ADAPTIVE SELECTION OF LORA COMPONENTS IN PRIVACY-PRESERVING FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Differentially private federated fine-tuning of large models with LoRA suffers from aggregation error caused by LoRA's multiplicative structure, which is further amplified by DP noise and degrades both stability and accuracy. Existing remedies apply a single update mode uniformly across all layers and all communication rounds (or alternate them on a fixed schedule), ignoring both the structural asymmetry between the two LoRA factors and the round-wise dynamics of training. We propose AS-LoRA, an adaptive framework defined by three axes *(i)* layer-wise freedom, in which each layer independently selects its active component, *(ii)* round-wise adaptivity, in which the selection updates over communication rounds, and *(iii)* a curvature-aware score derived from a second-order approximation of the loss. Theoretically, AS-LoRA eliminates the reconstruction-error floor of layer-tied schedules, accelerates convergence, implicitly biases solutions toward flatter minima, and incurs no additional privacy cost. Across GLUE, SQuAD, CIFAR-100, and Tiny-ImageNet under strict DP budgets and non-IID partitions, AS-LoRA improves over the federated LoRA baselines by up to $+9.68$ pp on MNLI (m) and $+4.58$ pp on SQuAD v2.0. Compared with SVD-based aggregation, it attains comparable accuracy with about 1% additional client-side compute, two orders of magnitude lower server-side aggregation cost, and negligible communication cost. Code for the proposed method is available at `https://anonymous.4open.science/r/as_lora-F75F/`.

1 INTRODUCTION

Federated learning (FL) enables collaborative training over decentralized data without exposing raw samples (McMahan et al., 2017; Bonawitz et al., 2019), yet the communicated updates remain vulnerable to attacks such as Deep Leakage from Gradients (Zhu et al., 2019) and Membership Inference (Shokri et al., 2017). Differential privacy (DP) has thus become a standard safeguard in this setting (Zheng et al., 2021; Naseri et al., 2020). Meanwhile, as foundation models such as GPT-4 (Achiam et al., 2023), LLaMA (Touvron et al., 2023), PaLM2 (Anil et al., 2023), ViT (Dosovitskiy et al., 2020), and Swin-T (Liu et al., 2021) continue to scale, fine-tuning them on resource-constrained clients quickly becomes prohibitive. Low-Rank Adaptation (LoRA) (Hu et al., 2022), which adapts a frozen weight W_0 via a small low-rank update $W = W_0 + BA$, has therefore become the predominant approach for Parameter-Efficient Fine-Tuning (PEFT) in FL with foundation models, incurring minimal performance degradation compared to full fine-tuning.

However, applying LoRA in private FL is non-trivial. Its multiplicative structure induces an *aggregation error* across clients, which is further amplified into a *quadratic noise term* under DP-SGD, and we defer the detailed mechanism to Section 2. *FedLoRA* (Zhang et al., 2024), which is the most direct adaptation of LoRA to FL, suffers from both effects and degrades sharply under tight privacy budgets and non-IID partitions. Several remedies have since been proposed to alleviate those issues. *FFA-LoRA* (Sun et al., 2024) freezes A and trains only B, *RoLoRA* (Chen et al., 2025) alternates updates between A and B across rounds, and *FedSVD* (Li et al., 2025) also trains only B while applying a server-side Singular Value Decomposition (SVD). Yet all of them adopt structurally simple update strategies that impose the *same mode on every layer* and follow *temporally static patterns*, which overlook both the structural asymmetry between A and B, and the round-wise variation of their relative importance. This motivates an adaptive framework in which the active LoRA component is selected per layer and per round, as depicted in Figure 1.

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.

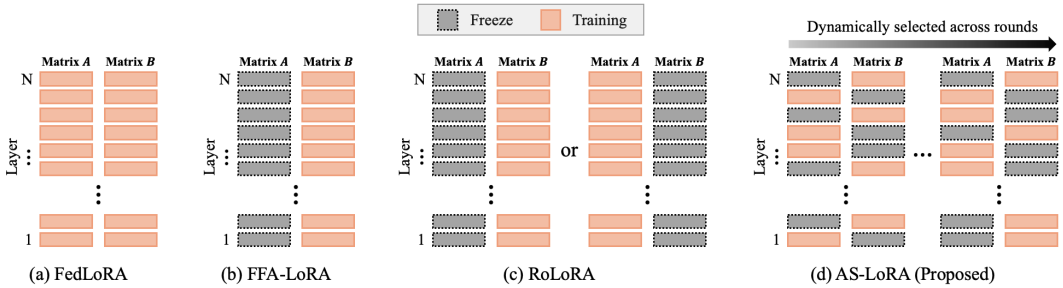


Figure 1: Comparison of update patterns in FedLoRA, FFA-LoRA, RoLoRA, and the proposed Adaptive Selection LoRA (AS-LoRA).

In fact, prior work (Hayou et al., 2024) shows that the two low-rank matrices contribute asymmetrically to the optimization process across training stages, owing to their different initialization schemes and structural characteristics. To examine whether this asymmetry can be handled by simple fixed schedules, we conduct preliminary experiments that compare several alternating patterns including BA, BBA, and BAA, as shown in Table 1. No single schedule attains the best performance across all tasks, which indicates that fixed and heuristic alternating schedules are inherently suboptimal, and that the choice of which low-rank component to optimize should instead be made adaptively according to current training dynamics. Consistent with this observation, the active LoRA component selected by our proposed AS-LoRA varies dynamically across both layers and communication rounds throughout training, as illustrated in Figure 2, which presents a representative trajectory whose full details are deferred to Appendix J.

Table 1: Performance comparison of fixed alternating LoRA optimization schedules on 3 GLUE tasks. Best results are shown in **bold** and second-best results are underlined.

Schedule	MNLI (m)	MNLI (mm)	SST-2	QQP
BA (Chen et al., 2025)	**81.17**	81.96	92.09	84.45
BAA	80.26	81.20	**92.89**	84.21
BBA	81.10	**81.98**	92.30	84.45
BBAA	80.78	81.55	92.66	**84.49**

Building on those observations, we propose AS-LoRA (Adaptive Selection LoRA), and the main contributions of this paper are summarized as follows.

Layer-wise adaptive LoRA optimization. To the best of our knowledge, this is the first work that extends alternating LoRA optimization to a adaptive selection framework. We further prove that this layer-wise freedom eliminates the irreducible reconstruction-error floor that any layer-tied schedule must incur.

Round-wise adaptive mode selection. We introduce a temporally adaptive mechanism that updates the selected LoRA component across communication rounds based on the scores aggregated in the previous round, which allows the model to adjust its optimization strategy as training progresses.

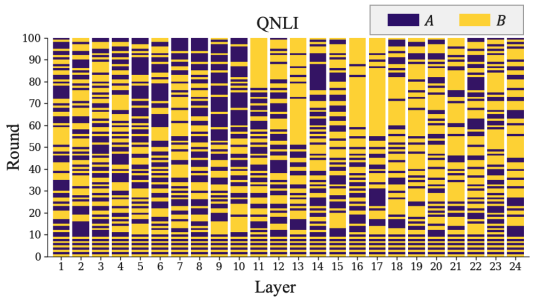


Figure 2: Selection of LoRA components with AS-LoRA (ours) on QNLI across layers/rounds. Different layers favor different components, and the preferred component varies across rounds, supporting the need for adaptive selection.

Curvature-aware scoring function. We design a scoring function derived from a second-order approximation of the loss, which captures curvature-aware component importance. We prove that this score-driven selection accelerates convergence with a quantifiable gain over fixed schedules and implicitly biases solutions toward flatter minima, which we further verify empirically through loss-landscape and sharpness analyses. To handle the channel-wise gradient outliers that are characteristic of LoRA (Jung et al., 2025), we further apply a Gaussian random projection (Johnson et al., 1984), which yields more reliable score estimates.

Privacy at no extra cost. We prove that the data-dependent component selection of AS-LoRA incurs no additional privacy cost beyond standard DP-SGD, which follows from the post-processing invariance of DP.

2 BACKGROUND

Differential privacy (Dwork & Roth, 2014). Formally, a randomized mechanism $\mathcal{M}$ is said to satisfy (ϵ, δ)-differential privacy if, for any two neighboring datasets D and D' that differ in exactly one data record, and for any measurable subset E of the output space, the following condition holds: $\Pr[\mathcal{M}(D) \in E] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in E] + \delta$. A common approach to achieving differential privacy in federated learning is differentially private stochastic gradient descent (DP-SGD) (Abadi et al., 2016). Formally, let $g_i = \nabla_\theta \ell(\theta; x_i)$ denote the gradient computed from a single training example x_i. DP-SGD first applies per-sample gradient clipping. And the clipped gradients are then aggregated and perturbed with Gaussian noise: *(i)* $g_i^{\text{clip}} = g_i \cdot \min\left(1, \frac{C}{\|g_i\|_2}\right)$, *(ii)* $g^{\text{noisy}} = \frac{1}{B}\left(\sum_{i=1}^{B} g_i^{\text{clip}} + \mathcal{N}\left(0, \sigma^2 C^2 \mathbf{I}\right)\right)$, where C is a predefined clipping norm that bounds the sensitivity, B is the batch size, σ is the noise multiplier, and $\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$ denotes isotropic Gaussian noise. Finally, the model parameters are updated using the noisy gradient: $\theta_{t+1} = \theta_t - \eta\, \bar{g}$, where η is the learning rate.

Aggregation error in federated LoRA (Wang et al., 2024). Given a pre-trained weight matrix $W_0 \in \mathbb{R}^{d_{out} \times d_{in}}$, LoRA parameterizes the adapted weight as $W = W_0 + BA$, where $B \in \mathbb{R}^{d_{out} \times r}$ and $A \in \mathbb{R}^{r \times d_{in}}$. In FL, the server aggregates local LoRAs $\{(A_k, B_k)\}_{k=1}^{K}$ via averaging as $\bar{A} = \frac{1}{K}\sum_{k=1}^{K} A_k, \quad \bar{B} = \frac{1}{K}\sum_{k=1}^{K} B_k$. However, due to the multiplicative structure of LoRA, $\frac{1}{K}\sum_{k=1}^{K} B_k A_k \neq \bar{B}\,\bar{A}$. This mismatch leads to an aggregation error that accumulates across communication rounds, and becomes more pronounced under heterogeneous client data distributions.

DP noise amplification in LoRA (Sun et al., 2024). When DP-SGD is applied to LoRA parameters, Gaussian noise is added to the gradients. Let ΔB and ΔA denote the updates to B and A in a given round, which consist of a true update component and a DP noise component: $\Delta B = \Delta B^{\star} + \mathcal{N}_B, \quad \Delta A = \Delta A^{\star} + \mathcal{N}_A$. The resulting update to the effective weight matrix is given by $\Delta W = B\Delta A^{\star} + \Delta B^{\star} A + B\mathcal{N}_A + \mathcal{N}_B A + \mathcal{N}_B \mathcal{N}_A$. As illustrated in Figure 3, different noise components exhibit distinct scaling behaviors with respect to the noise scale σ. The full-model noise $\|\mathcal{N}_W\|_F$ grows linearly, whereas the multiplicative cross-term $\|\mathcal{N}_B\mathcal{N}_A\|_F$ exhibits quadratic growth and becomes dominant as σ increases. This phenomenon leads to *noise amplification* in the effective model update, which can significantly degrade convergence and stability.

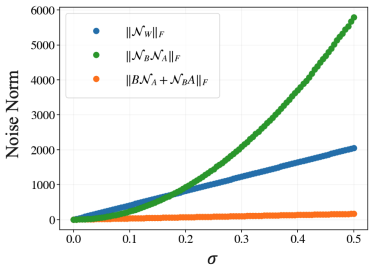


Figure 3: Scaling behavior of different noise components with respect to the noise scale σ.

Additional related works are introduced in Appendix A.

3 METHOD

In this section, we introduce our AS-LoRA framework. We first describe the problem formulation of FL with LoRA, and then present the score computation algorithm $\mathcal{S}(\cdot)$ and the mode selection algorithm $\mathcal{M}(\cdot)$ that together determine which LoRA component is activated at each layer and communication round. The full algorithmic procedure and visual illustration of AS-LoRA are deferred to Appendix B.

Problem formulation. We consider a FL setting with K clients and a central server. Each client holds a local dataset that is not shared with others, and the goal is to collaboratively fine-tune a large pre-trained model while preserving data privacy. Following Section 2, the adapted weight matrix is parameterized as $W = W_0 + BA$, where A and B denote the two low-rank components. Since LoRA is applied to multiple layers of the model, we further decompose the update into layer-wise adapters. Specifically, for the n-th layer at round t, the adapted weight can be written as $W_t(n) = W_0(n) + B^t(n)A^t(n),\ n \in [N]$, where N denotes the total number of LoRA layers.

Score computation algorithm. To decide which LoRA component to activate at each communication round, we score each candidate block $a \in \{A, B\}$ of layer n by the second-order Taylor approximation of the loss decrease after one gradient step $W_t - \eta \nabla_{a(n)} L(W_t)$, which yields

$$\mathcal{S}_{a(n)}(W_t) = \|\nabla_{a(n)} L(W_t)\|^2 - \frac{\eta}{2} \nabla_{a(n)} L(W_t)^{\top} H_{a(n)}(W_t)\, \nabla_{a(n)} L(W_t), \tag{1}$$

where $H_{a(n)}(W_t)$ is the Hessian with respect to block $a(n)$ and the client index k is omitted for clarity. The *first term* extends the Gauss–Southwell rule of block coordinate descent (BCD) (Nesterov, 2012), which selects the block with the largest gradient magnitude and converges strictly faster than uniform block selection; the *second term* is a curvature penalty that mirrors the implicit bias of sharpness-aware minimization (SAM) (Foret et al., 2021), discouraging steps that would overshoot a sharp region of the loss surface. Maximizing $\mathcal{S}_{a(n)}$ therefore selects the component with the largest expected one-step loss decrease while preferring flatter regions, which we formalize in Theorem 3 and Theorem 4.

Estimation from privatized gradients. Neither the gradient nor the Hessian in equation 1 may be read under DP-SGD; the only gradient a client may use is the privatized one released at each local step, $\bar{g}_a^i = g_a^{\mathrm{clip},i} + \xi_i,\ \xi_i \sim \mathcal{N}\big(0, (\sigma C/B)^2 I\big),\ i \in [\tau]$, with $g_a^{\mathrm{clip},i}$ the per-sample-clipped, batch-averaged gradient of Section 2. We estimate both terms from these outputs only. The noise inflates the squared norm by a public floor, $\mathbb{E}\|\bar{g}_a^i\|^2 = \|g_a^{\mathrm{clip},i}\|^2 + F_a$ with $F_a := D_a(\sigma C/B)^2$ and D_a the number of entries of the block, so the first term is estimated by

$$S_{g,a} = \frac{1}{\tau}\sum_{i=1}^{\tau} \max\big(\|\bar{g}_a^i\|^2 - F_a,\ 0\big). \tag{2}$$

For the curvature term we avoid any further query on the private data and use a secant estimate from consecutive steps (Barzilai & Borwein, 1988; Nocedal & Wright, 2006): since the active block moves by $s_i = -\eta\,\bar{g}_a^{i-1}$, the relation $H_a s_i \approx y_i := \bar{g}_a^i - \bar{g}_a^{i-1}$ gives the block Rayleigh quotient $\lambda_a = \nabla_a L^\top H_a \nabla_a L / \|\nabla_a L\|^2$ and the reported score as

$$\hat{\lambda}_a = \frac{\sum_{i=2}^{\tau}\big(\langle s_i, y_i\rangle - \eta F_a\big)}{\sum_{i=2}^{\tau}\big(\|s_i\|^2 - \eta^2 F_a\big)}, \qquad S_a = S_{g,a}\Big(1 - \frac{\eta}{2}\,\hat{\lambda}_a\Big), \tag{3}$$

where the numerator and the denominator are debiased by the floors they inherit from ξ_{i-1}. Every quantity in equation 2 and equation 3 is a DP-SGD output or a deterministic function of such outputs and the public constants $(\sigma, C, B, D_a, \eta)$, so the score adds no privacy cost (Theorem 1). Since only the active block $a = M^t(n)$ is updated in round t, only its score is refreshed; the inactive block keeps the score from its last active round. Implementation details, including the late-phase schedule of the curvature term, are given in Algorithm 3.

