{(q')}} \Delta_j^{(q')}[u] = D_{x_j^{(q')}} \left(\operatorname{Softplus}\left(W_\Delta^{(q')} x_j^{(q')} \right) \right)[u] = \operatorname{Softplus}'\left(W_\Delta^{(q')} x_j^{(q')} \right) \odot \left(W_\Delta^{(q')} u \right). \tag{285}$$

Next, since $\operatorname{Softplus}'(z) = \frac{1}{1+e^{-z}} \in (0,1)$, we get:

$$\left\| D_{x_j^{(q')}} \Delta_j^{(q')}[u] \right\|_2 \leq \left\| W_\Delta^{(q')} \right\|_2 \|u\|_2 = \left\| W_\Delta^{(q')} \right\|_2. \tag{286}$$

Then, the fact that $\|ab^\top\|_F = \|a\|_2 \|b\|_2$ (which follows from the definition of the Frobenius norm), the following holds:

$$\begin{aligned} \left\| D_{x_j^{(q')}} \bar{A}_j^{(q')}[u] \right\|_F &\leq \left\| \bar{A}_j^{(q')} \right\|_{\max} \left\| A^{(q')} \left(D_{x_j^{(q')}} \Delta_j^{(q')}[u] \right)^\top \right\|_F \\ &= \left\| \bar{A}_j^{(q')} \right\|_{\max} \left\| A^{(q')} \right\|_2 \left\| D_{x_j^{(q')}} \Delta_j^{(q')}[u] \right\|_2 \leq \rho_{q'} \left\| A^{(q')} \right\|_2 \left\| W_\Delta^{(q')} \right\|_2, \end{aligned} \tag{287}$$

Finally, combining equations (287) and (267) yields:

$$\left\| \left(D_{x_j^{(q')}} \bar{A}_j^{(q')}[u] \right) \odot h_{j-1}^{(q')} \right\|_F \leq \left\| D_{x_j^{(q')}} \bar{A}_j^{(q')}[u] \right\|_F \left\| h_{j-1}^{(q')} \right\|_F \leq \rho_{q'} \left\| A^{(q')} \right\|_2 \left\| W_\Delta^{(q')} \right\|_2 H_{q'}. \tag{288}$$

For the second term, from the definition of $\bar{B}_j^{(q')}$, and using the product rule, we have:

$$D_{x_j^{(q')}} \bar{B}_j^{(q')}[u] = \left(W_B^{(q')} u \right) \left(\Delta_j^{(q')} \right)^\top + \left(W_B^{(q')} x_j^{(q')} \right) \left(D_{x_j^{(q')}} \Delta_j^{(q')}[u] \right)^\top \tag{289}$$

Then, combining the result of (286) with

$$\left\|\Delta_j^{(q')}\right\|_2 \leq \left\|W_\Delta^{(q')} x_j^{(q')}\right\|_2 + \sqrt{D}\log(2) \leq K_{\Delta,q'}, \tag{290}$$

we get:

$$\begin{aligned}\left\|D_{x_j^{(q')}} \bar{B}_j^{(q')}[u]\right\|_F &\leq \left\|W_B^{(q')} u\right\|_2 \left\|\Delta_j^{(q')}\right\|_2 + \left\|W_B^{(q')} x_j^{(q')}\right\|_2 \left\|D_{x_j^{(q')}} \Delta_j^{(q')}[u]\right\|_2 \\ &\leq \left\|W_B^{(q')}\right\|_2 K_{\Delta,q'} + \left\|W_B^{(q')}\right\|_2 L_{x,q'} \left\|W_\Delta^{(q')}\right\|_2 .\end{aligned} \tag{291}$$

Finally, since $\|\mathrm{diag}\,(v)\|_2 = \max_d |v_d| = \|v\|_\infty \leq \|v\|_2$ (and therefore $\left\|\mathrm{diag}\left(x_j^{(q')}\right)\right\|_2 \leq L_{x,q'}$), the second term is bounded as:

$$\begin{aligned}\left\|\left(D_{x_j^{(q')}} \bar{B}_j^{(q')}[u]\right) \mathrm{diag}\left(x_j^{(q')}\right)\right\|_F &\leq \left\|D_{x_j^{(q')}} \bar{B}_j^{(q')}[u]\right\|_F \left\|\mathrm{diag}\left(x_j^{(q')}\right)\right\|_2 \\ &\leq \left\|W_B^{(q')}\right\|_2 L_{x,q'} \left(K_{\Delta,q'} + L_{x,q'} \left\|W_\Delta^{(q')}\right\|_2\right)\end{aligned} \tag{292}$$

For the remaining term, the bound is as follows:

$$\begin{aligned}\left\|\bar{B}_j^{(q')} \mathrm{diag}(u)\right\|_F &\leq \left\|\bar{B}_j^{(q')}\right\|_F \|\mathrm{diag}(u)\|_2 \\ &\leq \left\|W_B^{(q')} x_j^{(q')}\right\|_2 \left\|\Delta_j^{(q')}\right\|_2 \|u\|_2 \\ &\leq \left\|W_B^{(q')}\right\|_2 L_{x,q'} K_{\Delta,q'}\end{aligned} \tag{293}$$

Combining the three bounds, we get:

$$\left\|D_{x_j^{(q')}} h_j^{(q')}[u]\right\|_F \leq \rho_{q'} \left\|A^{(q')}\right\|_2 \left\|W_\Delta^{(q')}\right\|_2 H_{q'} + \left\|W_B^{(q')}\right\|_2 L_{x,q'} \left(2K_{\Delta,q'} + L_{x,q'} \left\|W_\Delta^{(q')}\right\|_2\right). \tag{294}$$

For ease of notation, for each layer q', we define:

$$\Lambda_{x,q'} := \rho_{q'} \left\|A^{(q')}\right\|_2 \left\|W_\Delta^{(q')}\right\|_2 H_{q'} + \left\|W_B^{(q')}\right\|_2 L_{x,q'} \left(2K_{\Delta,q'} + L_{x,q'} \left\|W_\Delta^{(q')}\right\|_2\right). \tag{295}$$

Using this notation, we have the following bound:

$$\left\|D_{x_j^{(q')}} h_j^{(q')}[u]\right\|_F \leq \Lambda_{x,q'}, \qquad \|u\|_2 = 1. \tag{296}$$

Substituting this bound into the recurrence established in (282), we get:

$$\left\|D_{x_j^{(q')}} h_t^{(q')}[u]\right\|_F \leq \rho_{q'}^{t-j} \left\|D_{x_j^{(q')}} h_j^{(q')}[u]\right\|_F \leq \Lambda_{x,q'} \rho_{q'}^{t-j}, \qquad 1 \leq j \leq t \leq S. \tag{297}$$

given $\|u\|_2 = 1$, and $\left\|D_{x_j^{(q')}} h_t^{(q')}[u]\right\|_F = 0$ for $t < j$.

Returning to the expression for $\mathbf{K}_{t,j}^{(q')} u$ in (277), we can bound the first term of the sum using the bound derived in (297) (and the fact that $\left\|Z^\top c\right\|_2 \leq \|Z\|_F \|c\|_2$) as follows:

$$\left\|\left(D_{x_j^{(q')}} h_t^{(q')}[u]\right)^\top C_t^{(q')}\right\|_2 \leq \left\|D_{x_j^{(q')}} h_t^{(q')}[u]\right\|_F \left\|C_t^{(q')}\right\|_2 \leq \left\|W_C^{(q')}\right\|_2 L_{x,q'} \Lambda_{x,q'} \rho_{q'}^{t-j}. \tag{298}$$

For the second term, we can bound this as:

$$\left\|\mathbb{1}_{j=t} \left(h_t^{(q')}\right)^\top W_C^{(q')} u\right\|_2 \leq \mathbb{1}_{j=t} \left\|h_t^{(q')}\right\|_F \left\|W_C^{(q')}\right\|_2 \|u\|_2 = \mathbb{1}_{j=t} \left\|h_t^{(q')}\right\|_F \left\|W_C^{(q')}\right\|_2 \tag{299}$$

Using the triangle inequality, and combining the derived bounds on the terms of the sum in (277), we get:

$$\left\|\mathbf{K}_{t,j}^{(q')}u\right\|_2 \le \left\|W_C^{(q')}\right\|_2 L_{x,q'}\Lambda_{x,q'}\rho_{q'}^{t-j} + \mathbb{1}_{j=t}\left\|W_C^{(q')}\right\|_2 H_{q'}. \tag{300}$$

Since the right hand side of (300) is independent of u, and therefore holds uniformly for all unit vectors u, taking the supremum over all u, we get:

$$\left\|\mathbf{K}_{t,j}^{(q')}\right\|_2 \le \left\|W_C^{(q')}\right\|_2 L_{x,q'}\Lambda_{x,q'}\rho_{q'}^{t-j} + \mathbb{1}_{j=t}\left\|W_C^{(q')}\right\|_2 H_{q'}, \qquad 1 \le j \le t \le S. \tag{301}$$

Next, for every fixed position t, summing $\left\|\mathbf{K}_{t,j}^{(q')}\right\|_2$ over j yields:

$$\begin{aligned}
\sum_{j=1}^{t}\left\|\mathbf{K}_{t,j}^{(q')}\right\|_2 &\le \sum_{j=1}^{t}\left(\left\|W_C^{(q')}\right\|_2 L_{x,q'}\Lambda_{x,q'}\rho_{q'}^{t-j} + \mathbb{1}_{j=t}\left\|W_C^{(q')}\right\|_2 H_{q'}\right)\\
&= \left\|W_C^{(q')}\right\|_2 L_{x,q'}\Lambda_{x,q'}\sum_{j=1}^{t}\rho_{q'}^{t-j} + \left\|W_C^{(q')}\right\|_2 H_{q'}\\
&= \left\|W_C^{(q')}\right\|_2 L_{x,q'}\Lambda_{x,q'}\frac{1-\rho_{q'}^{t}}{1-\rho_{q'}} + \left\|W_C^{(q')}\right\|_2 H_{q'}\\
&\le \left\|W_C^{(q')}\right\|_2\left(H_{q'} + \frac{L_{x,q'}\Lambda_{x,q'}}{1-\rho_{q'}}\right) = \kappa_{q'},
\end{aligned} \tag{302}$$

where, for each layer q', we define the uniform layerwise input-output sensitivity bound

$$\kappa_{q'} := \left\|W_C^{(q')}\right\|_2\left(H_{q'} + \frac{L_{x,q'}\Lambda_{x,q'}}{1-\rho_{q'}}\right) \tag{303}$$

Next, since $\kappa_{q'}$ is independent of t, it follows that:

$$\max_{1\le t\le S}\sum_{j=1}^{t}\left\|\mathbf{K}_{t,j}^{(q')}\right\|_2 \le \kappa_{q'}. \tag{304}$$

Since W satisfies the stated stability and bounded-input conditions, the bound holds uniformly over all admissible input-target sequences and token indices.

Now, returning to (272), consider the two inner-most sums (for $q \le Q-2$):

$$\begin{aligned}
\sum_{j_{q+1}=1}^{j_{q+2}}\sum_{j_q=1}^{j_{q+1}}\left[\prod_{q'=q+1}^{Q}\left\|\mathbf{K}_{j_{q'},j_{q'-1}}^{(q')}\right\|_2\right] &= \sum_{j_{q+1}=1}^{j_{q+2}}\sum_{j_q=1}^{j_{q+1}}\left\|\mathbf{K}_{j_{q+1},j_q}^{(q+1)}\right\|_2\left\|\mathbf{K}_{j_{q+2},j_{q+1}}^{(q+2)}\right\|_2\cdots\left\|\mathbf{K}_{j_Q,j_{Q-1}}^{(Q)}\right\|_2\\
&= \sum_{j_{q+1}=1}^{j_{q+2}}\left[\sum_{j_q=1}^{j_{q+1}}\left\|\mathbf{K}_{j_{q+1},j_q}^{(q+1)}\right\|_2\left\|\mathbf{K}_{j_{q+2},j_{q+1}}^{(q+2)}\right\|_2\cdots\left\|\mathbf{K}_{j_Q,j_{Q-1}}^{(Q)}\right\|_2\right]\\
&= \sum_{j_{q+1}=1}^{j_{q+2}}\left\|\mathbf{K}_{j_{q+2},j_{q+1}}^{(q+2)}\right\|_2\cdots\left\|\mathbf{K}_{j_Q,j_{Q-1}}^{(Q)}\right\|_2\left[\sum_{j_q=1}^{j_{q+1}}\left\|\mathbf{K}_{j_{q+1},j_q}^{(q+1)}\right\|_2\right]\\
&\le \sum_{j_{q+1}=1}^{j_{q+2}}\left\|\mathbf{K}_{j_{q+2},j_{q+1}}^{(q+2)}\right\|_2\cdots\left\|\mathbf{K}_{j_Q,j_{Q-1}}^{(Q)}\right\|_2\kappa_{q+1}\\
&= \kappa_{q+1}\sum_{j_{q+1}=1}^{j_{q+2}}\left\|\mathbf{K}_{j_{q+2},j_{q+1}}^{(q+2)}\right\|_2\cdots\left\|\mathbf{K}_{j_Q,j_{Q-1}}^{(Q)}\right\|_2,
\end{aligned} \tag{305}$$

where the inequality follows from (304). Since κ_{q+1} is independent of all remaining sums, and we can apply the same argument successively to each remaining innermost sum, it follows that:

$$\sum_{j_{Q-1}=1}^{j_Q}\cdots\sum_{j_q=1}^{j_{q+1}}\left[\prod_{q'=q+1}^{Q}\left\|\mathbf{K}_{j_{q'},j_{q'-1}}^{(q')}\right\|_2\right] \le \kappa_{q+1}\kappa_{q+2}\cdots\kappa_Q = \prod_{q'=q+1}^{Q}\kappa_{q'}. \tag{306}$$

for the fixed W and every admissible input-target sequence pair. For $q = Q - 1$, there is only one sum, and holds directly from definition: $\sum_{j_{Q-1}=1}^{t} \left\| \mathbf{K}^{(Q)}_{t,j_{Q-1}} \right\|_2 \leq \kappa_Q$.

Finally, combining the nested-sum bound in (306) with (264) and (270), we obtain:

$$\left\| \frac{\partial y_t^{(Q)}}{\partial W^{(q)}} \right\|_F \leq L_{x,q} \left(H_q + \left\| W_C^{(q)} \right\|_2 J_q \right) \prod_{q'=q+1}^{Q} \kappa_{q'}. \tag{307}$$

This bound holds for every $q \in \{1, \ldots, Q\}$ and $t \in \{1, \ldots, S\}$, where an empty product is interpreted as one. For the $q = Q$ case, the inequality follows directly from the own-layer bound in (270).

We now return to the loss gradient in (245). Under Assumption 4.2, applied at the fixed full parameter tuple W to final-layer predictions uniformly over admissible pairs and token indices, we have:

$$\left\| \nabla_y \ell_t \left(y_t^{(Q)} \right) \right\|_2 \leq L_\ell. \tag{308}$$

Finally, taking the Euclidean norm of the vectorized gradient in (245), which equals its block Frobenius norm, gives:

$$\begin{aligned}
\left\| \nabla_{W^{(q)}} \mathcal{L}^{S,(Q)}(W) \right\|_F &= \left\| \frac{1}{S} \sum_{t=1}^{S} \left(\frac{\partial y_t^{(Q)}}{\partial W^{(q)}} \right)^\top \nabla_y \ell_t \left(y_t^{(Q)} \right) \right\|_2 \\
&\leq \frac{1}{S} \sum_{t=1}^{S} \left\| \nabla_y \ell_t \left(y_t^{(Q)} \right) \right\|_2 \left\| \frac{\partial y_t^{(Q)}}{\partial W^{(q)}} \right\|_F \\
&\leq \frac{1}{S} \sum_{t=1}^{S} L_\ell L_{x,q} \left(H_q + \left\| W_C^{(q)} \right\|_2 J_q \right) \prod_{q'=q+1}^{Q} \kappa_{q'} \\
&= L_\ell L_{x,q} \left(H_q + \left\| W_C^{(q)} \right\|_2 J_q \right) \prod_{q'=q+1}^{Q} \kappa_{q'},
\end{aligned} \tag{309}$$

where we use (308), (307), the triangle inequality, and the fact that $\|M^\top v\|_2 \leq \|M\|_F \|v\|_2$. The bound is independent of the token index t.

Substituting the result of (309) back into (242), we get:

$$\left\| \nabla_W \mathcal{L}^{S,(Q)}(W) \right\|_F \leq L_\ell \left(\sum_{q=1}^{Q} L_{x,q}^2 \left(H_q + \left\| W_C^{(q)} \right\|_2 J_q \right)^2 \prod_{q'=q+1}^{Q} \kappa_{q'}^2 \right)^{1/2} = \mathcal{G}_{\text{SSM}}^{(Q)}(W). \tag{310}$$

Since $S \geq 1$ and the admissible length-S pair $\xi \in \mathcal{Z}$ were arbitrary, and the right-hand side is independent of S and ξ, the bound holds uniformly over all such lengths and admissible pairs for the fixed full parameter tuple W, proving Theorem C.3. □

C.2 Auxiliary Lemmas for Multi-Layer SSM Federated Convergence

First, we describe the following setup that will be utilized in the remainder of the proofs of the multi-layer SSM federated convergence.

Fix a reference full-model parameter tuple $W^r = (W^{r,(1)}, \ldots, W^{r,(Q)})$. For each layer $q \in \{1, \ldots, Q\}$ of the Q-layer deep SSM defined in C.1, choose some radius $R_q > 0$ and define:

$$\mathcal{N}_{R_q}^{(q)} \left(W^{r,(q)} \right) := \left\{ V^{(q)} : \left\| V^{(q)} - W^{r,(q)} \right\|_F \leq R_q \right\} \tag{311}$$

as the layerwise parameter neighborhood of radius R_q, centered at the parameter tuple of layer q in the central model W^r.

Furthermore, define:

$$\mathcal{N}_{\mathbf{R}_Q} := \mathcal{N}_{R_1}^{(1)}\left(W^{r,(1)}\right) \times \cdots \times \mathcal{N}_{R_Q}^{(Q)}\left(W^{r,(Q)}\right) \tag{312}$$

$$\mathbf{R}_Q := (R_1, \ldots, R_Q)\,. \tag{313}$$

The radii $R_1, \ldots, R_Q$ are chosen such that:

$$\mathcal{N}_{\mathbf{R}_Q} \subseteq \mathcal{S}_{\boldsymbol{\gamma},\boldsymbol{\delta}}^{(Q)}. \tag{314}$$

Furthermore, we assume that Assumption C.1 holds uniformly over this neighborhood with the same finite constants $L_{x,q}$. Furthermore, we let Assumption 4.2 hold at the final-layer predictions $y_t^{(Q)}(W; x)$ with the same finite constants L_ℓ and H_ℓ throughout this neighborhood.

All stability margins, layer-input bounds, and loss-regularity constants are fixed independently of the parameter tuple $W \in \mathcal{N}_{\mathbf{R}_Q}$, the admissible sequence pair, its finite length, and the token position. In particular, $\rho_q = \exp(-\gamma_q \delta_q) < 1$ is fixed for each layer.

For each $q \in \{1, \ldots, Q\}$, define the parameter envelopes as:

$$M_{X,q} := \left\|W_X^{r,(q)}\right\|_2 + R_q \qquad X \in \{\Delta, B, C\}, \tag{315}$$

$$M_{A,q} := \left\|A^{r,(q)}\right\|_2 + R_q. \tag{316}$$

By the definition of the layerwise neighborhoods, and the triangle inequality, every $W \in \mathcal{N}_{\mathbf{R}_Q}$ satisfies:

$$\begin{aligned} \left\|W_X^{(q)}\right\|_2 &\le M_{X,q} \qquad X \in \{\Delta, B, C\}, \\ \left\|A^{(q)}\right\|_2 &\le M_{A,q} \end{aligned} \tag{317}$$

Given these envelopes, let:

$$M_{K_\Delta,q} := M_{\Delta,q} L_{x,q} + \sqrt{D}\log(2), \tag{318}$$

$$M_{K_B,q} := M_{B,q} L_{x,q}^2, \tag{319}$$

$$\begin{aligned} M_{K_W,q} := NDM_{K_B,q}M_{K_\Delta,q}\left(M_{K_\Delta,q} + \sqrt{D}M_{A,q}L_{x,q}\right)\left(\frac{\rho_q}{1-\rho_q}\right) \\ + L_{x,q}^2\sqrt{N}M_{K_\Delta,q} + L_{x,q}\sqrt{D}M_{K_B,q}. \end{aligned} \tag{320}$$

These are the layerwise counterparts of the single-layer envelopes introduced in Appendix B.1, with the neighborhood radius R replaced by R_q.

Then, by the definitions of $K_{\Delta,q}$, $K_{B,q}$, and $K_{W,q}$, it follows that:

$$K_{\Delta,q} \le M_{K_\Delta,q}, \qquad K_{B,q} \le M_{K_B,q}, \qquad K_{W,q} \le M_{K_W,q} \tag{321}$$

over $W \in \mathcal{N}_{\mathbf{R}_Q}$.

Further, define:

$$H_{R_q} := \frac{M_{K_B,q}M_{K_\Delta,q}\sqrt{ND}}{1-\rho_q}, \tag{322}$$

$$J_{R_q} := \frac{M_{K_W,q}}{1-\rho_q}. \tag{323}$$

$$M_{\Lambda_x,q} := \rho_q M_{A,q} M_{\Delta,q} H_{R_q} + M_{B,q}L_{x,q}\left(2M_{K_\Delta,q} + L_{x,q}M_{\Delta,q}\right), \tag{324}$$

$$M_{\kappa,q} := M_{C,q}\left(H_{R_q} + \frac{L_{x,q}M_{\Lambda_x,q}}{1-\rho_q}\right). \tag{325}$$

By the definitions of $\Lambda_{x,q}$ and κ_q in the multi-layer gradient analysis, the following bounds hold

$$\Lambda_{x,q}(W) \le M_{\Lambda_x,q}, \qquad \kappa_q(W) \le M_{\kappa,q} \tag{326}$$

for every $W \in \mathcal{N}_{\mathbf{R}_Q}$. These bounds follow by replacing each parameter norm by its corresponding envelope, since all factors are nonnegative.

C.2.1 Local multi-layer SSM smoothness Lemma And Proof

Lemma C.4 (Local multi-layer SSM smoothness)**.** *Under the neighborhood assumptions and parameter envelopes introduced above, there exists a finite constant $L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}$ defined:*

$$L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} := H_\ell \Theta_Q^2 + L_\ell \Omega_Q, \tag{327}$$

with Θ_Q and Ω_Q defined in (330) *and* (331) *respectively, such that, for every admissible sequence pair ξ of finite length S and every $W, V \in \mathcal{N}_{\mathbf{R}_Q}$,*

$$\left\| \nabla_W \mathcal{L}^{S,(Q)}(W;\xi) - \nabla_W \mathcal{L}^{S,(Q)}(V;\xi) \right\|_F \le L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \|W - V\|_F, \tag{328}$$

independent of S and ξ.

Furthermore, if each client objective $\mathcal{L}_k$ is finite and differentiable on $\mathcal{N}_{\mathbf{R}_Q}$ and satisfies:

$$\nabla_W \mathcal{L}_k(W) = \mathbb{E}_{\xi\sim\mathcal{P}_k}\left[\nabla_W \mathcal{L}^{S,(Q)}(W;\xi) \right], \tag{329}$$

then every $\mathcal{L}_k$, and therefore $f = \sum_k p_k \mathcal{L}_k$, is also $L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}$-smooth on $\mathcal{N}_{\mathbf{R}_Q}$.

Θ_Q and Ω_Q have the following recursive definitions:

$$\Theta_0 := 0, \qquad \Theta_q := \left(L_{x,q}^2 \left(H_{R_q} + M_{C,q} J_{R_q} \right)^2 + M_{\kappa,q}^2 \Theta_{q-1}^2 \right)^{1/2}, \tag{330}$$

$$\begin{aligned} \Omega_0 := 0, \qquad \Omega_q := {} & \frac{M_{C,q} L_{x,q} \mathcal{E}_q}{1-\rho_q} + 2\left(L_{x,q} + M_{C,q}\Theta_{q-1}\right)\left(J_{R_q} + \frac{M_{\Lambda_x,q}\Theta_{q-1}}{1-\rho_q} \right) \\ & + 2H_{R_q}\Theta_{q-1} + M_{\kappa,q}\Omega_{q-1}. \end{aligned} \tag{331}$$

where:

$$\begin{aligned} \mathcal{E}_q := {} & \rho_q H_{R_q} \Bigg[\left(M_{K_\Delta,q} + M_{A,q}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right) \right)^2 + 2\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right) \\ & + M_{A,q}\left(\frac{1}{4}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)^2 + 2\Theta_{q-1} \right) \Bigg] \\ & + 2\rho_q \left(M_{K_\Delta,q} + M_{A,q}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right) \right)\left(J_{R_q} + \frac{M_{\Lambda_x,q}\Theta_{q-1}}{1-\rho_q} \right) \\ & + 2\Theta_{q-1} M_{K_\Delta,q} L_{x,q} \\ & + 2\left(L_{x,q} + M_{B,q}\Theta_{q-1}\right)\left[L_{x,q}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right) + M_{K_\Delta,q}\Theta_{q-1} \right] \\ & + M_{B,q} L_{x,q} \Bigg[L_{x,q}\left(\frac{1}{4}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)^2 + 2\Theta_{q-1} \right) + 2\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)\Theta_{q-1} \Bigg]. \end{aligned} \tag{332}$$

Proof of Lemma C.4: Let $W \in \mathcal{N}_{\mathbf{R}_Q}$ be some fixed arbitrary parameter tuple, and $\xi = (x, y^*)$ be some admissible fixed input-target sequence pair of length $S \ge 1$. Note that, while x and y^* remain fixed throughout the differentiation, the intermediate inputs and outputs $x_t^{(q)} = y_t^{(q-1)}$ depend on the parameters of the preceding layers for $q \ge 2$.

We introduce:

$$U = \left(U^{(1)}, \dots, U^{(Q)} \right), \qquad V = \left(V^{(1)}, \dots, V^{(Q)} \right) \tag{333}$$

as two arbitrary full-model perturbations such that:

$$U^{(q)} = \left(U_\Delta^{(q)}, U_B^{(q)}, U_C^{(q)}, U_A^{(q)} \right), \qquad V^{(q)} = \left(V_\Delta^{(q)}, V_B^{(q)}, V_C^{(q)}, V_A^{(q)} \right) \tag{334}$$

and:

$$\|U\|_F^2 := \sum_{q=1}^{Q} \left\| U^{(q)} \right\|_F^2 = 1, \qquad \|V\|_F^2 := \sum_{q=1}^{Q} \left\| V^{(q)} \right\|_F^2 = 1. \tag{335}$$

Note that each layerwise norm such as $\left\|U^{(q)}\right\|_F$ is defined as the Euclidean combination of the Frobenius norms of its four parameter blocks. As a result, every layerwise direction and every individual parameter-block direction has norm at most one.

Then, for a twice continuously differentiable function F of the full parameter tuple, we define the first and second directional derivatives by

$$DF(W)[U] := \frac{d}{d\epsilon} F(W + \epsilon U)\Big|_{\epsilon=0}, \tag{336}$$

$$D^2F(W)[U,V] := \frac{\partial^2}{\partial\epsilon\partial\zeta} F(W + \epsilon U + \zeta V)\Big|_{\epsilon=\zeta=0}. \tag{337}$$

Unless otherwise specified, all derivatives in the remaining part of this proof are taken with respect to the full parameter tuple W.

The first-order chain rule gives

$$D\left(\ell_t\left(y_t^{(Q)}\right)\right)(W)[U] = \left\langle \nabla_y \ell_t\left(y_t^{(Q)}(W)\right), Dy_t^{(Q)}(W)[U]\right\rangle. \tag{338}$$

where we used the notation $\ell_t(y) = \ell_t(y, y_t^*)$. By differentiating this expression in V direction, we get:

$$\begin{aligned} D^2\left(\ell_t\left(y_t^{(Q)}\right)\right)(W)[U,V] = &\left(Dy_t^{(Q)}(W)[U]\right)^\top \nabla_y^2 \ell_t\left(y_t^{(Q)}(W)\right) Dy_t^{(Q)}(W)[V] \\ &+ \left\langle \nabla_y \ell_t\left(y_t^{(Q)}(W)\right), D^2 y_t^{(Q)}(W)[U,V]\right\rangle. \end{aligned} \tag{339}$$

By the neighborhood-uniform loss-regularity bounds introduced in Assumption 4.2 and the Cauchy-Schwarz inequality, it follows that

$$\left|D^2\left(\ell_t\left(y_t^{(Q)}\right)\right)(W)[U,V]\right| \leq H_\ell \left\|Dy_t^{(Q)}(W)[U]\right\|_2 \left\|Dy_t^{(Q)}(W)[V]\right\|_2 + L_\ell \left\|D^2 y_t^{(Q)}(W)[U,V]\right\|_2. \tag{340}$$

As such, to bound the second derivative of the loss, it suffices to derive uniform bounds on the first and second directional derivatives of the final-layer output.

We first establish the first-order output sensitivity bounds needed for the second-order calculation.

For ease of notation, for a full-model direction U we write $U^{(\leq q)} = \left(U^{(1)}, \ldots, U^{(q)}\right)$ and $U^{(<q)} = (U^{(1)}, \ldots, U^{(q-1)})$. We further define the following recursion:

$$\Theta_0 := 0, \qquad \Theta_q := \left(L_{x,q}^2\left(H_{R_q} + M_{C,q} J_{R_q}\right)^2 + M_{\kappa,q}^2 \Theta_{q-1}^2\right)^{1/2} \tag{341}$$

First, by (270) and the definition of the neighborhood envelopes, we have:

$$\left\|\frac{\partial y_t^{(q)}}{\partial W^{(q)}}\right\|_F \leq L_{x,q}\left(H_{R_q} + M_{C,q} J_{R_q}\right). \tag{342}$$

Then, by (304) and the bound $\kappa_q(W) \leq M_{\kappa,q}$, for every $t \in \{1, \ldots, S\}$,

$$\sum_{j=1}^{t} \left\|\frac{\partial y_t^{(q)}}{\partial x_j^{(q)}}\right\|_2 = \sum_{j=1}^{t} \left\|\mathbf{K}_{t,j}^{(q)}\right\|_2 \leq \kappa_q(W) \leq M_{\kappa,q}. \tag{343}$$

We next prove by induction on q that

$$\left\|Dy_t^{(q)}(W)[U]\right\|_2 \leq \Theta_q \left\|U^{(\leq q)}\right\|_F \tag{344}$$

for every $q \in \{1, \ldots, Q\}$ and every $t \in \{1, \ldots, S\}$.

When $q = 1$, the input sequence is independent of W, and the output of the first layer depends only on $W^{(1)}$. As such:

$$Dy_t^{(1)}(W)[U] = \frac{\partial y_t^{(1)}}{\partial W^{(1)}} \operatorname{vec}\left(U^{(1)}\right), \tag{345}$$

Using $\|Az\|_2 \le \|A\|_F \|z\|_2$ and $\|\operatorname{vec}(U^{(1)})\|_2 = \|U^{(1)}\|_F$ gives

$$\left\|Dy_t^{(1)}(W)[U]\right\|_2 \le L_{x,1}\left(H_{R_1} + M_{C,1}J_{R_1}\right)\left\|U^{(1)}\right\|_F = \Theta_1 \left\|U^{(\le 1)}\right\|_F, \tag{346}$$

where the equality follows from $\Theta_0 = 0$ and the definition of Θ_1.

Next, for a fixed $q \ge 2$, Since $U^{(\le q-1)} = U^{(<q)}$, suppose (the induction hypothesis):

$$\left\|Dy_j^{(q-1)}(W)[U]\right\|_2 \le \Theta_{q-1}\left\|U^{(<q)}\right\|_F, \qquad j \in \{1, \dots, S\}. \tag{347}$$

For a fixed t, the output $y_t^{(q)}$ depends on the current-layer parameters $W^{(q)}$ and the input tokens $x_1^{(q)}, \dots, x_t^{(q)}$, which depend on the preceding layers through $x_j^{(q)} = y_j^{(q-1)}$. The chain rule therefore gives

$$Dy_t^{(q)}(W)[U] = \frac{\partial y_t^{(q)}}{\partial W^{(q)}} \operatorname{vec}\left(U^{(q)}\right) + \sum_{j=1}^{t} \frac{\partial y_t^{(q)}}{\partial x_j^{(q)}} Dy_j^{(q-1)}(W)[U]. \tag{348}$$

The first term accounts for the direct parameter perturbation in layer q, while the sum accounts for changes propagated through its input sequence.

Taking norms and applying the established Jacobian bounds and the induction hypothesis yields

$$\begin{aligned}\left\|Dy_t^{(q)}(W)[U]\right\|_2 \le& L_{x,q}\left(H_{R_q} + M_{C,q}J_{R_q}\right)\left\|U^{(q)}\right\|_F + \sum_{j=1}^{t}\left\|\frac{\partial y_t^{(q)}}{\partial x_j^{(q)}}\right\|_2 \left\|Dy_j^{(q-1)}(W)[U]\right\|_2 \\ \le& L_{x,q}\left(H_{R_q} + M_{C,q}J_{R_q}\right)\left\|U^{(q)}\right\|_F + M_{\kappa,q}\Theta_{q-1}\left\|U^{(<q)}\right\|_F. \end{aligned} \tag{349}$$

Applying Cauchy-Schwarz to the two terms gives

$$\begin{aligned}\left\|Dy_t^{(q)}(W)[U]\right\|_2 &\le \left(L_{x,q}^2\left(H_{R_q} + M_{C,q}J_{R_q}\right)^2 + M_{\kappa,q}^2\Theta_{q-1}^2\right)^{1/2} \times \left(\left\|U^{(q)}\right\|_F^2 + \left\|U^{(<q)}\right\|_F^2\right)^{1/2} \\ &= \Theta_q \left\|U^{(\le q)}\right\|_F, \end{aligned} \tag{350}$$

where the equality follows from the definition of Θ_q and the block Frobenius norm. This completes the induction.

The same argument applies to V. Finally, since $\|U^{(\le q)}\|_F \le \|U\|_F = 1$ and $\|V^{(\le q)}\|_F \le \|V\|_F = 1$, we obtain the following first-derivative bounds:

$$\left\|Dy_t^{(q)}(W)[U]\right\|_2 \le \Theta_q, \qquad \left\|Dy_t^{(q)}(W)[V]\right\|_2 \le \Theta_q, \tag{351}$$

uniformly over all layers $q \in \{1, \dots, Q\}$ and token positions $t \in \{1, \dots, S\}$.

Finally, applying this result to $q = Q$, we get the resulting first-derivative bounds:

$$\left\|Dy_t^{(Q)}(W)[U]\right\|_2 \le \Theta_Q, \qquad \left\|Dy_t^{(Q)}(W)[V]\right\|_2 \le \Theta_Q, \tag{352}$$

and substituting back into (340) we get:

$$\left|D^2\left(\ell_t\left(y_t^{(Q)}\right)\right)(W)[U,V]\right| \le H_\ell \Theta_Q^2 + L_\ell \left\|D^2 y_t^{(Q)}(W)[U,V]\right\|_2. \tag{353}$$

We next bound the second output derivative term $\left\|D^2 y_t^{(Q)}(W)[U,V]\right\|_2$. For any layer q, the second-order product rule gives:

$$D^2 y_t^{(q)}(W)[U,V] = \left(D^2 h_t^{(q)}(W)[U,V]\right)^\top C_t^{(q)} + \left(h_t^{(q)}\right)^\top D^2 C_t^{(q)}(W)[U,V]$$

$$+\left(Dh_t^{(q)}(W)[U]\right)^\top DC_t^{(q)}(W)[V]+\left(Dh_t^{(q)}(W)[V]\right)^\top DC_t^{(q)}(W)[U]. \tag{354}$$

We bound each term separately.

By the first-order bounds already established:

$$\left\|Dx_t^{(q)}(W)[U]\right\|_2=\left\|Dy_t^{(q-1)}(W)[U]\right\|_2\le\Theta_{q-1}, \tag{355}$$

vanishing for $q=1$, consistent with $\Theta_0=0$. The same holds for $\|Dx_t^{(q)}(W)[V]\|_2$. As a result, by the product rule:

$$\left\|DC_t^{(q)}(W)[U]\right\|_2\le L_{x,q}+M_{C,q}\Theta_{q-1}, \tag{356}$$

$$\left\|DC_t^{(q)}(W)[V]\right\|_2\le L_{x,q}+M_{C,q}\Theta_{q-1}, \tag{357}$$

$$\left\|D^2C_t^{(q)}(W)[U,V]\right\|_2\le 2\Theta_{q-1}+M_{C,q}\left\|D^2x_t^{(q)}(W)[U,V]\right\|_2. \tag{358}$$

Furthermore, by (268) and (297),

$$\left\|Dh_t^{(q)}(W)[U]\right\|_F\le J_{R_q}+M_{\Lambda_x,q}\Theta_{q-1}\sum_{j=1}^{t}\rho_q^{t-j}\le J_{R_q}+\frac{M_{\Lambda_x,q}\Theta_{q-1}}{1-\rho_q}, \tag{359}$$

with the same inequality holding for $\left\|Dh_t^{(q)}(W)[V]\right\|_F$.

Since $\left\|C_t^{(q)}\right\|_2\le M_{C,q}L_{x,q}$ and $\left\|h_t^{(q)}\right\|_F\le H_{R_q}$, equation (354) yields

$$\begin{aligned}\left\|D^2y_t^{(q)}(W)[U,V]\right\|_2\le&M_{C,q}L_{x,q}\left\|D^2h_t^{(q)}(W)[U,V]\right\|_F+M_{C,q}H_{R_q}\left\|D^2x_t^{(q)}(W)[U,V]\right\|_2\\&+2\left(L_{x,q}+M_{C,q}\Theta_{q-1}\right)\left(J_{R_q}+\frac{M_{\Lambda_x,q}\Theta_{q-1}}{1-\rho_q}\right)+2H_{R_q}\Theta_{q-1}.\end{aligned} \tag{360}$$

As a result, we need to bound the second derivative of the hidden state. By the definition of $h_t^{(q)}(W)$, differentiating it twice yields:

$$\begin{aligned}D^2h_t^{(q)}(W)[U,V]=&\bar{A}_t^{(q)}\odot D^2h_{t-1}^{(q)}(W)[U,V]+D^2\bar{A}_t^{(q)}(W)[U,V]\odot h_{t-1}^{(q)}\\&+D\bar{A}_t^{(q)}(W)[U]\odot Dh_{t-1}^{(q)}(W)[V]+D\bar{A}_t^{(q)}(W)[V]\odot Dh_{t-1}^{(q)}(W)[U]\\&+D^2\left[B_t^{(q)}\left(\Delta_t^{(q)}\odot x_t^{(q)}\right)^\top\right](W)[U,V].\end{aligned} \tag{361}$$

Then, since the absolute values of the first and second derivatives of $\mathrm{Softplus}$ are bounded above by 1 and $1/4$, respectively, the product and chain rule give:

$$\left\|D\Delta_t^{(q)}(W)[U]\right\|_2\le L_{x,q}+M_{\Delta,q}\Theta_{q-1}, \tag{362}$$

$$\left\|D\Delta_t^{(q)}(W)[V]\right\|_2\le L_{x,q}+M_{\Delta,q}\Theta_{q-1}, \tag{363}$$

$$\left\|D^2\Delta_t^{(q)}(W)[U,V]\right\|_2\le\frac{1}{4}\left(L_{x,q}+M_{\Delta,q}\Theta_{q-1}\right)^2+2\Theta_{q-1}+M_{\Delta,q}\left\|D^2x_t^{(q)}(W)[U,V]\right\|_2. \tag{364}$$

Then, applying the product rule and the exponential chain rule (per every entry of A), and using the established state and first-derivative bounds, we get:

$$\left\|D^2h_t^{(q)}(W)[U,V]\right\|_F\le\rho_q\left\|D^2h_{t-1}^{(q)}(W)[U,V]\right\|_F+\mathcal{E}_q+M_{\Lambda_x,q}\left\|D^2x_t^{(q)}(W)[U,V]\right\|_2, \tag{365}$$

where we collected all the terms independent of the second input derivative in $\mathcal{E}_q$ defined as

$$\begin{aligned}\mathcal{E}_q :=& \rho_q H_{R_q} \Bigg[\left(M_{K_\Delta,q} + M_{A,q}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)\right)^2 + 2\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right) \\ &+ M_{A,q}\left(\frac{1}{4}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)^2 + 2\Theta_{q-1}\right)\Bigg] \\ &+ 2\rho_q\left(M_{K_\Delta,q} + M_{A,q}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)\right)\left(J_{R_q} + \frac{M_{\Lambda_x,q}\Theta_{q-1}}{1-\rho_q}\right) \\ &+ 2\Theta_{q-1}M_{K_\Delta,q}L_{x,q} \\ &+ 2\left(L_{x,q} + M_{B,q}\Theta_{q-1}\right)\left[L_{x,q}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right) + M_{K_\Delta,q}\Theta_{q-1}\right] \\ &+ M_{B,q}L_{x,q}\Bigg[L_{x,q}\left(\frac{1}{4}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)^2 + 2\Theta_{q-1}\right) + 2\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)\Theta_{q-1}\Bigg].\end{aligned} \tag{366}$$

The first term in $\mathcal{E}_q$ comes from the second transition derivative multiplied entrywise by the state:

$$D^2\bar{A}_t^{(q)}(W)[U,V] \odot h_{t-1}^{(q)}.$$

The second comes from the two mixed transition-state terms,

$$D\bar{A}_t^{(q)}(W)[U] \odot Dh_{t-1}^{(q)}(W)[V] + D\bar{A}_t^{(q)}(W)[V] \odot Dh_{t-1}^{(q)}(W)[U].$$

The remaining three terms come from differentiating the additive state term

$$B_t^{(q)}\left(\Delta_t^{(q)} \odot x_t^{(q)}\right)^\top$$

twice. The contributions involving the second input derivative are collectively bounded by

$$M_{\Lambda_x,q}\left\|D^2x_t^{(q)}(W)[U,V]\right\|_2,$$

where $M_{\Lambda_x,q}$ was defined in (324).

Since the initial state and its derivatives equal $\mathbf{0}$, unrolling (365) and summing the geometric series gives

$$\left\|D^2h_t^{(q)}(W)[U,V]\right\|_F \le \frac{\mathcal{E}_q}{1-\rho_q} + \frac{M_{\Lambda_x,q}}{1-\rho_q}\max_{1\le j\le t}\left\|D^2x_j^{(q)}(W)[U,V]\right\|_2. \tag{367}$$

Next, we note that since $\left\|D^2x_t^{(q)}(W)[U,V]\right\|_2 \le \max_{1\le j\le t}\left\|D^2x_j^{(q)}(W)[U,V]\right\|_2$, substituting (367) and using this fact, the new coefficient of the $\max_{1\le j\le t}\left\|D^2x_j^{(q)}(W)[U,V]\right\|_2$ term becomes:

$$M_{C,q}H_{R_q} + \frac{M_{C,q}L_{x,q}M_{\Lambda_x,q}}{1-\rho_q} = M_{\kappa,q}. \tag{368}$$

Finally, we define a new recursive term Ω_q as follows:

$$\begin{aligned}&\Omega_0 := 0, \\ &\Omega_q := \frac{M_{C,q}L_{x,q}\mathcal{E}_q}{1-\rho_q} + 2\left(L_{x,q} + M_{C,q}\Theta_{q-1}\right)\left(J_{R_q} + \frac{M_{\Lambda_x,q}\Theta_{q-1}}{1-\rho_q}\right) \\ &\qquad + 2H_{R_q}\Theta_{q-1} + M_{\kappa,q}\Omega_{q-1}.\end{aligned} \tag{369}$$

For $q = 1$, second input derivative vanishes. Then, for $q \ge 2$, and using the similar induction argument as in the first-order derivative, as well as the fact $x_j^{(q)} = y_j^{(q-1)}$, the induction yields:

$$\left\|D^2y_t^{(q)}(W)[U,V]\right\|_2 \le \Omega_q \tag{370}$$

for every layer and token position. All constants are finite and independent of W, ξ, S, and t, because the neighborhood envelopes are fixed, $\rho_q < 1$, and Q is finite.

Finally, applying this bound to $q = Q$, and substituting the result into (340) it follows that:

$$\left| D^2 \left(\ell_t \left(y_t^{(Q)} \right) \right) (W)[U, V] \right| \leq H_\ell \Theta_Q^2 + L_\ell \Omega_Q =: L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}. \tag{371}$$

Averaging over the token losses preserves this bound. Since U, V are arbitrary unit directions, it is true that:

$$\left\| \nabla_W^2 \mathcal{L}^{S,(Q)}(W;\xi) \right\|_{\mathrm{op}} \leq L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}, \tag{372}$$

where the operator norm is induced by the block Frobenius norm.

Since $\mathcal{N}_{\mathbf{R}_Q}$ is convex, integrating the Hessian along the segment joining any $W_0, W_1 \in \mathcal{N}_{\mathbf{R}_Q}$ gives

$$\left\| \nabla_W \mathcal{L}^{S,(Q)}(W_1;\xi) - \nabla_W \mathcal{L}^{S,(Q)}(W_0;\xi) \right\|_F \leq L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)} \|W_1 - W_0\|_F. \tag{373}$$

Taking expectations, by Assumption 4.6 and Jensen's inequality, we get:

$$\begin{aligned} \|\nabla_W \mathcal{L}_k(W_1) - \nabla_W \mathcal{L}_k(W_0)\|_F &= \left\| \mathbb{E}_{\xi \sim \mathcal{P}_k} \left[\nabla_W \mathcal{L}^{S,(Q)}(W_1;\xi) - \nabla_W \mathcal{L}^{S,(Q)}(W_0;\xi) \right] \right\|_F \\ &\leq \mathbb{E}_{\xi \sim \mathcal{P}_k} \left[\left\| \nabla_W \mathcal{L}^{S,(Q)}(W_1;\xi) - \nabla_W \mathcal{L}^{S,(Q)}(W_0;\xi) \right\|_F \right] \\ &\leq L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)} \|W_1 - W_0\|_F, \end{aligned} \tag{374}$$

demonstrating that every client objective is $L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}$-smooth on $\mathcal{N}_{\mathbf{R}_Q}$.

Finally, since this inequality holds for every client objective, and $p_k \geq 0$ and $\sum_k p_k = 1$, it also holds for $f = \sum_k p_k \mathcal{L}_k$, demonstrating that f is also $L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}$-smooth on $\mathcal{N}_{\mathbf{R}_Q}$, completing the proof. □.

C.2.2 Multi-layer SSM gradient second-moment constant Lemma and Proof

Lemma C.5 (Multi-layer SSM gradient second-moment constant)**.** *Suppose the Assumptions 4.2 and C.1 hold. Let $\mathcal{N}_{\mathbf{R}_Q}(W^r) \subseteq \mathcal{S}_{\boldsymbol{\gamma},\boldsymbol{\delta}}^{(Q)}$ be a stable local model neighborhood around the round-r broadcast Q-deep model W^r. Let:*

$$g(W;\xi) := \nabla_W \mathcal{L}^{S,(Q)}(W;\xi) \tag{375}$$

denote the stochastic sequence-level gradient for a sampled input-target sequence pair $\xi = (x_{1:S}, y_{1:S}^*)$. Then, for every $W \in \mathcal{N}_{\mathbf{R}_Q}(W^r)$,

$$\mathbb{E}_\xi \|g(W;\xi)\|_F^2 \leq \left(G_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)} \right)^2, \tag{376}$$

where

$$G_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)} := L_\ell \left(\sum_{q=1}^{Q} L_{x,q}^2 \left(H_{R_q} + M_{C,q} J_{R_q} \right)^2 \prod_{q'=q+1}^{Q} M_{\kappa,q'}^2 \right)^{1/2} = L_\ell \Theta_Q. \tag{377}$$

where $H_{R_q}, J_{R_q}, M_{C,q}$ and $M_{\kappa,q}$ are defined in (322)*,* (323)*,* (315)*, and* (325) *respectively.*

Proof of Lemma C.5: By Theorem C.3, under the assumptions of the lemma, for every $W \in \mathcal{N}_{\mathbf{R}_Q}(W^r) \subseteq \mathcal{S}_{\boldsymbol{\gamma},\boldsymbol{\delta}}^{(Q)}$, every finite integer $S \geq 1$, and every admissible length-S sequence pair $\xi \in \mathcal{Z}$, we have:

$$\left\| \nabla_W \mathcal{L}^{S,(Q)}(W;\xi) \right\|_F \leq L_\ell \left(\sum_{q=1}^{Q} L_{x,q}^2 \left(H_q + \left\| W_C^{(q)} \right\|_2 J_q \right)^2 \prod_{q'=q+1}^{Q} \kappa_{q'}^2 \right)^{1/2}.$$

Since $g(W;\xi) = \nabla_W \mathcal{L}^{S,(Q)}(W;\xi)$, it follows that:

$$\|g(W;\xi)\|_F \le L_\ell \left(\sum_{q=1}^{Q} L_{x,q}^2 \left(H_q + \left\| W_C^{(q)} \right\|_2 J_q \right)^2 \prod_{q'=q+1}^{Q} \kappa_{q'}^2 \right)^{1/2}. \tag{378}$$

Restricting to $W \in \mathcal{N}_{\mathbf{R}_Q}(W^r)$, by the local neighborhood bounds defined previously ((322), (323), (315), and (325)),

$$H_q \le H_{R_q}, \qquad \left\| W_C^{(q)} \right\|_2 \le M_{C,q}, \qquad J_q \le J_{R_q}, \qquad \kappa_q \le M_{\kappa,q}. \tag{379}$$

Therefore,

$$\|g(W;\xi)\|_F \le L_\ell \left(\sum_{q=1}^{Q} L_{x,q}^2 \left(H_{R_q} + M_{C,q} J_{R_q} \right)^2 \prod_{q'=q+1}^{Q} M_{\kappa,q'}^2 \right)^{1/2} = G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}. \tag{380}$$

Since both values are non-negative, squaring both sides gives:

$$\|g(W;\xi)\|_F^2 \le \left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \right)^2 = L_\ell^2 \left(\sum_{q=1}^{Q} L_{x,q}^2 \left(H_{R_q} + M_{C,q} J_{R_q} \right)^2 \prod_{q'=q+1}^{Q} M_{\kappa,q'}^2 \right) \tag{381}$$

Finally, taking expectation over the sampled sequence ξ preserves the bound:

$$\mathbb{E}_\xi \left[\|g(W;\xi)\|_F^2 \right] \le \left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \right)^2 \tag{382}$$

proving the lemma. □

C.2.3 Multi-layer SSM stochastic gradient variance bound Lemma and Proof

Lemma C.6 (Multi-layer SSM stochastic gradient variance bound)**.** *Suppose the assumptions of Lemma C.5 and Assumption 4.6 hold. Let*

$$\mathcal{L}(W) := \mathbb{E}_{\xi\sim\mathcal{P}} \left[\mathcal{L}^{S,(Q)}(W;\xi) \right]$$

denote the expected sequence loss over some local data distribution $\mathcal{P}$, for a Q-layer deep SSM parameterized by W. Define

$$g(W;\xi) := \nabla_W \mathcal{L}^{S,(Q)}(W;\xi)$$

as the single-sequence stochastic gradient. For a mini-batch $\Xi_\xi = \{\xi_1, \dots, \xi_\Xi\}$ of Ξ independent sampled sequences, define

$$g_\Xi(W;\Xi_\xi) := \frac{1}{\Xi} \sum_{j=1}^{\Xi} g(W;\xi_j). \tag{383}$$

Then, for every $W \in \mathcal{N}_{\mathbf{R}_Q}(W^r)$ it follows that

$$\mathbb{E}_{\Xi_\xi} \left[\|g_\Xi(W;\Xi_\xi) - \nabla_W \mathcal{L}(W)\|_F^2 \right] \le \left(\sigma^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q,\Xi} \right)^2, \tag{384}$$

where

$$\left(\sigma^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q,\Xi} \right)^2 := \frac{1}{\Xi} \sup_{W\in\mathcal{N}_{\mathbf{R}_Q}(W^r)} \left(\mathbb{E}_{\xi\sim\mathcal{P}} \|g(W;\xi)\|_F^2 - \|\nabla_W \mathcal{L}(W)\|_F^2 \right). \tag{385}$$

Moreover, by Lemma C.5,

$$\left(\sigma^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q,\Xi} \right)^2 \le \frac{\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \right)^2}{\Xi}. \tag{386}$$

Proof of Lemma C.6: By unbiasedness and independence of the sampled sequences, the standard mini-batch variance identity gives:

$$\mathbb{E}_{\Xi_\xi}\left[\|g_\Xi(W;\Xi_\xi)-\nabla_W\mathcal{L}(W)\|_F^2\right]=\frac{1}{\Xi}\mathbb{E}_{\xi\sim\mathcal{P}}\left[\|g(W;\xi)-\nabla_W\mathcal{L}(W)\|_F^2\right].$$

Using the identity

$$\mathbb{E}_{\xi\sim\mathcal{P}}\left[\|g(W;\xi)-\nabla_W\mathcal{L}(W)\|_F^2\right]=\mathbb{E}_{\xi\sim\mathcal{P}}\|g(W;\xi)\|_F^2-\|\nabla_W\mathcal{L}(W)\|_F^2, \tag{387}$$

and taking the supremum over $W\in\mathcal{N}_{\mathbf{R}_Q}(W^r)$, we obtain

$$\mathbb{E}_{\Xi_\xi}\left[\|g_\Xi(W;\Xi_\xi)-\nabla_W\mathcal{L}(W)\|_F^2\right]\le\left(\sigma^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q,\Xi}\right)^2. \tag{388}$$

Finally, Lemma C.5 gives

$$\mathbb{E}_{\xi\sim\mathcal{P}}\|g(W;\xi)\|_F^2\le\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2$$

and since $\|\nabla_W\mathcal{L}(W)\|_F^2\ge 0$,

$$\left(\sigma^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q,\Xi}\right)^2\le\frac{\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2}{\Xi}. \tag{389}$$

This proves the claim. □

C.2.4 Multi-layer SSM client heterogeneity bound under total variation Lemma and Proof

Lemma C.7 (Multi-layer SSM client heterogeneity bound under total variation)**.** *Suppose the assumptions of Lemma C.5 and Assumption 4.6 hold. Let*

$$f(W):=\sum_{k=1}^K p_k\mathcal{L}_k(W) \tag{390}$$

be the global objective, where $p_k\ge 0$ and $\sum_{k=1}^K p_k=1$. Let $\mathcal{P}_k$ denote the local data distribution of client k, and define the global mixture distribution as:

$$\bar{\mathcal{P}}:=\sum_{k=1}^K p_k\mathcal{P}_k. \tag{391}$$

Define the local client heterogeneity constant

$$\left(\Gamma^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2:=\sup_{W\in\mathcal{N}_{\mathbf{R}_Q}(W^r)}\sum_{k=1}^K p_k\left\|\nabla_W\mathcal{L}_k(W)-\nabla_W f(W)\right\|_F^2. \tag{392}$$

Then:

$$\left(\Gamma^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2\le 4\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2\sum_{k=1}^K p_k\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2, \tag{393}$$

where

$$\mathrm{TV}(P,Q):=\sup_A|P(A)-Q(A)|. \tag{394}$$

Proof of Lemma C.7: Recall that for each client k, the local objective is:

$$\mathcal{L}_k(W)=\mathbb{E}_{\xi\sim\mathcal{P}_k}\left[\mathcal{L}^{S,(Q)}(W;\xi)\right],$$

and, by Assumption 4.6,

$$\nabla_W\mathcal{L}_k(W)=\mathbb{E}_{\xi\sim\mathcal{P}_k}\left[g(W;\xi)\right]. \tag{395}$$

Similarly, since

$$f(W) := \sum_{k=1}^{K} p_k \mathcal{L}_k(W),$$

we have

$$\nabla_W f(W) = \sum_{k=1}^{K} p_k \nabla_W \mathcal{L}_k(W) = \sum_{k=1}^{K} p_k \mathbb{E}_{\xi\sim\mathcal{P}_k}\left[g(W;\xi)\right]. \tag{396}$$

Next, by the definition of the mixture distribution $\bar{\mathcal{P}}$, we can write this as:

$$\nabla_W f(W) = \mathbb{E}_{\xi\sim\bar{\mathcal{P}}}\left[g(W;\xi)\right]. \tag{397}$$

Therefore, it follows that

$$\nabla_W \mathcal{L}_k(W) - \nabla_W f(W) = \mathbb{E}_{\xi\sim\mathcal{P}_k}[g(W;\xi)] - \mathbb{E}_{\xi\sim\bar{\mathcal{P}}}[g(W;\xi)]. \tag{398}$$

By the pointwise bound (380) established in the proof of Lemma C.5, for every $W \in \mathcal{N}_{\mathbf{R}_Q}(W^r)$ and every admissible sequence pair $\xi \in \mathcal{Z}$,

$$\|g(W;\xi)\|_F \le G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}.$$

Since $\mathcal{P}_k(\mathcal{Z}) = \bar{\mathcal{P}}(\mathcal{Z}) = 1$, the total-variation inequality for bounded vector-valued functions gives:

$$\left\|\mathbb{E}_{\xi\sim\mathcal{P}_k}[g(W;\xi)] - \mathbb{E}_{\xi\sim\bar{\mathcal{P}}}[g(W;\xi)]\right\|_F \le 2G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}}). \tag{399}$$

So:

$$\|\nabla_W \mathcal{L}_k(W) - \nabla_W f(W)\|_F^2 \le 4\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2 \mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2. \tag{400}$$

Multiplying both sides by p_k and summing over all clients gives:

$$\sum_{k=1}^{K} p_k \|\nabla_W \mathcal{L}_k(W) - \nabla_W f(W)\|_F^2 \le 4\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2 \sum_{k=1}^{K} p_k\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2. \tag{401}$$

Since the right-hand side of this inequality is independent of W, taking the supremum over $W \in \mathcal{N}_{\mathbf{R}_Q}(W^r)$ we get:

$$\left(\Gamma^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2 := \sup_{W\in\mathcal{N}_{\mathbf{R}_Q}(W^r)} \sum_{k=1}^{K} p_k \|\nabla_W \mathcal{L}_k(W) - \nabla_W f(W)\|_F^2 \le 4\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2 \sum_{k=1}^{K} p_k\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2. \tag{402}$$

This proves the multi-layer SSM client heterogeneity bound. □

C.3 Corollary C.8: FedAvg Multi-Layer SSM Convergence and Proof

Corollary C.8 (FedAvg convergence for multi-layer selective SSMs)**.** *Consider the distributed learning framework established in Section 3, with objective*

$$f(W) := \sum_{k=1}^{K} p_k \mathcal{L}_k(W), \qquad p_k \ge 0, \qquad \sum_{k=1}^{K} p_k = 1,$$

where the weights p_k are fixed and

$$W = \left(W^{(1)},\ldots,W^{(Q)}\right)$$

parameterizes the selective SSM of finite depth $Q \ge 1$ defined in Appendix C.1.

Each client objective is finite and differentiable on the neighborhoods under consideration, and satisfies

$$\nabla_W \mathcal{L}_k(W) = \mathbb{E}_{\xi\sim\mathcal{P}_k}\left[\nabla_W \mathcal{L}^{S,(Q)}(W;\xi)\right].$$

where S is the length of the sampled sequence pair ξ, and all clients use the same sequence-loss map.

Let W^0 be deterministic with $f(W^0) < \infty$, and suppose that f is bounded below by a finite constant f^*.

Consider full-participation FedAvg for T communication rounds, with E local steps per round, mini-batch size Ξ, and constant deterministic local step size $\eta > 0$, where $T, E, \Xi \in \mathbb{Z}_{>0}$:

$$\begin{aligned} W_k^{r,0} &= W^r, \\ W_k^{r,e+1} &= W_k^{r,e} - \eta g_{k,\Xi}^{r,e}, \qquad e = 0, \dots, E-1, \\ W^{r+1} &= \sum_{k=1}^{K} p_k W_k^{r,E}. \end{aligned}$$

The mini-batch gradient is

$$g_{k,\Xi}^{r,e} := \frac{1}{\Xi} \sum_{i=1}^{\Xi} \nabla_W \mathcal{L}^{S_i,(Q)}(W_k^{r,e}; \xi_{k,i}^{r,e}),$$

where S_i denotes the length of $\xi_{k,i}^{r,e}$.

Conditional on the complete training history before each local step, each client draws Ξ fresh independent samples from $\mathcal{P}_k$, and the mini-batches drawn by different clients are conditionally independent.

For fixed deterministic radii $R_q > 0$, define

$$\mathcal{N}_{\mathbf{R}_Q}(W^r) := \prod_{q=1}^{Q} \mathcal{N}_{R_q}^{(q)}(W^{r,(q)}), \qquad \mathbf{R}_Q := (R_1, \dots, R_Q).$$

Suppose that, almost surely, for every round $r < T$, client k, and local step $0 \le e \le E$,

$$W_k^{r,e} \in \mathcal{N}_{\mathbf{R}_Q}(W^r) \subseteq \mathcal{S}_{\boldsymbol{\gamma},\boldsymbol{\delta}}^{(Q)}.$$

Suppose also that finite deterministic envelopes $M_{X,q}$ satisfy, almost surely for every round,

$$\left\| W_X^{r,(q)} \right\|_2 + R_q \le M_{X,q}, \qquad X \in \{\Delta, B, C, A\}, \qquad q \in \{1, \dots, Q\},$$

where $W_A^{r,(q)} := A^{r,(q)}$.

The stability margins $\gamma_q, \delta_q > 0$, the layer-input bounds $L_{x,q}$, the loss-regularity constants L_ℓ, H_ℓ, and the parameter envelopes $M_{X,q}$ are deterministic and common across rounds.