Now, we introduce the Gaussian random projection (Johnson et al., 1984) applied to the gradients entering the score. Large language models exhibit channel-wise activation outliers (Xiao et al., 2023; Lin et al., 2024), and Jung et al. (2025) shows that the gradient of the A component concentrates in a few channels, so the squared-gradient term can overestimate the score of A. For each component we therefore fix a dimension-preserving random matrix $R_A(n)$, $R_B(n)$ with i.i.d. $\mathcal{N}(0, 1/d)$ entries and project the privatized gradients of client k at round t and step i as $\widetilde{g}_{A,k}^{t,i}(n) = \bar{g}_{A,k}^{t,i}(n)\,R_A(n), \quad \widetilde{g}_{B,k}^{t,i}(n) = R_B(n)\,\bar{g}_{B,k}^{t,i}(n)$, which redistributes outliers across coordinates; equation 2 and equation 3 are then evaluated on $\widetilde{g}$ in place of $\bar{g}$. Because $\mathbb{E}\|R_a(n)\|_F^2 = d$, the noise floor F_a is unchanged, and the fixed linear map preserves the post-processing property. The benefit of the projection is validated in Appendix J.1. We remark that the score is *single-pass*, obtained as a by-product of local training from gradients that DP-SGD releases anyway; a *two-pass inner-loop* alternative that re-scores the redistributed global model needs extra gradient evaluations on the private data and is compared only as a non-private reference (Appendix J.2).

Mode selection algorithm. We now design a mode selection algorithm, denoted by $\mathcal{M}(\cdot)$ which determines the active LoRA component for each layer at round t using the layer-wise scores aggregated from the local scores uploaded by the clients. The algorithm outputs the *mode vector* indicating which component should be optimized. We denote the mode vector at round t as $\{M^t(1), M^t(2), \ldots, M^t(N)\},\ M^t(n) \in \{0, 1\}$, where $M^t(n) = 0$ indicates that $A^t(n)$ is optimized and $M^t(n) = 1$ indicates that $B^t(n)$ is optimized during round t. Accordingly, we define the layer-wise score as $S_a^t(n) := \frac{1}{K}\sum_{k=1}^{K} S_{a,k}^t(n)$, the uniform average of the local scores. Instead of deterministically selecting the component with the larger score, we use a temperature-scaled softmax strategy. This approach encourages exploration during training and is consistent with the common practice of temperature annealing (Jang et al., 2016; Maddison et al., 2016). The probability of selecting A at round t is defined as

$$P^t(A, n) = \sigma\left(\frac{S_A^{t-1}(n) - S_B^{t-1}(n)}{T^t\big(|S_A^{t-1}(n)| + |S_B^{t-1}(n)|\big)}\right), \qquad P^t(B, n) = 1 - P^t(A, n),$$

where σ is the logistic function, the score difference is divided by the total score magnitude so that the temperature is comparable across layers and rounds, and T^t is the temperature parameter at round t. The temperature follows an annealing schedule $T^t = \begin{cases} T^0, & t \le t_w, \\ \max\big(T_{\min},\, T^0 \cdot \gamma^{(t-t_w)}\big), & t > t_w, \end{cases}$ where T^0 is the initial temperature, $T_{\min}$ is the minimum temperature, $\gamma \in (0,1)$ is the decay factor, and t_w denotes the warm-up period. During the warm-up stage ($t \le t_w$), the two LoRA components are alternately activated across rounds. The stochastic selection mechanism prevents premature convergence to a single component while allowing the framework to gradually favor the component with consistently higher scores as the temperature decreases. We remark that our mode selection operates on scores aggregated by *uniform averaging* across clients, which gives equal weight to every client and can be regarded as the standard choice. In Appendix J.3, we compare this design against two alternative aggregation rules, namely *majority voting* and *weighted averaging*.

4 Theoretical Analysis

We analyze AS-LoRA along two axes: *(i) layer-wise freedom* which is the ability to activate A or B independently per layer and *(ii) score-driven selection* which is $\arg\max_a S_a$ with $S_a = \|g_a\|^2 - \frac{\eta}{2} g_a^\top H_a g_a$. Each axis contributes a distinct and quantifiable advantage over the layer-tied or static baselines including FFA-LoRA (Sun et al., 2024), and RoLoRA (Chen et al., 2025). To focus our goals, we do not consider the effect of random projection, temperature-scaled softmax strategy, and the variance of training and scoring caused by distributed settings in our analysis. We first prove that AS-LoRA requires no extra privacy cost. In fact, AS-LoRA's selection at each layer depends on $\|g_a^{(n)}\|^2$ and $(g_a^{(n)})^\top H_a^{(n)} g_a^{(n)}$, both of which are gradient-dependent statistics that in principle *could* be privacy-relevant. The following theorem resolves the concern based on the post-processing invariance of DP (Dwork & Roth, 2014). The proof of Theorem 1 can be found in Appendix D.

Theorem 1 ((ϵ, δ)-DP at no privacy cost)**.** *With per-round DP-SGD (Abadi et al., 2016) (subsampled Gaussian, sensitivity C, noise scale σ, sampling rate q), AS-LoRA satisfies (ϵ, δ)-DP with $\epsilon \le \frac{q^2 C^2 T}{\sigma^2} \log(1/\delta)$, the same bound with FFA-LoRA/RoLoRA via moments accountant / Rényi-DP composition (Mironov, 2017; Wang et al., 2019).*

Now, we show that any schedule which ties the mode across all layers must incur an irreducible loss floor, whereas AS-LoRA provably eliminates it. For the analysis, we first introduce *layer-wise population-optimal LoRA pair* that minimizes the *population* fine-tuning loss f_n as

$$\big(A^\star(n),\, B^\star(n)\big) := \arg \min_{A \in \mathbb{R}^{r\times d},\, B \in \mathbb{R}^{d\times r}} \mathbb{E}_{X_n}\big[\ell\big(f(X_n)\big)\big]. \tag{4}$$

Next, we define the *subspace misalignment* $\delta^t(n)$ as

$$\delta^t(n) := \big\|B^\star(n)\, A^\star(n)\, (I - P_{A^t(n)})\big\|_F = \big\|\Delta W^\star(n)(I - P_{A^t(n)})\big\|_F \in [0, \|\Delta W^\star(n)\|_F], \tag{5}$$

where $P_{A^t(n)} := A^t(n)^\dagger A^t(n) \in \mathbb{R}^{d\times d}$ denote the orthogonal projector onto $\mathrm{row}(A^t(n))$, $A^\dagger$ denotes the Moore-Penrose pseudoinverse (Barata & Hussein, 2012). It quantifies the part of the optimal update $B^\star A^\star$ that lies outside $\mathrm{row}(A^t(n))$. The intermediate values with $r = 1$ was used by (Chen et al., 2025) to diagnose the fundamental obstruction of FFA-LoRA, where $A^t(n) \equiv A^0(n)$ so $\delta^t(n)$ persists forever. With $X_n \in \mathbb{R}^{d\times m}$ stacking the layer-n inputs, we define *per-layer reconstruction risk* as

$$\mathcal{R}_{\text{rec}}(\mathbf{W}) := \frac{1}{N} \sum_{n=1}^{N} \big\|B^\star(n) A^\star(n) X_n - B(n) A(n) X_n\big\|_F^2, \tag{6}$$

i.e. the average layer-wise squared Frobenius distance between each layer's effective update and its optimal LoRA pair defined in equation 4. While the reconstruction risk with one layer was also used in (Chen et al., 2025), $\mathcal{R}_{\text{rec}}$ in equation 6 is averaged layer-wise. Following Theorem 2 and Corollary 1 prove the strict dominance of layer-wise freedom over layer-tied schedules.

Theorem 2 (Irreducible reconstruction floor of layer-tied schedules)**.** *Under Assumptions 1–3 and a well-conditioned input regime that is $\sigma_{\max}(X_n)/\sigma^+_{\min}(X_n A^0(n)^\dagger) = O(1)$ uniformly over layers n, FFA-LoRA (Sun et al., 2024) which freezes $A^{(n)}$ at the random initialization $A^0(n)$ satisfies*

$$\lim_{T\to\infty} \mathbb{E}\big[\mathcal{R}_{rec}(\mathbf{W}_T^{FFA})\big] \;=\; \Theta\left(\frac{1}{N}\sum_{n=1}^{N} \big(\delta^0(n)\big)^2\right), \tag{7}$$

where $\delta^0(n) = \|B^\star(n)A^\star(n)(I - P_{A^0(n)})\|_F$ is the initial subspace misalignment, and the implicit $\Theta(1)$ constants depend on the per-layer conditioning $\sigma^2_{\max}(X_n)/\sigma^2_{\min}(X_n A^0(n)^\dagger) = O(1)$ and absorb both the correction $O(1/\sqrt{Km})$ and the SGD noise floor $O(\eta\sigma_g^2)$. Moreover, any RoLoRA-type layer-tied alternation inherits $\Omega\big(\frac{1}{N}\sum_n(\delta^0(n))^2\big)$ floor under adversarial layer configurations.

The preliminaries can be found in Appendix C, the proof of Theorem 2 can be found in Appendix E, and the proof of Corollary 1 can be found in Appendix F.

Corollary 1 (Strict improvement of AS-LoRA)**.** *Under the same well-conditioned input regime as Theorem 2: if AS-LoRA activates $A^{(n)}$ on $\Theta(T)$ rounds per layer, then for any $\varepsilon > 0$ there exists $T = O(\log(1/\varepsilon))$ with $\delta^T(n) \le \varepsilon$ for all n, and*

$$\lim_{T\to\infty} \mathbb{E}\big[\mathcal{R}_{rec}(\mathbf{W}_T^{AS})\big] \;=\; O\big(\varepsilon^2\big) \;<\; \lim_{T\to\infty} \mathbb{E}\big[\mathcal{R}_{rec}(\mathbf{W}_T^{FFA})\big] \;=\; \Theta\big(\tfrac{1}{N}\sum_n (\delta^0(n))^2\big) \tag{8}$$

whenever any $\delta^0(n_0) > \varepsilon$ for at least one layer n_0 (the FFA floor's $\Theta(1)$ constant is bounded below).

Having established that per-layer freedom is the right design axis, we now ask which block to pick at each round. Here, we discuss the improvement via the selection based on $S_a^{(n)}$. For this, we first define the *adaptive gain* $\Delta_{\mathcal{S}}^t(n) := \mathbb{E}\big[\max_{a\in\{A,B\}} S_a^t(n)\big] - \frac{1}{2}\,\mathbb{E}[S_A^t(n) + S_B^t(n)]$ the per-layer gap between the score of AS-LoRA's greedy pick and the uniform baseline. This is the score-space version of the *suboptimality gap* used in Gauss–Southwell analyses of BCD (Nesterov, 2012). In addition, we collect $1 - \frac{\eta}{2}\lambda_a^{(n)}$ across the trajectory and across layers, and define $\rho^\star := \inf_{t,n}\Big(1 - \frac{\eta}{2}\lambda^{(n)}_{M^t(n)}\Big) \in (0, 1]$, the *curvature-penalty ratio*, where $\lambda_a^{(n)} := (g_a^{(n)})^\top H_a^n g_a^{(n)}/\|g_a^{(n)}\|^2$ the block Rayleigh quotient. Intuitively, $\rho^\star$ is the fraction of pure-descent score that survives after the curvature penalty under the AS-LoRA selection rule. Note that the curvature-penalty ratio with AS-LoRA is larger than the value with static schedules in the late-training regime where $\|g_A^{(n)}\|^2 \approx \|g_B^{(n)}\|^2$. The following theorem provides the convergence of AS-LoRA and improvement compared to the static schedules characterized by the factors $\overline{\Delta_{\mathcal{S}}}$, and $\rho^\star$.

Theorem 3 (Accelerated Convergence with quantities $\overline{\Delta_{\mathcal{S}}}$, and $\rho^\star$)**.** *Under Assumptions 1–3 with $\eta \le 1/\beta$,*

$$\min_{t\le T} \mathbb{E}\|\nabla\mathcal{L}(W_t)\|^2 \;\le\; \frac{2(\mathcal{L}(W_0) - \mathcal{L}^\star)}{\eta\,\rho^\star\,T} \;-\; \frac{2\,\overline{\Delta_{\mathcal{S}}}}{\rho^\star} \;+\; \frac{\eta\beta\sigma_g^2}{\rho^\star}, \tag{9}$$

where $\overline{\Delta_{\mathcal{S}}} := \frac{1}{T}\sum_{t=1}^T \frac{1}{N}\sum_{n=1}^N \Delta_{\mathcal{S}}^t(n)$. Under the additional PL condition (Assumption 4) (Karimi et al., 2016), $\mathbb{E}[\mathcal{L}(W_T) - \mathcal{L}^\star] \le (1 - \eta\mu\rho^\star)^T(\mathcal{L}(W_0) - \mathcal{L}^\star) + O(\eta\sigma_g^2/\mu)$.

The non-convex bound equation 9 recovers the standard $O(1/(\eta\rho^\star T))$ gradient-norm rate of SGD (Karimi et al., 2016) but with two distinguishing features. The terms are multiplied by $1/\rho^\star$, and the second term $-2\overline{\Delta_{\mathcal{S}}}/\rho^\star$ guarantees that AS-LoRA's pick tightens the bound directly and beats the uniform baseline. Under the PL condition, the improvement compounds geometrically, yielding a $(1 - \eta\mu\rho^\star)^T$ rate that matches the best-known LoRA fine-tuning rate (Liu et al., 2022; Hayou et al., 2024) but with $\mu\rho^\star$ in place of μ, i.e. AS-LoRA provably converges generally faster than any layer-tied or static baselines. The proof can be found in Appendix G. Finally, we discuss that AS-LoRA's trajectory with $S_a^{(n)}$ is biased toward the *flatter* of the two blocks, which can be seen as an implicit SAM-style regularization (Foret et al., 2021) or cheap method to achieve the flatness without the SVD overhead in FedSVD (Li et al., 2025). The proof can be found in Appendix H.