Let Assumption C.1 hold uniformly on all these neighborhoods, and for each layer, define

$$\rho_q := \exp(-\gamma_q \delta_q) < 1.$$

Define:

$$\begin{aligned} M_{K_\Delta,q} &:= M_{\Delta,q} L_{x,q} + \sqrt{D} \log(2), \\ M_{K_B,q} &:= M_{B,q} L_{x,q}^2, \\ M_{K_W,q} &:= NDM_{K_B,q} M_{K_\Delta,q} \left(M_{K_\Delta,q} + \sqrt{D} M_{A,q} L_{x,q} \right) \left(\frac{\rho_q}{1-\rho_q} \right) \\ &\quad + L_{x,q}^2 \sqrt{N} M_{K_\Delta,q} + L_{x,q} \sqrt{D} M_{K_B,q}, \\ H_{R_q} &:= \frac{M_{K_B,q} M_{K_\Delta,q} \sqrt{ND}}{1-\rho_q}, \\ J_{R_q} &:= \frac{M_{K_W,q}}{1-\rho_q}, \\ M_{\Lambda_x,q} &:= \rho_q M_{A,q} M_{\Delta,q} H_{R_q} + M_{B,q} L_{x,q} \left(2M_{K_\Delta,q} + L_{x,q} M_{\Delta,q} \right), \end{aligned}$$

$$
\begin{aligned}
M_{\kappa,q} &:= M_{C,q}\left(H_{R_q} + \frac{L_{x,q}M_{\Lambda_x,q}}{1-\rho_q}\right),\\
\Theta_q &:= \left(L_{x,q}^2\left(H_{R_q} + M_{C,q}J_{R_q}\right)^2 + M_{\kappa,q}^2\Theta_{q-1}^2\right)^{1/2}, \qquad \Theta_0 := 0,\\
\mathcal{E}_q :=& \rho_q H_{R_q}\Bigg[\left(M_{K_\Delta,q} + M_{A,q}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)\right)^2 + 2\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)\\
&\quad + M_{A,q}\left(\frac{1}{4}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)^2 + 2\Theta_{q-1}\right)\Bigg]\\
&\quad + 2\rho_q\left(M_{K_\Delta,q} + M_{A,q}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)\right)\left(J_{R_q} + \frac{M_{\Lambda_x,q}\Theta_{q-1}}{1-\rho_q}\right)\\
&\quad + 2\Theta_{q-1}M_{K_\Delta,q}L_{x,q}\\
&\quad + 2\left(L_{x,q} + M_{B,q}\Theta_{q-1}\right)\left[L_{x,q}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right) + M_{K_\Delta,q}\Theta_{q-1}\right]\\
&\quad + M_{B,q}L_{x,q}\left[L_{x,q}\left(\frac{1}{4}\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)^2 + 2\Theta_{q-1}\right) + 2\left(L_{x,q} + M_{\Delta,q}\Theta_{q-1}\right)\Theta_{q-1}\right],\\
\Omega_q &:= \frac{M_{C,q}L_{x,q}\mathcal{E}_q}{1-\rho_q} + 2\left(L_{x,q} + M_{C,q}\Theta_{q-1}\right)\left(J_{R_q} + \frac{M_{\Lambda_x,q}\Theta_{q-1}}{1-\rho_q}\right) + 2H_{R_q}\Theta_{q-1} + M_{\kappa,q}\Omega_{q-1},\\
\Omega_0 &= 0
\end{aligned}
$$

Let:

$$
\begin{aligned}
L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)} &:= H_\ell\Theta_Q^2 + L_\ell\Omega_Q,\\
G_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)} &:= L_\ell\Theta_Q = L_\ell\left(\sum_{q=1}^{Q} L_{x,q}^2\left(H_{R_q} + M_{C,q}J_{R_q}\right)^2\prod_{q'=q+1}^{Q} M_{\kappa,q'}^2\right)^{1/2},
\end{aligned}
$$

define the common deterministic smoothness and gradient bounds on all the considered neighborhoods.

For the client distributions, define

$$
\bar{\mathcal{P}} := \sum_{k=1}^{K} p_k\mathcal{P}_k, \qquad \mathcal{H}_{\mathrm{TV}} := \sum_{k=1}^{K} p_k\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2,
$$

where

$$
\mathrm{TV}(P,Q) := \sup_{\mathcal{A}}|P(\mathcal{A}) - Q(\mathcal{A})|.
$$

If the step size satisfies

$$
\eta E L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)} \le \frac{1}{2}, \qquad 12\eta^2\left(L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}\right)^2 E(E-1) \le 1,
$$

then

$$
\begin{aligned}
\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\|\nabla f(W^r)\|_F^2 \le& \frac{4(f(W^0) - f^*)}{\eta ET} + \frac{4\eta L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}\left(G_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}\right)^2}{\Xi}\sum_{k=1}^{K} p_k^2\\
&+ \frac{6\eta^2\left(L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}\right)^2(E-1)\left(G_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}\right)^2}{\Xi}\\
&+ 48\eta^2\left(L_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}\right)^2 E(E-1)\left(G_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}\right)^2\mathcal{H}_{\mathrm{TV}}.
\end{aligned}
$$

Proof of Corollary C.8: We apply the federated optimization argument from the proof of Theorem B.5 to the full parameter tuple $W = \left(W^{(1)},\ldots,W^{(Q)}\right)$. After vectorization, the Euclidean norm of this tuple is exactly its block Frobenius norm.

The preceding neighborhood estimates remain valid when the reference-dependent parameter envelopes are replaced by the common deterministic upper envelopes $M_{X,q}$, because their defining expressions are nondecreasing in those envelopes.

Lemma C.4 therefore supplies the common smoothness constant $L = L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}$.

Lemma C.6, applied conditionally on the history before each mini-batch, supplies the common variance bound $\sigma^2 = \frac{\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2}{\Xi}$.

Conditional independence across clients gives the corresponding weighted-noise variance bound $\sigma^2 \sum_k p_k^2$.

Lemma C.7 supplies the common client-gradient disagreement bound $\Gamma^2 = 4\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2 \mathcal{H}_{\mathrm{TV}}$.

Each neighborhood $\mathcal{N}_{\mathbf{R}_Q}(W^r)$ is a product of convex balls. Since it contains all local iterates in round r, it also contains their weighted aggregate and the segments used in the smoothness argument.

Substituting these constants into (164) yields the desired inequality in Corollary C.8 under the specified step size conditions. □

C.4 Corollary C.9: FedProx Multi-Layer SSM Convergence and Proof

Corollary C.9 (FedProx convergence for multi-layer selective SSMs)**.** *Consider the distributed learning framework established in Section 3, with objective*

$$f(W) := \sum_{k=1}^{K} p_k \mathcal{L}_k(W), \qquad p_k \geq 0, \qquad \sum_{k=1}^{K} p_k = 1,$$

where the weights p_k are fixed and $W = \left(W^{(1)}, \ldots, W^{(Q)}\right)$ parameterizes the selective SSM of finite integer depth $Q \geq 1$ defined in Appendix C.1.

Each client objective is finite and differentiable on the neighborhoods under consideration and satisfies

$$\nabla_W \mathcal{L}_k(W) = \mathbb{E}_{\xi \sim \mathcal{P}_k}\left[\nabla_W \mathcal{L}^{S,(Q)}(W;\xi)\right],$$

where S is the length of the sampled sequence pair ξ, and all clients use the same sequence-loss map. Let W^0 be deterministic with $f(W^0) < \infty$, and suppose that f is bounded below by a finite constant f^*.

Consider full-participation FedProx for T communication rounds, with E local steps per round, mini-batch size Ξ, constant deterministic local step size $\eta > 0$, and constant deterministic proximal coefficient $\mu > 0$, where $T, E, \Xi \in \mathbb{Z}_{>0}$:

$$\begin{aligned} W_k^{r,0} &= W^r, \\ W_k^{r,e+1} &= W_k^{r,e} - \eta\left(g_{k,\Xi}^{r,e} + \mu(W_k^{r,e} - W^r)\right), \qquad e = 0, \ldots, E-1, \\ W^{r+1} &= \sum_{k=1}^{K} p_k W_k^{r,E}, \qquad r = 0, \ldots, T-1. \end{aligned}$$

The mini-batch gradient is

$$g_{k,\Xi}^{r,e} := \frac{1}{\Xi}\sum_{i=1}^{\Xi} \nabla_W \mathcal{L}^{S_i,(Q)}\left(W_k^{r,e}; \xi_{k,i}^{r,e}\right),$$

where S_i denotes the length of $\xi_{k,i}^{r,e}$. Conditional on the complete training history before each local step, each client draws Ξ fresh independent samples from $\mathcal{P}_k$, and the mini-batches drawn by different clients are conditionally independent.

For fixed deterministic radii $R_q > 0$, define

$$\mathcal{N}_{\mathbf{R}_Q}(W^r) := \prod_{q=1}^{Q} \mathcal{N}_{R_q}^{(q)}(W^{r,(q)}), \qquad \mathbf{R}_Q := (R_1, \ldots, R_Q).$$

Suppose that, almost surely, for every round $r < T$, client k, and local step $0 \le e \le E$,

$$W_k^{r,e} \in \mathcal{N}_{\mathbf{R}_Q}(W^r) \subseteq \mathcal{S}_{\boldsymbol{\gamma},\boldsymbol{\delta}}^{(Q)}.$$

Suppose also that finite deterministic envelopes $M_{X,q}$ satisfy, almost surely for every round,

$$\left\| W_X^{r,(q)} \right\|_2 + R_q \le M_{X,q}, \qquad X \in \{\Delta, B, C, A\}, \qquad q = 1, \dots, Q,$$

where $W_A^{r,(q)} := A^{r,(q)}$.

Let Assumption C.1 hold uniformly on all these neighborhoods, and let Assumption 4.2 hold uniformly at the final-layer predictions. The stability margins $\gamma_q, \delta_q > 0$, layer-input bounds $L_{x,q}$, loss-regularity constants L_ℓ, H_ℓ, and parameter envelopes $M_{X,q}$ are deterministic and common across rounds, admissible sequence pairs, finite sequence lengths, and token positions.

For each layer, define

$$\rho_q := \exp(-\gamma_q \delta_q) < 1.$$

Let $G_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}$ be the common gradient bound defined in Corollary (C.8)*, constructed using the original envelopes $M_{X,q}$.*

Assume $0 < \eta\mu \le 1$, and define

$$R_{\mu,q} := \min\left\{ R_q, \frac{G_{\mathrm{SSM},\mathbf{R}_Q}^{(Q)}}{\mu}\left[1 - (1-\eta\mu)^E\right] \right\}, \qquad q = 1, \dots, Q,$$

$$\mathbf{R}_{Q,\mu} := (R_{\mu,1}, \dots, R_{\mu,Q}),$$

$$M_{X,q}^{\mu} := M_{X,q} - R_q + R_{\mu,q}, \qquad X \in \{\Delta, B, C, A\}.$$

Using these reduced parameter envelopes, define

$$\begin{aligned}
M_{K_\Delta,q}^{\mu} &:= M_{\Delta,q}^{\mu} L_{x,q} + \sqrt{D}\log 2,\\
M_{K_B,q}^{\mu} &:= M_{B,q}^{\mu} L_{x,q}^2,\\
M_{K_W,q}^{\mu} &:= NDM_{K_B,q}^{\mu} M_{K_\Delta,q}^{\mu}\left(M_{K_\Delta,q}^{\mu} + \sqrt{D}M_{A,q}^{\mu}L_{x,q}\right)\frac{\rho_q}{1-\rho_q}\\
&\quad + L_{x,q}^2\sqrt{N}M_{K_\Delta,q}^{\mu} + L_{x,q}\sqrt{D}M_{K_B,q}^{\mu},\\
H_{R_{\mu,q}} &:= \frac{M_{K_B,q}^{\mu} M_{K_\Delta,q}^{\mu}\sqrt{ND}}{1-\rho_q},\\
J_{R_{\mu,q}} &:= \frac{M_{K_W,q}^{\mu}}{1-\rho_q},\\
M_{\Lambda_x,q}^{\mu} &:= \rho_q M_{A,q}^{\mu} M_{\Delta,q}^{\mu} H_{R_{\mu,q}} + M_{B,q}^{\mu} L_{x,q}\left(2M_{K_\Delta,q}^{\mu} + L_{x,q}M_{\Delta,q}^{\mu}\right),\\
M_{\kappa,q}^{\mu} &:= M_{C,q}^{\mu}\left(H_{R_{\mu,q}} + \frac{L_{x,q}M_{\Lambda_x,q}^{\mu}}{1-\rho_q}\right).
\end{aligned}$$

Initialize $\Theta_0^{\mu} = \Omega_0^{\mu} = 0$, and recursively define

$$\Theta_q^{\mu} := \left[L_{x,q}^2\left(H_{R_{\mu,q}} + M_{C,q}^{\mu}J_{R_{\mu,q}}\right)^2 + (M_{\kappa,q}^{\mu})^2(\Theta_{q-1}^{\mu})^2\right]^{1/2},$$

$$\begin{aligned}
\mathcal{E}_q^{\mu} := \rho_q H_{R_{\mu,q}}\Bigg[&\left(M_{K_\Delta,q}^{\mu} + M_{A,q}^{\mu}\left(L_{x,q} + M_{\Delta,q}^{\mu}\Theta_{q-1}^{\mu}\right)\right)^2\\
&+ 2\left(L_{x,q} + M_{\Delta,q}^{\mu}\Theta_{q-1}^{\mu}\right) + M_{A,q}^{\mu}\left(\frac{1}{4}\left(L_{x,q} + M_{\Delta,q}^{\mu}\Theta_{q-1}^{\mu}\right)^2 + 2\Theta_{q-1}^{\mu}\right)\Bigg]
\end{aligned}$$

$$
\begin{aligned}
&+ 2\rho_q \left(M^{\mu}_{K_\Delta,q} + M^{\mu}_{A,q}\left(L_{x,q} + M^{\mu}_{\Delta,q}\Theta^{\mu}_{q-1}\right)\right)\left(J_{R_{\mu,q}} + \frac{M^{\mu}_{\Lambda_x,q}\Theta^{\mu}_{q-1}}{1-\rho_q}\right)\\
&+ 2\Theta^{\mu}_{q-1}M^{\mu}_{K_\Delta,q}L_{x,q}\\
&+ 2\left(L_{x,q} + M^{\mu}_{B,q}\Theta^{\mu}_{q-1}\right)\left[L_{x,q}\left(L_{x,q} + M^{\mu}_{\Delta,q}\Theta^{\mu}_{q-1}\right) + M^{\mu}_{K_\Delta,q}\Theta^{\mu}_{q-1}\right]\\
&+ M^{\mu}_{B,q}L_{x,q}\Bigg[L_{x,q}\left(\frac{1}{4}\left(L_{x,q} + M^{\mu}_{\Delta,q}\Theta^{\mu}_{q-1}\right)^2 + 2\Theta^{\mu}_{q-1}\right)\\
&\qquad + 2\left(L_{x,q} + M^{\mu}_{\Delta,q}\Theta^{\mu}_{q-1}\right)\Theta^{\mu}_{q-1}\Bigg],
\end{aligned}
$$

$$
\begin{aligned}
\Omega^{\mu}_{q} :=\ & \frac{M^{\mu}_{C,q}L_{x,q}\mathcal{E}^{\mu}_{q}}{1-\rho_q}\\
&+ 2\left(L_{x,q} + M^{\mu}_{C,q}\Theta^{\mu}_{q-1}\right)\left(J_{R_{\mu,q}} + \frac{M^{\mu}_{\Lambda_x,q}\Theta^{\mu}_{q-1}}{1-\rho_q}\right)\\
&+ 2H_{R_{\mu,q}}\Theta^{\mu}_{q-1} + M^{\mu}_{\kappa,q}\Omega^{\mu}_{q-1}.
\end{aligned}
$$

Let

$$
\begin{aligned}
L^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}} &:= H_\ell(\Theta^{\mu}_{Q})^2 + L_\ell\Omega^{\mu}_{Q},\\
L^{(Q)}_{\mathrm{prox},\mathbf{R}_{Q,\mu}} &:= L^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}} + \mu,\\
G^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}} &:= L_\ell\Theta^{\mu}_{Q}\\
&= L_\ell\left[\sum_{q=1}^{Q} L^2_{x,q}\left(H_{R_{\mu,q}} + M^{\mu}_{C,q}J_{R_{\mu,q}}\right)^2 \prod_{q'=q+1}^{Q}(M^{\mu}_{\kappa,q'})^2\right]^{1/2},
\end{aligned}
$$

where empty products equal one. These are common deterministic smoothness and gradient bounds on the reduced neighborhoods

$$
\mathcal{N}_{\mathbf{R}_{Q,\mu}}(W^r) := \prod_{q=1}^{Q}\mathcal{N}^{(q)}_{R_{\mu,q}}(W^{r,(q)}).
$$

For the client distributions, define

$$
\bar{\mathcal{P}} := \sum_{k=1}^{K} p_k\mathcal{P}_k, \qquad \mathcal{H}_{\mathrm{TV}} := \sum_{k=1}^{K} p_k\,\mathrm{TV}(\mathcal{P}_k,\bar{\mathcal{P}})^2,
$$

where $\mathrm{TV}(P,Q) := \sup_{\mathcal{A}}|P(\mathcal{A}) - Q(\mathcal{A})|$*, with the supremum taken over measurable sets* $\mathcal{A}$*.*

If the step size satisfies

$$
\eta E L^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}} \le \frac{1}{2}, \qquad 12\eta^2\left(L^{(Q)}_{\mathrm{prox},\mathbf{R}_{Q,\mu}}\right)^2 E(E-1) \le 1,
$$

then

$$
\begin{aligned}
\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\|\nabla f(W^r)\|_F^2 \le\ & \frac{3(f(W^0) - f^*)}{\eta E T}\\
&+ \frac{3\eta L^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}}\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}}\right)^2}{\Xi}\sum_{k=1}^{K}p_k^2\\
&+ \frac{3\eta^2\left(L^{(Q)}_{\mathrm{prox},\mathbf{R}_{Q,\mu}}\right)^2(E-1)\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}}\right)^2}{\Xi}
\end{aligned}
$$

$$+ 24\eta^2 \left(L^{(Q)}_{\mathrm{prox},\mathbf{R}_{Q,\mu}}\right)^2 E(E-1)\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}}\right)^2 \mathcal{H}_{\mathrm{TV}}.$$

Moreover, almost surely, for every round $r < T$, client k, layer q, and local step $0 \le e \le E$,

$$\left\|W_k^{r,e,(q)} - W^{r,(q)}\right\|_F \le \min\left\{R_q, \frac{G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}}{\mu}\left[1-(1-\eta\mu)^e\right]\right\} \le R_{\mu,q}.$$

Consequently, all local iterates and the aggregated model W^{r+1} lie in

$$\mathcal{N}_{\mathbf{R}_{Q,\mu}}(W^r) \subseteq \mathcal{N}_{\mathbf{R}_Q}(W^r) \subseteq \mathcal{S}^{(Q)}_{\boldsymbol{\gamma},\boldsymbol{\delta}}.$$

Proof of Corollary C.9: We apply the federated optimization argument from the proof of Theorem B.6 to the full parameter tuple $W = \left(W^{(1)}, \ldots, W^{(Q)}\right)$. After vectorization, the Euclidean norm of this tuple is exactly its block Frobenius norm.

First, the pointwise multi-layer gradient bound implies $\left\|g^{r,e}_{k,\Xi}\right\|_F \le G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}$, which also bounds each layer block of the mini-batch gradient. Applying the displacement recursion from the single-layer FedProx proof separately to each layer, and using $0 < \eta\mu \le 1$, gives

$$\left\|W_k^{r,e,(q)} - W^{r,(q)}\right\|_F \le \frac{G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}}{\mu}\left[1-(1-\eta\mu)^e\right]. \tag{403}$$

Combining this with the assumed containment in the original neighborhood yields the stated bound by $R_{\mu,q}$.

Consequently, all local iterates lie in $\mathcal{N}_{\mathbf{R}_{Q,\mu}}(W^r)$. This neighborhood is a product of convex balls, so it also contains their weighted aggregate and the segments used in the smoothness argument.

For every $W \in \mathcal{N}_{\mathbf{R}_{Q,\mu}}(W^r)$,

$$\left\|W_X^{(q)}\right\|_2 \le \left\|W_X^{r,(q)}\right\|_2 + R_{\mu,q} \le M_{X,q} - R_q + R_{\mu,q} = M^{\mu}_{X,q}.$$

The preceding neighborhood estimates therefore apply with the common deterministic envelopes $M^{\mu}_{X,q}$, because their defining expressions are nondecreasing in those envelopes.

Lemma C.4 supplies the common smoothness constant $L^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}}$ for each client objective and for f. Adding the proximal penalty makes each local proximal objective $L^{(Q)}_{\mathrm{prox},\mathbf{R}_{Q,\mu}}$-smooth, where

$$L^{(Q)}_{\mathrm{prox},\mathbf{R}_{Q,\mu}} = L^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}} + \mu.$$

Lemma C.6, applied conditionally on the complete history before each mini-batch, supplies

$$\sigma^2 = \frac{\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}}\right)^2}{\Xi}.$$

The proximal term is deterministic under this conditioning and does not alter the gradient-noise variance. Conditional independence across clients gives the corresponding weighted-noise variance bound $\sigma^2 \sum_k p_k^2$.

Lemma C.7 supplies the common client-gradient disagreement bound

$$\Gamma^2 = 4\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}}\right)^2 \mathcal{H}_{\mathrm{TV}}.$$

The common proximal term cancels when comparing client gradients with their weighted average at the same parameter tuple.

The remaining local-drift and descent calculations in the proof of Theorem B.6 depend only on these smoothness, variance, and disagreement bounds, the conditional sampling assumptions, and convexity of the neighborhoods. Applying those calculations with the multi-layer constants above under the state learning-rate conditions and substituting the defined values of σ^2 and Γ^2 yields the desired inequality in Corollary C.9, completing the proof. □

D SUPPLEMENTARY COROLLARIES

In this section, we provide supplementary corollaries to the theorems presented in the preceding sections and in the main manuscript. These corollaries simplify the preceding results and make explicit their dependence on SSM-specific quantities, including stability, parameter norms, network depth, and related architectural properties.

D.1 SINGLE-LAYER CONVERGENCE RATE COROLLARIES

In this section, we provide convergence rate guarantees for FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020) learning frameworks. Every convergence corollary retains the assumptions of its referenced result. Furthermore, we establish and retain the following conventions:

$$F_0 := f(W^0) - f^*, \tag{404}$$

$$P_2 := \sum_{k=1}^{K} p_k^2, \tag{405}$$

$$\mathcal{H}_{\mathrm{TV}} := \sum_k p_k \,\mathrm{TV}(\mathcal{P}_k, \bar{\mathcal{P}})^2 \tag{406}$$

All $\mathcal{O}$ expressions denote upper bounds on nonnegative quantities.

Note that Corollary 4.8 is stated in Section 4, with this section containing its proof.

D.1.1 COROLLARY 4.8: SINGLE-LAYER FEDAVG CONVERGENCE RATE PROOF

Proof of Corollary 4.8:

By Lemmas 4.7 and B.2:

$$G_{\mathrm{SSM},R} = \mathcal{O}\left(\frac{M_R}{(1-\rho)^2}\right), \qquad L_{\mathrm{SSM},R} = \mathcal{O}\left(\frac{M_R^2}{(1-\rho)^4}\right) \tag{407}$$

Substituting these bounds into Theorem B.5, collecting the two local-drift terms, and using the fact that $P_2 \le 1$, yields the inequality in Corollary 4.8. □

D.1.2 COROLLARY D.1: SINGLE-LAYER FEDPROX CONVERGENCE RATE AND PROOF

Corollary D.1 (Single-layer FedProx convergence rate)**.** *Under the assumptions and step-size conditions of Theorem B.6, given its reduced envelopes $M_X^\mu = M_X - R + R_\mu$, define:*

$$M_{R_\mu} := (1 + M_B^\mu)(1 + M_C^\mu)(1 + M_\Delta^\mu)(1 + M_\Delta^\mu + M_A^\mu).$$

Then

$$\frac{1}{T}\sum_{r=0}^{T-1} \mathbb{E}\,\|\nabla f(W^r)\|_F^2 = \mathcal{O}\left(\frac{F_0}{\eta E T} + \frac{\eta M_{R_\mu}^4}{\Xi(1-\rho)^8} + \eta^2 \frac{M_{R_\mu}^2}{(1-\rho)^4}\left(\frac{M_{R_\mu}^2}{(1-\rho)^4} + \mu\right)^2 (E-1)\left(\frac{1}{\Xi} + E\mathcal{H}_{\mathrm{TV}}\right)\right). \tag{408}$$

The implicit constant has the same dependencies as in Corollary 4.8 and is independent of μ.

Proof of Corollary D.1: The single-layer scaling argument, evaluated with M_X^μ, gives $G_{\mathrm{SSM},R_\mu} = \mathcal{O}\left(\frac{M_{R_\mu}}{(1-\rho)^2}\right)$ and $L_{\mathrm{SSM},R_\mu} = \mathcal{O}\left(\frac{M_{R_\mu}^2}{(1-\rho)^4}\right)$.

Consequently $L_{\mathrm{prox},R_\mu} = \mathcal{O}\left(\frac{M_{R_\mu}^2}{(1-\rho)^4} + \mu\right)$. Substituting these results into Theorem B.6 and using the fact that $P_2 \le 1$ proves the bound. □

Note that although $\frac{M_{R_\mu}}{(1-\rho)^2} \le \frac{M_R}{(1-\rho)^2}$ for the same original envelopes, the bound in Corollary D.1 also contains additional μ terms. As such, these bounds expose a trade-off and do not imply that every positive proximal coefficient improves the convergence guarantee.

D.2 Multi-Layer SSM Gradient Scaling Bounds

In this section we provide simplified bounds on the multi-layer gradient, where all per-layer quantities are scaled to a single, common value. For the remainder of this section, we will utilize the following notation: for a fixed model in Theorem C.3, let

$$\begin{aligned} L_x &:= \max_{1\le q\le Q} L_{x,q}, \qquad \rho := \max_{1\le q\le Q} \rho_q < 1, \\ M_X &:= \max_{1\le q\le Q} \left\| W_X^{(q)} \right\|_2, \qquad X \in \{\Delta, B, C, A\}, \end{aligned} \tag{409}$$

where $W_A^{(q)} := A^{(q)}$.

For neighborhood-uniform statements, take $M_X := \max_q M_{X,q}$ using the deterministic envelopes in the multi-layer convergence results. The bounds on L_x and ρ are then uniform on those neighborhoods as well. Note that common upper bounds suffice - the layers do not need to have identical weights or identical stability factors.

Using these common quantities, define

$$M_{K_\Delta} := M_\Delta L_x + \sqrt{D}\log 2, \tag{410}$$

$$M_{K_B} := M_B L_x^2, \tag{411}$$

$$\begin{aligned} M_{K_W} := {} & NDM_{K_B} M_{K_\Delta}(M_{K_\Delta} + \sqrt{D} M_A L_x)\frac{\rho}{1-\rho} \\ & + L_x^2\sqrt{N} M_{K_\Delta} + L_x\sqrt{D} M_{K_B}, \end{aligned} \tag{412}$$

$$H := \frac{M_{K_B} M_{K_\Delta}\sqrt{ND}}{1-\rho}, \tag{413}$$

$$J := \frac{M_{K_W}}{1-\rho}, \tag{414}$$

$$\Lambda := \rho M_A M_\Delta H + M_B L_x(2M_{K_\Delta} + L_x M_\Delta), \tag{415}$$

$$\kappa := M_C\left(H + \frac{L_x\Lambda}{1-\rho}\right). \tag{416}$$

D.2.1 Corollary D.2: Multi-layer gradient under common layer bounds and Proof

Corollary D.2 (Multi-layer gradient under common layer bounds)**.** *Under the assumptions of Theorem C.3, the quantities in* (410)-(416) *satisfy, for every admissible sequence pair ξ of finite length S,*

$$\left\| \nabla_W \mathcal{L}^{S,(Q)}(W;\xi) \right\|_F \le L_\ell L_x (H + M_C J)\sqrt{\sum_{q=0}^{Q-1} \kappa^{2q}}.$$

Furthermore, the following inequalities hold for different values of κ:

$$\left\| \nabla_W \mathcal{L}^{S,(Q)}(W;\xi) \right\|_F \le L_\ell L_x (H + M_C J) \begin{cases} \frac{1}{\sqrt{1-\kappa^2}}, & 0 \le \kappa < 1, \\ \sqrt{Q}, & \kappa = 1, \\ \sqrt{(\kappa^{2Q}-1)/(\kappa^2-1)}, & \kappa > 1. \end{cases}$$

Proof of Corollary D.2: All defining expressions 410- 416 are nondecreasing in the nonnegative parameter, input, and stability upper bounds. As a result, each own-layer factor in Theorem C.3 is at most $L_x(H + M_C J)$, and each downstream sensitivity is at most κ. Substitution into the inequality from Theorem C.3 gives the right-hand side bounded by $L_\ell L_x(H + M_C J)\left(\sum_{q=1}^{Q} \kappa^{2(Q-q)}\right)^{1/2}$. Evaluating the geometric sum proves both claimed inequalities. □

For common envelopes independent of Q, the upper bound is uniform in depth when $\kappa < 1$, grows at most as $\sqrt{Q}$ when $\kappa = 1$, and can grow exponentially when $\kappa > 1$. Temporal stability $\rho < 1$ alone does not imply $\kappa < 1$, and an exponentially growing upper bound does not establish that actual gradients grow exponentially.

D.2.2 Corollary D.3: Multi-layer gradient rate under common layer bounds and Proof

Corollary D.3 (Multi-layer gradient rate under common layer bounds)**.** *Under the assumptions of Theorem C.3, define*

$$\mathfrak{M}_q := \left(1 + \|W_B^{(q)}\|_2\right)\left(1 + \|W_C^{(q)}\|_2\right)\left(1 + \|W_\Delta^{(q)}\|_2\right)\left(1 + \|W_\Delta^{(q)}\|_2 + \|A^{(q)}\|_2\right).$$

Then, given the κ_q defined in that theorem, it holds that:

$$\left\|\nabla_W \mathcal{L}^{S,(Q)}(W;\xi)\right\|_F = \mathcal{O}\left[\left(\sum_{q=1}^{Q} \frac{\mathfrak{M}_q^2}{(1-\rho_q)^4} \prod_{j=q+1}^{Q} \kappa_j^2\right)^{1/2}\right].$$

Furthermore, with the common envelopes defined above, let $\mathfrak{M} := (1+M_B)(1+M_C)(1+M_\Delta)(1+M_\Delta+M_A)$. Then the following holds:

$$\begin{aligned}\left\|\nabla_W \mathcal{L}^{S,(Q)}(W;\xi)\right\|_F &= \mathcal{O}\left(\frac{\mathfrak{M}}{(1-\rho)^2}\sqrt{\sum_{q=0}^{Q-1}\kappa^{2q}}\right) \\ &= \mathcal{O}\left(\frac{\mathfrak{M}}{(1-\rho)^2}\begin{cases}\frac{1}{\sqrt{1-\kappa^2}}, & 0 \le \kappa < 1, \\ \sqrt{Q}, & \kappa = 1, \\ \sqrt{(\kappa^{2Q}-1)/(\kappa^2-1)}, & \kappa > 1.\end{cases}\right).\end{aligned}$$

The implicit constants depend only on N, D, L_x, L_ℓ, where $L_x \ge \max_q L_{x,q}$, and are independent of Q, sequence length, parameter norms, and stability factors.

Proof of Corollary D.3: The single-layer estimates give $L_{x,q}\left(H_q + \left\|W_C^{(q)}\right\|_2 J_q\right) \le C\mathfrak{M}_q/(1-\rho_q)^2$ for some common constant C. Substituting this into Theorem C.3 we obtain the first equality of in Corollary D.3.

Additionally, substituting $L_x(H + M_C J) = \mathcal{O}\left(\mathfrak{M}/(1-\rho)^2\right)$ into Corollary D.2 proves the second equality. □

D.3 Multi-layer convergence under common layer bounds

In the following corollaries and their proof, we utilize the neighborhood-uniform common envelopes defined in (409) and 410- 416, with the same L_ℓ, H_ℓ throughout all considered neighborhoods. The following reduction replaces the original layer-dependent sensitivity recursions by three finite geometric sums.

We further define the common auxiliary constants

$$d_* := L_x + M_\Delta, \tag{417}$$

$$b_* := \frac{1}{4}d_*^2 + 2, \tag{418}$$

$$c_* := M_{K_\Delta} + M_A d_*, \tag{419}$$

$$j_* := J + \frac{\Lambda}{1-\rho}, \tag{420}$$

$$\begin{aligned}\mathcal{E}_* := {} & \rho H(c_*^2 + 2d_* + M_A b_*) + 2\rho c_* j_* + 2M_{K_\Delta}L_x \\ & + 2(L_x + M_B)(L_x d_* + M_{K_\Delta}) + M_B L_x(L_x b_* + 2d_*),\end{aligned} \tag{421}$$

$$B_* := \frac{M_C L_x \mathcal{E}_*}{1-\rho} + 2(L_x + M_C)j_* + 2H. \tag{422}$$

For $Q \ge 1$, define

$$\widehat{G}_Q := L_\ell L_x(H + M_C J)\sqrt{\sum_{q=0}^{Q-1}\kappa^{2q}}, \tag{423}$$

$$\widehat{L}_Q := H_\ell L_x^2 (H + M_C J)^2 \sum_{q=0}^{Q-1} \kappa^{2q} + L_\ell B_* \left[\sum_{q=0}^{Q-1} \kappa^q + L_x^2 (H + M_C J)^2 \sum_{q=1}^{Q} \kappa^{Q-q} \left(\sum_{j=0}^{q-1} \kappa^{2j} \right) \right]. \tag{424}$$

In particular, for $\kappa \neq 1$,

$$\sum_{q=0}^{Q-1} \kappa^q = \frac{1-\kappa^Q}{1-\kappa}, \qquad \sum_{q=1}^{Q} \kappa^{Q-q} \left(\sum_{j=0}^{q-1} \kappa^{2j} \right) = \frac{(1-\kappa^Q)(1-\kappa^{Q+1})}{(1-\kappa)(1-\kappa^2)}, \qquad \kappa \neq 1. \tag{425}$$

For $\kappa = 1$, the two sums reduce to Q and $Q(Q+1)/2$, respectively.

D.3.1 Corollary D.4: Common-layer smoothness and depth dependence and Proof

Corollary D.4 (Common-layer smoothness and depth dependence)**.** *Under the neighborhood assumptions of Lemma C.4, the constants $G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}$ and $L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}$ defined in Corollary C.8 satisfy:*

$$G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \leq \widehat{G}_Q, \qquad L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \leq \widehat{L}_Q.$$

If all common envelopes and regularity constants are fixed independently of Q, their depth dependence is

$$\left(\widehat{G}_Q, \widehat{L}_Q\right) = \begin{cases} (\mathcal{O}(1), \mathcal{O}(1)), & 0 \leq \kappa < 1, \\ \left(\mathcal{O}(\sqrt{Q}), \mathcal{O}(Q^2)\right), & \kappa = 1, \\ \left(\mathcal{O}(\kappa^Q), \mathcal{O}(\kappa^{2Q})\right), & \kappa > 1. \end{cases}$$

Here only Q varies: the implicit constants may depend on the fixed common envelopes and on κ.

Proof of Corollary D.4: After replacing the layerwise quantities by their common upper bounds, the first-sensitivity recursion yields, for every $q \geq 1$,

$$\Theta_q^2 \leq L_x^2 (H + M_C J)^2 \sum_{j=0}^{q-1} \kappa^{2j}. \tag{426}$$

Hence,

$$G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} = L_\ell \Theta_Q \leq L_\ell L_x (H + M_C J) \sqrt{\sum_{j=0}^{Q-1} \kappa^{2j}} = \widehat{G}_Q. \tag{427}$$

For the second-sensitivity recursion, the terms independent of Ω_{q-1} are bounded by

$$B_* \left(1 + \Theta_{q-1}^2\right), \tag{428}$$

This follows because, after replacing the layerwise quantities by their common upper bounds, the terms independent of Ω_{q-1} form a polynomial of degree at most two in Θ_{q-1} with nonnegative coefficients. The sum of these coefficients is B_*, obtained by setting $\Theta_{q-1} = 1$. Since each of 1, Θ_{q-1}, and Θ_{q-1}^2 is at most $1 + \Theta_{q-1}^2$, the displayed bound follows. Therefore, unrolling the recursion gives

$$\begin{aligned} \Omega_Q &\leq B_* \sum_{q=1}^{Q} \kappa^{Q-q} \left(1 + \Theta_{q-1}^2\right) \\ &\leq B_* \left[\sum_{q=0}^{Q-1} \kappa^q + L_x^2 (H + M_C J)^2 \sum_{q=1}^{Q} \kappa^{Q-q} \left(\sum_{j=0}^{q-1} \kappa^{2j} \right) \right], \end{aligned} \tag{429}$$

where the first-sensitivity bound was used in the second inequality. Substituting the bounds on Θ_Q and Ω_Q into

$$L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} = H_\ell \Theta_Q^2 + L_\ell \Omega_Q, \tag{430}$$

we get:

$$L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \le \widehat{L}_Q. \tag{431}$$

Finally, evaluating the geometric sums gives

$$\left(\widehat{G}_Q, \widehat{L}_Q\right) = \begin{cases} \left(\mathcal{O}(1), \mathcal{O}(1)\right), & 0 \le \kappa < 1, \\ \left(\mathcal{O}(\sqrt{Q}), \mathcal{O}(Q^2)\right), & \kappa = 1, \\ \left(\mathcal{O}(\kappa^Q), \mathcal{O}(\kappa^{2Q})\right), & \kappa > 1, \end{cases} \tag{432}$$

which proves the stated depth dependence. □

D.3.2 Corollary D.5: Multi-layer FedAvg under common layer bounds and Proof

Corollary D.5 (Multi-layer FedAvg under common layer bounds)**.** *Under the assumptions of Corollary C.8, let $\widehat{G}_Q$ and $\widehat{L}_Q$ be the common upper bounds in* (423) *and* (424)*, evaluated with the neighborhood-uniform common envelopes introduced above. If*

$$\eta E \widehat{L}_Q \le \frac{1}{2}, \qquad 12\eta^2 \widehat{L}_Q^2 E(E-1) \le 1,$$

then

$$\frac{1}{T}\sum_{r=0}^{T-1} \mathbb{E}\left\|\nabla f(W^r)\right\|_F^2 = \mathcal{O}\Bigg(\frac{F_0}{\eta E T} + \frac{\eta \widehat{L}_Q \widehat{G}_Q^2 P_2}{\Xi} + \eta^2 \widehat{L}_Q^2 \widehat{G}_Q^2 (E-1)\left[\frac{1}{\Xi} + E\mathcal{H}_{\mathrm{TV}}\right]\Bigg).$$

The implicit constant is numerical. For common envelopes and loss-regularity constants fixed independently of Q, the depth dependence of $\widehat{G}_Q$ and $\widehat{L}_Q$ is explicit in (423) *and* (424) *and Corollary D.4.*

Proof of Corollary D.5: Corollary D.4 gives

$$G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \le \widehat{G}_Q, \qquad L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \le \widehat{L}_Q.$$

Consequently, the displayed step-size conditions imply those of Corollary C.8. Substituting these upper bounds into that corollary and grouping the two local-drift terms proves the stated claim. □

D.3.3 Corollary D.6: Multi-layer FedProx under common layer bounds and Proof

Corollary D.6 (Multi-layer FedProx under common layer bounds)**.** *Under the assumptions of Corollary C.9, retain its original radii R_q, effective radii $R_{\mu,q}$, and reduced envelopes $M^{\mu}_{X,q} = M_{X,q} - R_q + R_{\mu,q}$.*

For this corollary and its proof, evaluate the existing common bounds $\widehat{G}_Q$ and $\widehat{L}_Q$ in (423) *and* (424) *using*

$$L_x := \max_{1\le q\le Q} L_{x,q}, \qquad \rho := \max_{1\le q\le Q} \rho_q,$$
$$M_X := \max_{1\le q\le Q} M^{\mu}_{X,q}, \qquad X \in \{\Delta, B, C, A\}.$$

All auxiliary quantities in (410)-(416) *and* (417)-(422)*, including κ and B_*, are evaluated with these reduced common envelopes. The loss-regularity constants L_ℓ, H_ℓ are unchanged.*

If $0 < \eta\mu \le 1$ and

$$\eta E \widehat{L}_Q \le \frac{1}{2}, \qquad 12\eta^2 \left(\widehat{L}_Q + \mu\right)^2 E(E-1) \le 1,$$

then

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\left\|\nabla f(W^r)\right\|_F^2 = \mathcal{O}\Bigg(\frac{F_0}{\eta ET} + \frac{\eta\widehat{L}_Q\widehat{G}_Q^2 P_2}{\Xi} + \eta^2(\widehat{L}_Q+\mu)^2\widehat{G}_Q^2(E-1)\left[\frac{1}{\Xi}+E\mathcal{H}_{\mathrm{TV}}\right]\Bigg).$$

The implicit constant is numerical and independent of μ.

Proof of Corollary D.6: The displacement argument in the proof of Corollary C.9 places all local iterates in $\mathcal{N}_{\mathbf{R}_{Q,\mu}}(W^r)$. This product of convex balls also contains their weighted aggregate and the segments used in the smoothness argument.

Since $M_{X,q}^{\mu} \le M_X$ for every layer, the common-envelope construction of Corollary D.4 gives

$$G^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}} \le \widehat{G}_Q, \qquad L^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}} \le \widehat{L}_Q.$$

Therefore,

$$L^{(Q)}_{\mathrm{prox},\mathbf{R}_{Q,\mu}} = L^{(Q)}_{\mathrm{SSM},\mathbf{R}_{Q,\mu}} + \mu \le \widehat{L}_Q + \mu.$$

The displayed step-size conditions consequently imply those of Corollary C.9. Substituting these upper bounds into that corollary and grouping the two local-drift terms proves the stated claim. □

D.4 SSM Convergence under Distributed Learning

In this section we provide and prove Corollary D.7: SSM Convergence under Distributed Learning and the corresponding rate corollary - a setting where every client runs only one step of gradient descent between model weights exchange.

Corollary D.7 (SSM Convergence under Distributed Learning)**.** *Under the assumptions of Theorem B.5, let $E = 1$. If $\eta L_{\mathrm{SSM},R} \le 1/2$, then*

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\left\|\nabla f(W^r)\right\|_F^2 \le \frac{4F_0}{\eta T} + \frac{4\eta L_{\mathrm{SSM},R} G^2_{\mathrm{SSM},R} P_2}{\Xi}.$$

Similarly, under the assumptions of Corollary C.8 with $E = 1$, if $\eta L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q} \le 1/2$, then

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\left\|\nabla f(W^r)\right\|_F^2 \le \frac{4F_0}{\eta T} + \frac{4\eta L^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\left(G^{(Q)}_{\mathrm{SSM},\mathbf{R}_Q}\right)^2 P_2}{\Xi}.$$

Both bounds also apply to FedProx with one local step, under the same FedAvg assumptions and step-size conditions, for every $\mu > 0$. No restriction on $\eta\mu$ is needed for this one-step conclusion.

Proof of Corollary D.7: Setting $E = 1$ in Theorem B.5 or Corollary C.8 eliminates both terms containing $E - 1$ and makes the second step-size condition automatic.

For FedProx, initialization at the broadcast model gives

$$W_k^{r,1} = W^r - \eta\left(g_{k,\Xi}^{r,0} + \mu(W^r - W^r)\right) = W^r - \eta g_{k,\Xi}^{r,0}. \tag{433}$$

As such, for every $\mu > 0$, its one-step update and weighted aggregate coincide with those of FedAvg. Applying the corresponding FedAvg result proves the claim. □

Corollary D.8 (SSM Convergence Rate under Distributed Learning)**.** *Under the assumptions of Corollary 4.8, let $E = 1$. Under $\eta L_{\mathrm{SSM},R} \le 1/2$, let M_B, M_C, M_Δ, and M_A denote uniform upper bounds on $\|W_B\|_2, \|W_C\|_2, \|W_\Delta\|_2$, and $\|A\|_2$, respectively, over the R-local neighborhoods $\mathcal{N}_R(W^r)$ and define:*

$$M_R := (1+M_B)(1+M_C)(1+M_\Delta)(1+M_\Delta+M_A).$$

Under these conditions, the global optimization trajectory $\{W^r\}_{r=0}^T$ *satisfies*

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\left\|\nabla f(W^r)\right\|_F^2 = \mathcal{O}\left(\frac{F_0}{\eta E T} + \frac{\eta M_R^4}{\Xi(1-\rho)^8}\right).$$

The quantities N, D, L_x, L_ℓ, *and* H_ℓ *affect only the implicit constant and do not alter the displayed scaling with the optimization parameters, parameter envelopes, stability, or sequence length.*

Proof of Corollary D.8: Setting $E = 1$ in Corollary 4.8 eliminates both terms containing $E - 1$, yielding the equality in the Corollary. □

E Discussion on Additional Distributed Learning Frameworks

The FedAvg and FedProx results (Theorems B.5 and B.6) relate convergence of selective SSM training to recurrent stability, parameter norms, local client drift, and data heterogeneity. In this section, we use these quantities to interpret additional federated learning methods, identifying how their local updates and server aggregation interact with the SSM-specific bounds. While we do not establish convergence guarantees for these methods, the discussion provides a qualitative, theory-guided characterization of their convergence behavior, offering intuition and motivating expectations about their performance under different training conditions.

The discussion below assumes SGD-style updates, containment in the stated neighborhood, and regularity assumptions. The connection to the AdamW-based Mamba2 experiments is therefore qualitative rather than a direct application of the convergence results.

SCAFFOLD. SCAFFOLD (Karimireddy et al., 2020) introduces server and client control variates (correction terms), which act as drift-correction terms for local client updates. In its classical SGD form, the local update is

$$W_k^{r,e+1} = W_k^{r,e} - \eta\left(g_k(W_k^{r,e};\xi_k^{r,e}) + c^r - c_k^r\right), \tag{434}$$

where $c^r = \sum_k p_k c_k^r$ approximates the global update direction and c_k^r approximates the client-specific drift direction. More generally, allowing a correction scale $\lambda_c > 0$, define the corrected local direction

$$d_k^{r,e} := g_k(W_k^{r,e};\xi_k^{r,e}) + \lambda_c(c^r - c_k^r). \tag{435}$$

The correction term $c^r - c_k^r$ is intended to reduce the component of the local update that points toward the client-specific objective rather than the global objective.

Assume that the correction term satisfies:

$$\|c^r - c_k^r\|_F \le C_R, \tag{436}$$

then the corrected direction satisfies:

$$\|d_k^{r,e}\|_F \le G_{\mathrm{SSM},R} + \lambda_c C_R := G_{\mathrm{SCAF},R}, \tag{437}$$

and the corresponding local drift radius can be bounded by

$$R_{\mathrm{SCAF}} \le \eta E G_{\mathrm{SCAF},R}. \tag{438}$$

While this mechanism does not directly change the SSM smoothness constant $L_{\mathrm{SSM},R}$, and could even increase the worst-case local drift radius through the additional correction term, the main effect of SCAFFOLD is to reduce the component of each client update that is biased toward the local objective rather than the global objective. In other words, SCAFFOLD can be interpreted as modifying the effective heterogeneity term. Specifically, define the corrected heterogeneity measure

$$\Gamma^2_{\mathrm{SCAF},R} := \sup_{W\in\mathcal{N}_R(W^r)}\sum_{k=1}^K p_k\left\|\nabla\mathcal{L}_k(W) - \nabla f(W) + \lambda_c(c^r - c_k^r)\right\|_F^2. \tag{439}$$

When the control terms accurately estimate the client-specific drift directions, the correction $\lambda_c(c^r - c_k^r)$ partially cancels the difference $\nabla\mathcal{L}_k(W) - \nabla f(W)$. In this case, one expects

$$\Gamma^2_{\mathrm{SCAF},R} < \Gamma^2_{\mathrm{SSM},R}. \tag{440}$$

This motivates a potentially reduced client disagreement, although a convergence bound must also account for the evolution and accuracy of the control variates.

This does not imply that SCAFFOLD has a smaller drift radius than FedAvg. In fact, in nearly IID settings, the uncorrected client gradients are already approximately aligned with the global update direction. As a result, the reduction from $\Gamma^2_{\mathrm{SSM},R}$ to $\Gamma^2_{\mathrm{SCAF},R}$ may be limited, while the additional correction term can increase the effective update radius and potentially degrade performance. In contrast, under non-IID data, the heterogeneity correction can provide a meaningful benefit by reducing the statistical drift term, which is amplified in the SSM-specific bound through the architecture-dependent gradient and smoothness quantities.

FedAdam. FedAdam (Reddi et al., 2021) keeps the same FedAvg-style local client training phase, but replaces the direct server averaging step with an adaptive server optimizer. Let

$$\bar{W}^{r+1} = \sum_{k=1}^{K} p_k W_k^{r,E} \tag{441}$$

denote the standard FedAvg aggregated parameters, and define the server pseudo-gradient:

$$\Delta^r := \bar{W}^{r+1} - W^r. \tag{442}$$

FedAdam tracks moving averages of the server pseudo-gradient and its coordinatewise square as:

$$m^r = \beta_1 m^{r-1} + (1-\beta_1)\Delta^r, \tag{443}$$

$$v^r = \beta_2 v^{r-1} + (1-\beta_2)(\Delta^r \odot \Delta^r), \tag{444}$$

and updates the server model coordinatewise as

$$W^{r+1} = W^r + \alpha_s \frac{m^r}{\sqrt{v^r} + \tau_s}, \tag{445}$$

where α_s is the server learning rate and $\tau_s > 0$ is the denominator offset.

Since the client-side training phase is unchanged from FedAvg, FedAdam does not directly reduce the local SSM drift radius. The local client trajectories remain controlled by the same worst-case bound

$$R_{\mathrm{local}} \le \eta E G_{\mathrm{SSM},R}. \tag{446}$$

The main change is instead introduced at the server: since $\sqrt{v^r} + \tau_s \ge \tau_s$, the server update satisfies the coarse bound

$$\|W^{r+1} - W^r\|_F \le \frac{\alpha_s}{\tau_s} \|m^r\|_F. \tag{447}$$

As such, even though the adaptive server mechanism can be beneficial when it rescales poorly conditioned coordinates in a useful way, it can also increase the effective global update radius when $\frac{\alpha_s}{\tau_s}$ is large or when the moment estimates are not well aligned with a stable descent direction.

This can be particularly detrimental for selective SSMs - an aggressive server-side adaptive step can increase the effective neighborhood radius, resulting in larger potential values of $L_{\mathrm{SSM},R}$ and $G_{\mathrm{SSM},R}$, and making the global trajectory more sensitive to non-IID client updates. Compared to FedProx, FedAdam does not explicitly constrain local client drift, and compared to SCAFFOLD, FedAdam does not correct the client heterogeneity term. Instead, it rescales the aggregated update after local drift has already occurred.

FedAlign. The FedAlign baseline we consider is an adaptation of the state-alignment method proposed in (Keçeci et al., 2025). The original FedAlign algorithm was developed for federated system identification with classical linear state-space models of the form

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}u_k, \qquad \tilde{y}_k = \tilde{C}\tilde{x}_k + \tilde{D}u_k \tag{448}$$

where the system matrices $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ are fixed model parameters. In that setting, the main alignment issue is that different clients may learn dynamically equivalent SSMs in different state-coordinate bases, so directly averaging their parameters can distort the global system dynamics. FedAlign addresses this by using similarity transformations to map local SSMs into a common parameter basin before aggregation.

Our setting differs from this original SYSID formulation because Mamba-style selective SSMs do not have a single fixed set of state-space matrices applied uniformly across the sequence. Instead, the effective transition and input terms are coupled to learned projections and input-dependent discretization quantities. Therefore, our FedAlign implementation adapts the same state-alignment principle to Mamba2 by aligning coupled SSM state, group, and head-indexed parameter blocks before aggregation.

As such, we can write the adapted FedAlign aggregation as

$$W^{r+1} = \sum_{k=1}^{K} p_k \, \mathrm{Align}_k(W_k^{r,E}), \tag{449}$$

where Align_k is the client-specific alignment map applied to the coupled Mamba2 SSM blocks.

Next, as long as Align_k is a norm-preserving permutation or matching of coupled coordinates, then

$$\left\| \mathrm{Align}_k(W_k^{r,E}) - \mathrm{Align}_k(W^r) \right\|_F = \left\| W_k^{r,E} - W^r \right\|_F. \tag{450}$$

The identity preserves distance when the same alignment is applied to both models, not distance to the fixed broadcast model W^r. FedAlign can reduce aggregation error from mismatched state coordinates, but norm preservation alone does not control local drift, statistical heterogeneity, or stability-sensitive gradient growth in selective SSMs. FedAvg-like behavior is therefore expected when coordinate mismatch is limited. Containment of aligned models requires separate verification.

FedDyn. FedDyn (Acar et al., 2021) combines proximal regularization with a dynamic gradient correction. In the full-participation, weighted formulation, client k maintains a state ν_k^r, initialized at zero, and optimizes

$$g_k^r(W) := \mathcal{L}_k(W) - \langle \nu_k^r, W \rangle_F + \frac{\alpha}{2} \|W - W^r\|_F^2, \qquad \nu_k^{r+1} := \nu_k^r - \alpha(W_k^{r,E} - W^r), \tag{451}$$

where $\alpha > 0$. Writing $\nu^r := \sum_k p_k \nu_k^r$ and $\bar{W}^{r+1} := \sum_k p_k W_k^{r,E}$, the server updates are $W^{r+1} = \bar{W}^{r+1} - \nu^{r+1}/\alpha$.

For local SGD, under the stable-neighborhood conditions of Theorem B.6, given the bound $\|\nu_k^r\|_F \leq C_\nu$, and with the learning rate satisfying $0 < \eta\alpha \leq 1$, the proximal displacement recursion gives:

$$\max_{e \in \{0,\dots,E\}} \|W_k^{r,e} - W^r\|_F \leq R_{\mathrm{dyn}} := \min\left\{ R, \frac{G_{\mathrm{SSM},R} + C_\nu}{\alpha} \left[1 - (1 - \eta\alpha)^E\right] \right\}. \tag{452}$$

The geometric factor therefore damps accumulated local displacement. While this bound includes C_ν, it ignores cancellation between the local gradient and the dual correction. Accurate tracking can on the other hand produce smaller corrected updates than the inequality suggests, and given a tighter local neighborhood, the SSM parameter envelopes could be reduced.

Furthermore, note that the client-gradient disagreement becomes:

$$\Gamma_{\mathrm{dyn},R}^2 := \sup_{W \in \mathcal{N}_R(W^r)} \sum_k p_k \, \|\nabla \mathcal{L}_k(W) - \nabla f(W) - (\nu_k^r - \nu^r)\|_F^2 \,. \tag{453}$$

As a result, when the centered states accurately track the client-gradient differences, this residual can be smaller than $\Gamma_{\mathrm{SSM},R}^2$. Therefore, FedDyn combines local-displacement damping with a mechanism for reducing the disagreement highlighted by the FedAvg bound.

However, these benefits compete with the additional local smoothness term α and the server displacement:

$$\|W^{r+1} - W^r\|_F \leq R_{\mathrm{dyn}} + \frac{\|\nu^{r+1}\|_F}{\alpha}. \tag{454}$$

As such, suitable regularization and accurate tracking can make the damping and disagreement beneficial while preserving stable containment, if these improvements outweigh the smoothness constant increase provided the server update also remains within the stable neighborhood. As such, for a well-tuned FedDyn, we expect it to improve on FedAvg, with the benefits being especially highlighted under heterogeneous data conditions through disagreement correction.

FedSAM. FedSAM (Qu et al., 2022) retains FedAvg aggregation but uses a sharpness-aware local SGD step with two gradient evaluations:

$$u_k^{r,e} := \mathfrak{R}_{\mathrm{SAM}} \frac{g_k(W_k^{r,e}; \xi_k^{r,e})}{\|g_k(W_k^{r,e}; \xi_k^{r,e})\|_F}, \qquad W_k^{r,e+1} = W_k^{r,e} - \eta\, g_k(W_k^{r,e} + u_k^{r,e}; \xi_k^{r,e}), \tag{455}$$

where $\mathfrak{R}_{\mathrm{SAM}} > 0$ is the perturbation radius, with $u_k^{r,e} = 0$ when the first gradient vanishes.

Suppose the local iterates lie in $\mathcal{N}_R(W^r)$ and the SSM assumptions hold uniformly on $\mathcal{N}_{R+\mathfrak{R}_{\mathrm{SAM}}}(W^r) \subseteq \mathcal{S}_{\gamma,\delta}$. The expanded neighborhood contains the perturbed evaluation points, resulting in the local-displacement bound

$$R_{\mathrm{SAM}} := \max_{e\in\{0,\dots,E\}} \|W_k^{r,e} - W^r\|_F \le \eta E G_{\mathrm{SSM},R+\mathfrak{R}_{\mathrm{SAM}}}. \tag{456}$$

The sequence-wise smoothness argument in the proof of Lemma 4.7 also gives

$$\|g_k(W_k^{r,e} + u_k^{r,e}; \xi_k^{r,e}) - g_k(W_k^{r,e}; \xi_k^{r,e})\|_F \le L_{\mathrm{SSM},R+\mathfrak{R}_{\mathrm{SAM}}} \mathfrak{R}_{\mathrm{SAM}}. \tag{457}$$

As a result, relative to an SGD step from the same iterate and sample, SAM changes the update by at most $\eta L_{\mathrm{SSM},R+\mathfrak{R}_{\mathrm{SAM}}} \mathfrak{R}_{\mathrm{SAM}}$. When the perturbation changes the SSM constants only slightly and its accumulated effect over training remains small, we expect FedSAM to perform similarly compared to FedAvg under both nearly IID and heterogeneous data. Note that this expectation relies on the small-perturbation approximation, and possible generalization benefits from flatter solutions are not quantified by these worst-case bounds.

FedExP. FedExP (Jhunjhunwala et al., 2023) retains the FedAvg local updates but rescales the aggregated update:

$$\begin{aligned} W^{r+1} &= W^r + \eta_g^r \sum_{k=1}^K p_k (W_k^{r,E} - W^r), \\ \eta_g^r &:= \max\left\{1, \frac{\sum_{k=1}^K p_k \|W_k^{r,E} - W^r\|_F^2}{2\left\|\sum_{k=1}^K p_k (W_k^{r,E} - W^r)\right\|_F^2 + \epsilon_g}\right\}, \end{aligned} \tag{458}$$

where $\epsilon_g > 0$ stabilizes the denominator. By the weighted variance identity, the server scale factor η_g^r exceeds 1 when

$$\sum_{k=1}^K p_k \left\| W_k^{r,E} - \sum_{j=1}^K p_j W_j^{r,E} \right\|_F^2 > \left\| \sum_{k=1}^K p_k (W_k^{r,E} - W^r) \right\|_F^2 + \epsilon_g. \tag{459}$$

As a result, rescaling is activated when disagreement among client updates is sufficiently large relative to the squared norm of their average. Otherwise, $\eta_g^r = 1$ and the server update reduces to FedAvg.

For local SGD under the same stable-neighborhood assumptions, neighborhood radius remains $R_{\mathrm{local}} \le \eta E G_{\mathrm{SSM},R}$, however, the server step satisfies:

$$\|W^{r+1} - W^r\|_F = \eta_g^r \left\| \sum_{k=1}^K p_k (W_k^{r,E} - W^r) \right\|_F \le \eta_g^r \eta E G_{\mathrm{SSM},R}, \tag{460}$$

Since $\eta_g^r \ge 1$, this server-step upper bound is no smaller than the corresponding FedAvg bound, although the actual server displacement need not exceed the local radius. Therefore, server scaling may require a larger neighborhood to cover the server update, with larger parameter envelopes entering the SSM gradient and smoothness bounds. Containment of this expanded neighborhood in $\mathcal{S}_{\gamma,\delta}$ is further not guaranteed.

The frequency of rescaling depends on disagreement among the client updates, which can reflect both data heterogeneity and stochastic sampling. For IID data, we expect sufficiently similar client updates for the server scaling to remain mostly inactive, resulting in FedAvg-like behavior. However, given heterogeneous data, greater dispersion relative to the average update can activate scaling more often. The resulting server steps also amplify any local-drift error in the averaged update, without correcting client disagreement. For the considered SSM setting, we therefore expect worse performance than

FedAvg when these displacement and stability costs outweigh the acceleration benefit. This is a qualitative expectation, not a convergence comparison.

DiLoCo-style. The original DiLoCo framework (Douillard et al., 2024) combines AdamW local optimization with a Nesterov momentum server update. Our DiLoCo-style baseline retains the shared local phase and updates the server as follows:

$$\begin{aligned} m_o^r &= \beta_o m_o^{r-1} + \sum_{k=1}^{K} p_k (W_k^{r,E} - W^r), \\ W^{r+1} &= W^r + \alpha_o \left(\beta_o m_o^r + \sum_{k=1}^{K} p_k (W_k^{r,E} - W^r) \right), \end{aligned} \tag{461}$$

where $\alpha_o > 0$, $\beta_o \in [0, 1)$, and $m_o^{-1} = 0$. Setting $\beta_o = 0$ and $\alpha_o = 1$ reduces to FedAvg.

For the SGD-based comparison, the same stable-neighborhood conditions with a common gradient bound $G_{\mathrm{SSM},R}$ across rounds hold. Each averaged client update then has norm at most $\eta E G_{\mathrm{SSM},R}$.