Theorem 4 (Implicit flatness via curvature-penalty score)**.** *For each layer n, let $\overline{H}_t^{(n)} := \mathbb{E}[H^{(n)}_{M^t(n)}]$ denote the expected block Hessian at the selected mode under AS-LoRA's argmax rule. Under Assumptions 1–3,*

$$\mathbb{E}\big[\lambda_{\max}\big(\overline{H}_t^{(n),AS}\big)\big] \;\le\; \mathbb{E}\big[\lambda_{\max}\big(\overline{H}_t^{(n),static}\big)\big] \quad \textit{for every } n \in [N], \tag{10}$$

with strict inequality at every layer at which the two per-block top-spectra $\{\lambda_{\max}(H_A^{(n)}), \lambda_{\max}(H_B^{(n)})\}$ differ. AS-LoRA's trajectory is therefore biased toward the flatter *of the two blocks, yielding an implicit SAM-style regularization (Foret et al., 2021).*

5 EXPERIMENTS

Experimental setup. For language tasks, we use RoBERTa-large (Liu et al., 2019) as the backbone model, while ViT-large (Dosovitskiy et al., 2020) is used for image tasks. The LoRA rank is set to $r = 8$, and the scaling factor is set to $\alpha = 8$. We consider a federated setting with $K = 6$ clients for language tasks and $K = 3$ clients for the vision tasks, with full client participation in every communication round. Data are partitioned using a Dirichlet distribution with $\alpha = 0.5$, except for SQuAD (Rajpurkar et al., 2016; 2018), where data are equally partitioned across clients. Training is conducted for $T = 100$ communication rounds, with $\tau = 10$ local steps per round. Following the empirical setup of (Sun et al., 2024), the learning rate η is selected from $\{0.05, 0.1, 0.25\}$ for FedLoRA and from $\{0.1, 0.25, 0.5, 1.0\}$ for all other methods. The batch size is set to $B = 128$. For DP-SGD, we set the target failure probability to $\delta = 10^{-5}$ and fix the clipping norm at $C = 2$. The differential privacy budget ϵ is computed using Opacus (Yousefpour et al., 2021). All experiments were run on RTX Pro 6000 GPUs. Additional details are provided in Appendix I. Unless otherwise specified, all experiments follow the settings described in Section 5 and Appendix I.

Our score uses the secant estimator of equation 3: curvature is read off the consecutive privatized gradients that DP-SGD already produces, so the score remains a post-processing of DP outputs (Theorem 1) and requires no additional backward pass. To further reduce client-side computation, we use the first-order term $S_{g,a}$ alone for the first T^* rounds and add the curvature term only in the remaining $T - T^*$ rounds, since $S_{g,a}$ is a by-product of the backward pass and incurs no additional cost. More accurate curvature could be obtained with Hessian-vector products (HVP) (Pearlmutter, 1994) or finite differences (FD) (Nocedal & Wright, 2006), but both evaluate curvature on the raw mini-batch outside the DP-SGD mechanism, so the resulting score is no longer a post-processing of privatized quantities and forfeits the guarantee of Theorem 1; they also cost an extra backward pass per estimate. We therefore did not adopt them and report them only as non-private references. Table 2 quantifies what this choice costs under the same schedule ($T^* = 90$): the secant estimator stays within 0.7 pp of HVP, the most accurate estimator, on both MNLI (m) and MNLI (mm), and matches FD, while being the only one that preserves the privacy guarantee. All subsequent experiments use the secant estimator with $T^* = 90$.

Table 2: Comparison of curvature estimators on MNLI under DP-SGD ($\epsilon = 3$) with the same late-phase schedule ($T^* = 90$).

Method	DP guarantee	MNLI (m)	MNLI (mm)
FD	✗	79.00 ± 0.30	79.78 ± 0.25
HVP	✗	$\mathbf{79.36 \pm 0.27}$	$\mathbf{80.62 \pm 0.13}$
Secant (Proposed)	✓	78.83 ± 0.27	79.97 ± 0.55

Baselines. We compare our method with the following baselines, which directly align with the design objective of AS-LoRA. Methods that address only the aggregation error (Wang et al., 2024; Singhal et al., 2024; Bai et al., 2024) are excluded from comparison, as they have been shown to suffer significant performance degradation under DP noise (Li et al., 2025).

FedLoRA (Zhang et al., 2024): Both A and B are locally trained and independently aggregated at the server.

FFA-LoRA (Sun et al., 2024): A is frozen, and only B is locally trained and aggregated.

RoLoRA (Chen et al., 2025): In odd-numbered rounds, A is frozen and only B is locally trained and aggregated. In even-numbered rounds, B is frozen and only A is locally trained and aggregated.

Comparison results. All results are reported as mean $\pm$ standard deviation over 4 runs with different seeds. Best results are shown in **bold** and second-best results are underlined. Table 3 reports the performance on the GLUE benchmark (Wang et al., 2018) under different privacy budgets. The proposed method consistently outperforms all baselines on every task except QNLI at $\epsilon = 1$, where it is second best within one standard deviation. Under a privacy budget of $\epsilon = 3$, AS-LoRA achieves the best average score of $\mathbf{82.30}$, outperforming RoLoRA (75.27) by $\mathbf{+7.03}$ percentage points (pp). Under a budget of $\epsilon = 1$, where training instability is more severe, AS-LoRA attains $\mathbf{77.59}$, surpassing FFA-LoRA ($\mathbf{71.89}$) by $\mathbf{+5.70}$ pp. In particular, significant improvements are observed on MNLI (mm) (+13.07 pp over FFA-LoRA) and consistent gains across SNLI and QQP. As shown in Table 4, these improvements extend beyond classification tasks. On SQuAD, AS-LoRA achieves higher performance across both SQuAD v1.1 and SQuAD v2.0. Moreover, the method consistently improves vision benchmarks, achieving +1.16 pp on CIFAR-100 and +1.33 pp on Tiny-ImageNet. Extended results on LoRA rank and heterogeneity of data distribution can be found in Appendix J.4.

Table 3: Performance comparison on the GLUE benchmark (Wang et al., 2018) under different privacy budgets.

DP Budget	Method	QNLI	MNLI (m)	MNLI (mm)	SST-2	QQP	SNLI	AVG
$\epsilon = 3$	FedLoRA	72.43 ± 10.56	63.76 ± 8.24	64.98 ± 8.65	91.37 ± 1.63	67.86 ± 10.47	78.19 ± 2.04	73.10 ± 6.93
	FFA-LoRA	68.18 ± 11.72	65.06 ± 5.40	67.51 ± 5.34	91.00 ± 0.98	75.79 ± 1.28	77.91 ± 7.49	74.24 ± 5.37
	RoLoRA	65.02 ± 13.32	69.15 ± 4.50	71.02 ± 3.93	91.37 ± 0.63	72.86 ± 1.21	82.22 ± 1.49	75.27 ± 4.18
	Proposed	**80.48** ± 0.17	**78.83** ± 0.27	**79.97** ± 0.55	**91.88** ± 0.25	**78.00** ± 0.67	**84.65** ± 0.65	**82.30** ± 0.38
$\epsilon = 1$	FedLoRA	55.72 ± 1.40	58.03 ± 8.05	59.37 ± 8.94	57.57 ± 0.37	65.72 ± 9.90	74.79 ± 1.63	61.87 ± 5.05
	FFA-LoRA	**71.95** ± 7.92	62.70 ± 7.67	65.04 ± 7.72	79.27 ± 11.71	71.00 ± 2.41	81.37 ± 1.57	71.89 ± 6.50
	RoLoRA	53.85 ± 5.34	59.91 ± 10.66	61.37 ± 10.73	62.99 ± 13.43	73.10 ± 1.59	80.97 ± 0.79	65.37 ± 7.09
	Proposed	70.36 ± 9.36	**77.12** ± 1.11	**78.11** ± 1.12	**81.33** ± 2.46	**75.28** ± 2.27	**83.36** ± 1.32	**77.59** ± 2.94

Table 4: Performance comparison on SQuAD benchmarks (Rajpurkar et al., 2016; 2018),CIFAR-100 (Krizhevsky et al., 2009) and Tiny-ImageNet (Deng et al., 2009) under different privacy budgets.

DP Budget	Method	Question Answering				Image Classification	
		SQuAD v1.1 (EM)	SQuAD v1.1 (F1)	SQuAD v2.0 (EM)	SQuAD v2.0 (F1)	CIFAR-100	Tiny-ImageNet
$\epsilon = 3$	FFA-LoRA	72.30 ± 3.52	82.83 ± 2.83	55.90 ± 3.66	59.92 ± 3.93	87.37 ± 0.26	85.50 ± 0.15
	RoLoRA	69.75 ± 2.70	80.63 ± 2.30	53.58 ± 1.22	56.88 ± 1.16	87.75 ± 0.40	84.62 ± 0.13
	Proposed	**72.77** ± 1.42	**83.00** ± 1.18	**60.48** ± 1.08	**64.47** ± 0.61	**88.91** ± 0.07	**86.83** ± 0.35
$\epsilon = 1$	FFA-LoRA	70.42 ± 3.91	81.24 ± 3.26	54.33 ± 3.91	58.58 ± 4.16	84.79 ± 0.23	83.52 ± 0.26
	RoLoRA	61.69 ± 3.81	73.94 ± 3.10	51.48 ± 1.47	53.59 ± 1.80	84.64 ± 0.38	82.34 ± 0.14
	Proposed	**71.48** ± 0.91	**82.31** ± 1.21	**57.69** ± 1.94	**61.52** ± 2.25	**85.31** ± 0.23	**83.56** ± 0.21

Granularity of mode selection scope. While AS-LoRA performs layer-wise mode selection that is shared across all clients, we evaluate two alternatives to justify this design choice. Both are round-wise adaptive as AS-LoRA, but differ in selection granularity. *Global* shares a single mode across all layers and clients, whereas *Per-client* selects modes independently for each layer *and* each client. As shown in Table 5, *Per-client*, yields the worst performance across all tasks and privacy budgets. It is because the aggregated model suffers from aggregation error. *Global* also underperforms AS-LoRA. Sharing a single mode across all layers prevents the model from exploiting layer-specific dynamics. AS-LoRA strikes the right balance, reserving model expressiveness without compromising aggregation stability.

Table 5: Comparison across mode selection granularity. (*Per-client*: layer-wise and client-wise, *AS-LoRA*: layer-wise and client-tied, *Global*: layer-tied and client-tied)

DP Budget	Method	MNLI (m)	MNLI (mm)	QQP
$\epsilon = 3$	Per-client	72.73 ± 2.07	73.73 ± 2.23	63.18 ± 0.00
	Proposed	**78.83** ± 0.27	**79.97** ± 0.55	**78.00** ± 0.67
	Global	75.74 ± 2.53	76.98 ± 2.44	73.61 ± 2.75
$\epsilon = 1$	Per-client	34.60 ± 1.72	34.53 ± 1.37	63.18 ± 0.00
	Proposed	**77.12** ± 1.11	**78.11** ± 1.12	**75.28** ± 2.27
	Global	72.54 ± 3.85	73.79 ± 3.87	72.62 ± 3.26

Comparison with SVD-based method. Table 6 compares AS-LoRA with *FedSVD* (Li et al., 2025), a representative SVD-based method. FedSVD improves performance by introducing an additional SVD computation at the server. FedSVD(n) applies SVD at every n-th communication round and standard FedAvg otherwise. Under $\epsilon = 3$, the proposed method achieves the best performance without any server-side SVD. Under $\epsilon = 1$, the proposed method is on par with FedSVD(5) (77.12 vs. 76.97 on MNLI (m) and 78.11 vs. 78.16 on MNLI (mm)). Moreover, the proposed method incurs only negligible communication and client-side overhead; a detailed overhead analysis, including the aggregation cost relative to FedSVD, is provided in Section 5.

Table 6: Comparison with FedSVD (Li et al., 2025) on MNLI.

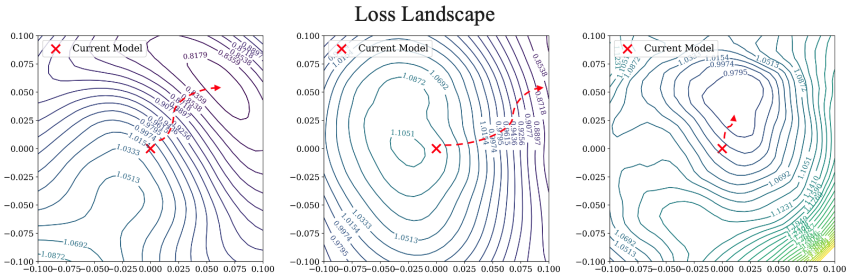
DP Budget	Method	MNLI (m)	MNLI (mm)
$\epsilon = 3$	FedSVD(10)	78.16 ± 1.01	79.43 ± 1.21
	FedSVD(5)	77.06 ± 1.47	79.07 ± 1.10
	FedSVD(1)	78.49 ± 0.89	79.75 ± 0.68
	Proposed	**78.83** ± 0.26	**79.97** ± 0.55
$\epsilon = 1$	FedSVD(10)	74.24 ± 1.48	75.69 ± 1.45
	FedSVD(5)	76.97 ± 1.25	**78.16** ± 1.13
	FedSVD(1)	76.62 ± 0.77	77.79 ± 0.60
	Proposed	**77.12** ± 1.11	78.11 ± 1.12

Sharpness and loss landscape analysis. To empirically validate Theorem 4, which establishes that AS-LoRA's curvature-penalty score implicitly biases the trajectory toward flatter regions of the loss landscape, we measure the SAM-style perturbation sharpness $PS = L(w + \rho\hat{g}) - L(w)$, $\hat{g} = \nabla L(w)/\|\nabla L(w)\|$, together with the induced accuracy drop (ΔAcc) (Foret et al., 2021).

The results in Table 7 indicates that the solution found by AS-LoRA is more robust to local perturbations and lies in a flatter region of the loss surface. This observation is further supported by the loss landscape visualization in Figure 4. Compared with FFA-LoRA and RoLoRA, the contour around the AS-LoRA solution appears to suggest a flatter optimization geometry. Such flatness is known to correlate with improved stability and generalization (Kaddour et al., 2022), which is consistent with the empirical results in Table 3 and Table 4.

Table 7: Comparison of SAM-style perturbation sharpness on MNLI.

Method	PS	ΔAcc
FFA-LoRA	0.03608 $\pm$ 0.04565	0.981 $\pm$ 1.182
RoLoRA	0.01972 $\pm$ 0.02530	0.321 $\pm$ 0.045
Proposed	**0.00992** $\pm$ 0.01281	**0.247** $\pm$ 0.282



(a) FFA-LoRA (b) RoLoRA (c) AS-LoRA

Figure 4: Loss landscape visualization of FFA-LoRA, RoLoRA, and AS-LoRA trained on MNLI with $T = 100$ and $\tau = 10$ under DP-SGD ($\epsilon = 3$).

Overhead analysis. Table 8 reports the measured wall-clock cost of each component of AS-LoRA, together with the server-side SVD of FedSVD, on the same machine (2× NVIDIA H200 NVL, RoBERTa-large, batch size 128). On the client side, the full score adds 6.85 ms per step, about 1.2% of a DP-SGD step. On the server side, AS-LoRA aggregation takes 9.03 ms per round, whereas FedSVD performs 48 full SVDs per round (7636.62 ms), which run serially and grow with the number of layers. Even when client-side scoring is counted, the total per-round overhead of AS-LoRA is at most about 78 ms, two orders of magnitude below a single SVD round of FedSVD. Communication overhead is negligible: AS-LoRA introduces only N scalar scores (uplink) and N mode bits (downlink) per round, accounting for less than 0.02% of the active LoRA parameter traffic in our 24-layer configuration. Detailed analyses are provided in Appendix J.5.

Table 8: Measured per-component wall-clock cost (mean $\pm$ std).

Component	Time (ms)
Client side (per step)	
DP-SGD step	565.14 $\pm$ 55.91
First-order score	3.53 $\pm$ 0.85
Secant curvature score	3.32 $\pm$ 0.14
Server side (per round)	
AS-LoRA aggregation	9.03 $\pm$ 0.10
FedSVD (48× full SVD)	7636.62 $\pm$ 2.61

Effect of gradient smoothing under DP-SGD. Motivated by prior work (Liu et al., 2026; Liang et al., 2024; Wang et al., 2020) demonstrating that applying smoothing after Gaussian noise injection in DP-SGD improves utility, we investigate several smoothing strategies such as Gaussian, Laplacian, and EMA applied to our method. Detailed results are provided in Appendix J.6.

6 Conclusion

In this paper, we proposed AS-LoRA, an adaptive optimization framework for LoRA in differentially private FL. Unlike prior work on federated LoRA, whose update schedule is fixed or shared across layers, AS-LoRA introduces a layer-wise adaptive selection mechanism that determines which component to update based on a theoretically grounded score function. We provided theoretical analysis showing that AS-LoRA eliminates the irreducible reconstruction error floor of static methods and achieves accelerated convergence with implicit flatter solution, without incurring additional privacy cost. Empirically, AS-LoRA consistently outperforms prior methods across language and vision tasks under non-IID and differentially private settings.

Limitations and future works. Our current analysis and algorithm focus on layer-wise greedy selection. While this design enables tractable analysis and strong empirical performance, it does not explicitly account for possible dependencies across layers. In practice, the mode selection dynamics in Appendix J.7 suggest that, particularly for language tasks, cross-layer structure may exist in the selection process. Extending AS-LoRA to a joint layer-wise decision framework that models such inter-layer correlations is an important direction for future work.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure reproducibility. Source code for AS-LoRA and all baselines, together with the scripts used to produce every table and figure, is available at `https://anonymous.4open.science/r/as_lora-F75F/`. The complete training procedure, including score computation and mode selection, is specified in Algorithms 1–3. All datasets are public (GLUE, SQuAD v1.1/v2.0, CIFAR-100, Tiny-ImageNet); the non-IID partitioning, the DP-SGD configuration (σ, C, δ, Poisson subsampling, and the Opacus PRV accountant), model and LoRA settings, and all hyperparameters are given in Section 5 and Appendix I.

STATEMENT ON THE USE OF AI

We used an LLM (Claude by Anthropic) as a coding, mathematical, and writing assistant. Among the tasks with required disclosure, it was used to assist in implementing the method from the authors' algorithm specification, and to assist in developing and writing the proofs of the theoretical results in the appendix, including suggesting proof steps and intermediate lemmas; the theorem statements were formulated by the authors, and every proof was checked step by step and verified by the authors. It was not used to develop the conceptual framework, to formulate the mathematical claims, to propose hypotheses, to design experiments, or to generate or process data; the remaining required-disclosure tasks are not applicable. Among the tasks with recommended disclosure, it was used to draft and edit parts of the manuscript for readability, resolve LaTeX issues, format references, and write experiment scripts. We have reviewed all AI-assisted work: LLM-generated code was tested by the authors, all proofs and claims were verified by the authors, and all reported numbers come from our experiment records. We take responsibility for the final content of this work.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. *Advances in Neural Information Processing Systems*, 37:14457–14483, 2024.

João Carlos Alves Barata and Mahir Saleh Hussein. The moore–penrose pseudoinverse: A tutorial review of the theory. *Brazilian Journal of Physics*, 42(1):146–165, 2012.

Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1: 374–388, 2019.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Shuangyi Chen, Yuanxin Guo, Yue Ju, Harik Dalal, Zhongwen Zhu, and Ashish Khisti. Robust federated finetuning of llms via alternating optimization of lora. *arXiv preprint arXiv:2502.01755*, 2025.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. Heterogeneous lora for federated fine-tuning of on-device foundation models. *arXiv preprint arXiv:2401.06432*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Zihan Fang, Zheng Lin, Zhe Chen, Xianhao Chen, Yue Gao, and Yuguang Fang. Automated federated pipeline for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2404.06448*, 2024.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024.

Roger A Horn and Charles R Johnson. Matrix analysis. *Cambridge University Press*, 2012.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Ajay Kumar Jaiswal, Yifan Wang, Lu Yin, Shiwei Liu, Runjin Chen, Jiawei Zhao, Ananth Grama, Yuandong Tian, and Zhangyang Wang. From low rank gradient subspace stabilization to low-rank weights: Observations, theories, and applications. In *International Conference on Machine Learning*, pp. 26740–26756. PMLR, 2025.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

William B Johnson, Joram Lindenstrauss, et al. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

Yeonjoon Jung, Daehyun Ahn, Hyungjun Kim, Taesu Kim, and Eunhyeok Park. Gralora: Granular low-rank adaptation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2505.20355*, 2025.

Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J Kusner. When do flat minima optimizers work? *Advances in Neural Information Processing Systems*, 35:16577–16595, 2022.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition. *ECML-PKDD*, 2016.

Jabin Koo, Minwoo Jang, and Jungseul Ok. Towards robust and efficient federated low-rank adaptation with heterogeneous clients. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 416–429, 2025.

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

Kevin Kuo, Arian Raje, Kousik Rajesh, and Virginia Smith. Federated lora with sparse communication. *arXiv preprint arXiv:2406.05233*, 2024.

Seanie Li et al. FedSVD: Adaptive orthogonalization for private federated learning with LoRA. 2025.

Zhicong Liang, Bao Wang, Quanquan Gu, Stanley Osher, and Yuan Yao. Differentially private federated learning with laplacian smoothing. *Applied and Computational Harmonic Analysis*, 72: 101660, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6: 87–100, 2024.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59: 85–116, 2022.

Jin Liu, Yinbin Miao, Ning Xi, and Junkang Liu. Rethinking lora for privacy-preserving federated learning in large models. *arXiv preprint arXiv:2602.19926*, 2026.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

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

Ilya Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium (CSF)*, pp. 263–275, 2017.

Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. *arXiv preprint arXiv:2009.03561*, 2020.

Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Yurii Nesterov. *Lectures on Convex Optimization*. Springer, 2nd edition, 2018.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

Roger Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, 2018.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Raghav Singhal, Kaustubh Ponkshe, and Praneeth Vepakomma. Fedex-lora: Exact aggregation for federated and efficient fine-tuning of foundation models. *arXiv preprint arXiv:2410.09432*, 2024.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving lora in privacy-preserving federated learning. *arXiv preprint arXiv:2403.12313*, 2024.

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

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Bao Wang, Quanquan Gu, March Boedihardjo, Lingxiao Wang, Farzin Barekat, and Stanley J Osher. Dp-lssgd: A stochastic optimization method to lift the utility in privacy-preserving erm. In *Mathematical and Scientific Machine Learning*, pp. 328–351. PMLR, 2020.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. In *AISTATS*, 2019.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *Advances in Neural Information Processing Systems*, 37:22513–22533, 2024.

Per-Åke Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pp. 38087–38099. PMLR, 2023.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. PyHessian: Neural networks through the lens of the Hessian. In *IEEE International Conference on Big Data*, 2020.

Liping Yi, Han Yu, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pfedlora: Model-heterogeneous personalized federated learning with lora tuning. *arXiv preprint arXiv:2310.13283*, 2023.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6915–6919. IEEE, 2024.

Qinqing Zheng, Shuxiao Chen, Qi Long, and Weijie Su. Federated f-differential privacy. In *International conference on artificial intelligence and statistics*, pp. 2251–2259. PMLR, 2021.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

A Related Works

A.1 Federated learning with LoRA

Recent studies have explored integrating LoRA into FL to alleviate the communication and computation overhead associated with fine-tuning large-scale models (Kuo et al., 2024; Cho et al., 2024; Fang et al., 2024; Yi et al., 2023). FedIT (Zhang et al., 2024) is one of the earliest attempts to incorporate LoRA into FL, demonstrating that low-rank adapters can effectively reduce communication costs while maintaining competitive performance. However, FedIT aggregates the two low-rank components in a straightforward manner and does not explicitly account for aggregation errors that arise during the aggregation process. To mitigate aggregation mismatch caused by heterogeneous client updates, several extensions have been proposed. FLoRA (Wang et al., 2024) addresses heterogeneity and aggregation errors by stacking client-specific LoRA adapters at the server for aggregation. FedEx-LoRA (Singhal et al., 2024) alleviates aggregation error by introducing a residual correction term added to a fixed pre-trained weight matrix. FlexLoRA (Bai et al., 2024) further addresses heterogeneity by reconstructing an ideal weight update matrix from the aggregated client adapters at the server, followed by singular value decomposition (SVD) to redistribute LoRA adapters with different ranks to individual clients.

A.2 Differentially private federated LoRA

More recent studies have investigated federated LoRA under differential privacy (DP) constraints. FFA-LoRA (Sun et al., 2024) provides a detailed analysis of the noise amplification problem that arises when DP is applied to federated LoRA training, and mitigates this issue by updating only the B component. However, restricting updates to the B component leads to a loss of expressiveness in LoRA. To address this limitation, FedSVD (Li et al., 2025) similarly updates only the B component, while leveraging orthogonalization and low-rank decomposition to alleviate the expressiveness loss and improve training stability under DP constraints. Another line of work, which is most closely related to our study, aims to alleviate aggregation error by adopting alternating optimization strategies. RoLoRA (Chen et al., 2025) and LoRA-A^2 (Koo et al., 2025) alternate the optimization of the two LoRA components across communication rounds, updating only one component at a time to reduce aggregation mismatch. La-LoRA (Liu et al., 2026) further extend alternating optimization from communication rounds to local update steps. While these methods empirically improve training stability, they rely on fixed and symmetric alternating schedules and do not explicitly account for the asymmetric roles of the two LoRA components during training. In summary, existing federated LoRA-based approaches either restrict updates to a single component or adopt fixed, symmetric alternating schedules, both of which overlook the structural asymmetry between the two LoRA components and the round-wise dynamics of training. This motivates AS-LoRA, which makes principled, adaptive per-layer decisions based on training dynamics.

B AS-LoRA Algorithm

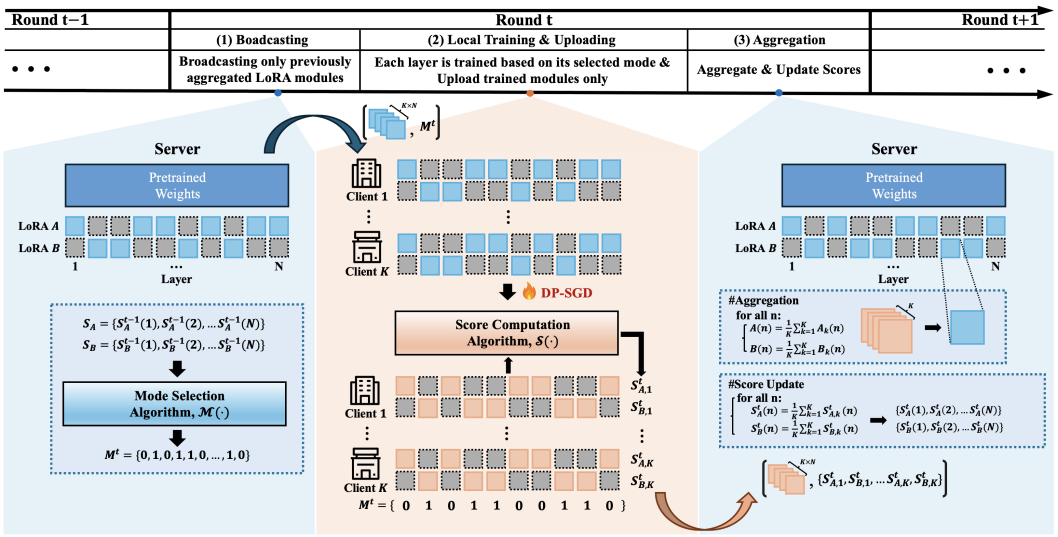
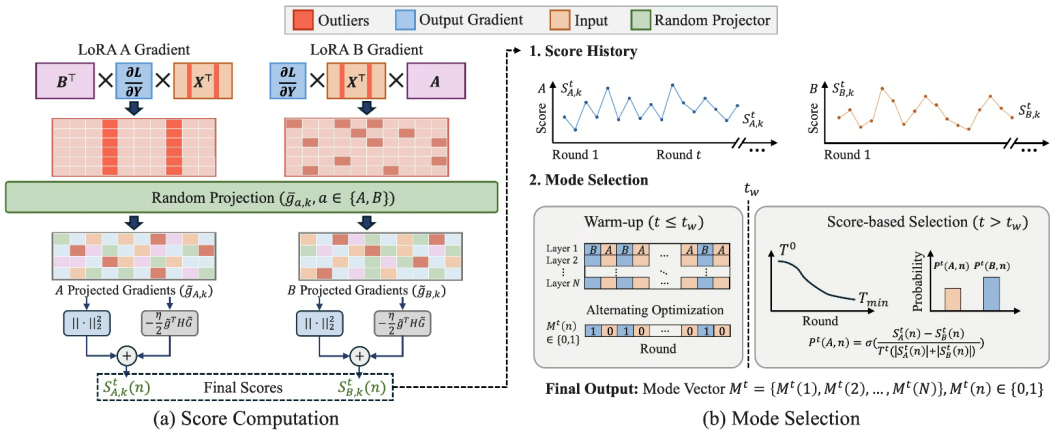


Figure 5: Detail of the proposed AS-LoRA framework.



(a) Score Computation (b) Mode Selection

Figure 6: Detailed visualization of the proposed AS-LoRA score computation and mode selection procedures. (a) Component-wise scores are computed from the projected privatized gradients of LoRA A and B matrices. Random projection mitigates the effect of channel-wise gradient outliers before applying the debiased squared-gradient and secant curvature terms. (b) The server averages the uploaded scores across clients and performs layer-wise mode selection. During warm-up, AS-LoRA alternates between components, while after warm-up it uses a temperature-scaled score-based selection rule to produce the final mode vector.