As such, unrolling the server momentum recursion we get $\|m_o^r\|_F \leq \eta E G_{\mathrm{SSM},R}/(1-\beta_o)$, and therefore

$$\|W^{r+1} - W^r\|_F \leq \frac{\alpha_o}{1-\beta_o} \eta E G_{\mathrm{SSM},R}. \tag{462}$$

When $\alpha_o/(1-\beta_o) > 1$, this upper bound exceeds the corresponding FedAvg bound. Furthermore, unlike FedExP, momentum can accumulate even when clients agree, because it combines averaged updates across rounds. Persistent update directions can increase server displacement, while changes in direction can leave the momentum contribution misaligned with the current averaged update. The server trajectory may therefore require a larger neighborhood and larger SSM parameter envelopes, with stable containment requiring separate verification. The within-round local update rule remains unchanged, and outer momentum does not directly correct client disagreement.

As such, depending on the momentum parameters, the displacement and history effects could outweigh the acceleration benefit of momentum, leading to worse performance than FedAvg, although the effects could potentially balance out.

Summary. The FedAvg bound (Theorem B.5) connects convergence to $L_{\mathrm{SSM},R}$, $G_{\mathrm{SSM},R}$, and $\mathcal{H}_{\mathrm{TV}}$ through local displacement, stable containment, and client disagreement. Under the stated assumptions, FedProx (Theorem B.6) and FedDyn's proximal term damp displacement bounds, at a smoothness cost. SCAFFOLD and FedDyn can reduce corrected disagreement when their control states accurately track client-gradient differences, though IID data leave less disagreement to correct. FedAdam, FedExP, and DiLoCo-style retain the local phase but modify server updates through coordinatewise adaptation, disagreement-dependent extrapolation, and momentum, respectively. These updates may require larger stable neighborhoods. FedAlign targets coordinate mismatch through norm-preserving permutations, while FedSAM perturbs gradient evaluation points, so FedAvg-like behavior is expected when coordinate mismatch or accumulated perturbation effects, respectively, are small. These theorem-motivated expectations are collected in Table 2 and compared with observations in Section 5.

F Discussion on Model Merging

F.1 Model Merging with SSMs Motivation

Section 3 formulates distributed learning as the minimization of the aggregated expected sequence loss in (4) through repeated local adaptation and parameter aggregation. The optimization proceeds over T communication rounds, each consisting of E local SGD steps followed by aggregation of the updated model parameters. This framework also provides a natural formulation of one-shot model merging (Wortsman et al., 2022; Ilharco et al., 2023).

Consider the case of task or domain experts adapted from a common pretrained initialization. This setting shares the central operation studied in our distributed framework: combining parameter updates learned from heterogeneous data. Specifically, let M experts share the same selective SSM architecture, parameterization, and pretrained initialization W^0. Expert m adapts to a local

dataset $\mathcal{D}_m$ associated with a distribution $\mathcal{P}_m$, with all experts using the sequence-loss formulation of Section 3. To make the connection explicit, suppose each expert starts from $W_m^{0,0} = W^0$ and performs E local SGD steps following the update rule in that section. Denote its final parameters and task vector by

$$W_m := W_m^{0,E}, \qquad \mathbf{T}_m := W_m - W^0. \tag{463}$$

The independently adapted experts are then combined using a merging operator,

$$W_{\text{merge}} = \text{Merge}\big(\{W_m\}_{m=1}^M; W^0\big), \tag{464}$$

which may use both the expert parameters and the pretrained reference. If the merging operator is chosen to be weighted parameter averaging, the final model parameters are obtained as

$$W_{\text{avg}} = \sum_{m=1}^M p_m W_m = W^0 + \sum_{m=1}^M p_m \mathbf{T}_m, \qquad p_m > 0, \quad \sum_{m=1}^M p_m = 1, \tag{465}$$

which reduces exactly to one round of FedAvg with $K = M$ and $T = 1$ when the weights in the objective and aggregation are kept the same.

This demonstrates that averaging task vectors around a shared initialization provides a direct connection between one-shot model merging and the distributed optimization setting considered in this work. More general merging operators retain this sequence of independent adaptation followed by aggregation, although the differences in how the expert updates are combined limit the applicability of the current theoretical results.

Additionally, practical model merging often involves large pretrained networks and extensive independent fine-tuning. Although our formulation permits pretrained initialization and includes a multilayer extension in Appendix C, applying the results to complete pretrained architectures and expert-specific optimization protocols requires verifying additional assumptions. Applying our federated convergence results to one-shot merging requires a neighborhood around W^0 containing the full independent adaptation trajectories, without intermediate synchronization. This may require a large radius R and correspondingly large parameter envelopes, increasing $L_{\text{SSM},R}$ and $G_{\text{SSM},R}$ and making the bounds loose or potentially uninformative. The required stability and step-size conditions may also become harder to satisfy. When a common stable neighborhood exists, convex averaging preserves containment within it, but stability alone does not guarantee predictive performance across the contributing domains.

Nevertheless, the shared structure of independent adaptation followed by aggregation motivates including model merging as a related setting for interpreting our results. In the remainder of this section, we discuss connections between representative merging methods and our analysis, clarify their applicability and limitations, and highlight potential takeaways for merging selective SSMs. We further provide supplementary model merging results in Appendix H.3.

F.2 Model Merging Methods: Connections to Selective SSM Analysis

For a merged model parameterized by W_{merge} with $W_{\text{merge}} \in \mathcal{N}_R(W^0) \subseteq \mathcal{S}_{\gamma,\delta}$, the uniform regularity conditions of Lemma 4.7 imply

$$\begin{aligned} f\left(W_{\text{merge}}\right) &= f\left(W^0 + \left(W_{\text{merge}} - W^0\right)\right) \\ &\le f\left(W^0\right) + \left\langle \nabla f\left(W^0\right), W_{\text{merge}} - W^0 \right\rangle_F + \frac{L_{\text{SSM},R}}{2} \left\| W_{\text{merge}} - W^0 \right\|_F^2. \end{aligned} \tag{466}$$

Similarly, for a Q-layer model with $W_{\text{merge}} \in \mathcal{N}_{\mathbf{R}_Q}(W^0) \subseteq \mathcal{S}^{(Q)}_{\boldsymbol{\gamma},\boldsymbol{\delta}}$, the assumptions of Lemma C.4 give

$$f\left(W_{\text{merge}}\right) \le f\left(W^0\right) + \left\langle \nabla f\left(W^0\right), W_{\text{merge}} - W^0 \right\rangle_F + \frac{L^{(Q)}_{\text{SSM},\mathbf{R}_Q}}{2} \left\| W_{\text{merge}} - W^0 \right\|_F^2, \tag{467}$$

where the inner product and norm are taken over the full parameter tuple, and $L^{(Q)}_{\text{SSM},\mathbf{R}_Q} = H_\ell \Theta_Q^2 + L_\ell \Omega_Q$ is the existing multi-layer smoothness constant.

These inequalities highlight important factors that determine the loss bound after merging. Firstly, the alignment of $W_{\text{merge}} - W^0$ with $-\nabla f(W^0)$ indicates whether the update initially moves toward

a lower aggregated loss. In addition, the merging operator determines the direction and magnitude of this update - different merging methods might rescale or modify the model parameters during aggregation. Finally, the SSM smoothness constant bounds the error in this local loss prediction and depends on the neighborhood over which it is evaluated. The radius R therefore plays two roles: it specifies the maximum displacement of the model parameters $\left\|W_{\text{merge}} - W^0\right\|_F \leq R$ (or $\left\|W^{(q)}_{\text{merge}} - W^{0,(q)}\right\|_F \leq R_q \forall q \in \{1, \ldots, Q\}$) covered by the analysis, and determines the parameter range over which the smoothness bound holds. A larger radius can require larger parameter envelopes M_X, leading to larger smoothness constants $L_{\text{SSM},R}$ (or $L^{(Q)}_{\text{SSM},\mathbf{R}_Q}$) and a looser loss bound.

This motivates examining how different merging methods affect the merged model's displacement from W^0 and the neighborhood needed to contain it. We discuss several representative methods below.

Weighted parameter averaging. If all the experts lie in a common stable neighborhood $\mathcal{N}_R(W^0)$, their weighted average satisfies

$$\left\|W_{\text{avg}} - W^0\right\|_F \leq \sum_{m=1}^{M} p_m \left\|W_m - W^0\right\|_F \leq R. \tag{468}$$

So an averaging merge operation does not enlarge the neighborhood containing the experts, and the same local smoothness bound applies. This property also holds for the multi-layer product neighborhood. Furthermore, Lemma B.4 bounds the weighted squared gradient disagreement at a common parameter tuple by $4G^2_{\text{SSM},R}\mathcal{H}_{\text{TV}}$, showing how data heterogeneity and SSM sensitivity jointly determine this bound.

Task Arithmetic. Task Arithmetic (Ilharco et al., 2023) scales combined task vectors. In a weighted, normalized form,

$$W_{\text{TA}} = W^0 + \lambda\overline{\mathbf{T}}, \qquad \overline{\mathbf{T}} := \sum_{m=1}^{M} p_m \mathbf{T}_m, \qquad \lambda \geq 0. \tag{469}$$

Here $\lambda = 1$ gives parameter averaging, while $0 \leq \lambda \leq 1$ preserves containment in the common neighborhood. Larger scales require a separate containment check. In (466), the directional term scales with λ and the quadratic remainder with λ^2. Thus, scaling controls displacement relative to SSM sensitivity, but shrinking the update can also suppress useful adaptation and cannot recover directions lost through cancellation.

TIES-Merging. TIES-Merging (Yadav et al., 2023) trims small task-vector entries, elects each coordinate's sign according to the total retained movement, and averages the nonzero entries agreeing with that sign. The resulting vector is scaled and added to W^0. Because coordinates may select different expert subsets, the result need not remain in the experts' common neighborhood. Moreover, raw magnitude and sign do not account for how changes to A, W_Δ, W_B, and W_C interact through the SSM recurrence. Our bounds therefore motivate sensitivity-informed trimming and conflict resolution, but do not establish that TIES improves stability or reduces the merged model's loss.

Practical Takeaways and Limitations. Overall, the model merging discussion motivates jointly considering expert adaptation, merge strength, and SSM sensitivity. The dependence of the bounds on neighborhood radii, recurrent stability, and interactions between layers motivates considering SSM-specific properties when evaluating and designing merging methods. Additionally, the derived relationships remain conditional on the stated assumptions - the local loss bounds require uniform regularity over a common convex stable neighborhood, while applying the federated convergence results additionally requires containment of the full adaptation trajectories. Extensive fine-tuning may therefore require large parameter envelopes and yield loose bounds. These limitations motivate extending the analysis to complete pretrained architectures, practical optimizers, and general merging operators as a promising direction for future work. We complement this discussion with supplementary model-merging experiments and an analysis of their results in Appendix H.3.

G Detailed Experimental Setup

In this section, we describe the experimental setup in detail. The experiments fall into three groups: (i) synthetic dataset validation of the derived theoretical bounds, (ii) federated language modeling with

practical selective state-space models on RedPajama, (iii) one-shot model merging supplementary experiments with pretrained Large Mamba2-based models. We detail the datasets, models, as well as training and validation procedures needed for reproducibility.

G.1 DATASETS

Synthetic Validation. The synthetic datasets are autoregressive token sequences sampled from frozen teacher selective SSMs that implement the recurrence of Section 3 exactly, so that every assumption of Theorem 4.4 holds for the data-generating model (rather than being assumed to hold). A teacher consists only of the four parameter blocks $W = (W_\Delta, W_B, W_C, A)$ of a layer (no embedding, convolution, residual path, normalization, MLP, or Transformer components). The inputs are one-hot encoded: x_t is one of the standard basis vectors $e_1, \ldots, e_D$ of $\mathbb{R}^D$ for every t. This formulation makes the input dimension D also act as the vocabulary size (since there are D different basis vectors). Since $\|x_t\|_2 = 1$, Assumption 4.3 holds with $L_x = 1$.

The teacher SSM is turned into a generative model as follows. For a partially generated sequence $x_1, \ldots, x_t$, running it through the recurrence, starting from $h_0 = \mathbf{0}_{N\times D}$, produces an output $y_t \in \mathbb{R}^D$, which has one coordinate per token in the vocabulary. We therefore treat the d-th coordinate $y_{t,d}$ as the logit of the event that the next input is token d, i.e., that $x_{t+1} = e_d$. We then take a softmax with temperature $\tau > 0$ to convert these D logits into a probability vector over the tokens, and draw the next input as follows:

$$\Pr\left[x_{t+1} = e_d \,|\, x_1, \ldots, x_t\right] = \left[\mathrm{softmax}\left(y_t/\tau\right)\right]_d = \frac{\exp\left(y_{t,d}/\tau\right)}{\sum_{d'=1}^{D} \exp\left(y_{t,d'}/\tau\right)}, \qquad d = 1, \ldots, D.$$

Table 3 summarizes the single-layer dataset generation over different settings.

Table 3: Generation settings of the synthetic teacher datasets. Grid entries are searched per configuration in the fixed order given in the text.

Setting	Value
Input/vocabulary dimension D	$\{4, 8, 16\}$
State dimension N	$\{4, 8, 16\}$
Stability limit ρ	$\{0.3, 0.5, 0.7, 0.9, 0.95, 0.99, 0.995, 0.999\}$
Configurations (datasets)	72
Sequence length S	64 stored tokens (16 burn-in tokens discarded, 80 generated)
Training / test sequences	4,096 / 1,024
Sampling seeds (train / test)	1,000 / 1,001,003
First input x_1	uniform over $\{e_1, \ldots, e_D\}$
Teacher seeds s	$\{0, 1, 2\}$ (grid)
Gain g, temperature τ	grid
W_Δ, W_C	i.i.d. $\mathcal{N}(0, 0.75^2)$ entries
W_B	i.i.d. $\mathcal{N}(0, (0.75\,g)^2)$ entries
A	$A_n = -\gamma(1 + 0.25\lvert z_n\rvert)$, $z_n \sim \mathcal{N}(0, 1)$
Step margin δ	$\lfloor 10^5 \hat{\delta} \rfloor / 10^5$ (realized minimum step of W_Δ)
Transition margin γ	$-\ln(\rho)/\delta$
Parameter precision	single precision (float32)

Single-layer synthetic datasets. To generate the single-layer datasets, we instantiate one frozen teacher SSM for every combination of the input dimension $D \in \{4, 8, 16\}$, the state dimension $N \in \{4, 8, 16\}$, and the stability limit $\rho \in \{0.3, 0.5, 0.7, 0.9, 0.95, 0.99, 0.995, 0.999\}$ (from Definition 4.1), which gives $3 \times 3 \times 8 = 72$ different frozen teachers. Each such teacher, with its specific combination of D, N, and ρ, then generates 4,096 training and 1,024 test sequences of length $S = 64$ according to the procedure described above.

Teacher initialization and stability targeting. For a teacher SSM corresponding to a combination of D, N, and ρ, its weights are drawn with i.i.d. Gaussian entries as follows: $(W_B)_{n,d} \sim \mathcal{N}\big(0, (0.75g)^2\big)$, $(W_C)_{n,d} \sim \mathcal{N}\left(0, 0.75^2\right)$, $(W_\Delta)_{d,d'} \sim \mathcal{N}\left(0, 0.75^2\right)$, where g is a gain multiplier on the input projection described below. The margins (γ, δ) are chosen such that the teacher belongs to $\mathcal{S}_{\gamma,\delta}$ with $\rho = \exp(-\gamma\delta)$ equal to the specified value while satisfying both constraints of Definition 4.1. Since

the inputs are one-hot, $W_\Delta x_t$ is simply a column of W_Δ, so the time-scale constraint of Definition 4.1 reduces to the entry-wise condition $\mathrm{Softplus}\left((W_\Delta)_{d,d'}\right) \geq \delta$ for all d, d'. We therefore draw W_Δ first and set δ to the smallest step it produces,

$$\hat{\delta} := \min_{d,d'} \mathrm{Softplus}\left((W_\Delta)_{d,d'}\right) = \mathrm{Softplus}\left(\min_{d,d'} (W_\Delta)_{d,d'}\right), \qquad \delta := \left\lfloor 10^5\, \hat{\delta} \right\rfloor / 10^5,$$

rounded down so that $\delta \leq \hat{\delta}$ holds exactly, and then $\gamma := -\ln(\rho)/\delta$ for the designated ρ. The parameters of the transition block are then generated as follows:

$$A_n = -\gamma\,(1 + 0.25\,|z_n|)\,, \qquad z_n \sim \mathcal{N}(0,1), \quad n = 1, \ldots, N,$$

so that $A_n \leq -\gamma$ holds with a multiplicative margin. With this construction, the realized stability limit $\hat{\rho} := \max_{n,d,d'} \left(\bar{A}_t(e_{d'})\right)_{n,d} = \exp(-\hat{\gamma}\hat{\delta})$ with $\hat{\gamma} := \min_n(-A_n)$ satisfies $\hat{\rho} = \rho^{(\hat{\delta}/\delta)(1+0.25\min_n |z_n|)}$ and tracks ρ, including as $\rho \to 1$. Across all generated datasets $\rho - \hat{\rho} \in [0, 0.03]$. By construction, $A_n \leq -\gamma$, $\mathrm{Softplus}\left((W_\Delta)_{d,d'}\right) \geq \delta$, and $\hat{\rho} < 1$ hold. Since the admissible inputs form the finite set $\{e_1, \ldots, e_D\}$, these conditions are also verified exhaustively over all D one-hot inputs before a teacher is accepted. A clamping projection onto $\mathcal{S}_{\gamma,\delta}$ is available as a safeguard, but it was not active for any of the 72 accepted teachers.

Sampling. Given the frozen teacher with a combination of D, N, and ρ and the frozen weights constructed as described above, a sequence is generated by drawing x_1 uniformly from $\{e_1, \ldots, e_D\}$, starting from $h_0 = \mathbf{0}_{N\times D}$, and sampling x_{t+1} as described above for 79 further steps. The first 16 tokens are discarded as burn-in and the remaining $S = 64$ tokens are stored as a single sequence. The burn-in aims to reduce the stored prefix's dependence on the first input. The recurrent state at the start of the stored window is not persisted, and every subsequent teacher-forced or learner evaluation restarts from $h_0 = \mathbf{0}_{N\times D}$ at the first stored token, which is the convention of the theory.

The training and test splits are sampled with separate fixed seeds (1,000 and 1,001,003), reused in every configuration. Each dataset stores the token indices as 8-bit integers together with the frozen teacher parameters, the generation settings, and the stability certificate, and is never modified afterwards. The learner only sees the stored tokens and never accesses the teacher.

Acceptance criteria. A generated candidate dataset is kept only if it passes the fixed criteria of Table 4, which exclude degenerate teachers (whose sequences collapse into a dominant token, or become deterministic, constant, or uninformative), while ensuring that the next-token task carries predictive information. All diagnostics of the table are computed on the union of the training and test splits, unless stated otherwise. The central quantity is the teacher-forced negative log-likelihood (NLL): the stored tokens of a sequence are fed to the teacher as inputs, starting from $h_0 = \mathbf{0}_{N\times D}$, instead of re-sampling from it, and at every position $t = 1, \ldots, S-1$ we record $-\ln\left(\Pr[x_{t+1} \mid x_1, \ldots, x_t]\right)$, i.e., minus the log-probability that the teacher (at the sampling temperature τ) assigns to the stored next token. The NLL is the average of these values over all positions and sequences and measures how predictable the generated data are for the very model that produced them. A model that assigns probability $1/D$ to every token has NLL $\ln D$, which serves as the uniform baseline. The informativeness criterion therefore requires the sequences to be predictable beyond chance, while the perplexity criterion $\exp(\mathrm{NLL}) \geq 1.20$ excludes near-deterministic teachers. The mutual information is estimated from the empirical frequencies of consecutive token pairs. To avoid generating full datasets for clearly degenerate candidates, each candidate is first screened on a small draw of 256 training and 128 test sequences, and only if this screen passes is the full dataset generated and checked again against the same thresholds, which are never relaxed.

Gain and temperature choices. As $\rho \to 1$, the hidden state of a random teacher accumulates input contributions over a horizon of order $1/(1-\rho)$, so its logits grow and sampling collapses into a few tokens, which violates the balance and entropy criteria. Instead of relaxing the criteria, we search a fixed grid of input-projection gains and sampling temperatures,

$$(g, \tau) \in \big\{(1, 0.7), (1, 1), (0.5, 0.7), (0.5, 1), (0.25, 0.7), (0.25, 1), (0.15, 0.7),$$
$$(0.15, 1), (0.1, 0.7), (0.1, 1), (0.05, 0.7), (0.05, 1), (0.03, 1)\big\},$$

in this order for the teacher seeds $s \in \{0, 1, 2\}$ (outer loop), and accept the first candidate that passes every criterion in Table 4. Reducing g scales W_B (and therefore $\bar{B}_t$) and the accumulated state without touching W_Δ or A, so the margins (γ, δ) and ρ remain unchanged. Over the 72 dataset

Table 4: Acceptance criteria applied to every synthetic candidate dataset. Statistics are computed on the union of the training and test splits unless noted. The token marginal is the empirical distribution of the D tokens over all stored positions, ignoring position and context. NLL denotes the teacher-forced negative log-likelihood at the sampling temperature τ with $h_0 = \mathbf{0}$ per stored sequence.

Criterion	Threshold
Global token balance	largest probability in the token marginal $\leq 1/D + 0.10$
Marginal entropy	Shannon entropy of the token marginal $\geq 0.90 \ln D$ (the maximum $\ln D$ is attained by the uniform distribution)
Per-sequence majority (test split)	for each test sequence, the fraction of its S positions occupied by its most frequent token; the 95th percentile of this fraction over the test sequences $\leq 1/D + 0.20$
Class coverage	each of the D tokens occurs at least once in the training split and at least once in the test split
Constant sequences	fraction of sequences whose S tokens are all identical ≤ 0.01
Sequence diversity	number of distinct sequences divided by the number of sequences ≥ 0.95
Teacher informativeness	$\ln D - \mathrm{NLL} \geq 0.02$, i.e., the teacher beats the uniform predictor
No deterministic collapse	teacher perplexity $\exp(\mathrm{NLL}) \geq 1.20$
Predictive information	mutual information between consecutive tokens x_t and x_{t+1}, estimated from their empirical joint frequencies, ≥ 0.005
Finiteness	all diagnostics, logits, steps Δ_t, transitions $\bar{A}_t$, and hidden states are finite (no overflow)
Stability	the teacher is certified to lie in $\mathcal{S}_{\gamma,\delta}$ with realized stability limit $\hat{\rho} < 1$

configurations, 18 were accepted at the first attempt and the longest search took 32 attempts. Teacher seed 0 was accepted in 63 configurations, seed 1 in 5, and seed 2 in 4. The accepted gains were $g = 1$ (30 configurations), 0.5 (9), 0.25 (15), 0.15 (8), 0.1 (2), 0.05 (6), and 0.03 (2). A gain below one was required in every configuration with $\rho \geq 0.99$ and in only 7 of the 27 configurations with $\rho \leq 0.7$. The accepted temperatures were $\tau = 0.7$ (39 configurations) and $\tau = 1$ (33 configurations). The resulting teachers improve on the uniform NLL by 0.02 to 0.97, have a consecutive-token mutual information between 0.008 and 0.90 and a teacher perplexity between 1.5 and 15.7. The most frequent token has frequency at most $1.76/D$, the normalized marginal entropy is at least 0.94, and at least 98% of the stored sequences are unique.

Federated language modeling (RedPajama). The federated experiments use the SlimPajama-627B (Together Computer, 2023) release of RedPajama, available as `cerebras/SlimPajama-627B`.. Of the seven SlimPajama domains: `arxiv`, `book`, `c4`, `common_crawl` (`cc`), `github`, `stackexchange`, `wikipedia`, we use four-domain combinations to instantiate four federated clients:

- Setting 1:`arxiv`, `github`, `stackexchange`, `wikipedia`;
- Setting 2:`arxiv`, `c4`, `stackexchange`, `wikipedia`;
- Setting 3:`c4`, `common_crawl`, `github`, `wikipedia`;
- Setting 4:`arxiv`, `common_crawl`, `github`, `stackexchange`.

Due to copyright restrictions, the `book` domain is no longer accessible, and therefore not used in our experiments.

For every client, we set the maximum number of sequences to `sequences_per_client = 10,000`. Each client's dataset is split into 80% training and 20% validation data using `train_val_split = 0.8`. Due to the limited number of available sequences in certain domains, the effective per-client budgets are reduced for some settings. The resulting dataset statistics are summarized in Table 5.

We use β to label three domain-allocation regimes:

- $\beta = 0.0$ (IID): each client receives an equal mixture over all $K = 4$ domains.
- $\beta = 1.0$ (pure non-IID): each client receives data exclusively from its primary domain.

Table 5: Effective client train/validation budgets and server validation sizes across heterogeneity settings.

Setting	β	Train cap	Val cap	Train limiter data	Valid limiter data	Server val
Setting 1	0.0	7,427	1,857	`stackexchange`	`stackexchange`	7,424
Setting 1	0.7	7,427	1,857	`stackexchange`	`stackexchange`	7,431
Setting 1	1.0	7,427	1,857	`stackexchange`	`stackexchange`	7,428
Setting 2	0.0	7,367	1,842	`c4`	`c4`	7,360
Setting 2	0.7	7,367	1,842	`c4`	`c4`	7,364
Setting 2	1.0	7,367	1,842	`c4`	`c4`	7,368
Setting 3	0.0	7,367	1,842	`c4`	`c4`	7,360
Setting 3	0.7	7,367	1,842	`c4`	`c4`	7,364
Setting 3	1.0	7,367	1,842	`c4`	`c4`	7,368
Setting 4	0.0	7,427	1,857	`stackexchange`	`stackexchange`	7,424
Setting 4	0.7	7,427	1,857	`stackexchange`	`stackexchange`	7,431
Setting 4	1.0	7,427	1,857	`stackexchange`	`stackexchange`	7,428

- $\beta = 0.7$ (mixed): each client receives 0.7 of its data from its primary domain and 0.1 from each of the other three.

Concretely, with `sequences_per_client = 10,000` and an 80/20 train/validation split, each client typically contains 8,000 training chunks and 2,000 validation chunks. The server held-out evaluation set is the union of all client validation pools, giving approximately 8,000 validation chunks for $K = 4$ clients.

For each configuration, we use the Hugging Face gpt2 tokenizer (50,257 entries), with EOS reused for padding. The model configuration sets `vocab_size` = 50,277 and `pad_vocab_size_multiple` = 8, yielding 50,280 embedding/output rows.

Model Merging. For model merging, we evaluate on four text classification datasets: AG News (Zhang et al., 2015), DBpedia (Lehmann et al., 2015), TREC (Li and Roth, 2002), and IMDB (Maas et al., 2011). The dataset statistics and evaluation limits are summarized in Table 6.

Each task is formulated as causal-LM scoring without a classification head. At evaluation time, we score `prompt + ' ' + label` for every candidate label and select the label with the highest mean per-token log-probability over the label tokens. We use the mean, rather than the sum, so that labels of different lengths remain comparable. The evaluation pool is sub-sampled with a fixed seed of `99991` so that the evaluation split is identical across training seeds. The training pool is sub-sampled with the per-run training seed (`seed` $\in 0, 1, 2$) so that the three runs differ both in optimization noise and in their training subsamples.

Table 6: Task datasets and sequence limits for causal-LM evaluation.

Dataset	Task	# Classes	Train cap	Eval cap	`max_seq_len`
`ag_news`	News topic	4	40,000	5,000	256
`dbpedia_14`	Wikipedia topic	14	40,000	5,000	256
`trec`	Question type	6	$\approx 5{,}500$ (full)	5,000	256
`imdb`	Sentiment	2	40,000	5,000	256

The prompt templates are defined in Table 7:

Tokenization uses each model's own tokenizer, namely GPT-2 BPE. Prompts are truncated or padded to `max_seq_len = 256`, with 16 tokens reserved for the label completion and template tail. During tokenization, we construct a boolean `label_mask`. This mask is used during fine-tuning to compute the loss only over label-token positions, and during evaluation to average scores only over label tokens.

Table 7: Prompt templates and candidate labels used for causal-LM scoring.

Dataset	Prompt template and labels
ag_news	`Article: {text}\nCategory:` Labels: `World`, `Sports`, `Business`, `Tech`
dbpedia_14	`<title-aware template>...:Category:` Labels: 14-way Wikipedia topic labels
trec	`Question: {text}\nType:` Labels: `Description`, `Entity`, `Abbreviation`, `Person`, `Location`, `Number`
imdb	`Review: {text}\nSentiment:` Labels: `negative`, `positive`

All datasets are used under their original licenses and terms of use.

G.2 MODELS

Synthetic Validation. The learner SSM has the exact same architecture as the teacher: a single-layer selective SSM as defined in Section 3, with the four parameter blocks $W = (W_\Delta, W_B, W_C, A)$ of sizes $D \times D$, $N \times D$, $N \times D$, and $N \times 1$, one-hot inputs, and no embedding, convolution, residual path, normalization, MLP, or Transformer components. The output $y_t \in \mathbb{R}^D$ is used directly as the logit vector for next-token prediction. In every configuration, the learner uses the same input dimension D and state dimension N as the teacher that generated its dataset (D must match the vocabulary), so its parameter count is $D^2 + 2ND + N$ (Table 8). The learner never receives the teacher parameters - the only teacher-related quantities it uses are the margins (γ, δ) of its configuration, which define the stability set $\mathcal{S}_{\gamma,\delta}$ in which it is trained.

Table 8: Learner parameter counts $D^2 + 2ND + N$ for the synthetic configurations.

	$N = 4$	$N = 8$	$N = 16$
$D = 4$	52	88	160
$D = 8$	132	200	336
$D = 16$	388	520	784

Initialization. For a learner seed $s \in \{0, 1, 2\}$, drawn independently of the teacher seed, the projections are initialized with i.i.d. Gaussian entries $(W_\Delta)_{d,d'}, (W_B)_{n,d}, (W_C)_{n,d} \sim \mathcal{N}(0, 0.1^2)$, and the transition block as

$$A_n = -(\gamma + \epsilon) - 0.1\,|z_n|, \qquad z_n \sim \mathcal{N}(0, 1), \quad n = 1, \ldots, N,$$

with a safety margin $\epsilon = 10^{-4}$. The initial parameters are then projected onto $\mathcal{S}_{\gamma+\epsilon,\delta+\epsilon}$ by clamping $A_n \le -(\gamma + \epsilon)$ and $(W_\Delta)_{d,d'} \ge \mathrm{Softplus}^{-1}(\delta + \epsilon)$, and membership in $\mathcal{S}_{\gamma,\delta}$ is verified over all D one-hot inputs, exactly as for the teachers. The initialization scale 0.1 is much smaller than the teacher's 0.75, so every learner starts close to a uniform next-token predictor. The same projection is applied after every optimizer update during training, so the learner remains in $\mathcal{S}_{\gamma,\delta}$ with realized transitions bounded by ρ throughout training. All learner parameters are stored and updated in single precision (`float32`).

Federated Training. All federated runs train Mamba-2 selective state-space models from scratch using `mamba_ssm` 's `MambaLMHeadModel` with the configuration:

- `d_model` = 256
- `ssm_cfg.layer` = `Mamba2`
  - default `d_state` = 128
  - `d_conv` = 4
  - `expand` = 2

  - `headdim` = 64
  - `ngroups` = 1
- `vocab_size` = 50,277
- `pad_vocab_multiple` = 8, giving padded vocabulary size $50{,}280$
- `rms_norm` = `True`
- `residual_in_fp32` = `True`
- `fused_add_norm` = `True`
- `dtype` = `float32` (with `fp16` autocast inside forward pass)
- `depth` $\in \{4, 16\}$ (observed depths: 4 and 16)

i.e. we consider custom Mamba2 models of depth 4 or depth 16. The parameter counts are summarized in Table 9.

Table 9: Parameter breakdown by model depth.

Depth	Total params	SSM (mixer) params	Embedding params	Per-layer mixer
4	14,733,664	1,860,704	12,871,680	465,176
16	20,318,848	7,442,816	12,871,680	465,176

Notes: the input/output embedding are shared by the LM head ($50{,}280 \times 256 = 12,871,680$ parameters). The SSM (mixer) params denotes the parameter count restricted to tensors matching `*.mixer.*`, including `in_proj`, `conv1d`, `A_log`, `D`, `dt_bias`, the block-gated `RMSNorm`, and `out_proj`.

Table 10: Backbones of the depth-4 ablation. Layer indices run from 0 to 3, every layer not listed as attention is a Mamba2 mixer.

Backbone	Layout	SSM fraction	Total params
`mamba2`	$4\times$ Mamba2	1.00	14,733,664
`hybrid75`	attention at $\{2\}$	0.75	14,531,656
`hybrid50`	attention at $\{1, 3\}$	0.50	14,329,648
`hybrid50alt`	attention at $\{0, 2\}$	0.50	14,329,648
`hybrid25`	attention at $\{0, 2, 3\}$	0.25	14,127,640
`attn0`	$4\times$ attention	0.00	13,925,632

Hybrid and Attention Backbones. The backbone ablation (see the ablations in the Practical Distributed Learning Framework subsection) replaces some or all of the four Mamba2 mixers of the depth-4 model by causal multi-head attention layers, keeping `d_model` $= 256$, the tied GPT-2 embedding and LM head, the RMSNorm layers, and the residual structure unchanged, so that the mixer of every layer is either a Mamba2 block with the configuration above or an attention block. Attention layers are `mamba_ssm`'s `MHA` (Multi-Head Attention) with 8 heads of dimension 32, rotary position embeddings over the full head dimension, causal masking, biases on the input (`qkv`) and output projections, and no feed-forward sublayer. The attention layers are spread evenly through the stack. With n_{attn} attention layers among the $L = 4$ layers, the j-th attention layer (counting from $j = 0$) is placed at index $\lfloor (j + 0.5)\, L/n_{\text{attn}} \rfloor$, which puts a single attention layer in the middle of the stack (index 2), two attention layers at indices 1 and 3, and three at indices 0, 2, and 3, with the remaining layers being Mamba2 mixers. To check that the placement itself does not drive the results, `hybrid50alt` keeps two attention layers but moves them to indices 0 and 2. Table 10 lists the resulting layouts and parameter counts: an attention layer has $263{,}168$ parameters against $465{,}176$ for a Mamba2 mixer, so the total parameter count decreases with the attention fraction. The attention-only model (`attn0`) serves as the zero-SSM control (since it has no SSM state to align).

Model Merging. The model merging experiments use public Mamba2 checkpoints loaded from Hugging Face with `MambaLMHeadModel.from_pretrained`. The results evaluate the 130M, 370M, and 780M checkpoints.

Table 11: Public Mamba2 checkpoints used for model merging.

Checkpoint	Total params
`state-spaces/mamba2-130m`	130M
`state-spaces/mamba2-370m`	370M
`state-spaces/mamba2-780m`	780M

All checkpoints are loaded with `dtype = bfloat16`. The base state θ^0 is the unmodified pretrained checkpoint and is fixed once at the start of each experiment. The per-task specialists θ_m are produced by task fine-tuning (details in G.5), and saved to the per-experiment directory.

The pretrained models and the `mamba_ssm` module are used under their original licenses and terms of use.

G.3 Synthetic Distributed Learning Framework

For every configuration (D, N, ρ) and learner seed $s \in \{0, 1, 2\}$, the synthetic pipeline runs four stages on the dataset corresponding to that configuration: (i) centralized training of the learner of the Models subsection, (ii) evaluation of the gradient bound of Theorem 4.4 at the stored checkpoints, (iii) evaluation of the smoothness constants of Lemma 4.7 at the same checkpoints, and (iv) federated training with FedAvg and FedProx, which measures the quantities entering Theorem B.5 and Theorem B.6. Stages (ii) and (iii) never train, they only read checkpoints. All stages use the margins (γ, δ) of the configuration. Table 12 lists the shared settings.

Table 12: Shared settings of the synthetic centralized and federated runs.

Component	**Value**
Loss	mean softmax cross-entropy over the $S-1$ next-token positions
Optimizer	SGD, no momentum, weight decay, or gradient clipping
Learning rate η	0.05
Batch size	64 sequences
Centralized updates	$2{,}000$ (epochs of 64 updates, reshuffled every epoch)
Checkpoints and evaluation	every 100 updates; bound and smoothness checkpoints at updates 0, 500, $1{,}000$, $1{,}500$, $2{,}000$
Stability	projection onto $\mathcal{S}_{\gamma+\epsilon,\delta+\epsilon}$ with $\epsilon = 10^{-4}$ and certificate check after every update
Learner seeds	$\{0, 1, 2\}$ per configuration
Precision	training in `float32`; bound, smoothness, and certificate evaluation in `float64`

Centralized Training. The learner minimizes the mean teacher-forced next-token cross-entropy over the $S-1$ shifted positions of every training sequence, i.e., the sequence loss of (3) with ℓ_t the softmax cross-entropy and $y_t^* = x_{t+1}$. SGD uses no momentum, weight decay, or clipping, while projection and epoch reshuffling differ from the analyzed independent-sampling protocol. Each epoch shuffles the $4{,}096$ training sequences with a seed derived from the learner seed and the epoch index and cuts them into batches of 64, running for $2{,}000$ updates. After every update the parameters are projected onto $\mathcal{S}_{\gamma+\epsilon,\delta+\epsilon}$ as at initialization, and membership in $\mathcal{S}_{\gamma,\delta}$ is re-verified over all D one-hot inputs. Checkpoints are written every 100 updates, and the test NLL, perplexity, and next-token accuracy are evaluated on the $1{,}024$ test sequences at the same frequency, always restarting from $h_0 = \mathbf{0}_{N\times D}$ at the first stored token.

Gradient-Bound Evaluation. At the five checkpoints of Table 12 and at the frozen teacher, every one of the $1{,}024$ test sequences is processed in `float64`, and the gradient norm of its sequence loss with respect to the four blocks is computed as follows: $\|\nabla_W \mathcal{L}^S\|_F = \big(\sum_{X\in\{\Delta,B,C,A\}} \|\nabla_{W_X} \mathcal{L}^S\|_F^2\big)^{1/2}$. It is compared with two constants evaluated at the same parameters. The *configured* bound $\mathcal{G}_{\mathrm{SSM}}(W)$ is Theorem 4.4 with its analytic constants, the configured (γ, δ, ρ), $L_x = 1$, and

$L_\ell = \sqrt{2}$ (the gradient of the softmax cross-entropy with respect to the logits has norm at most $\sqrt{2}$). The *tightened* bound $\mathcal{G}^{\text{tight}}_{\text{SSM}}(W)$ uses realized stability margins, exact one-hot projection bounds, and finite-history sums: the realized margins $\hat{\gamma} = \min_n(-A_n)$, $\hat{\delta}$, and $\hat{\rho} = \exp(-\hat{\gamma}\hat{\delta})$ replace (γ, δ, ρ), $\max_d \|\operatorname{Softplus}(W_\Delta e_d)\|_2$ replaces K_Δ, the largest column norms $\max_d \|W_B e_d\|_2$ and $\max_d \|W_C e_d\|_2$ replace $\|W_B\|_2$ and $\|W_C\|_2$, and every geometric factor $1/(1-\rho)$ is truncated to $(1-\hat{\rho}^{S-1})/(1-\hat{\rho})$, since a stored sequence contributes at most $S-1$ terms to each history sum. These replacements upper-bound the same one-hot projection and history terms and do not exceed the configured constants. A violation is recorded when the measured norm exceeds the bound by more than a relative tolerance of 10^{-8}. The utilization is the ratio of the measured norm to the bound. The gradient of the batch-mean loss over all test sequences is evaluated as well and checked against the mean of the per-sequence norms (triangle inequality) and against the bound.

Smoothness Evaluation. At the same checkpoints, 32 test sequences drawn once with a fixed seed are processed in `float64`. For every sequence, (a) the operator norm of the Hessian of the sequence loss is estimated by power iteration with exact Hessian-vector products (at most 15 iterations, stopping when the Rayleigh quotient changes by less than a relative 10^{-9}) and compared with $L_{\text{SSM},R}$ at $R = 0$, (b) the finite-difference ratio $\|\nabla\mathcal{L}^S(W + hU) - \nabla\mathcal{L}^S(W)\|_F / h$ with $h = 10^{-3}$ is evaluated along three unit directions U (two random and the normalized gradient) and compared with $L_{\text{SSM},h}$ whenever the ball of radius h is guaranteed to lie in $\mathcal{S}_{\gamma,\delta}$. Between consecutive learner checkpoints $W_a \to W_b$, the per-sequence ratio $\max_i \|\nabla\ell_i(W_b) - \nabla\ell_i(W_a)\|_F / \|W_b - W_a\|_F$ is compared with $L_{\text{SSM},R}$ centered at W_a with $R = \|W_b - W_a\|_F$, and $|\mathcal{G}_{\text{SSM}}(W_b) - \mathcal{G}_{\text{SSM}}(W_a)| / \|W_b - W_a\|_F$ with $L_{\mathcal{G}_{\text{SSM}},R}$ of Lemma B.1. The loss-regularity constants are $L_\ell = \sqrt{2}$ and $H_\ell = 1/2$.

Federated Training. FedAvg and FedProx are run on the same learner with K clients holding disjoint, equal-size shards of the 4,096 training sequences and weights $p_k = 1/K$. Under the IID partition, the sequences are randomly permuted (seeded by the learner seed) and cut into K shards. Under the dominant-token partition, the sequences are first sorted by their most frequent token, then by the fraction of positions that token occupies, and then cut into K shards, so that clients differ systematically in token composition. Every client starts each round from the broadcast model and performs E local SGD updates with the settings of Table 12, each on a fresh batch of 64 sequences drawn with replacement from the client's shard, followed by the same projection and verification as in centralized training. FedProx adds $\mu(W - W^r)$ to every local gradient, taking projected SGD steps on F_k^r from Section 3. The server forms $W^{r+1} = \sum_k p_k W_k^{r,E}$ in `float64` and applies the projection once more. All runs start from the centralized learner initialization of the same seed and use $T = 60$ rounds. Table 13 lists the settings: the base setting is run on all 72 configurations, and the one-factor variations of K, E, μ, and the partition are run on the 16 core configurations, namely the ρ axis at $D = N = 8$ and all (D, N) pairs at $\rho = 0.95$. With three seeds this gives 1,200 federated runs.

Table 13: Synthetic federated settings. The base setting is run on all 72 configurations, each variation changes one factor of the base setting and is run on the 16 core configurations.

Component	**Value**
Algorithms	FedAvg; FedProx
Base setting	$K = 4$ clients, $E = 16$ local updates, IID partition
Communication rounds T	60
Client-count variation	$K \in \{2, 8, 32\}$
Local-step variation	$E \in \{1, 4, 64\}$
Proximal variation (FedProx)	$\mu \in \{0.01, 1\}$
Partition variation	dominant-token (non-IID)
Evaluation	at round 0 and every 5 rounds; $\|\nabla f(W^r)\|_F^2$ at every broadcast
Seeds	$\{0, 1, 2\}$, shared across the settings of a configuration

The following quantities are recorded. At every evaluated round, the subset gradient norm $\|\nabla f(W^r)\|_F$ on a fixed subset of 1,024 training sequences is compared with the configured and tightened $\mathcal{G}_{\text{SSM}}(W^r)$. The client drift $\|W_k^{r,E} - W^r\|_F$ (mean and maximum over clients) and the largest displacement of any local iterate $\|W_k^{r,e} - W^r\|_F$ are compared with the radius R. The heterogeneity $\sum_k p_k \|\nabla\mathcal{L}_k(W^r) - \nabla f(W^r)\|_F^2$, with client gradients over full shards, is compared with

$4\,\mathcal{G}_{\mathrm{SSM}}(W^r)^2 \sum_k p_k\,\mathrm{TV}_k^2$ of Lemma B.4, where TV_k is the exact total variation between the empirical distribution of client k over whole sequences and the pooled distribution (unigram and bigram variants are recorded for reference but do not enter the lemma). The test NLL of the global model is evaluated on the $1{,}024$ test sequences. Reference radii and envelopes are set from initialization: $R = \eta E\,\mathcal{G}_{\mathrm{SSM}}(W^0)$ and $M_X = \|W_X^0\|_2 + (T+1)R$ for each block X, from which $G_{\mathrm{SSM},R}$, $L_{\mathrm{SSM},R}$, and $L_{\mathcal{G}_{\mathrm{SSM}},R}$ are computed. For FedProx the radius is $R_\mu = \min\left\{R,\, G_{\mathrm{SSM},R}[1-(1-\eta\mu)^E]/\mu\right\}$ with correspondingly reduced envelopes. Beyond the sufficient step-size range, we log iterate containment, projection activity, and the tests $\eta E L \le 1/2$ and $12\eta^2(L+\mu)^2E(E-1) \le 1$. We compare the right-hand side of Theorem B.5 (coefficients $4, 4, 6, 48$) or Theorem B.6 (coefficients $3, 3, 3, 24$) with the subset estimate of $\frac{1}{T}\sum_{r<T}\|\nabla f(W^r)\|_F^2$, using the full-training initial loss $f(W^0)$ and $f^* = 0$.

G.4 Practical Distributed Learning Framework

The federated framework uses $K = 4$ clients unless stated otherwise (see the client-count ablation) and initializes all clients from the same randomly initialized model θ^0. At the start of every round r, the server broadcasts the global model θ^r, each client k runs E local optimization steps on its private data starting from θ^r, and returns its updated parameters $\theta_k^{r,E}$ to a server module that performs algorithm-specific aggregation. After aggregation, the new global model θ^{r+1} is evaluated on the union of all clients' local validation data (server validation pool). The per-round per-client training perplexities and the server validation perplexity are saved.

Shared Training Schedule. All algorithms are evaluated under the same overall optimization schedule. Across communication rounds, we keep the number of clients, participation rate, local optimizer, batch construction, sequence length, and random seeds fixed. The only differences across methods are the algorithm-specific server or client update rule and the corresponding method-specific hyperparameters. The shared hyperparameters are summarized in Table 14. Local training runs in single precision with mixed-precision (`fp16`) autocast and dynamic loss scaling, and gradients are clipped after unscaling. FedSAM is the one exception and runs in full `float32`. A fresh AdamW optimizer is created at the start of every round, so no optimizer state carries across rounds.

Table 14: Shared training hyperparameters used across all federated algorithms.

Component	**Value**
Number of clients, K	4
Participation rate	1.0; all clients participate each round
Communication rounds, T	50
Local steps per round, E	$\{64, 128, 256\}$
Per-step batch size	8
Sequence length, S	512
Local optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
Local weight decay	0.01
Local learning rate	10^{-3}
Gradient clipping	Global ℓ_2 norm clipping with threshold 1.0
Sampling	Infinite `RandomSampler` with replacement
Random seeds	$\{0, 1, 2\}$ for PyTorch and NumPy

The following metrics are saved for every round:

- **Train Perplexity** (per client). During each local step, the client computes the token-level cross-entropy of the next-token objective: the logits at positions $1, \ldots, S-1$ are scored against the tokens at positions $2, \ldots, S$, with $S = 512$. The pre-aggregation training perplexity of client k in round r is $\exp(\overline{\mathrm{CE}})$, where $\overline{\mathrm{CE}}$ is the mean cross-entropy over the E local steps of the round.
- **Validation Perplexity**. After aggregation, the global model θ^{r+1} is evaluated on the server validation pool. We report $\exp(\overline{\mathrm{CE}})$ with $+\infty$ saturation on overflow.

Federated Algorithms. We evaluate nine federated learning algorithms, all sharing the schedule of Table 14. Throughout, p_k denotes the sample-count weight of client k, and since every client processes the same number of sequences per round, $p_k = 1/K$ in all runs.

- **FedAvg** (McMahan et al., 2017). After each client completes its E local AdamW updates, the server sets $\theta^{r+1} = \sum_{k=1}^{K} p_k \theta_k^{r,E} =: \bar{\theta}^{r+1}$. This method serves as the main baseline and introduces no algorithm-specific hyperparameters.
- **FedProx** (Li et al., 2020). FedProx adds the uniform proximal penalty $\frac{\mu}{2}\|\theta - \theta^r\|_F^2$ to the local objective, which discourages local models from drifting far from the broadcast model. Server aggregation is identical to FedAvg.
- **SCAFFOLD** (Karimireddy et al., 2020). SCAFFOLD corrects each client's local update with the difference between a server control variate c^r and the client control variate c_k^r: the local gradient is shifted by $\lambda_c\,(c^r - c_k^r)$ with correction scale λ_c. The control variates are refreshed every f_c rounds with control learning rate η_c, and their norm is bounded by τ_c. Aggregation of the model parameters is identical to FedAvg.
- **FedAdam** (Reddi et al., 2021). Clients run the unchanged AdamW local phase. The server applies Adam to the pseudo-gradient $\bar{\theta}^{r+1} - \theta^r$: it maintains first- and second-moment estimates m^r and v^r with coefficients (β_1^s, β_2^s) and updates $\theta^{r+1} = \theta^r + \alpha_s\, m^r/(\sqrt{v^r} + \tau_s)$ coordinatewise, with server learning rate α_s and offset τ_s.
- **FedAlign** (Keçeci et al., 2025). FedAlign aligns the Mamba2 state coordinates of the client models before averaging. In the default mode (`align_mode = mamba2_ref`), the B/C state features of every client are matched to the current global model as the reference within each group, applying the same permutation to all coupled parameters. Optionally (`align_mamba2_heads = 1`), heads are matched as well, permuting every head-indexed parameter. After alignment, the client parameters are averaged with the same weights as FedAvg.
- **FedDyn** (Acar et al., 2021). Each client keeps a persistent dual state ν_k^r, initialized at zero, and trains the local objective $\mathcal{L}_k(\theta) - \langle \nu_k^r, \theta\rangle_F + \frac{\alpha}{2}\|\theta - \theta^r\|_F^2$, with the two regularization terms computed in `float32` outside autocast. After local training, the client sets $\nu_k^{r+1} = \nu_k^r - \alpha(\theta_k^{r,E} - \theta^r)$. The server keeps the average dual state $\nu^{r+1} = \nu^r - \alpha \sum_k p_k(\theta_k^{r,E} - \theta^r)$ and updates $\theta^{r+1} = \bar{\theta}^{r+1} - \nu^{r+1}/\alpha$. The local optimizer remains AdamW, and the gradient of the regularized objective is clipped like every other local gradient.
- **FedSAM** (Qu et al., 2022). Every local step is a first-order sharpness-aware step: the client computes the gradient g at the current iterate θ, perturbs the weights by $u = \mathfrak{R}_{\mathrm{SAM}}\, g/(\|g\|_F + 10^{-12})$ with the norm taken jointly over all trainable parameters, computes the gradient at $\theta + u$, restores θ exactly, and applies the perturbed gradient (clipped) with AdamW. Aggregation is identical to FedAvg. Because the perturbation must be formed from exact gradients across two backward passes, FedSAM runs without mixed precision at roughly twice the wall-clock cost of the other methods.
- **FedExP** (Jhunjhunwala et al., 2023). Local training is identical to FedAvg. With the client displacements $\theta_k^{r,E} - \theta^r$ and their weighted average $\sum_k p_k \left(\theta_k^{r,E} - \theta^r\right)$, the server applies $\theta^{r+1} = \theta^r + \eta_g^r \sum_k p_k \left(\theta_k^{r,E} - \theta^r\right)$ with the scaling factor $\eta_g^r = \max\left\{1,\ \sum_k p_k \|\Delta_k^r\|_F^2 \big/ \big(2\|\Delta^r\|_F^2 + \epsilon_g\big)\right\}$, a single scalar shared by all parameters, with norms taken over all floating-point tensors. The factor has no upper cap and is logged every round. We do not average the last two server iterates at evaluation time.
- **DiLoCo-style** (Douillard et al., 2024). Clients run the same AdamW local phase as FedAvg. The server applies an outer Nesterov step to the pseudo-gradient $\bar{\theta}^{r+1} - \theta^r$: $m_o^r = \beta_o m_o^{r-1} + \bar{\theta}^{r+1} - \theta^r$ and $\theta^{r+1} = \theta^r + \alpha_o(\beta_o m_o^r + \bar{\theta}^{r+1} - \theta^r)$, with $m_o^{-1} = 0$, outer learning rate α_o, outer momentum β_o, and per-parameter momentum buffers kept in `float32`. Unlike DiLoCo, whose workers keep their inner AdamW state across outer steps, the inner optimizer is re-created every round, as for all other methods.

Hyperparameter Tuning. For each algorithm, we ran every value of the grid in Table 15 at every (depth, E, β, setting) configuration and over three seeds.

Table 15: Hyperparameter grids of the federated algorithms, run at every configuration and seed. Values shared by all members of a grid are fixed.

Algorithm	Hyperparameters and values
FedAvg	none
FedProx	$\mu \in \{5{\times}10^{-5}, 10^{-4}, 5{\times}10^{-4}, 10^{-3}, 5{\times}10^{-2}\}$
SCAFFOLD	$f_c \in \{4, 8, 16\}$; $\eta_c = 0.01$; $\lambda_c = 0.01$; $\tau_c = 0.1$
FedAdam	$\alpha_s \in \{10^{-3}, 5{\times}10^{-3}, 10^{-2}\}$; $\beta_1^s = 0.9$; $\beta_2^s = 0.99$; $\tau_s = 10^{-3}$
FedAlign	`align_mode` = `mamba2_ref`; `align_mamba2_heads` $\in \{0, 1\}$
FedDyn	$\alpha \in \{10^{-3}, 3{\times}10^{-3}, 10^{-2}, 3{\times}10^{-2}, 10^{-1}\}$
FedSAM	$\mathfrak{R}_{\mathrm{SAM}} \in \{10^{-3}, 3{\times}10^{-3}, 10^{-2}, 5{\times}10^{-2}, 10^{-1}\}$
FedExP	$\epsilon_g \in \{10^{-4}, 3{\times}10^{-4}, 10^{-3}, 3{\times}10^{-3}, 10^{-2}\}$
DiLoCo-style	$\alpha_o \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1.0\}$; $\beta_o = 0.9$

The final reported metrics correspond, per configuration, to the hyperparameter value with the lowest validation perplexity averaged over the last 10 rounds (rounds 41-50) and over the three seeds. Runs that end with a non-finite perplexity are counted as failures and are never replaced.

Reporting Metrics. For every (depth, E, β, setting) configuration we report:

- `train_ppl`: the per-round, per-client training perplexity $\exp(\overline{\mathrm{CE}})$ averaged over rounds and clients, then averaged over seeds (with its standard deviation).
- `val_ppl`: the server validation perplexity $\exp(\overline{\mathrm{CE}})$ averaged over all 50 rounds, then over seeds (with its standard deviation).
- `val_ppl_last10`: the server validation perplexity averaged over rounds 41–50, then over seeds (with its standard deviation).

Ablations. We further add two ablation experiments to the main grid, both at the constant settings $\beta = 0.7$, depth 4, $E = 64$, $T = 50$ with three seeds and one fixed hyperparameter value per algorithm: FedProx $\mu = 10^{-3}$ ($5{\times}10^{-4}$ in the backbone ablation), SCAFFOLD $f_c = 8$, FedAdam $\alpha_s = 10^{-2}$, FedAlign without head matching (with head matching in the backbone ablation), FedDyn $\alpha = 10^{-2}$, FedSAM $\mathfrak{R}_{\mathrm{SAM}} = 10^{-2}$, FedExP $\epsilon_g = 10^{-3}$, and DiLoCo-style $(\alpha_o, \beta_o) = (0.7, 0.9)$.

The *client-count ablation* varies $K \in \{4, 8, 16, 32\}$ on all four domain settings. Clients are assigned primary domains round-robin over the four domains of the setting, so $K/4$ clients share each primary domain, and each client draws a fraction β of its data from its primary domain and $(1-\beta)/3$ from each of the other three. Since the demand on every domain grows as $K/4$, the per-client budget is capped at the available domain pool divided by this demand: the $7{,}427/1{,}857$ training/validation sequences per client at $K = 4$ become approximately $3{,}713/928$, $1{,}856/464$, and $928/232$ at $K = 8$, 16, and 32 for Settings 1 and 4, and slightly fewer for Settings 2 and 3 (which are limited by by the size of `c4`).

The *backbone ablation* keeps $K = 4$ and Setting 1 and replaces some or all Mamba2 mixers of the depth-4 model by attention layers. The six backbones are described in the Models subsection.

G.5 Model Merging Framework

The merging framework is a two-phase pipeline. In the first phase, we fine-tune the pretrained base θ^0 on each of the four downstream tasks independently, producing four task specialists θ_m. In the second phase, we merge the specialists with each configured method and evaluate the merged state on every task's evaluation split.

Per-Task Local Fine-Tuning For each task $m \in \{$`ag_news`, `dbpedia_14`, `trec`, `imdb`$\}$ the pretrained base θ^0 is fine-tuned on the task's training pool with the causal-LM cross-entropy objective restricted to label-token positions. Specialists are checkpointed at every epoch in "Checkpoint epochs" so that phase 2 can be repeated at multiple training horizons without re-paying the fine-tuning cost. Table 16 lists the optimization parameters.

The cross-entropy loss is masked to label-token positions only, prompt and template tokens are ignored. The training-pool subsample varies with the seed (so each seed sees a different 40k subset

Table 16: Per-task fine-tuning recipe (identical across tasks and seeds).

Component	Value
Optimizer	AdamW
Learning rate	2×10^{-4}
Weight decay	0.0
Gradient clipping	ℓ_2 norm, threshold 1.0
Train batch size	4 for `mamba2-130m`, `mamba2-370m`, 8 for `mamba2-780m`
Eval batch size	4
LR schedule	linear warmup + cosine decay
Warmup ratio	0.05 (5% of total steps)
Total epochs	4
Checkpoint epochs	$\{1, 2, 4\}$
Compute / storage dtype	`bfloat16`
Seeds per (model, task)	$\{0, 1, 2\}$
Sequence length	256 tokens

of the underlying task), while the evaluation-pool subsample is held fixed across seeds (eval seed = 99,991) so that all reported accuracies are comparable across seeds and methods.

G.6 Reproducibility

Deterministic Data and Seed Handling. All three frameworks share a single PyTorch + NumPy seed. The federated harness deterministically caches the tokenized RedPajama subsets to a single binary cache file shared across all algorithms, settings, and seeds. The merging harness deterministically samples train and eval pools from each downstream dataset, with the eval seed (99,991) held fixed across all training seeds so that all reported accuracies are directly comparable.

Hardware and Runtime. All reported experiments were run on single-GPU workers using NVIDIA A100-40GB and NVIDIA H100 GPUs. The A100-40GB workers were used for prototyping, debugging, and smaller validation runs, as well as the synthetic dataset generation and validation experiments. The remaining main sweeps of federated and model merging experiments were run on the H100 workers. Each run used one GPU worker, with host CPU/RAM used only for data loading, tokenization-cache access, logging, and checkpoint I/O. The tokenized RedPajama subsets were cached once and reused across algorithms, settings, and seeds.

For federated language-modeling experiments, the dominant cost was local client training. Runtime scaled approximately linearly with the number of local steps E. In our setup, single federated runs ranged from approximately 30 minutes for $E = 64$ to at most roughly 2 hours for the longest $E = 256$ runs on an H100. All federated algorithms used the same base hardware and data pipeline. Their runtime differences were small relative to the effect of E. Naturally, the 16-layer model scaled the computation compared to the 4-layer one.

For model merging, each Phase-1 task-expert fine-tuning configuration required up to roughly 8 hours on a single H100, depending on checkpoint size.

H Additional Experimental Results and Ablation Studies

H.1 Single Layer Synthetic Distributed Learning

Note that we validate the single-layer statements only. The multi-layer bound of Theorem C.3 composes the same per-layer constants with the layerwise sensitivities κ_q of (303): the contribution of layer q is the single-layer constant of that layer (with its parameters adjusted to the multi-layer setting) multiplied by $\prod_{j>q} \kappa_j$ over the layers above it. When $\kappa_j \geq 1$, these factors do not reduce

the per-layer bounds, but this does not establish greater relative looseness. Useful layers can also contract perturbations. Numerical evaluation is restricted to the single-layer theory.

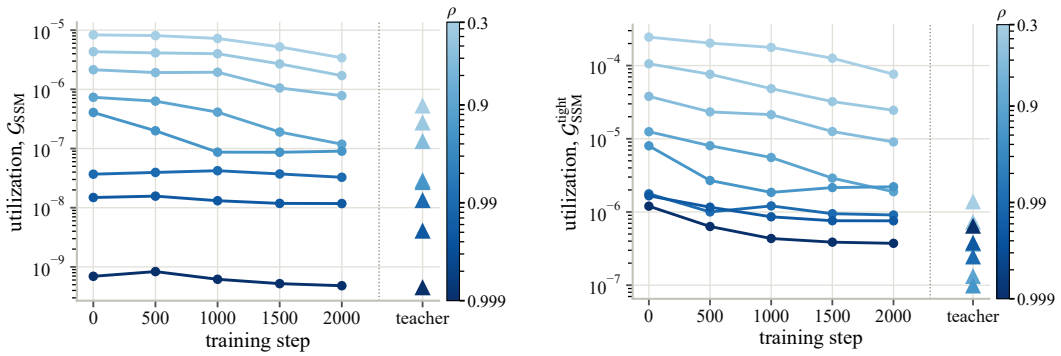
Validity and tightness. Table 17 reports, per ρ, the number of evaluated cells, parameter points and sequences, the violation counts for the configured and tightened bounds (all zero), the median utilization with confidence intervals, the largest utilization and the median realized $\hat{\rho}$. Table 18 gives the same statistics along training. Figure 3 shows the utilization along training, one curve per ρ, and Figure 4 the tightened-constant companion of Figure 2a together with the full per-sequence utilization distributions: at every ρ the distribution ends at least two orders of magnitude below the bound.

Table 17: Validity and tightness of the gradient bound of Theorem 4.4 by stability limit (single layer). Utilization is the measured per-sequence gradient norm divided by $\mathcal{G}_{\mathrm{SSM}}$ (conf.) or $\mathcal{G}^{\mathrm{tight}}_{\mathrm{SSM}}$ (tight.). The row marked * is the one $\rho = 0.3$ cell whose realized step $\hat{\delta}$ differs from the other eight and is grouped separately by the evaluation.