Algorithm 1 AS-LoRA

Input: communication rounds T, clients K, LoRA layers N, local steps τ, warm-up period t_w
Initialize $A^0(n), B^0(n)$ and server scores $S_A^0(n), S_B^0(n)$ for all n
for $t = 1$ **to** T **do**
  $M^t \leftarrow \mathcal{M}(S_A^{t-1}, S_B^{t-1})$ (Algorithm 2)
  Broadcast M^t and the components aggregated in round $t-1$
  for $k = 1$ **to** K **in parallel do**
    Run τ local DP-SGD steps on the active component $M^t(n)$ of every layer n; step i releases the privatized gradient $\bar{g}^{t,i}_{M^t(n),k}(n)$
    Compute the local score of the active component from $\{\bar{g}^{t,i}_{M^t(n),k}(n)\}_{i=1}^{\tau}$: $S^t_{M^t(n),k}(n) \leftarrow \mathcal{S}(\cdot)$ (Algorithm 3)
    Upload the updated active components and the scores $\{S^t_{M^t(n),k}(n)\}_{n=1}^{N}$ to the server
  end for
  for $n = 1$ **to** N **do**
    if $M^t(n) = 0$ **then**
      $A^t(n) \leftarrow \frac{1}{K}\sum_{k=1}^{K} A_k^t(n), \quad B^t(n) \leftarrow B^{t-1}(n)$
      $S_A^t(n) \leftarrow \frac{1}{K}\sum_{k=1}^{K} S_{A,k}^t(n), \quad S_B^t(n) \leftarrow S_B^{t-1}(n)$
    else
      $B^t(n) \leftarrow \frac{1}{K}\sum_{k=1}^{K} B_k^t(n), \quad A^t(n) \leftarrow A^{t-1}(n)$
      $S_B^t(n) \leftarrow \frac{1}{K}\sum_{k=1}^{K} S_{B,k}^t(n), \quad S_A^t(n) \leftarrow S_A^{t-1}(n)$
    end if
  end for
end for
Output: final LoRA parameters $\{A^T(n), B^T(n)\}_{n=1}^{N}$

Algorithm 2 Mode Selection $\mathcal{M}(\cdot)$

Input: aggregated scores of the previous round $\{S_A^{t-1}(n), S_B^{t-1}(n)\}_{n=1}^{N}$, round t, warm-up period t_w, temperature parameters $(T^0, T_{\min}, \gamma)$
Output: mode vector $M^t = \{M^t(n)\}_{n=1}^{N}$, where $M^t(n) = 0$ activates A and $M^t(n) = 1$ activates B
if $t \le t_w$ **then**
  // warm-up: alternating schedule
  for $n = 1$ **to** N **do**
    $M^t(n) \leftarrow (t \bmod 2)$
  end for
else
  Compute temperature:

$$T^t \leftarrow \max\Big(T_{\min},\, T^0 \cdot \gamma^{(t-t_w)}\Big)$$

  for $n = 1$ **to** N **do**
    Compute selection probability:

$$P^t(A,n) \leftarrow \sigma\left(\frac{S_A^{t-1}(n) - S_B^{t-1}(n)}{T^t\left(|S_A^{t-1}(n)| + |S_B^{t-1}(n)|\right)}\right), \qquad \sigma(x) = \frac{1}{1+e^{-x}}$$

    Sample mode:

$$M^t(n) \sim \text{Bernoulli}\big(1 - P^t(A,n)\big)$$

  end for
end if
return M^t

Algorithm 3 Score Computation $\mathcal{S}(\cdot)$ from Privatized Gradients (client k, round t)

Input: mode vector M^t; for each layer n with $a = M^t(n)$, the privatized gradients $\{\bar{g}_{a,k}^{t,i}(n)\}_{i=1}^{\tau}$ released by the τ local DP-SGD steps; projection matrices $\{R_A(n), R_B(n)\}_{n=1}^{N}$; public DP parameters (σ, C, B); learning rate η; curvature start round T^*; clamp $\lambda_{\max}$

Output: local scores of the active components $\{S_{M^t(n),k}^t(n)\}_{n=1}^{N}$

// Only DP-SGD outputs and public constants are used; no gradient or loss is evaluated here.

for $n = 1$ **to** N **do**

  $a \leftarrow M^t(n), \quad F_a \leftarrow D_a\,(\sigma C/B)^2$ // noise floor from public constants

  Apply random projection to every step $i = 1, \ldots, \tau$:

$$\widetilde{g}^{\,i} \leftarrow \begin{cases} \bar{g}_{A,k}^{t,i}(n)\, R_A(n), & a = A, \\ R_B(n)\, \bar{g}_{B,k}^{t,i}(n), & a = B \end{cases}$$

  First-order score:

$$S_g \leftarrow \frac{1}{\tau} \sum_{i=1}^{\tau} \max\big(\|\widetilde{g}^{\,i}\|^2 - F_a,\ 0\big)$$

  if $t \le T^*$ **then**

    $S_{a,k}^t(n) \leftarrow S_g$ // first-order score only

  else

    $\mathrm{num} \leftarrow 0, \quad \mathrm{den} \leftarrow 0$

    for $i = 2$ **to** τ **do**

      $s_i \leftarrow -\eta\, \widetilde{g}^{\,i-1}, \quad y_i \leftarrow \widetilde{g}^{\,i} - \widetilde{g}^{\,i-1}$ // secant pair of consecutive steps

      $\mathrm{num} \leftarrow \mathrm{num} + \langle s_i, y_i \rangle - \eta F_a, \quad \mathrm{den} \leftarrow \mathrm{den} + \|s_i\|^2 - \eta^2 F_a$

    end for

    if $\mathrm{den} > 0$ **then**

      $\hat{\lambda}_a \leftarrow \mathrm{clip}(\mathrm{num}/\mathrm{den},\ -\lambda_{\max},\ \lambda_{\max})$

      $S_{a,k}^t(n) \leftarrow S_g \left(1 - \frac{\eta}{2}\, \hat{\lambda}_a\right)$ // curvature-aware score

    else

      $S_{a,k}^t(n) \leftarrow S_g$

    end if

  end if

end for

return $\{S_{M^t(n),k}^t(n)\}_{n=1}^{N}$

B.1 VISUALIZATION OF AS-LORA FRAMEWORK

Fig 5 illustrates the overall workflow of the proposed AS-LoRA framework. At each communication round, the server first determines the layer-wise mode from the aggregated scores of the previous round (Algorithm 2) and broadcasts the mode vector M^t together with the current global LoRA parameters. Each client then updates only the selected component of each layer using DP-SGD, computes the score of that component from its privatized gradients (Algorithm 3), and uploads the updated component and its score to the server. The server aggregates the received components and scores, while the score of the inactive component is carried over from the previous round; the resulting scores are used for mode selection in the next round. This process enables adaptive, layer-wise optimization across communication rounds while mitigating aggregation error and unnecessary noise amplification. The overall training procedure of AS-LoRA is summarized in Algorithm 1.

B.2 SCORE COMPUTATION AND MODE SELECTION

Figure 6 provides a detailed illustration of the score computation and mode selection procedures in AS-LoRA.

Score Computation. Figure 6(a) depicts the score computation procedure performed on each client. As illustrated, the gradient of LoRA-A exhibits channel-wise outliers, where the gradient magnitude is highly concentrated in a small subset of channels. To mitigate this, a Gaussian random projection is applied to each gradient, redistributing the concentrated magnitude across dimensions and enabling more reliable score estimation. The score computation procedures with random projection are described in Algorithm 3.

Mode Selection. Figure 6(b) illustrates the mode selection procedure. During the warm-up phase ($t \leq t_w$), all layers follow a simple alternating schedule, where the A and B components are activated in alternation across rounds, and the mode vector $M^t(n) \in \{0, 1\}$ changes accordingly. After the warm-up phase ($t > t_w$), score-based selection is activated. The selection probability for each component is computed via a temperature-scaled softmax.

Here, the temperature T^t follows an annealing schedule that decays from an initial value T^0 to a minimum $T_{\min}$ over training rounds. This annealing encourages exploration of both components in early rounds while gradually shifting toward exploitation of the higher-scoring component as training progresses. The final output is the mode vector $M^t = \{M^t(1), M^t(2), \ldots, M^t(N)\}$, $M^t(n) \in \{0, 1\}$, which specifies the active LoRA component for each layer at round t. The full mode selection procedure is summarized in Algorithm 2.

C PRELIMINARIES FOR THEORETICAL ANALYSIS

C.1 BACKGROUND

We analyze AS-LoRA along two axes: *(i) layer-wise freedom*—the ability to activate $A^{(n)}$ or $B^{(n)}$ independently per layer—and *(ii) score-driven selection*—the greedy rule $\arg\max_{a\in\{A,B\}} S_a^{(n)}$ with $S_a^{(n)} = \|g_a^{(n)}\|^2 - \frac{\eta}{2}(g_a^{(n)})^\top H_a^{(n)} g_a^{(n)}$. Each axis contributes a distinct and quantifiable advantage over the layer-tied or fixed-schedule baselines (FFA-LoRA (Sun et al., 2024), RoLoRA (Chen et al., 2025)). The analysis combines block-coordinate descent theory (Nesterov, 2012), Polyak–Łojasiewicz convergence (Karimi et al., 2016), and DP-SGD privacy accounting (Abadi et al., 2016; Mironov, 2017; Wang et al., 2019).

C.2 SETUP

Let $\mathcal{L}(W) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{L}_k(W)$ be the federated loss over K clients with per-client dataset size m (total Km samples). For layer $n \in [N]$, the frozen weight is $W_0^{(n)} \in \mathbb{R}^{d\times d}$ and the LoRA update factorizes as $B^{(n)}A^{(n)}$ with $B^{(n)} \in \mathbb{R}^{d\times r}$, $A^{(n)} \in \mathbb{R}^{r\times d}$, and rank $r \ll d$, so that the effective weight is $W^{(n)} = W_0^{(n)} + B^{(n)}A^{(n)}$ acting on an input $X \in \mathbb{R}^{d\times m}$ as $W^{(n)}X^{(n)} = W_0^{(n)}X + B^{(n)}A^{(n)}X^{(n)}$

Stochastic gradient and population Hessian. For each block label $a \in \{A, B\}$, let $g_a^{(n)}$ denote the *mini-batch stochastic gradient* of $\mathcal{L}$ with respect to the factor matrix $a^{(n)}$, satisfying $\mathbb{E}[g_a^{(n)}] = \nabla_{a^{(n)}}\mathcal{L}$ over the random sampling. Concretely, $g_A^{(n)} \in \mathbb{R}^{r\times d}$ and $g_B^{(n)} \in \mathbb{R}^{d\times r}$ are matrix-valued. The corresponding *deterministic population Hessian* $H_a^{(n)}(W) := \nabla^2_{a^{(n)}}\mathcal{L}(W)$ acts on the vectorized factor and is treated as a $(\dim a^{(n)})\times(\dim a^{(n)})$ matrix throughout (with $\dim A^{(n)} = \dim B^{(n)} = rd$). Randomness enters only through $g_a^{(n)}$ and DP-SGD noise $\mathcal{N}_a \sim \mathcal{N}(0, \sigma^2C^2I)$ (clipping bound C), and $\nabla_{a^{(n)}}\mathcal{L}$ and $H_a^{(n)}(W)$ are deterministic functions of W as in the standard convention for second-order SGD/BCD analyses (Karimi et al., 2016; Nesterov, 2012). With $\|\cdot\|$ denoting the Frobenius norm for matrices and the Euclidean norm for vector objects, the per-layer block Rayleigh quotient is $\lambda_a^{(n)} := \langle g_a^{(n)}, H_a^{(n)} g_a^{(n)}\rangle / \|g_a^{(n)}\|^2$ and the per-layer block-greedy score is $S_a^{(n)} := \|g_a^{(n)}\|^2 - \frac{\eta}{2}\langle g_a^{(n)}, H_a^{(n)} g_a^{(n)}\rangle$.

Remark 1 (Practical estimation of $H_a^{(n)} g_a^{(n)}$)**.** *All theoretical statements use the population Hessian $H_a^{(n)}(W) = \nabla^2_{a^{(n)}}\mathcal{L}(W)$. In practice, the curvature term $\langle g_a^{(n)}, H_a^{(n)} g_a^{(n)}\rangle$ is computed via a single Hessian-vector product on the same mini-batch used for $g_a^{(n)}$ (Pearlmutter trick) or via a Hutchinson-style sketch (Yao et al., 2020); the resulting estimation bias is $O(\sigma_g^2/m)$ and is absorbed into the $\eta\beta\sigma_g^2$ noise term of Theorem 3 without altering any rate.*

Moore–Penrose pseudoinverse and orthogonal projectors. Several reconstruction-floor and subspace-misalignment quantities are stated in terms of the Moore–Penrose pseudoinverse, which we briefly recall. For any matrix $M \in \mathbb{R}^{p\times q}$, the *Moore–Penrose pseudoinverse* $M^\dagger \in \mathbb{R}^{q\times p}$ is the unique matrix satisfying the four Penrose conditions $MM^\dagger M = M$, $M^\dagger MM^\dagger = M^\dagger$, $(MM^\dagger)^\top = MM^\dagger$, $(M^\dagger M)^\top = M^\dagger M$ (Penrose, 1955). Three properties are used repeatedly. (i) When M is square and invertible, $M^\dagger = M^{-1}$. (ii) When M is row full-rank (the generic case for our LoRA factors $A^t(n) \in \mathbb{R}^{r\times d}$ with $r < d$), $M^\dagger = M^\top(MM^\top)^{-1}$, and $A^t(n)\,A^t(n)^\dagger = I_r$ acts as a right inverse. (iii) The two products $MM^\dagger \in \mathbb{R}^{p\times p}$ and $M^\dagger M \in \mathbb{R}^{q\times q}$ are precisely the orthogonal projectors onto $\mathrm{col}(M) \subseteq \mathbb{R}^p$ and $\mathrm{row}(M) \subseteq \mathbb{R}^q$ respectively. Operationally, the least-squares minimum-norm solution of $Mx = b$ is $x^\star = M^\dagger b$, which is the property used to obtain the closed-form FFA-LoRA minimizer in the proof of Theorem 2. All projectors $P_{A^t(n)} = A^t(n)^\dagger A^t(n)$ used in the analysis are of form (iii) and project onto the row-span of the corresponding factor matrix.

C.3 Assumptions

First, we collect here all assumptions invoked by the main text. Assumptions 1–4 are the standard block / per-layer counterparts of the smoothness, bounded-variance, Hessian-spectrum, and Polyak–Łojasiewicz (PL) conditions used in modern SGD/BCD/PL analyses (Nesterov, 2012; Karimi et al., 2016). Strong convexity of the Assumption 3 is invoked only on the per-layer reconstruction quadratic, not on the full non-convex loss, and the PL condition of Assumption 4 replaces it for end-to-end statements.

Assumption 1 (Block smoothness)**.** *Each $\mathcal{L}_k$ is L-smooth and block-smooth with constants $L_A^{(n)}, L_B^{(n)}$, and $\beta := \max_{a,n} L_a^{(n)}$.*

Assumption 2 (Bounded stochastic variance)**.** $\mathbb{E}\|g_a^{(n)} - \nabla_{a(n)}\mathcal{L}\|^2 \le \sigma_g^2$ *for every layer n and block label $a \in \{A, B\}$, and per-sample clipping $\|g_{a,i}^{(n)}\| \le C$ a.s.*

Assumption 3 (Block strong convexity and Lipschitz Hessian)**.** $\mu_a^{(n)} I \preceq H_a^{(n)} \preceq L_a^{(n)} I$*, and H is L_H-Lipschitz.*

Assumption 4 (PL condition, non-convex)**.** $\|\nabla\mathcal{L}\|^2 \ge 2\mu(\mathcal{L} - \mathcal{L}^\star)$ *for some $\mu > 0$ (Karimi et al., 2016).*

C.4 Mathematical Preliminaries

This appendix collects the standard tools from matrix analysis (Horn & Johnson, 2012), high-dimensional probability (Vershynin, 2018), and convex optimization (Nesterov, 2018).