ρ	cells	points	sequences	viol. (conf./tight.)	median util. (conf.)	median util. (tight.)	max util. (tight.)	median $\hat{\rho}$
0.3*	1	18	18,432	0 / 0	7.6×10^{-5}	7.2×10^{-4}	8.3×10^{-3}	0.0602
0.3	8	144	147,456	0 / 0	2.0×10^{-6}	1.2×10^{-4}	4.1×10^{-3}	0.0345
0.5	9	162	165,888	0 / 0	1.6×10^{-6}	3.7×10^{-5}	5.1×10^{-3}	0.146
0.7	9	162	165,888	0 / 0	7.7×10^{-7}	1.6×10^{-5}	2.1×10^{-3}	0.369
0.9	9	162	165,888	0 / 0	2.3×10^{-7}	4.1×10^{-6}	1.0×10^{-3}	0.746
0.95	9	162	165,888	0 / 0	9.1×10^{-8}	2.0×10^{-6}	5.7×10^{-4}	0.868
0.99	9	162	165,888	0 / 0	3.2×10^{-8}	7.3×10^{-7}	1.6×10^{-4}	0.97
0.995	9	162	165,888	0 / 0	1.2×10^{-8}	7.6×10^{-7}	2.4×10^{-4}	0.982
0.999	9	162	165,888	0 / 0	5.3×10^{-10}	6.2×10^{-7}	1.2×10^{-4}	0.996
all	72	1,296	1,327,104	0 / 0	1.2×10^{-7}	3.5×10^{-6}	8.3×10^{-3}	0.823

Table 18: Utilization along training (single layer).

checkpoint	points	median util. (conf.)	median util. (tight.)	max util. (tight.)	median $\mathcal{G}_{\mathrm{SSM}}$	median gradient norm
frozen_teacher	216	2.7×10^{-8}	4.0×10^{-7}	1.8×10^{-4}	9.0×10^{6}	0.177
learner_final	216	1.2×10^{-7}	4.4×10^{-6}	3.5×10^{-3}	6.5×10^{5}	0.0612
learner_initialization	216	3.4×10^{-7}	1.9×10^{-5}	8.3×10^{-3}	1.1×10^{5}	0.0293
learner_intermediate	648	1.4×10^{-7}	5.8×10^{-6}	7.4×10^{-3}	4.0×10^{5}	0.0438



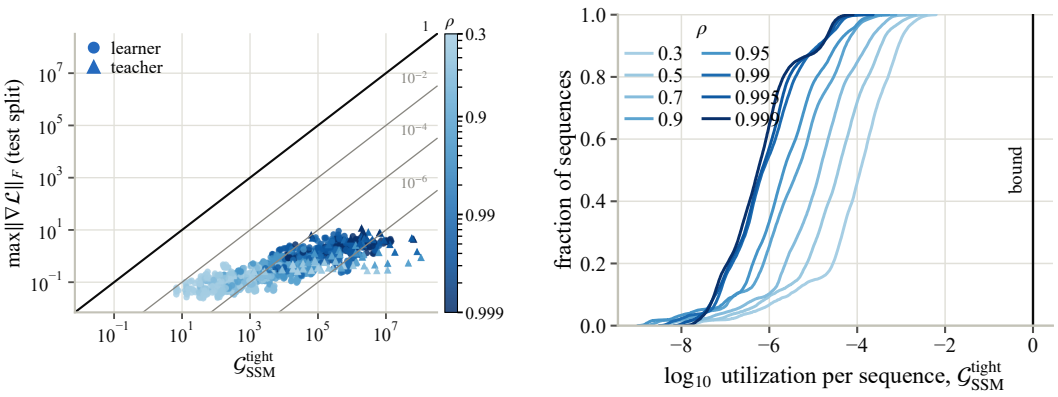
(a) Configured constants

(b) Tightened constants

Figure 3: Median bound utilization along training, one curve per ρ (colour), with the frozen teacher at the last position. The bound is tightest at initialization and loosest at the frozen teacher at every ρ.

Anatomy of the bound and of the gradient. Table 19 lists the median share of the squared gradient norm carried by each parameter block at the final learner checkpoint and at the frozen teacher, next to the share of $\mathcal{G}_{\mathrm{SSM}}$ carried by its recurrent-Jacobian term.

Dimension. Table 20 and Figure 2d give the log-log exponents of the configured bound, the tightened bound and the measured gradient in D (at fixed N, seed and ρ) and in N at the final learner checkpoint. The certified exponents follow the $N\,D^2$-type structure of K_W, the measured gradient is flat in N and decreasing in D.



(a) Gradient versus $\mathcal{G}_{\mathrm{SSM}}^{\mathrm{tight}}$

(b) Per-sequence utilization

Figure 4: (a) Largest measured gradient norm over the test split against the tightened $\mathcal{G}_{\mathrm{SSM}}^{\mathrm{tight}}$ for all 1,296 parameter points (thin lines mark utilizations 10^{-2} to 10^{-6}). (b) Empirical distribution of the per-sequence tightened utilization per ρ (one in eight sequences shown), the vertical line is the bound.

Table 19: Where the measured gradient lives (median share of squared Frobenius norm per parameter block) against the share of the certified bound carried by its recurrent-Jacobian term.

checkpoint	ρ	W_Δ	W_B	W_C	A	recurrent share of $\mathcal{G}_{\mathrm{SSM}}$
final (step 2000)	0.3	0.148	0.428	0.431	0.000	0.994
	0.5	0.145	0.434	0.433	0.000	0.996
	0.7	0.121	0.432	0.428	0.002	0.997
	0.9	0.044	0.369	0.415	0.039	0.999
	0.95	0.039	0.387	0.429	0.079	0.999
	0.99	0.006	0.187	0.204	0.600	1.000
	0.995	0.005	0.096	0.122	0.780	1.000
	0.999	0.003	0.067	0.072	0.852	1.000
frozen teacher	0.3	0.050	0.594	0.331	0.000	0.999
	0.5	0.077	0.568	0.404	0.000	0.999
	0.7	0.065	0.548	0.350	0.003	0.999
	0.9	0.043	0.482	0.301	0.111	1.000
	0.95	0.028	0.354	0.124	0.325	1.000
	0.99	0.002	0.393	0.015	0.581	1.000
	0.995	0.002	0.420	0.008	0.570	1.000
	0.999	0.001	0.362	0.004	0.631	1.000

Realized stability and training. Table 21 and Figure 5 demonstrate that the bounds use learners that generally outperform uniform prediction and approach the teachers' test Negative Log Likelihood (NLL). At short memory (ρ far away from 1), the teachers' small predictive advantage over uniform makes normalized gap recovery sensitive to small absolute errors, while the learners remain well inside the stability limit. At long memory (ρ approaches 1), learners nearly match the teachers and train close to that limit.

Federated ingredients. Figure 6 shows how the quantities that enter Theorem B.5 scale with the federated hyperparameters. Client drift, the end-of-round displacement $\|W_k^{r,E} - W^r\|_F$, grows as $E^{0.87}$ while the initialization-based reference radius $R = \eta E \mathcal{G}_{\mathrm{SSM}}(W^0)$ grows as E^1 (Figure 6a). The client disagreement $\sum_k p_k \|\nabla\mathcal{L}_k(W^r) - \nabla f(W^r)\|_F^2$, the quantity whose supremum $\Gamma^2_{\mathrm{SSM},R}$ Lemma B.4 bounds, grows as $K^{1.48}$ (Figure 6b) and as $(1-\rho)^{-1.15}$, whereas drift is flat in ρ and $\|\nabla f\|_F^2$ grows only as $(1-\rho)^{-0.2}$ (Figure 6c), so disagreement grows faster than the squared gradient norm, consistent with the lemma's SSM-sensitivity dependence. Figure 7 shows the per-round curves behind these tail means. Figure 8 examines the heterogeneity lemma: measured disagreement is 10^{-11} to 10^{-14} of $4\mathcal{G}^2_{\mathrm{SSM}}\mathcal{H}_{\mathrm{TV}}$ (Figure 8a). The sequence-level squared-TV aggregate is 0.56 at $K = 4$ (0.25 to 0.94 across K) and unchanged by the dominant-token partition, whereas a token-level squared-TV analogue rises from 1.6×10^{-5} to 2.8×10^{-3} and disagreement by $226\times$ (Figure 8b). Thus, the sequence-level term is insensitive to this partition; certifying a token-level or moment-based alternative requires a separate bound.

Table 20: Log-log exponents in the state and vocabulary sizes at the final learner checkpoint (median over one-factor fits with 95% confidence intervals).

ρ	$\mathcal{G}$ in D	$\mathcal{G}_{\mathrm{SSM}}^{\mathrm{tight}}$ in D	$\|\nabla\mathcal{L}\|$ in D	$\mathcal{G}$ in N	$\mathcal{G}_{\mathrm{SSM}}^{\mathrm{tight}}$ in N	$\|\nabla\mathcal{L}\|$ in N
0.3	1.78 [1.71, 2.84]	-0.29 [-1.28, 0.11]	-0.92 [-0.95, 0.41]	1.96 [1.81, 2.10]	1.81 [0.82, 2.30]	0.17 [0.14, 0.22]
0.5	1.75 [1.48, 2.80]	0.38 [-0.79, 0.77]	-0.97 [-1.09, 0.34]	2.05 [1.76, 2.21]	1.86 [1.21, 2.44]	0.19 [0.12, 0.29]
0.7	1.39 [1.30, 2.47]	0.04 [-0.40, 0.89]	-1.23 [-1.49, -0.07]	1.70 [0.94, 1.80]	1.41 [1.32, 1.96]	0.03 [-0.37, 0.27]
0.9	1.86 [1.70, 2.19]	0.48 [0.33, 1.64]	-1.28 [-1.66, -1.09]	1.32 [0.74, 2.23]	1.55 [0.30, 3.18]	0.10 [-0.50, 0.87]
0.95	1.74 [1.25, 2.01]	0.31 [0.10, 1.40]	-1.63 [-1.84, -1.58]	1.09 [0.87, 1.14]	1.37 [0.28, 1.55]	0.05 [-0.07, 0.23]
0.99	1.85 [0.88, 2.23]	1.05 [0.03, 1.96]	-1.38 [-1.58, -1.27]	1.08 [0.95, 1.11]	1.38 [0.75, 1.66]	0.06 [-0.12, 0.12]
0.995	2.93 [0.71, 3.40]	2.09 [0.12, 2.94]	-0.30 [-1.39, -0.21]	1.70 [0.99, 2.15]	1.55 [1.10, 2.48]	0.50 [-0.09, 0.56]
0.999	2.90 [1.17, 3.85]	2.33 [0.91, 3.71]	0.16 [-0.06, 0.45]	1.27 [0.63, 2.20]	1.34 [0.81, 2.75]	0.17 [-0.41, 0.49]

Table 21: Centralized learners by stability limit: fraction of runs whose test Negative Log Likelihood (NLL) falls below the uniform baseline, median excess NLL over the teacher, and the fraction of the uniform-to-teacher gap recovered.

ρ	runs	recovered fraction	median excess NLL	median gap recovered [95% CI]	median final $\hat{\rho}$
0.3	27	0.93	0.0180	0.80 [0.52, 0.90]	0.031
0.5	27	0.93	0.0212	0.79 [0.49, 0.82]	0.134
0.7	27	0.93	0.0203	0.81 [0.40, 0.87]	0.357
0.9	27	0.93	0.0187	0.69 [0.64, 0.96]	0.724
0.95	27	1.00	0.0179	0.74 [0.52, 0.99]	0.855
0.99	27	1.00	0.0027	0.95 [0.42, 1.03]	0.966
0.995	27	1.00	0.0004	1.00 [0.75, 1.02]	0.981
0.999	27	1.00	-0.0019	1.04 [0.95, 1.05]	0.996

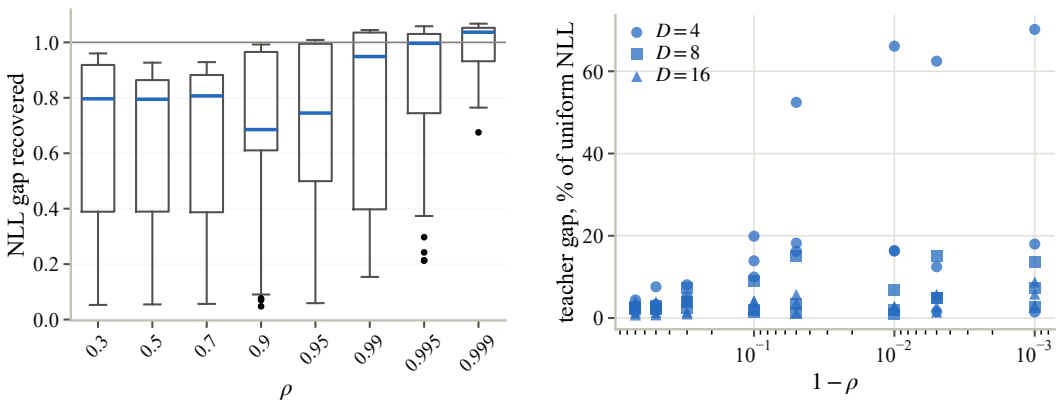
What drives the bound. Table 22 and Figure 9 give signed covariance attribution across the bank. The recurrent-history term carries at least 99% of the bound, so $\log_{10}\mathcal{G}_{\mathrm{SSM}}$ is the sum of log-factors $\rho/(1-\rho)^2$, ND, $\|W_C\|_2$, K_B, K_Δ, and $K_\Delta + \sqrt{D}\|A\|_2$, plus a residual below 0.1 decades. Each share is the covariance of a log-factor with log-bound magnitude divided by the latter's variance. The memory factor spans 6.2 decades and contributes 93%. Dimension, readout and step contribute 7%, 6% and 4%, while the transition factor contributes -13%, offsetting memory growth. This offset helps explain the exponent above -2 in Figure 2b. Teacher initialization gives $\|A\|_2 = \gamma\sqrt{\sum_n (1+0.25|z_n|)^2}$ with $\gamma = -\ln\rho/\delta$. For log-utilization, memory contributes 102% and measured gradient growth -18%. Recurrence therefore dominates variation in tightness across this bank, not necessarily absolute looseness.

Table 22: What drives the bound across the bank. $\log_{10}\mathcal{G}_{\mathrm{SSM}}$ is the sum of the logarithms of the factors of its recurrent-history term (plus a residual for the remaining terms), so the covariance of each factor with $\log_{10}\mathcal{G}_{\mathrm{SSM}}$ divided by its variance is an additive attribution that sums to one over all 1,296 parameter points. The last column attributes the variation of the utilization $\|\nabla\mathcal{L}\|_F/\mathcal{G}_{\mathrm{SSM}}$ in the same way, with the measured gradient as an additional term. A negative share means the factor moves against the total: $\|A\|_2$ shrinks as $\rho \to 1$.

factor	range (decades)	std (decades)	share of $\mathrm{Var}(\log_{10}\mathcal{G}_{\mathrm{SSM}})$	share of $\mathrm{Var}(\log_{10}$ util.)
memory $\rho/(1-\rho)^2$	6.2	2.06	93%	102%
dimension $N\,D$	1.2	0.35	7%	12%
readout $\|W_C\|_2$	1.5	0.31	6%	6%
input K_B	1.8	0.28	3%	3%
step K_Δ	0.8	0.18	4%	6%
transition $K_\Delta + \sqrt{D}\,\|A\|_2$	2.0	0.50	-13%	-11%
remaining terms of $\mathcal{G}_{\mathrm{SSM}}$	0.1	0.01	0%	0%
measured $\|\nabla\mathcal{L}\|_F$	2.7	0.62	–	-18%

H.2 Practical Distributed Learning

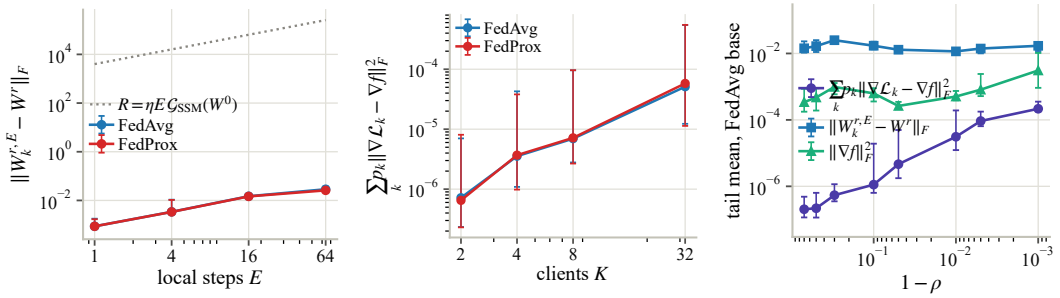
In this subsection we provide supplemental results to the Section 5 Mamba2 federated learning experiments. Every table reports the three metrics defined in Appendix G: the client training perplexity, the server validation perplexity (both averaged over all 50 rounds), and the server validation perplexity averaged over the last 10 rounds. Appendix H.2.1 evaluates the 9 federated algorithms in all the



(a) Gap recovered by the learners

(b) Predictive gap of the teachers

Figure 5: (a) Fraction of the uniform-to-teacher NLL gap recovered by the learners, per ρ. (b) The teachers' predictive gap over the uniform baseline as a percentage of the uniform NLL, per cell.



(a) Client drift against E

(b) Dissimilarity against K

(c) Base setting against $1-\rho$

Figure 6: Federated scaling directions (medians with bootstrap 95% intervals). (a) End-of-round client drift against E for FedAvg and FedProx on the core cells, with the initialization-based reference radius $R = \eta E\, \mathcal{G}_{\mathrm{SSM}}(W^0)$. (b) Client dissimilarity $\sum_k p_k \|\nabla \mathcal{L}_k(W^r) - \nabla f(W^r)\|_F^2$ against K on the core cells. (c) $\sum_k p_k \|\nabla \mathcal{L}_k(W^r) - \nabla f(W^r)\|_F^2$, drift and $\|\nabla f\|_F^2$ against $1-\rho$ at the base setting (all 72 cells).

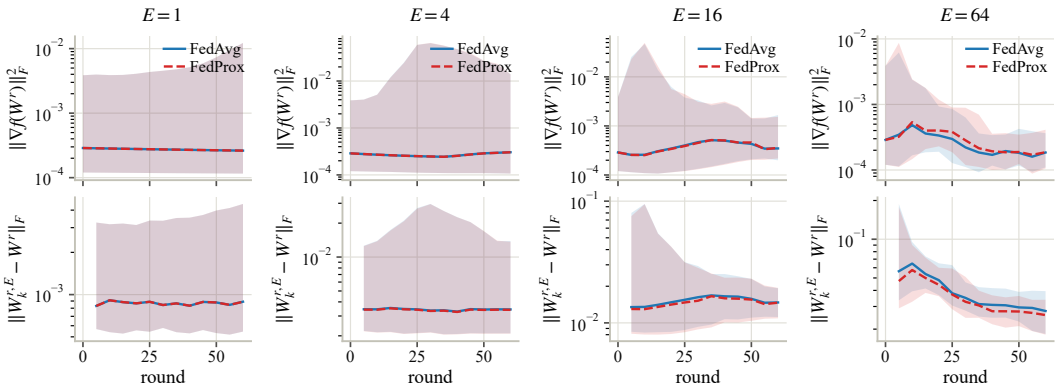
72 main configurations in details, demonstrating how the different local rounds E, model depths, heterogeneity degrees β, and data compositions affect their performance. The reported standard deviation measures the per-seed average across configurations for 3 seeds (not the perplexity spread across configurations). Appendices H.2.2 and H.2.3 provide two ablation studies, on different client counts, and different model backbones respectively. Both use fixed hyperparameters listed in the Ablations paragraph of Appendix G.

H.2.1 Detailed Results Across the 72 Main Configurations

Tables 23-26 report the nine algorithms averaged over the 72 configurations along each factor of the grid.

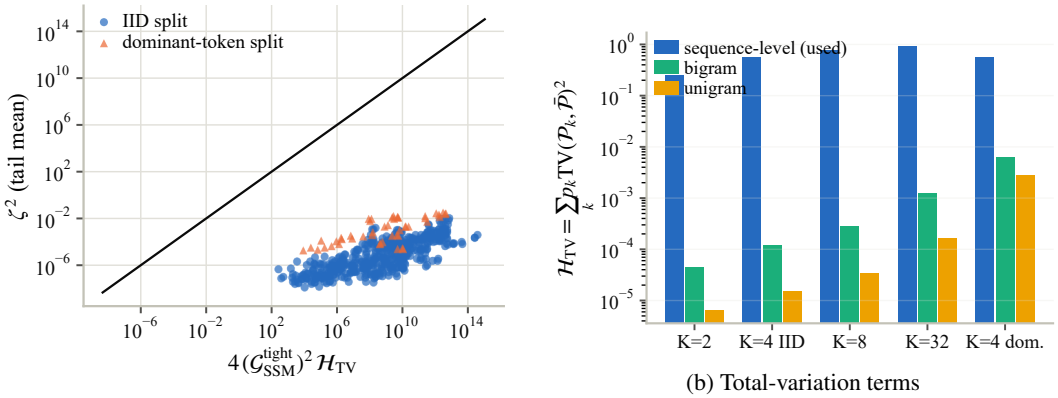
Comments. The results averages show a consistent overall ranking. FedProx achieves the lowest mean perplexity in 34 of the 36 metric columns. The two exceptions are last-10 validation perplexity at $\beta = 1$, where FedDyn is marginally lower (40.0 versus 40.2), and all-round validation perplexity at $E = 64$, where SCAFFOLD is lower (105.6 versus 106.6). On the last-10 metric, FedProx, FedDyn, and SCAFFOLD rank first, second, and third, respectively, at every local-step count, domain setting, and model depth in Tables 24-26.

Heterogeneity produces the clearest changes in relative performance. FedProx reduces last-10 perplexity relative to FedAvg by approximately 19%, 20%, and 34% at $\beta = 0$, 0.7, and 1, respectively.



(a) $E = 1$ (b) $E = 4$ (c) $E = 16$ (d) $E = 64$

Figure 7: Per-round gradient (top) and client drift (bottom) on the core cells for each E



(a) Heterogeneity lemma

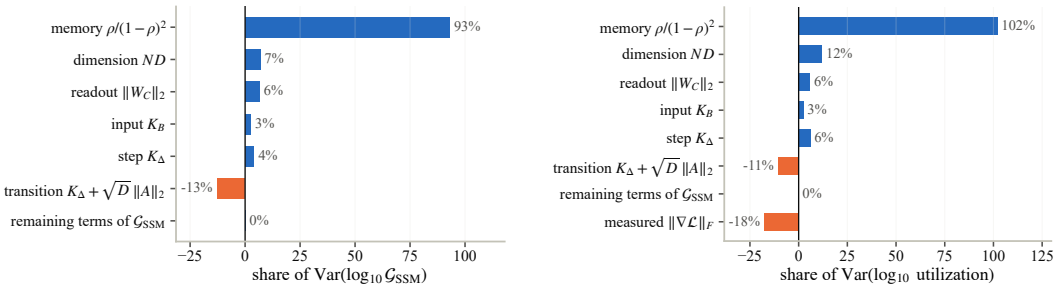
(b) Total-variation terms

Figure 8: (a) Measured client dissimilarity against the certified heterogeneity bound with the tightened constant, for IID and dominant-token splits. (b) Sequence-level, bigram and unigram weighted total variation for the client counts and partitions of the bank, only the sequence-level value enters the lemma.

SCAFFOLD moves from eighth place at $\beta = 0$ (46.9 versus 45.4 for FedAvg) to third at $\beta = 1$, where it improves on FedAvg by 18%. FedDyn narrowly overtakes FedProx at $\beta = 1$, whereas FedExP moves from near parity with FedAvg at $\beta \leq 0.7$ to last place at $\beta = 1$, with 13.5% higher last-10 perplexity. Across the same increase in β, last-10 perplexity rises by approximately 32-34% for FedAvg, FedAlign, FedSAM, and DiLoCo-style, by 52% for FedExP, and by 16% for FedAdam. The corresponding increases are only 9% for FedProx and 6% for SCAFFOLD. FedDyn instead improves by approximately 9%, although this comparison uses a separately selected α at each β.

Increasing the number of local steps strengthens the relative advantages of FedDyn and SCAFFOLD. Their reductions in last-10 perplexity relative to FedAvg grow from approximately 12% and 4.5% at $E = 64$ to 22% and 10% at $E = 256$, respectively. FedProx maintains a comparatively uniform reduction of 24-27% across the three values of E. These trends are qualitatively consistent with the role of the $E(E - 1)$ drift term in the bounds and the benefits of radius control or heterogeneity correction discussed in our analysis. Naturally, larger E results in longer training, so all nine algorithms achieve approximately 35-45% lower last-10 perplexity when E increases from 64 to 256 at fixed $T = 50$ communication rounds.

The remaining methods generally remain close to, or underperform, FedAvg on the last-10 metric. FedAlign and FedSAM stay within 2% of FedAvg throughout the factor tables, while DiLoCo-style stays within 5%. FedAdam has approximately 9% higher last-10 perplexity at every local-step count and model depth, with gaps of approximately 5-12% across domain settings.



(a) What drives $\mathcal{G}_{\mathrm{SSM}}$

(b) What drives the utilization $\|\nabla\mathcal{L}\|_F/\mathcal{G}_{\mathrm{SSM}}$

Figure 9: Additive attribution of the variance of (a) $\log_{10}\mathcal{G}_{\mathrm{SSM}}$ and (b) $\log_{10}$ utilization over all 1,296 parameter points to the factors of the bound (and, for the utilization, to the measured gradient). Bars sum to 100%, a negative bar moves against the total.

Table 23: **Effect of heterogeneity.** Training, validation, and last-10-round validation perplexity of the nine federated learning algorithms on Mamba2 at every β, averaged over the 24 configurations.

	$\beta = 0$			$\beta = 0.7$			$\beta = 1$		
Algorithm	Train	Val	Last-10	Train	Val	Last-10	Train	Val	Last-10
FedAvg	75.4 ± 0.5	63.7 ± 0.3	45.4 ± 0.2	99.3 ± 0.2	66.7 ± 0.6	45.8 ± 0.3	143.5 ± 0.7	136.5 ± 3.8	60.7 ± 0.5
FedProx	**69.0** ± 0.2	**57.6** ± 0.3	**36.9** ± 0.4	**88.0** ± 0.4	**60.4** ± 0.4	**36.5** ± 0.2	**119.3** ± 0.4	**121.3** ± 2.9	40.2 ± 0.3
SCAFFOLD	78.3 ± 0.5	66.3 ± 0.2	46.9 ± 0.1	102.4 ± 0.4	67.3 ± 0.3	45.2 ± 0.1	149.0 ± 0.5	131.6 ± 3.6	49.7 ± 0.2
FedAdam	88.4 ± 0.4	164.8 ± 1.6	53.0 ± 0.3	112.7 ± 0.4	160.0 ± 2.1	50.8 ± 0.1	157.2 ± 0.6	355.2 ± 13.8	61.7 ± 0.3
FedAlign	75.3 ± 0.4	63.6 ± 0.2	45.2 ± 0.0	99.2 ± 0.4	66.6 ± 0.3	45.7 ± 0.2	143.5 ± 0.5	135.3 ± 5.4	60.1 ± 0.4
FedDyn	78.9 ± 0.1	106.4 ± 2.1	44.1 ± 0.5	95.5 ± 0.2	120.1 ± 3.9	41.6 ± 0.1	127.1 ± 0.0	814.4 ± 50.2	**40.0** ± 0.2
FedSAM	75.4 ± 0.4	63.7 ± 0.3	45.2 ± 0.1	99.3 ± 0.4	66.7 ± 0.3	45.7 ± 0.1	144.5 ± 0.7	135.5 ± 3.4	59.8 ± 0.3
FedExP	75.3 ± 0.4	63.6 ± 0.2	45.2 ± 0.1	97.6 ± 0.4	66.5 ± 0.4	45.2 ± 0.2	137.8 ± 0.6	154.9 ± 7.1	68.9 ± 0.5
DiLoCo-style	72.4 ± 0.6	66.4 ± 0.4	46.1 ± 0.3	96.2 ± 0.5	71.7 ± 0.4	46.2 ± 0.2	140.5 ± 0.6	202.2 ± 9.8	61.3 ± 0.2

All-round validation perplexity provides an important contrast to these last-10 results. Despite its strong late-training performance, FedDyn has substantially higher all-round validation perplexity than FedAvg, as does FedAdam. Across the factor tables, their all-round averages are approximately 1.7-6.4 times those of FedAvg, reaching 814.4 for FedDyn and 355.2 for FedAdam at $\beta = 1$. The round-wise results place these averages in context: after round 1, FedDyn and FedAdam have validation perplexities approximately 8 and 9 times that of FedAvg, respectively, using geometric means across configurations. FedDyn subsequently falls below FedAvg from round 4 onward and finishes 18% lower, whereas FedAdam remains above FedAvg throughout training and finishes 9% higher.

Domain composition and model depth change the absolute perplexity levels without changing the ordering of the three leading methods. Setting 1 yields the lowest last-10 perplexity and Setting 4 the highest for every algorithm. For FedAvg, these values differ by approximately a factor of 2.7. FedProx improves on FedAvg by approximately 24-26% in each setting. Increasing depth from 4 to 16 reduces last-10 perplexity by approximately 15-18% for FedAvg, FedAlign, FedSAM, FedExP, DiLoCo-style, and FedAdam, compared with 11% for SCAFFOLD, 8% for FedProx, and 4% for FedDyn. Consequently, the relative improvements over FedAvg narrow from 28% to 22% for FedProx, from 22% to 12% for FedDyn, and from 9% to 4% for SCAFFOLD.

H.2.2 Client-Count Ablation

Table 27 reports the client-count ablation described in Appendix G. We consider $K \in \{4, 8, 16, 32\}$ client counts at fixed $\beta = 0.7$, depth 4, and $E = 64$, averaged over the four domain settings, with $K/4$ clients sharing each primary domain and every client holding a $4/K$ share of the same domain pools.

Comments. With per-client work held fixed, FedAvg's last-10 validation perplexity decreases by approximately 4% from $K = 4$ to $K = 8$ (64.2 to 61.6), with little further change for the larger

Table 24: **Effect of local steps.** Training, validation, and last-10-round validation perplexity of the nine federated learning algorithms on Mamba2 at every E, averaged over the 24 configurations.