Lemma 1 (Norm submultiplicativity and SVD-based reverse bound)**.** *For any $M \in \mathbb{R}^{p\times d}$ and $X \in \mathbb{R}^{d\times m}$ of compatible dimensions (so that $MX \in \mathbb{R}^{p\times m}$ is well defined),*

$$\|MX\|_F \;\le\; \sigma_{\max}(X)\,\|M\|_F, \tag{11}$$

where $\sigma_{\max}(X) = \|X\|_{op}$ is the largest singular value of X. We refer to this as the standard inequality. If the rows of M lie in $\mathrm{col}(X^\top) = \mathrm{row}(X)$, equivalently $M = M\,P_X$ where $P_X = X^\top(XX^\top)^{-1}X$ when X has full row rank, then the matching reverse inequality holds:

$$\|MX\|_F \;\ge\; \sigma^{+}_{\min}(X)\,\|M\|_F, \tag{12}$$

with $\sigma^{+}_{\min}(X)$ the smallest nonzero *singular value of X.*

Proof. The upper bound is directly from (Horn & Johnson, 2012). For the proof of lower bound, let $X = U\Sigma V^\top$ be the SVD with $\Sigma = \mathrm{diag}(\sigma_1,\dots,\sigma_k,0,\dots,0)$, $\sigma_1 \ge \cdots \ge \sigma_k > 0$, $k = \mathrm{rank}(X)$. Then $\|MX\|_F^2 = \|MU\Sigma V^\top\|_F^2 = \|MU\Sigma\|_F^2$ (Frobenius is invariant under right multiplication by orthogonal $V^\top$). Writing $MU =: N$ with columns $n_1,\dots,n_d$, $\|MU\Sigma\|_F^2 = \sum_{i=1}^d \sigma_i^2\|n_i\|^2$. If rows of M lie in $\mathrm{row}(X)$, then $n_i = 0$ for all $i > k$, so $\sum_i \sigma_i^2\|n_i\|^2 \ge \sigma_k^2 \sum_{i\le k}\|n_i\|^2 = (\sigma^{+}_{\min}(X))^2\|M\|_F^2$. □

Lemma 2 (High-probability bound of covariance estimation (Vershynin, 2018), Rem. 4.7.3)**.** *Let $x_1,\dots,x_n \in \mathbb{R}^d$ be i.i.d. zero-mean random vectors with population covariance $\Sigma := \mathbb{E}[x_i x_i^\top]$, satisfying*

$$\big\|\langle X, v\rangle\big\|_{\psi_2} \;\le\; \kappa\,\big\|\langle X, v\rangle\big\|_{L^2} \qquad \textit{for every } v \in \mathbb{R}^d, \tag{13}$$

with parameter κ. Then for every $u \ge 0$, with probability at least $1 - 2e^{-u}$,

$$\Big\|\tfrac{1}{n}\sum_{i=1}^n x_i x_i^\top - \Sigma\Big\|_{op} \;\le\; C\,\kappa^2\,\|\Sigma\|_{op}\Big(\sqrt{(d+u)/n} + (d+u)/n\Big), \tag{14}$$

for an absolute constant $C > 0$. In particular, for $n \gtrsim d$ and any fixed confidence level $1 - 2e^{-u}$,

$$\Big\|\tfrac{1}{n}\sum_{i=1}^n x_i x_i^\top - \Sigma\Big\|_{op} \;=\; O_p\big(\kappa^2\|\Sigma\|_{op}\sqrt{d/n}\big). \tag{15}$$

Lemma 3 (Linear convergence of GD on strongly-convex smooth functions (Nesterov, 2018), Thm. 2.3.4)**.** *Let $f : \mathbb{R}^{p\times q} \to \mathbb{R}$ be twice differentiable with Hessian satisfying $\mu I \preceq \nabla^2 f \preceq LI$ for $0 < \mu \le L$. Let $X^\star = \arg\min f$. Then gradient descent $X_{t+1} = X_t - \eta\nabla f(X_t)$ with step size $\eta \le 1/L$ satisfies*

$$\|X_{t+1} - X^\star\|_F^2 \;\le\; (1 - \eta\mu)\,\|X_t - X^\star\|_F^2 \;\le\; (1 - \mu/L)\,\|X_t - X^\star\|_F^2. \tag{16}$$

Lemma 4 (Second-order Taylor with cubic remainder (Nesterov, 2018), Lemma 1.2.4)**.** *Let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ be twice differentiable with L_H-Lipschitz Hessian: $\|\nabla^2\mathcal{L}(x) - \nabla^2\mathcal{L}(y)\|_{op} \le L_H\|x - y\|$. Then for any W, h,*

$$\Big|\mathcal{L}(W + h) - \mathcal{L}(W) - \langle\nabla\mathcal{L}(W), h\rangle - \tfrac{1}{2}h^\top\nabla^2\mathcal{L}(W)h\Big| \;\le\; \tfrac{L_H}{6}\|h\|^3. \tag{17}$$

D Proof of Theorem 1

The DP-SGD mechanism of (Abadi et al., 2016) uses subsampled Gaussian noise: at each round, sample a fraction q of clients, compute clipped gradients $\tilde g_i = \Pi_{\|\cdot\|\le C}(g_i)$, aggregate $\bar g = \sum_{i\in\text{batch}} \tilde g_i + \mathcal{N}(0, \sigma^2C^2I)$. By the moments accountant / Rényi-DP analysis of (Mironov, 2017; Wang et al., 2019), each round contributes at most $\frac{q^2C^2}{\sigma^2}$ to the cumulative Rényi-DP $(\alpha, \epsilon_\alpha)$ budget at order α. Rényi-DP composes additively. T rounds give cumulative Rényi-DP cost at most $T\cdot q^2C^2/\sigma^2$ at order α. Converting to (ϵ,δ)-DP via the standard tail bound $\epsilon = \epsilon_\alpha + \log(1/\delta)/(\alpha - 1)$ and optimising over α yields $\epsilon \le q^2C^2T\log(1/\delta)/\sigma^2$. The AS-LoRA selection $\arg\max_{a\in\{A,B\}} S_a^{(n),t}$ uses the per-block scores $S_a^{(n),t} = \|\bar g_a^{(n)}\|^2 - \frac{\eta}{2}\langle g_a^{(n)}, H_a^{(n)}\bar g_a^{(n)}\rangle$, where $\bar g_a^{(n)}$ is the already-clipped, already-noised, before being aggregated. Since $H_a^{(n)}$ is also computed on the same mini-batch,

the score is a deterministic function of $\bar{g}_a^{(n)}$. The post-processing invariance of differential privacy (Dwork & Roth, 2014) says that *any function applied to the output of an (ϵ, δ)-DP mechanism remains (ϵ, δ)-DP*. Since $M^t(n)$ is computed deterministically from $\{\bar{g}_a^{(n)}\}_{a,n}$, which are already DP-protected, no additional privacy cost is incurred. The total AS-LoRA privacy ledger therefore matches FFA-LoRA / RoLoRA exactly. □

E Proof of Theorem 2

Fix layer n. Under FFA-LoRA, $A^{(n)}$ is permanently fixed at $A^0(n)$ and only $B^{(n)}$ is updated by SGD with step size η. Because the per-layer reconstruction loss is convex quadratic in $B^{(n)}$ (when $A^{(n)}$, X_n are fixed), standard SGD theory on convex quadratics guarantees that the iterates $B_t^{(n)}$ converge in expectation to the static minimizer $B_{\text{FFA}}(n)$ of

$$\ell^{(n)}(B) \;=\; \tfrac{1}{2}\big\|B^\star(n)A^\star(n)X_n - B\,A^0(n)X_n\big\|_F^2, \tag{18}$$

up to an $O(\eta\sigma_g^2)$ noise floor (Bottou et al., 2018). Consequently,

$$\lim_{T\to\infty} \mathbb{E}[\mathcal{R}_{\text{rec}}^{(n),\text{FFA}}] \;=\; \min_B \ell^{(n)}(B)/(Km) \;+\; O(\eta\sigma_g^2), \tag{19}$$

and the first term is fully determined by the static least-squares minimization. The $O(\eta\sigma_g^2)$ correction is a standard SGD noise-floor term, not a function of the misalignment $\delta^0(n)$, so it does not affect the qualitative shape of the floor. We absorb it into $\tilde{c}$ for the remainder of the proof. The function $\ell^{(n)}(B)$ is convex quadratic in B with Hessian $A^0(n)X_nX_n^\top A^0(n)^\top \otimes I$, which is positive definite when $A^0(n)X_n$ has full row rank (the generic case). Setting $\nabla_B \ell^{(n)}(B) = 0$ gives the normal equation

$$\big(B\,A^0(n)X_n - B^\star(n)A^\star(n)X_n\big)\,(A^0(n)X_n)^\top \;=\; 0, \tag{20}$$

which solves to

$$B_{\text{FFA}}(n) \;=\; B^\star(n)A^\star(n)\,X_nX_n^\top A^0(n)^\top\big(A^0(n)\,X_nX_n^\top A^0(n)^\top\big)^{-1}. \tag{21}$$

This is the matrix Moore–Penrose form of the least-squares solution: writing $Z = A^0(n)X_n$, the optimal B is precisely $B^\star A^\star X_n \cdot Z^\dagger$. Recall $Z^\dagger = Z^\top(ZZ^\top)^{-1}$ for row full-rank Z. Substituting back,

$$\min_B \ell^{(n)}(B) \;=\; \tfrac{1}{2}\big\|B^\star(n)A^\star(n)\,X_n\,(I_m - P^{\text{row}}_{A^0(n)X_n})\big\|_F^2, \tag{22}$$

where

$$P^{\text{row}}_{A^0(n)X_n} \;:=\; X_n^\top A^0(n)^\top\big(A^0(n)\,X_nX_n^\top A^0(n)^\top\big)^{-1}A^0(n)\,X_n \;\in\; \mathbb{R}^{m\times m} \tag{23}$$

is the orthogonal projector onto $\text{row}(A^0(n)X_n)$ inside $\mathbb{R}^m$ (the sample-index space), confirming that $B^\star(n)A^\star(n)\,X_n\,(I_m - P^{\text{row}}_{A^0(n)X_n})$ is the part of $B^\star A^\star X_n$ orthogonal to $\text{row}(A^0(n)X_n)$.Bring the residual into the parameter space via the trace identity

$$\big\|B^\star A^\star X_n(I - P^{\text{row}}_{A^0(n)X_n})\big\|_F^2 = \text{tr}\big((B^\star A^\star)\,M_n\,(B^\star A^\star)^\top\big), \quad M_n := X_n(I - P^{\text{row}}_{A^0(n)X_n})X_n^\top. \tag{24}$$

Setting $S := X_nX_n^\top$ and expanding $P^{\text{row}}_{A^0(n)X_n}$ explicitly, M_n admits the *Schur complement representation*

$$M_n \;=\; S - S\,A^0(n)^\top\big(A^0(n)\,S\,A^0(n)^\top\big)^{-1}\,A^0(n)\,S. \tag{25}$$

We assume the per-layer input distribution $\mathcal{D}_X^{(n)}$ produces zero-mean random vectors $x \sim \mathcal{D}_X^{(n)}$ satisfying the shape-aware sub-Gaussian condition equation 13 in Lemma 2 with parameter $\kappa^{(n)} = O(1)$ uniformly across layers and clients, which is the standard condition in or sample-covariance concentration in operator norm (Vershynin, 2018). In specific, it is satisfied by Gaussian and log-concave distributions (Vershynin, 2018). Throughout we treat $\kappa^{(n)}$, $\|\Sigma_X^{(n)}\|_{\text{op}}$, and $\|B^\star A^\star\|_{\text{op}}$ as $\Theta(1)$ constants in the rate notation.

Lemma 2 gives $S/(Km) = \Sigma_X^{(n)} + O_p(1/\sqrt{Km})$, which under the well-conditioned input regime propagates through the Schur complement to

$$M_n/(Km) = \Sigma_X^\perp + O_p(1/\sqrt{Km}), \qquad \Sigma_X^\perp := \Sigma_X^{(n)} - \Sigma_X^{(n)}A^0(n)^\top\big(A^0(n)\Sigma_X^{(n)}A^0(n)^\top\big)^{-1}A^0(n)\,\Sigma_X^{(n)}. \tag{26}$$

Here, the Schur complement $\Sigma_X^{\perp} \in \mathbb{R}^{d\times d}$ has kernel $\mathrm{row}(A^0(n))$, which can be easily derived. We have

$$\tfrac{1}{Km}\,\|B^\star A^\star X_n(I - P^{\mathrm{row}}_{A^0(n)X_n})\|_F^2 \;=\; \mathrm{tr}\big((B^\star A^\star)\,\Sigma_X^{\perp}\,(B^\star A^\star)^\top\big) \;+\; O_p(1/\sqrt{Km}), \tag{27}$$

by combining equation 24–equation 26. Because $\Sigma_X^{\perp}$ has kernel $\mathrm{row}(A^0(n))$, only the part of $B^\star A^\star$ outside $\mathrm{row}(A^0(n))$ contributes, *i.e.* $B^\star A^\star(I - P_{A^0(n)})$. Now, we apply Lemma 1 to

$$\mathrm{tr}\big((B^\star A^\star)\,\Sigma_X^{\perp}\,(B^\star A^\star)^\top\big) = \|B^\star A^\star(\Sigma_X^{\perp})^{1/2}\|_F^2 = \|B^\star A^\star(I - P_{A^0(n)})(\Sigma_X^{\perp})^{1/2}\|_F^2 \tag{28}$$

with $M = B^\star A^\star(I - P_{A^0(n)})$ and $X = (\Sigma_X^{\perp})^{1/2}$ and obtain

$$\lambda^+_{\min}(\Sigma_X^{\perp})\,(\delta^0(n))^2 \;\le\; \mathrm{tr}\big((B^\star A^\star)\,\Sigma_X^{\perp}\,(B^\star A^\star)^\top\big) \;\le\; \lambda_{\max}(\Sigma_X^{\perp})\,(\delta^0(n))^2, \tag{29}$$

where $\lambda^+_{\min}(\Sigma_X^{\perp}) > 0$ is the smallest nonzero eigenvalue (acting on $\ker A^0(n)$) and $\lambda_{\max}(\Sigma_X^{\perp}) \le \|\Sigma_X^{(n)}\|_{\mathrm{op}} = O(1)$. Under the well-conditioned input regime, both eigenvalue bounds are $\Theta(1)$, so the sandwich's two endpoints have the same order $\Theta((\delta^0(n))^2)$. Combined with the $O_p(1/\sqrt{Km})$ correction equation 27, we have

$$\min_B \ell^{(n)}(B) \cdot \tfrac{1}{Km} \;=\; \Theta\big((\delta^0(n))^2\big). \tag{30}$$

Because $\mathcal{R}_{\mathrm{rec}}$ is layer-additive *by definition*, summing the per-layer Θ statements and dividing by N gives, with no further independence assumption,

$$\lim_{T\to\infty} \mathbb{E}\big[\mathcal{R}_{\mathrm{rec}}(\mathbf{W}_T^{\mathrm{FFA}})\big] \;=\; \Theta\Big(\tfrac{1}{N}\sum_n (\delta^0(n))^2\Big), \tag{31}$$

the implicit constant absorbing the per-layer conditioning $\sigma^2_{\max}(X_n)/\sigma^2_{\min}(X_n A^0(n)^\dagger)$, the correction $O_p(1/\sqrt{Km})$, and any SGD noise floor $O(\eta\sigma_g^2)$.

RoLoRA lower bound via pigeonhole. For any layer-tied schedule $M^t \in \{\mathbf{1}_A, \mathbf{1}_B\}$ with A-active set $\mathcal{R}_A := \{t \le T : M^t = \mathbf{1}_A\}$ of size $|\mathcal{R}_A| = \lfloor T/2 \rfloor$, consider an adversarial layer configuration where the per-layer *optimal A-active sets* $\mathcal{S}_1, \dots, \mathcal{S}_N \subseteq [T]$ partition $[T]$ into N equal parts of size T/N each (assuming $N \mid T$ for clarity; the general case differs by $O(1)$). Define the *mismatch set for layer n* as $\mathcal{S}_n \setminus \mathcal{R}_A$, which are the rounds where layer n's optimal protocol calls for A-active but the schedule has B active globally, so $A^t(n)$ remains unchanged and $\delta^t(n)$ cannot contract on these rounds.

Since $\{\mathcal{S}_n\}$ partition $[T]$,

$$\sum_{n=1}^{N} |\mathcal{R}_A \cap \mathcal{S}_n| \;=\; |\mathcal{R}_A| \;=\; T/2, \tag{32}$$

so by averaging, at least one layer $n^\star$ has $|\mathcal{R}_A \cap \mathcal{S}_{n^\star}| \le T/(2N)$. The mismatch count for this layer is

$$|\mathcal{S}_{n^\star} \setminus \mathcal{R}_A| \;=\; |\mathcal{S}_{n^\star}| - |\mathcal{R}_A \cap \mathcal{S}_{n^\star}| \;\ge\; T/N - T/(2N) \;=\; T/(2N) \;=\; \Omega(T). \tag{33}$$

On each mismatch round, $M^t = \mathbf{1}_B$ so $A^t(n^\star)$ is unchanged; the contracting (matched) rounds for layer $n^\star$ are at most $T/(2N)$, half of what AS-LoRA-style adaptive selection could achieve. In the regime where this bounded number of contractions is insufficient to drive δ to zero (e.g., finite-T analysis or contraction ratio close to 1), $\delta^T(n^\star) \ge (1-\epsilon)\,\delta^0(n^\star)$ persists for some $\epsilon \in (0,1)$. Plugging into the per-layer Θ floor yields

$$\mathbb{E}\big[\mathcal{R}_{\mathrm{rec}}(\mathbf{W}_T^{\mathrm{RoLoRA}})\big] \;=\; \Omega\Big(\tfrac{1}{N}\sum_n (\delta^0(n))^2\Big), \tag{34}$$

matching the FFA Θ floor up to the multiplicative factor $(1-\epsilon)^2 \in (0,1]$. □

F Proof of Corollary 1

Fix layer n and consider a round t at which AS-LoRA activates the down-projection block $A^{(n)}$. Recall $\delta^t(n) = \|B^\star(n)A^\star(n)(I - P_{A^t(n)})\|_F$. The per-layer reconstruction loss $\ell^{(n)}(A,B) =$

$\frac{1}{2}\|B^\star A^\star X_n - BAX_n\|_F^2$, viewed as a function of A alone (with B fixed at any value achieving the layer-n optimum given A), is convex quadratic in A with Hessian sandwich $\mu_A^{(n)} I \preceq H_A^{(n)} \preceq L_A^{(n)} I$ (Assumptions 1, 3). Applying Lemma 3 with step size $\eta \le 1/L_A^{(n)}$ yields

$$\|A^{t+1}(n) - A_{\text{opt}}(n)\|_F^2 \;\le\; \big(1 - \mu_A^{(n)}/L_A^{(n)}\big)\, \|A^t(n) - A_{\text{opt}}(n)\|_F^2, \tag{35}$$

where $A_{\text{opt}}(n)$ is the local minimizer at layer n and a particular gauge representative of $A^\star(n)$. As we have seen in the proof of Theorem 2, the map $A \mapsto P_A = A^\dagger A$ is locally Lipschitz on the manifold of full-row-rank matrices. By Davis–Kahn–Wedin sinΘ theorem (Wedin, 1972), for $A, \widetilde{A} \in \mathbb{R}^{r\times d}$ both row-full-rank with smallest singular values bounded below by $\sigma > 0$,

$$\big\|P_A - P_{\widetilde{A}}\big\|_F \;=\; \sqrt{2}\big\|\sin\Theta\big\|_F \;\le\; \frac{C}{\sigma}\,\big\|A - \widetilde{A}\big\|_F \tag{36}$$

for an absolute constant C. Applying with $A = A^{t+1}(n)$, $\widetilde{A} = A_{\text{opt}}(n)$, and $\sigma = \sigma_{\min}(A_{\text{opt}}(n))$ gives

$$\|P_{A^{t+1}(n)} - P_{A_{\text{opt}}(n)}\|_F \;\le\; \frac{C}{\sigma_{\min}(A_{\text{opt}}(n))}\,\|A^{t+1}(n) - A_{\text{opt}}(n)\|_F. \tag{37}$$

Using the fact that $B^\star A^\star (I - P_{A_{\text{opt}}}) = 0$ ($\ker(A^\star) = \ker(A_{\text{opt}})$), we have $\delta^t(n) = \|B^\star A^\star (P_{A_{\text{opt}}} - P_{A^t})\|_F$, so

$$\delta^{t+1}(n) \;\le\; \|B^\star A^\star\|_{\text{op}}\, \|P_{A_{\text{opt}}} - P_{A^{t+1}}\|_F. \tag{38}$$

Combining equation 35 and equation 37, using $\|B^\star A^\star\|_{\text{op}} = \Theta(1)$, $\sigma_{\min}(A_{\text{opt}}) = \Theta(1)$ under the assumption and discussion with Theorem 2, one obtains the per-layer geometric contraction

$$\delta^{t+1}(n) \;\le\; \sqrt{1 - \mu_A^{(n)}/L_A^{(n)}}\cdot \delta^t(n) \;=:\; \gamma_n \cdot \delta^t(n), \qquad \gamma_n \in (0,1). \tag{39}$$

On rounds where $M^t(n) = B$ instead, $A^{(n)}$ is unchanged, so $P_{A^t(n)}$ and hence $\delta^t(n)$ are unchanged. Thus $\delta^t(n)$ is monotonically non-increasing across the trajectory, and contracts strictly only on A-active rounds.

By the hypothesis of the corollary, there exists $c \in (0,1)$ such that $|\{t \le T : M^t(n) = A\}| \ge cT$ for every n (with probability one for argmax, in expectation for softmax). Combining with equation 39, we have

$$\delta^T(n) \;\le\; \gamma_n^{cT} \cdot \delta^0(n). \tag{40}$$

This is the geometric-decay form, where every A-active round multiplies δ by at most $\gamma_n < 1$, and we have at least cT such rounds. Solve $\gamma_n^{cT}\delta^0(n) \le \varepsilon$ for T as

$$T \;\ge\; \frac{\log(\delta^0(n)/\varepsilon)}{c\,\log(1/\gamma_n)}. \tag{41}$$

To guarantee this for every layer n simultaneously, take

$$T \;\ge\; \max_n\; \frac{\log(\delta^0(n)/\varepsilon)}{c\log(1/\gamma_n)} \;=\; O\big(\log(1/\varepsilon)\big), \tag{42}$$

where the absolute constant hides c^{-1} and $\max_n(\log\gamma_n^{-1})^{-1}$, both finite by the assumption of this corollary itself and Assumptions 1–3.

The per-layer Θ floor derived in Theorem 2's proof depends on the current iterate only through $\delta(n)$. Substituting $\delta^T(n) \le \varepsilon$ uniformly into the same sandwich (with ε in place of $\delta^0(n)$):

$$\lim_{T\to\infty} \mathbb{E}\big[\mathcal{R}_{\text{rec}}(\mathbf{W}_T^{\text{AS}})\big] \;=\; O(\varepsilon^2), \tag{43}$$

where the implicit $O(1)$ constant matches the $\Theta(1)$ constant of Theorem 2's FFA floor (same per-layer conditioning $\sigma_{\max}^2(X_n)/\sigma_{\min}^2(X_n A^0(n)^\dagger)$). Whenever $\delta^0(n_0) > \varepsilon$ for some layer n_0, the FFA floor of Theorem 2 is $\Theta(\frac{1}{N}\sum_n(\delta^0(n))^2) \ge \Theta((\delta^0(n_0))^2/N) = \Theta(\delta^0(n_0)^2)$ (treating N as $\Theta(1)$ in the rate notation), so the AS-LoRA bound $O(\varepsilon^2)$ is strictly smaller in order whenever $\varepsilon \ll \delta^0(n_0)$. This proves the strict inequality. □

G PROOF OF THEOREM 3

Fix layer n and block label $a \in \{A, B\}$. Write $\Delta := \mathcal{L}(W) - \mathbb{E}[\mathcal{L}(W - \eta\, g_a^{(n)})]$ and let $h := \eta\, g_a^{(n)}$ (the perturbation at the $a^{(n)}$ block). Apply the Taylor expansion of Lemma 4 to $\mathcal{L}$ as a function of the block factor $a^{(n)}$:

$$\mathcal{L}(W - h) = \mathcal{L}(W) - \langle \nabla_{a^{(n)}} \mathcal{L}, h\rangle + \tfrac{1}{2} h^\top H_a^{(n)}(W) h + R, \qquad |R| \le \tfrac{L_H}{6}\|h\|^3. \tag{44}$$

Substituting $h = \eta g_a^{(n)}$ and rearranging,

$$\mathcal{L}(W) - \mathcal{L}(W - \eta g_a^{(n)}) \;=\; \eta\langle \nabla_{a^{(n)}} \mathcal{L}, g_a^{(n)}\rangle - \tfrac{\eta^2}{2} (g_a^{(n)})^\top H_a^{(n)} g_a^{(n)} - R, \tag{45}$$

with $|R| = O(\eta^3 \|g_a^{(n)}\|^3)$ uniformly bounded under Assumption 3.

Now take expectation over the gradient noise. Decompose $g_a^{(n)} = \nabla_{a^{(n)}}\mathcal{L} + \xi_a^{(n)}$ with $\mathbb{E}[\xi_a^{(n)}] = 0$ and $\mathbb{E}\|\xi_a^{(n)}\|^2 \le \sigma_g^2$ (Assumption 2). Then we have *first-order term.* as $\mathbb{E}[\langle \nabla_{a^{(n)}}\mathcal{L}, g_a^{(n)}\rangle] = \|\nabla_{a^{(n)}}\mathcal{L}\|^2$ by unbiasedness, and *second-order term* as $\mathbb{E}[(g_a^{(n)})^\top H_a^{(n)} g_a^{(n)}] \;=\; (\nabla_{a^{(n)}}\mathcal{L})^\top H_a^{(n)} \nabla_{a^{(n)}}\mathcal{L} \;+\; \mathrm{tr}(H_a^{(n)}\Sigma_\xi)$ where $\Sigma_\xi = \mathbb{E}[\xi\xi^\top]$ has $\mathrm{tr}\Sigma_\xi \le \sigma_g^2$ and $H_a^{(n)} \preceq \beta I$, so $\mathrm{tr}(H_a^{(n)}\Sigma_\xi) \le \beta\sigma_g^2$.

Plugging back to equation 45, we have

$$\Delta \;=\; \eta\|\nabla_{a^{(n)}}\mathcal{L}\|^2 - \tfrac{\eta^2}{2}(\nabla_{a^{(n)}}\mathcal{L})^\top H_a^{(n)} \nabla_{a^{(n)}}\mathcal{L} - \tfrac{\eta^2}{2}\beta\sigma_g^2 + O(\eta^3). \tag{46}$$

The score satisfies $\mathbb{E}[S_a(n)] = \|\nabla_{a^{(n)}}\mathcal{L}\|^2 - \tfrac{\eta}{2}(\nabla_{a^{(n)}}\mathcal{L})^\top H_a^{(n)}\nabla_{a^{(n)}}\mathcal{L} - \tfrac{\eta}{2}\mathrm{tr}(H_a^{(n)}\Sigma_\xi) + O(\sigma_g^2)$. Accordingly, modulo $O(\eta\sigma_g^2)$ correction which is absorbed for $\eta \le 1/\beta$,

$$\Delta \;\ge\; \eta\, S_a(n) - \tfrac{\eta^2\beta}{2}\sigma_g^2. \tag{47}$$

Therefore, we can say that the greedy rule $\arg\max_a S_a(n)$ thus maximizes Δ up to $O(\eta^2\sigma_g^2)$.

Applying equation 47 at round t with the selected block $M^t(n)$, summing across layers $n \in [N]$ and rounds $t \in [T]$, and using $\mathcal{L}(W_T) \ge \mathcal{L}^\star$, we have

$$\mathcal{L}(W_0) - \mathcal{L}^\star \;\ge\; \mathbb{E}[\mathcal{L}(W_0)] - \mathbb{E}[\mathcal{L}(W_T)] \;\ge\; \eta \sum_{t=1}^{T}\sum_{n=1}^{N} \mathbb{E}\big[S^t_{M^t(n)}(n)\big] - \tfrac{\eta^2\beta NT}{2}\sigma_g^2. \tag{48}$$

Decompose AS-LoRA's chosen score against the *uniform-random baseline* $\frac{1}{2}(S_A^t(n) + S_B^t(n))$:

$$S^t_{M^t(n)}(n) \;=\; \tfrac{1}{2}\big(S_A^t(n) + S_B^t(n)\big) + \Big(S^t_{M^t(n)}(n) - \tfrac{1}{2}\big(S_A^t(n) + S_B^t(n)\big)\Big). \tag{49}$$

Under AS-LoRA's argmax rule, $S^t_{M^t(n)}(n) = \max_a S_a^t(n)$, so the second term has expectation exactly $\Delta_{\mathcal{S}}^t(n)$ (the adaptive gain, defined as the gap between $\mathbb{E}[\max_a S_a^t(n)]$ and the uniform-baseline expectation $\frac{1}{2}\mathbb{E}[S_A^t(n) + S_B^t(n)]$ in the main text). Hence

$$\mathbb{E}[S^t_{M^t(n)}(n)] \;=\; \tfrac{1}{2}\mathbb{E}[S_A^t(n) + S_B^t(n)] + \Delta_{\mathcal{S}}^t(n). \tag{50}$$

To convert the score expression into a gradient-norm bound, we apply Assumption 3 to each individual block-score $S_a(n)$. Using $H_a^{(n)} \preceq L_a^{(n)} I \preceq \beta I$ and $\eta \le 1/\beta$, the curvature term satisfies $(g_a^{(n)})^\top H_a^{(n)} g_a^{(n)} \le \beta\|g_a^{(n)}\|^2$, so

$$S_a(n) \;=\; \|g_a^{(n)}\|^2 - \tfrac{\eta}{2}\lambda_a^{(n)}\|g_a^{(n)}\|^2 \;=\; \big(1 - \tfrac{\eta}{2}\lambda_a^{(n)}\big)\|g_a^{(n)}\|^2 \;\ge\; \rho^\star\, \|g_a^{(n)}\|^2, \tag{51}$$

by definition of $\rho^\star = \inf_{t,n}(1 - \frac{\eta}{2}\lambda^{(n)}_{M^t(n)}) \ge 1/2$. Combining equation 50–equation 51 and $\mathbb{E}\|g_a^{(n)}\|^2 \ge \|\nabla_{a(n)}\mathcal{L}\|^2$ (since $\mathbb{E}[g_a^{(n)}] = \nabla_{a(n)}\mathcal{L}$ and variance ≥ 0), we have

$$\sum_n \mathbb{E}[S^t_{M^t(n)}(n)] \;\ge\; \tfrac{\rho^\star}{2}\sum_n \mathbb{E}\|\nabla_{M^t(n)}\mathcal{L}\|^2 + \sum_n \Delta_{\mathcal{S}}^t(n). \tag{52}$$

Combining equation 48, equation 52 and $\min_{t\le T}\mathbb{E}\|\nabla\mathcal{L}(W_t)\|^2 \le \frac{1}{T}\sum_t \mathbb{E}\|\nabla\mathcal{L}(W_t)\|^2$, we obtain

$$\min_{t\le T}\mathbb{E}\|\nabla\mathcal{L}(W_t)\|^2 \;\le\; \frac{2(\mathcal{L}(W_0)-\mathcal{L}^\star)}{\eta\rho^\star T} \;-\; \frac{2\,\overline{\Delta_{\mathcal{S}}}}{\rho^\star} \;+\; \frac{\eta\beta\sigma_g^2}{\rho^\star}, \tag{53}$$

by dividing by $\eta\rho^\star T/2$ where $\overline{\Delta_{\mathcal{S}}} = \frac{1}{T}\sum_t \frac{1}{N}\sum_n \Delta_{\mathcal{S}}^t(n)$. This is the non-convex bound.