	$E = 64$			$E = 128$			$E = 256$		
Algorithm	Train	Val	Last-10	Train	Val	Last-10	Train	Val	Last-10
FedAvg	151.2 ± 0.5	108.5 ± 1.7	63.9 ± 0.2	100.9 ± 0.5	85.5 ± 1.1	48.7 ± 0.2	66.1 ± 0.3	72.9 ± 2.0	39.2 ± 0.4
FedProx	**137.7** ± 0.3	106.6 ± 1.7	**48.6** ± 0.2	**85.1** ± 0.4	**74.9** ± 0.9	**35.6** ± 0.4	**53.6** ± 0.3	**57.8** ± 1.0	**29.4** ± 0.2
SCAFFOLD	150.8 ± 0.4	**105.6** ± 1.7	61.0 ± 0.3	104.5 ± 0.3	85.3 ± 0.7	45.4 ± 0.3	74.5 ± 0.4	74.3 ± 2.0	35.4 ± 0.0
FedAdam	167.3 ± 0.5	261.5 ± 5.6	69.6 ± 0.5	113.0 ± 0.4	217.6 ± 6.6	53.1 ± 0.1	78.1 ± 0.3	200.9 ± 5.2	42.8 ± 0.1
FedAlign	151.2 ± 0.5	108.3 ± 2.1	63.6 ± 0.2	100.8 ± 0.4	85.1 ± 1.7	48.4 ± 0.3	66.0 ± 0.3	72.0 ± 2.3	39.0 ± 0.2
FedDyn	149.6 ± 0.1	257.9 ± 5.0	56.2 ± 0.3	90.6 ± 0.2	351.7 ± 43.0	38.8 ± 0.4	61.3 ± 0.1	431.3 ± 48.0	30.7 ± 0.0
FedSAM	151.3 ± 0.5	109.0 ± 1.5	63.8 ± 0.3	101.1 ± 0.4	85.6 ± 0.8	48.3 ± 0.1	66.8 ± 0.5	71.3 ± 1.8	38.6 ± 0.2
FedExP	147.1 ± 0.6	111.7 ± 1.2	65.3 ± 0.2	98.6 ± 0.5	91.5 ± 1.1	51.9 ± 0.3	65.0 ± 0.4	81.8 ± 5.6	42.1 ± 0.4
DiLoCo-style	144.6 ± 0.5	140.8 ± 5.4	63.5 ± 0.3	97.5 ± 0.6	113.4 ± 4.7	49.3 ± 0.3	67.0 ± 0.5	86.1 ± 3.2	40.9 ± 0.1

Table 25: **Effect of the domain assignment.** Training, validation, and last-10-round validation perplexity of the nine federated learning algorithms on Mamba2 for every domain setting (Setting 1: `arxiv`, `github`, `stackexchange`, `wikipedia`; Setting 2: `arxiv`, `c4`, `stackexchange`, `wikipedia`; Setting 3: `c4`, `common_crawl`, `github`, `wikipedia`; Setting 4: `arxiv`, `common_crawl`, `github`, `stackexchange`), averaged over the 18 configurations.

	Setting 1			Setting 2			Setting 3			Setting 4		
Algorithm	Train	Val	Last-10	Train	Val	Last-10	Train	Val	Last-10	Train	Val	Last-10
FedAvg	45.1 ± 0.2	53.2 ± 1.5	25.2 ± 0.3	107.3 ± 0.0	96.5 ± 4.2	51.7 ± 0.6	140.6 ± 1.0	91.4 ± 3.3	56.9 ± 0.1	131.3 ± 0.5	114.8 ± 1.1	68.7 ± 0.8
FedProx	**42.3** ± 0.2	**46.4** ± 0.2	**19.0** ± 0.2	**93.8** ± 0.3	**86.4** ± 0.3	**38.9** ± 0.3	**117.7** ± 0.6	**87.3** ± 3.1	**43.0** ± 0.3	**114.6** ± 0.5	**98.9** ± 1.7	**50.5** ± 0.9
SCAFFOLD	45.9 ± 0.2	51.6 ± 1.5	22.6 ± 0.4	112.0 ± 0.0	95.6 ± 3.4	47.5 ± 0.2	146.5 ± 0.7	93.1 ± 3.2	55.6 ± 0.1	135.2 ± 0.5	113.3 ± 1.3	63.3 ± 0.7
FedAdam	51.4 ± 0.1	169.0 ± 4.3	27.2 ± 0.2	118.2 ± 0.4	260.3 ± 6.6	54.2 ± 0.2	158.0 ± 1.1	237.0 ± 6.4	62.2 ± 0.2	150.2 ± 0.2	240.3 ± 7.6	77.1 ± 1.3
FedAlign	45.0 ± 0.0	50.8 ± 1.6	24.8 ± 0.2	107.2 ± 0.3	96.7 ± 3.4	51.4 ± 0.3	140.6 ± 0.9	91.8 ± 2.7	56.7 ± 0.1	131.3 ± 0.6	114.6 ± 1.2	68.5 ± 0.9
FedDyn	46.1 ± 0.0	339.0 ± 57.0	20.8 ± 0.2	101.3 ± 0.2	502.0 ± 35.6	42.3 ± 0.3	127.1 ± 0.5	248.9 ± 70.4	47.7 ± 0.2	127.4 ± 0.2	297.9 ± 23.3	56.7 ± 0.7
FedSAM	45.4 ± 0.2	51.8 ± 1.6	25.0 ± 0.2	107.8 ± 0.3	96.2 ± 3.6	50.9 ± 0.2	141.0 ± 0.9	91.9 ± 2.2	56.6 ± 0.1	131.6 ± 0.5	114.8 ± 1.2	68.6 ± 0.9
FedExP	43.9 ± 0.3	60.7 ± 3.9	27.0 ± 0.1	104.8 ± 0.2	105.5 ± 3.8	55.0 ± 0.3	137.3 ± 1.0	97.5 ± 4.2	60.2 ± 0.1	128.3 ± 0.5	116.3 ± 1.1	70.2 ± 0.7
DiLoCo-style	43.2 ± 0.3	62.8 ± 2.5	25.6 ± 0.3	104.6 ± 0.2	130.7 ± 9.2	52.3 ± 0.3	137.6 ± 1.1	110.2 ± 3.2	58.4 ± 0.5	126.9 ± 0.5	149.9 ± 5.4	68.6 ± 0.8

client counts (60.9 and 61.1). The training perplexity continues to decrease, from 144 at $K = 4$ to 129 at $K = 32$, potentially reflecting more frequent reuse of smaller local datasets. The three leading methods in the main grid -FedProx, FedDyn, and SCAFFOLD - remain the strongest on last-10 validation perplexity at every K. FedAlign and FedSAM remain within 1% of FedAvg, whereas FedAdam has 8-9$\%$ higher perplexity.

FedProx maintains a reduction of 22-24$\%$ compared to FedAvg across the evaluated values of K. In contrast, the reductions achieved by SCAFFOLD and FedDyn increase from 3% to 13% and from 7% to 18%, respectively, between $K = 4$ and $K = 32$. DiLoCo-style moves from 3% higher perplexity than FedAvg to 8% lower, while FedExP moves from approximate parity to a 6% reduction. As such, although the leading methods remain unchanged, their relative gains and the ordering of the remaining methods vary with client count.

One possible explanation for these method-dependent trends is that having multiple clients per primary domain at larger K produces a more stable aggregate update than the one-client-per-domain partition at $K = 4$. However, this interpretation is not directly tested by the ablation. Increasing K also changes the total client work, local dataset sizes, and data partitioning. These results therefore describe performance under the evaluated client-count protocol at a single (β, depth, E) setting, rather than isolating the effect of client count or establishing a particular algorithmic mechanism.

H.2.3 Model Backbone Ablation

Table 28 reports the different model backbones ablation described in Appendix G: the depth-4 model with some or all Mamba2 mixers replaced by attention layers (Table 10), trained on Setting 1 at $\beta = 0.7$, $E = 64$, and $K = 4$.

Comments. Under FedAvg the backbones order as `hybrid75` (26.5) $<$ `hybrid50` (29.3) $<$ `mamba2` (30.7) $<$ `hybrid50alt` (31.3) $<$ `hybrid25` (32.7) $<$ `attn0` (42.3) on the last-10 validation perplexity. Notably, one attention layer in the middle of the stack improves on the pure Mamba2 model by 14%. Attention layers placed first (`hybrid50alt`, `hybrid50`) or a mostly-

Table 26: **Effect of model depth.** Training, validation, and last-10-round validation perplexity of the nine federated learning algorithms for the depth-4 and depth-16 Mamba2 models, averaged over the 36 configurations.

	Depth 4 (14.7M)			Depth 16 (20.3M)		
Algorithm	Train	Val	Last-10	Train	Val	Last-10
FedAvg	109.8 ± 0.5	88.8 ± 0.5	54.8 ± 0.3	102.3 ± 0.4	89.1 ± 2.6	46.4 ± 0.2
FedProx	$\mathbf{93.2} \pm 0.4$	$\mathbf{79.0} \pm 1.3$	$\mathbf{39.5} \pm 0.4$	$\mathbf{91.0} \pm 0.3$	$\mathbf{80.5} \pm 1.2$	$\mathbf{36.2} \pm 0.1$
SCAFFOLD	112.8 ± 0.4	88.0 ± 1.4	50.1 ± 0.0	107.0 ± 0.4	88.8 ± 2.0	44.5 ± 0.2
FedAdam	122.6 ± 0.3	235.1 ± 4.1	59.8 ± 0.3	116.2 ± 0.5	218.2 ± 7.4	50.5 ± 0.3
FedAlign	109.7 ± 0.5	89.1 ± 1.9	54.4 ± 0.2	102.4 ± 0.4	87.9 ± 2.2	46.3 ± 0.3
FedDyn	100.2 ± 0.2	398.3 ± 43.9	42.8 ± 0.3	100.8 ± 0.3	295.7 ± 15.5	41.0 ± 0.1
FedSAM	110.1 ± 0.6	89.8 ± 1.7	54.3 ± 0.1	102.7 ± 0.4	87.5 ± 1.1	46.2 ± 0.4
FedExP	107.5 ± 0.4	93.1 ± 2.5	57.6 ± 0.4	99.6 ± 0.6	96.9 ± 2.6	48.5 ± 0.3
DiLoCo-style	108.8 ± 0.6	112.2 ± 0.7	56.3 ± 0.1	97.3 ± 0.5	114.7 ± 6.8	46.2 ± 0.4

Table 27: **Client-count ablation.** Training, validation, and last-10-round validation perplexity of the nine algorithms with $K \in \{4, 8, 16, 32\}$ clients at $\beta = 0.7$, depth 4, $E = 64$, averaged over the four domain settings.

	$K = 4$			$K = 8$			$K = 16$			$K = 32$		
Algorithm	Train	Val	Last-10	Train	Val	Last-10	Train	Val	Last-10	Train	Val	Last-10
FedAvg	144.4 ± 0.3	89.7 ± 0.9	64.2 ± 0.4	141.4 ± 0.4	87.3 ± 0.5	61.6 ± 0.2	136.9 ± 0.6	86.3 ± 0.7	60.9 ± 0.3	129.0 ± 0.3	86.2 ± 0.6	61.1 ± 0.2
FedProx	$\mathbf{131.6} \pm 0.2$	$\mathbf{82.4} \pm 0.8$	$\mathbf{49.0} \pm 0.2$	$\mathbf{129.1} \pm 0.5$	$\mathbf{80.6} \pm 0.3$	$\mathbf{47.3} \pm 0.3$	$\mathbf{125.2} \pm 0.4$	$\mathbf{79.9} \pm 0.5$	$\mathbf{46.8} \pm 0.2$	$\mathbf{118.5} \pm 0.3$	80.5 ± 0.4	$\mathbf{47.6} \pm 0.1$
SCAFFOLD	143.7 ± 0.2	87.6 ± 0.9	62.4 ± 0.4	139.5 ± 0.5	83.8 ± 0.5	58.7 ± 0.2	134.2 ± 0.7	81.2 ± 0.6	55.7 ± 0.4	126.8 ± 0.4	$\mathbf{79.7} \pm 0.5$	53.0 ± 0.2
FedAdam	159.9 ± 0.1	187.1 ± 2.0	69.5 ± 0.6	156.2 ± 0.6	183.0 ± 1.2	67.0 ± 0.4	151.3 ± 0.5	181.7 ± 2.0	66.1 ± 0.4	143.7 ± 0.2	182.8 ± 1.7	66.5 ± 0.5
FedAlign	144.4 ± 0.2	89.7 ± 0.9	64.2 ± 0.4	141.4 ± 0.4	87.3 ± 0.5	61.6 ± 0.2	136.9 ± 0.6	86.3 ± 0.7	60.9 ± 0.3	129.0 ± 0.3	86.2 ± 0.6	61.1 ± 0.2
FedDyn	164.5 ± 0.4	161.8 ± 15.9	59.4 ± 0.5	163.0 ± 0.7	156.9 ± 5.7	54.1 ± 0.4	162.2 ± 0.2	151.4 ± 10.3	51.6 ± 0.4	162.0 ± 0.2	151.9 ± 11.1	50.4 ± 0.3
FedSAM	145.1 ± 0.4	90.3 ± 0.8	64.3 ± 0.2	142.2 ± 0.5	88.1 ± 0.5	62.1 ± 0.2	137.5 ± 0.7	86.9 ± 0.7	61.2 ± 0.3	129.8 ± 0.4	86.8 ± 0.6	61.5 ± 0.2
FedExP	141.7 ± 0.1	89.9 ± 1.0	63.9 ± 0.7	137.1 ± 0.5	85.9 ± 0.4	59.5 ± 0.5	131.6 ± 0.8	84.0 ± 0.8	57.7 ± 0.4	123.6 ± 0.5	83.2 ± 0.5	57.3 ± 0.4
DiLoCo-style	138.7 ± 0.2	94.6 ± 1.4	65.8 ± 0.1	134.0 ± 0.8	90.2 ± 0.8	61.0 ± 0.3	128.3 ± 0.3	87.4 ± 0.9	58.0 ± 0.6	120.8 ± 0.6	86.5 ± 0.5	56.5 ± 0.1

attention stack lose that gain, and the attention-only control is 38% worse than pure Mamba2 at this model size and sequence length. The ranking of the algorithms transfers from the pure-SSM backbone to every hybrid, with two exceptions. FedProx is the best method on all six backbones (-17 to -21% relative to FedAvg), and FedAlign and FedSAM stay within 3% of FedAvg everywhere. SCAFFOLD, in contrast, gains 13–15% on the hybrids with two or three Mamba2 layers and 10% on `hybrid25`, but only 4% on `mamba2` and 1% on `attn0`, and FedDyn at its fixed α is unstable across architectures (-2 to -6% on `mamba2`, `hybrid50`, and `attn0`, $+6$ to $+11\%$ on the other three hybrids). The server-step methods are the second exception: their penalty grows as Mamba2 layers are replaced, from $+8\%$ (FedAdam) and $+1\%$ (DiLoCo-style) on `mamba2` to $+40\%$ and $+14\%$ on `attn0`, whereas FedExP, neutral on `mamba2`, is 3-6% better than FedAvg on every backbone that contains attention.

H.3 Model Merging

This subsection reports the one-shot merging experiments referenced in Appendix F. Following the framework of Appendix G.5, the pretrained Mamba2-130M, 370M, and 780M checkpoints (Table 11) are fine-tuned separately on AG News, DBpedia, TREC, and IMDB (Table 6) with the recipe of Table 16, and the four specialists $\theta_m = \theta^0 + \mathbf{T}_m$ saved after 1, 2, and 4 epochs are merged by weight averaging ($p_m = 1/4$), Task Arithmetic ($\theta^0 + \lambda \sum_m \mathbf{T}_m$), and TIES-Merging (trim each task vector to the fraction d of its largest-magnitude entries, elect signs, average the agreeing entries, and scale the result by λ). Every merged model is evaluated on all four tasks by causal-LM label scoring, and we report the accuracy averaged over the four tasks (mean $\pm$ standard deviation over three fine-tuning seeds), as well as the accuracies of the pretrained model and the specialists evaluated on their own and on the other tasks. Task Arithmetic and TIES-Merging are shown at the best of the evaluated λ and (λ, d) for each checkpoint and epoch. Tables 31 and 32 give the full sweeps, Table 30 the per-task accuracies after four epochs, and Table 33 a variant of the 130M experiment in which the specialists are fine-tuned with the FedProx proximal term.

Comments. After four epochs, specialists average 89-91% accuracy on their own tasks but show little transfer to the other tasks. Task Arithmetic and TIES-Merging outperform weight averaging, with the

Table 28: **Backbone ablation.** Validation and last-10-round validation perplexity of the nine algorithms on the six depth-4 backbones of Table 10 (Setting 1, $\beta = 0.7$, $E = 64$, $K = 4$, the second header row gives the fraction of Mamba2 layers).

	mamba2 SSM 1.00		hybrid75 SSM 0.75		hybrid50 SSM 0.50		hybrid50alt SSM 0.50		hybrid25 SSM 0.25		attn0 SSM 0.00	
Algorithm	Val	Last-10	Val	Last-10	Val	Last-10	Val	Last-10	Val	Last-10	Val	Last-10
FedAvg	43.4 ± 0.4	30.7 ± 0.3	41.5 ± 0.8	26.5 ± 0.5	43.3 ± 0.6	29.3 ± 0.4	52.6 ± 1.2	31.3 ± 0.2	54.4 ± 1.2	32.7 ± 0.7	69.6 ± 1.1	42.3 ± 1.1
FedProx	**41.9** ± 0.4	**25.4** ± 0.2	39.8 ± 0.4	**21.0** ± 0.2	41.4 ± 0.3	**23.6** ± 0.2	50.4 ± 0.7	**24.7** ± 0.4	**52.2** ± 1.3	**26.8** ± 0.9	**67.7** ± 1.0	**34.8** ± 0.9
SCAFFOLD	42.2 ± 0.4	29.4 ± 0.3	**39.2** ± 1.0	22.5 ± 0.5	**41.3** ± 0.7	25.4 ± 0.8	**50.3** ± 1.2	27.1 ± 0.3	52.5 ± 1.2	29.6 ± 1.3	68.3 ± 1.0	41.8 ± 0.9
FedAdam	116.0 ± 1.1	33.2 ± 0.2	116.5 ± 3.4	30.7 ± 0.5	120.4 ± 2.8	32.5 ± 0.5	141.7 ± 3.9	38.8 ± 0.2	144.4 ± 2.3	40.3 ± 0.6	180.8 ± 4.6	59.4 ± 1.1
FedAlign	43.4 ± 0.4	30.7 ± 0.3	41.7 ± 1.0	26.7 ± 0.6	43.3 ± 0.6	29.3 ± 0.4	52.5 ± 1.2	31.2 ± 0.1	54.3 ± 1.1	32.7 ± 0.7	69.6 ± 1.2	42.0 ± 0.9
FedDyn	76.2 ± 6.4	29.6 ± 0.5	68.4 ± 2.1	28.2 ± 0.4	69.5 ± 5.5	27.5 ± 1.0	114.0 ± 18.4	34.8 ± 1.0	101.8 ± 16.0	35.5 ± 0.6	132.3 ± 22.7	41.6 ± 1.4
FedSAM	43.8 ± 0.4	30.8 ± 0.3	41.8 ± 0.8	26.7 ± 0.6	43.6 ± 0.7	29.6 ± 0.5	53.7 ± 0.9	32.1 ± 0.1	55.2 ± 1.4	33.4 ± 1.2	70.5 ± 0.9	42.7 ± 0.4
FedExP	43.6 ± 0.2	30.5 ± 0.0	41.0 ± 0.7	25.7 ± 0.4	43.1 ± 0.5	28.3 ± 0.3	51.8 ± 1.2	30.0 ± 0.2	53.5 ± 1.1	30.9 ± 0.7	68.9 ± 0.9	40.3 ± 1.3
DiLoCo-style	44.8 ± 0.2	31.1 ± 0.4	42.0 ± 0.9	26.3 ± 0.5	43.7 ± 0.4	29.0 ± 0.4	56.6 ± 2.6	33.5 ± 0.9	57.9 ± 0.9	35.2 ± 0.5	78.6 ± 2.5	48.1 ± 1.2

Table 29: **One-shot merging of Mamba2 task specialists: task-averaged accuracy (%).** For each public checkpoint, the accuracy averaged over AG News, DBpedia, TREC, and IMDB of the pretrained model, of the four specialists on their own task and on the other three tasks, and of the merged model produced by weight averaging, Task Arithmetic, and TIES-Merging from specialists fine-tuned for 1, 2, or 4 epochs.

Checkpoint	Model	1 epoch	2 epochs	4 epochs	Selected λ / (λ, d)
Mamba2-130M	Zero-shot (pretrained)		25.7 ± 0.0		–
	Specialists (own task)	81.8 ± 1.8	88.4 ± 0.5	89.2 ± 0.2	–
	Specialists (other tasks)	23.0 ± 0.2	23.6 ± 0.1	23.4 ± 0.3	–
	Weight averaging	34.4 ± 2.1	36.3 ± 1.7	37.4 ± 0.6	–
	Task Arithmetic	**53.5** ± 4.6	**61.9** ± 5.1	**61.3** ± 3.8	1 / 1 / 0.7
	TIES-Merging	47.0 ± 2.0	55.0 ± 3.9	56.6 ± 3.6	$(1.3, 0.1)$ / $(1.3, 0.1)$ / $(1.3, 0.1)$
Mamba2-370M	Zero-shot (pretrained)		24.2 ± 0.1		–
	Specialists (own task)	86.6 ± 2.3	90.7 ± 0.1	90.8 ± 0.1	–
	Specialists (other tasks)	24.8 ± 0.5	24.8 ± 0.3	25.4 ± 0.1	–
	Weight averaging	45.6 ± 1.0	49.6 ± 0.5	50.4 ± 0.4	–
	Task Arithmetic	61.1 ± 2.5	**71.0** ± 1.7	**71.8** ± 1.2	0.7 / 0.7 / 0.7
	TIES-Merging	**62.6** ± 1.6	69.8 ± 2.0	70.9 ± 1.8	$(1.3, 0.5)$ / $(1.3, 0.1)$ / $(1.3, 0.1)$
Mamba2-780M	Zero-shot (pretrained)		24.7 ± 0.0		–
	Specialists (own task)	82.4 ± 3.3	89.4 ± 0.5	90.8 ± 0.3	–
	Specialists (other tasks)	24.5 ± 1.0	24.6 ± 0.1	24.9 ± 0.2	–
	Weight averaging	39.4 ± 2.8	45.4 ± 1.4	51.7 ± 1.5	–
	Task Arithmetic	59.9 ± 2.8	61.9 ± 5.6	71.4 ± 2.5	0.7 / 0.7 / 0.7
	TIES-Merging	**60.3** ± 3.0	**64.5** ± 7.6	**72.1** ± 3.5	$(1.3, 0.1)$ / $(1.3, 0.1)$ / $(1.3, 0.1)$

best evaluated merges reaching 61.3, 71.8, and 72.1% for the 130M, 370M, and 780M checkpoints, respectively, compared with 37.4, 50.4, and 51.7% for averaging (Table 29). Larger checkpoints narrow the gap to specialists, although substantial deficits remain, especially on DBpedia and TREC (Table 30). The sweeps favor Task Arithmetic scales of $\lambda = 0.7$-1, rather than the averaging scale $\lambda = 0.25$, and sparse, rescaled TIES updates, with $(\lambda, d) = (1.3, 0.1)$ performing best among the four-epoch settings evaluated (Tables 31 and 32). Additionally, fine-tuning with the proximal penalty $\frac{\mu}{2}\|\theta - \theta^0\|_F^2$ reduces four-epoch averaging accuracy from 37.4% at $\mu = 0$ to 26.2% at $\mu = 10^{-1}$, near the pretrained baseline (Table 33).

Relation to the analysis. These results are consistent with the trade-off in the local loss control (466) between the linear alignment term $\langle \nabla f(\theta^0), \theta_{\text{merge}} - \theta^0 \rangle_F$ and the quadratic smoothness remainder. Weight averaging preserves neighborhood containment but rescales each task vector by $1/M$. For the 780M model parameters, $\|\sum_m \mathbf{T}_m\|_F^2 / \sum_m \|\mathbf{T}_m\|_F^2 = 1.04$-$1.07$, indicating little net alignment among the task vectors. Together with the scale sweeps, this suggests that averaging suppresses useful task-specific adaptation: larger displacements can improve accuracy despite weaker containment guarantees, illustrating why containment alone does not ensure predictive performance (Appendix F.1). The proximal results suggest a similar trade-off. Limiting specialist drift can also suppress useful adaptation, and, unlike iterative federated learning, one-shot merging has no subsequent rounds to recover it. Finally, the uniform smoothness bound does not explain the larger accuracy gaps on DBpedia and TREC, motivating further investigation of the sensitivity-aware per-block scales discussed in Appendix F.2.

Table 30: **Per-task accuracy (%) of the merged models after 4 specialist epochs.** Columns are the evaluation tasks for each checkpoint, rows are the pretrained model, the specialist fine-tuned on that task, and the three merges.

	Mamba2-130M				Mamba2-370M				Mamba2-780M			
Model	AG News	DBpedia	TREC	IMDB	AG News	DBpedia	TREC	IMDB	AG News	DBpedia	TREC	IMDB
Zero-shot (pretrained)	25.0 ± 0.0	9.9 ± 0.0	16.8 ± 0.0	51.1 ± 0.2	24.3 ± 0.0	8.1 ± 0.0	13.4 ± 0.2	51.1 ± 0.1	25.3 ± 0.0	8.7 ± 0.0	15.1 ± 0.1	49.9 ± 0.1
Specialists (own task)	89.5 ± 0.3	96.4 ± 0.2	86.7 ± 0.6	83.9 ± 0.4	90.8 ± 0.2	97.6 ± 0.2	88.7 ± 0.5	86.0 ± 0.1	90.8 ± 0.1	97.9 ± 0.2	88.4 ± 1.0	86.1 ± 0.5
Weight averaging	37.9 ± 2.6	21.6 ± 3.9	34.9 ± 5.3	55.3 ± 0.7	42.4 ± 2.3	36.8 ± 0.9	63.5 ± 4.4	59.1 ± 5.4	44.4 ± 0.6	41.1 ± 3.3	54.5 ± 1.6	66.8 ± 3.2
Task Arithmetic	64.7 ± 8.5	**53.3** ± 13.8	**61.7** ± 7.3	**65.5** ± 2.6	75.3 ± 7.0	**74.4** ± 2.8	**65.1** ± 2.8	**72.5** ± 2.2	80.6 ± 2.3	65.9 ± 7.9	**57.9** ± 7.7	81.1 ± 1.7
TIES-Merging	**72.2** ± 8.4	51.7 ± 17.9	46.2 ± 13.1	56.2 ± 0.1	**77.3** ± 6.8	73.4 ± 3.6	62.9 ± 3.0	69.9 ± 0.8	**83.4** ± 1.4	**69.5** ± 8.2	52.4 ± 11.6	**83.1** ± 1.1

Table 31: **Task Arithmetic scale sweep: task-averaged accuracy (%).** $\theta_{\mathrm{TA}} = \theta^0 + \lambda \sum_m \mathbf{T}_m$ over the four specialists, so $\lambda = 1/M = 0.25$ would be weight averaging and $\lambda = 1$ adds every task vector at full length. The 780M sweep evaluated only $\lambda \in \{0.7, 1.0\}$.

Checkpoint	Epochs	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.7$	$\lambda = 1$	$\lambda = 1.3$
Mamba2-130M	1	31.2 ± 1.2	38.1 ± 3.2	48.3 ± 3.3	52.4 ± 4.2	**53.5** ± 4.6	51.7 ± 3.8
	2	32.6 ± 1.1	41.3 ± 1.9	55.5 ± 2.0	61.9 ± 4.1	**61.9** ± 5.1	55.8 ± 5.4
	4	32.8 ± 0.8	42.7 ± 0.8	55.8 ± 2.1	**61.3** ± 3.8	57.2 ± 6.2	49.3 ± 7.0
Mamba2-370M	1	40.2 ± 1.9	49.5 ± 0.6	59.4 ± 2.9	**61.1** ± 2.5	58.6 ± 3.4	54.1 ± 5.6
	2	43.5 ± 1.5	54.7 ± 0.4	67.6 ± 1.0	**71.0** ± 1.7	68.2 ± 2.2	59.6 ± 2.7
	4	44.5 ± 1.5	55.4 ± 0.5	68.6 ± 0.5	**71.8** ± 1.2	66.3 ± 2.5	54.0 ± 3.3
Mamba2-780M	1	–	–	–	**59.9** ± 2.8	58.1 ± 3.9	–
	2	–	–	–	**61.9** ± 5.6	57.9 ± 5.2	–
	4	–	–	–	**71.4** ± 2.5	69.5 ± 4.7	–

Table 32: **TIES-Merging sweep after 4 specialist epochs: task-averaged accuracy (%).** Rows are the output scale λ applied to the task vector, columns the fraction d of largest-magnitude entries kept per task vector. The 780M sweep evaluated only $\lambda = 1.3$ with $d \in \{0.1, 0.5\}$ in addition to the default $(\lambda, d) = (1.0, 0.2)$.

Checkpoint	λ	$d = 0.1$	$d = 0.2$	$d = 0.3$	$d = 0.5$
Mamba2-130M	0.5	34.7 ± 1.5	33.7 ± 0.9	32.6 ± 0.8	32.1 ± 0.8
	0.8	46.1 ± 1.3	44.0 ± 1.1	42.2 ± 1.1	41.2 ± 1.4
	1	51.7 ± 1.6	49.6 ± 1.7	47.9 ± 1.9	46.9 ± 1.3
	1.3	**56.6** ± 3.6	54.7 ± 2.6	53.7 ± 2.3	52.9 ± 1.8
Mamba2-370M	0.5	45.4 ± 0.6	46.4 ± 0.3	45.0 ± 1.5	42.0 ± 1.3
	0.8	63.0 ± 1.7	62.6 ± 0.8	60.9 ± 1.1	57.5 ± 1.3
	1	68.0 ± 1.7	67.8 ± 1.2	67.1 ± 1.4	64.7 ± 1.8
	1.3	**70.9** ± 1.8	70.4 ± 1.7	70.6 ± 1.5	70.2 ± 1.6
Mamba2-780M	1	–	67.8 ± 4.2	–	–
	1.3	**72.1** ± 3.5	–	–	67.2 ± 3.3

Table 33: **Weight averaging of proximally fine-tuned specialists (Mamba2-130M): task-averaged accuracy (%).** Each specialist is fine-tuned with the FedProx local objective $\mathcal{L}_m(\theta) + \frac{\mu}{2}\|\theta - \theta^0\|_F^2$ for 1, 2, or 4 epochs and the four specialists are merged by weight averaging

	Specialists (own task)			Weight averaging		
μ	1 epoch	2 epochs	4 epochs	1 epoch	2 epochs	4 epochs
0 (reference)	81.8 ± 1.8	**88.4** ± 0.5	**89.2** ± 0.2	**34.4** ± 2.1	**36.3** ± 1.7	**37.4** ± 0.6
10^{-5}	**82.6** ± 3.0	88.1 ± 0.9	88.9 ± 0.6	33.3 ± 0.9	35.7 ± 0.6	37.3 ± 0.6
10^{-4}	82.5 ± 3.5	87.6 ± 0.3	88.1 ± 0.3	32.5 ± 2.0	33.1 ± 1.1	34.0 ± 0.8
10^{-3}	80.6 ± 2.6	85.3 ± 0.6	86.1 ± 0.1	30.4 ± 1.1	30.3 ± 0.5	30.6 ± 0.5
10^{-2}	75.3 ± 3.1	79.7 ± 0.9	81.9 ± 0.4	27.1 ± 0.7	27.6 ± 0.1	28.0 ± 0.1
10^{-1}	65.0 ± 3.3	72.6 ± 0.9	74.3 ± 0.5	26.3 ± 0.3	26.2 ± 0.2	26.2 ± 0.3