PL refinement to geometric rate. Under Assumption 4 ($\|\nabla\mathcal{L}\|^2 \ge 2\mu(\mathcal{L}-\mathcal{L}^\star)$), the per-round inequality equation 47 sharpens from a gradient-norm lower bound to a function-value recursion. Combining equation 47 with equation 51, we have

$$\mathbb{E}[\mathcal{L}(W_t)-\mathcal{L}^\star] \;\le\; \mathbb{E}[\mathcal{L}(W_{t-1})-\mathcal{L}^\star] - \tfrac{\eta\rho^\star}{2}\,\mathbb{E}\|\nabla\mathcal{L}(W_{t-1})\|^2 + \tfrac{\eta^2\beta\sigma_g^2}{2}. \tag{54}$$

Applying PL to the negative-coefficient gradient-norm term which is $-\frac{\eta\rho^\star}{2}\,\mathbb{E}\|\nabla\mathcal{L}\|^2 \le -\eta\mu\rho^\star\,\mathbb{E}[\mathcal{L}-\mathcal{L}^\star]$, equation 54 can be rewritten as

$$\mathbb{E}[\mathcal{L}(W_t)-\mathcal{L}^\star] \;\le\; (1-\eta\mu\rho^\star)\,\mathbb{E}[\mathcal{L}(W_{t-1})-\mathcal{L}^\star] + \tfrac{\eta^2\beta\sigma_g^2}{2}. \tag{55}$$

This is a first-order recursion in $\mathbb{E}[\mathcal{L}(W_t)-\mathcal{L}^\star]$, contracted by $\gamma := 1-\eta\mu\rho^\star \in (0,1)$ each round (provided $\eta < 1/(\mu\rho^\star)$, automatically satisfied under $\eta \le 1/\beta$ in the typical regime $\beta \ge \mu\rho^\star$). Considering the iteration, the first-order recusrion is extended to

$$\mathbb{E}[\mathcal{L}(W_T)-\mathcal{L}^\star] \;\le\; (1-\eta\mu\rho^\star)^T(\mathcal{L}(W_0)-\mathcal{L}^\star) + \tfrac{\eta\beta\sigma_g^2}{2\mu\rho^\star}. \tag{56}$$

□

H Proof of Theorem 4

Fix layer n and let $\kappa(H_a^{(n)}) := \lambda_{\max}(H_a^{(n)})/\lambda_{\min}(H_a^{(n)})$ denote the condition number of the per-block Hessian. Under AS-LoRA's argmax rule $M^t(n) = \arg\max_{a\in\{A,B\}} S_a(n)$, the selected block is a deterministic function of the stochastic gradients $\{g_a^{(n)}\}_{a\in\{A,B\}}$. Since $g_a^{(n)}$ is random (mini-batch noise, Assumption 2), $M^t(n)$ inherits this randomness, and we define the *expected Hessian at the selected mode*

$$\overline{H}_t^{(n)} \;:=\; \mathbb{E}\big[H_{M^t(n)}^{(n)}\big] \;=\; \sum_{a\in\{A,B\}} p^t(a,n)\,H_a^{(n)}, \tag{57}$$

where $p^t(a,n) := \Pr\big[M^t(n)=a\big]$ is the *induced selection probability* which is the probability that argmax picks block a, taken over the distribution of stochastic gradients. $\overline{H}_t^{(n)}$ is a convex combination of $H_A^{(n)}, H_B^{(n)}$ with weights $p^t(\cdot,n)$. The score $S_a(n) = \|g_a^{(n)}\|^2 - \frac{\eta}{2}(g_a^{(n)})^\top H_a^{(n)} g_a^{(n)}$ contains a curvature-penalty term that is large when $\lambda_a^{(n)} = (g_a^{(n)})^\top H_a^{(n)} g_a^{(n)}/\|g_a^{(n)}\|^2$ is large, i.e. when block a has high effective curvature along the gradient direction. Hence among two blocks with comparable gradient norms, the block with smaller $\lambda_a^{(n)}$ has larger $S_a(n)$ and is therefore selected by argmax. Formally, under heterogeneity $\lambda_A^{(n)} \ne \lambda_B^{(n)}$, the argmax-induced selection probability satisfies $p^t(a^\star,n) > 1/2$ where $a^\star := \arg\min_a \lambda_a^{(n)}$, with the gap $p^t(a^\star,n) - 1/2$ strictly positive and growing with the curvature heterogeneity $|\lambda_A^{(n)} - \lambda_B^{(n)}|$. Accordingly, we can say that AS-LoRA favors to select $a \in A, B$ with low $\lambda_a^{(n)}$, and also maximum eigenvalue of $H_a^{(n)}$ following the observation in (Jaiswal et al., 2025) that demonstrates that *the gradient subspace aligns with the dominant directions in the Hessian.*

For symmetric matrices the maximum eigenvalue is convex (Horn & Johnson, 2012), so by Jensen's inequality

$$\lambda_{\max}\big(\overline{H}_t^{(n)}\big) \;=\; \lambda_{\max}\big(\mathbb{E}_a[H_a^{(n)}]\big) \;\le\; \mathbb{E}_a[\lambda_{\max}(H_a^{(n)})] \;=\; \sum_a p^t(a,n)\,\lambda_{\max}(H_a^{(n)}). \tag{58}$$

Comparing AS-LoRA's induced distribution p^t against the uniform baseline,

$$\sum_a p^t(a,n)\,\lambda_{\max}(H_a^{(n)}) \;\le\; \tfrac{1}{2}(\lambda_{\max}(H_A^{(n)}) + \lambda_{\max}(H_B^{(n)})) = \mathbb{E}_a^{\text{unif}}[\lambda_{\max}(H_a^{(n)})], \tag{59}$$

because AS-LoRA favors low maximum eigenvalue as we discussed above. Composing,

$$\mathbb{E}[\lambda_{\max}(\overline{H}_t^{(n),\mathrm{AS}})] \;\le\; \mathbb{E}[\lambda_{\max}(\overline{H}_t^{(n),\mathrm{static}})], \tag{60}$$

strict whenever the per-block top-spectra differ. This is the SAM-style implicit-flatness statement of Theorem 4. □

I Experimental Detail

I.1 Differential Privacy Implementation

We implement differentially private training using the Opacus library. To track the privacy budget, we primarily adopt the PRV (Privacy Random Variable) accountant, which provides tighter estimates compared to the standard RDP accountant. For a subset of kernel-based experiments, we instead use the RDP accountant due to technical limitations in the PRV implementation under those settings.

We employ Poisson subsampling using the `UniformWithReplacementSampler` provided by Opacus. Each data point is independently sampled with probability $q = B/N_k$, where B is the expected batch size and N_k denotes the dataset size of client k. This sampling scheme ensures correct privacy accounting for DP-SGD.

I.2 Model and LoRA Configuration

For language tasks, we apply LoRA to all transformer layers, specifically to the query and value projection matrices. The LoRA modules are configured with a dropout rate of 0.05 and without bias terms (i.e., `bias="none"`).

For vision tasks, LoRA is similarly applied to the query and value projections across all layers. In addition, the classifier head is trained jointly without LoRA adaptation.

I.3 Tokenization and Input Processing

For GLUE tasks, the maximum sequence length is set to 128, while for SQuAD tasks it is set to 384. For question answering, we tokenize the input using the following setting:

```
tokenizer(question, context, truncation="only_second",
max_length=L, padding="max_length", return_offsets_mapping=True)
```

I.4 Implementation Details

All experiments are conducted in float32 precision to ensure numerical stability under differential privacy. We reinitialize the model, optimizer, and privacy engine at each communication round to avoid unintended state accumulation in the privacy accountant.

During training, only the selected LoRA components are optimized according to the layer-wise mode configuration, while all other parameters remain frozen.

J Additional Experiments

Unless otherwise specified, all experiments follow the settings described in Section 5 and Appendix I.

J.1 Effect of random projection

To examine whether the benefit of random projection is related to channel-wise gradient outliers, we vary the batch size in DP-SGD, since larger batches can average stochastic gradient fluctuations and reduce the effect of outliers. As shown in Table 9, the performance gap between Proposed and Proposed w/o RP decreases as the batch size increases. At B=32, random projection improves QQP accuracy by 2.03 pp, while the improvement reduces to 0.79 pp at B=128. This trend suggests

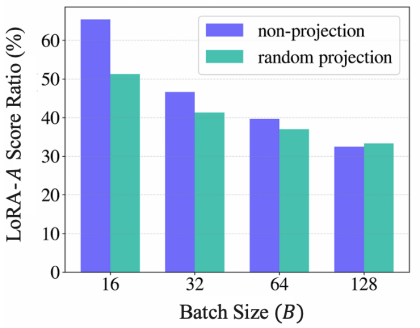


Figure 7: Comparison of LoRA-A score ratio with and without random projection across different batch sizes.

that random projection is particularly effective when smaller batch sizes make channel-wise outliers less likely to be canceled through batch averaging. As the batch size increases and this implicit averaging effect becomes stronger, the additional benefit of random projection becomes smaller.

Figure 7 further corroborates this interpretation by visualizing the LoRA-A score ratio across different batch sizes. At small batch sizes (e.g., B=16), the LoRA-A score ratio under non-projection is substantially higher than that under random projection, indicating that channel-wise outliers inflate the A component's score and cause it to be disproportionately favored in mode selection. As the batch size increases, the gap between non-projection and random projection narrows steadily, and at B=128 the two nearly coincide. This confirms that larger batches implicitly suppress outlier effects through averaging, reducing the discrepancy between the two scoring strategies. Taken together, these results support that the proposed random projection effectively mitigates channel-wise gradient outliers and leads to more balanced and reliable score estimation, particularly in the small-batch regime typical of DP-SGD training.

Table 9: Comparison with/without random projection under DP-SGD ($\epsilon = 6$).

Method	QQP (B=32)	QQP (B=64)	QQP (B=128)
Proposed	**63.19** $\pm$ 0.01	**73.99** $\pm$ 1.41	**79.94** $\pm$ 0.11
Proposed w/o RP	61.16 $\pm$ 2.03	70.90 $\pm$ 2.78	79.15 $\pm$ 0.69

J.2 Single-Pass vs. Two-Pass Score Computation

We compare two designs for score computation that differ in when and how the scores are evaluated relative to the communication rounds.

In the *single-pass* design (ours), each client computes the scores as a by-product of local training and uploads them to the server together with the updated LoRA parameters. The server then averages the uploaded scores across clients and uses them to determine the mode vector for the *next* round, requiring no additional communication beyond the standard parameter exchange. As a result, the mode selection at round t is based on the scores of round $t-1$ rather than the current global model state.

Table 10: Comparison of single-pass and two-pass under DP-SGD ($\epsilon = 3$).

Method	**MNLI (m)**	**MNLI (mm)**
single-pass (Proposed)	78.83 $\pm$ 0.27	79.97 $\pm$ 0.55
two-pass inner-loop computation	79.37 $\pm$ 0.31	80.29 $\pm$ 0.26

In the *two-pass inner-loop* design, the server first broadcasts both the A and B components of the current global model to all clients. Each client then evaluates the scores for both components locally and sends them back to the server. The server aggregates the received scores and determines the mode vector based on the current global model, after which the selected components are redistributed to the clients for local training. This design enables greedy, round-wise mode selection based on the most up-to-date global model state, at the cost of an additional communication round and extra local computation per round.

As shown in Table 10, the two-pass inner-loop computation achieves slightly higher performance than the single-pass design, improving MNLI (m) and MNLI (mm) by $+0.54$ and $+0.32$ pp, respectively. However, the performance gap is marginal, and the single-pass design avoids the additional commu-

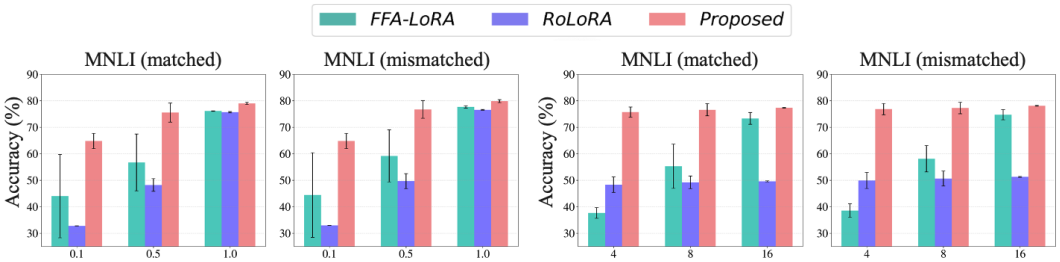
nication and computation overhead introduced by the two-pass approach. These results demonstrate that the single-pass design provides a favorable trade-off between performance and efficiency.

J.3 Effect of Aggregation Rule

We further study alternative rules for aggregating layer-wise mode decisions across clients. Specifically, we consider three variants: (i) *majority voting*, where each client independently selects its preferred mode and the server adopts the mode receiving the majority of votes; (ii) *weighted averaging*, which computes a data-size weighted average of client scores before selecting a mode; and (iii) *uniform averaging* (ours), which averages client scores uniformly and determines the mode accordingly. As shown in Table 11, majority voting and weighted averaging achieve higher performance than uniform averaging on both MNLI (m) and MNLI (mm), suggesting that the aggregation rule for mode selection can meaningfully affect the quality of the selected modes. We adopt uniform averaging as the default aggregation strategy throughout this paper. Even with this simple aggregation, AS-LoRA outperforms prior federated LoRA methods; further gains are expected when combined with more sophisticated aggregation rules including majority rule, weighted aggregation.

Table 11: Comparison of mode selection rules. Experiments are conducted under DP-SGD ($\epsilon = 6$) with $B = 64$.

Mode Selection Strategy	MNLI (m)	MNLI (mm)
Majority voting	**75.56** $\pm$ 1.51	76.31 $\pm$ 2.38
Weighted averaging	75.50 $\pm$ 1.58	**76.60** $\pm$ 1.97
Uniform averaging (Proposed)	74.48 $\pm$ 0.04	75.15 $\pm$ 0.35



(a) α for Dirichlet distribution (b) LoRA Rank r

Figure 8: (a): Comparison under varying dirichlet distribution $\alpha \in \{0.1, 0.5, 1.0\}$ on the MNLI dataset. (b): Comparison under varying LoRA rank $r \in \{4, 8, 16\}$ on the MNLI dataset. The number of clients is fixed at $K = 3$ under DP-SGD ($\epsilon = 6$).

J.4 Effect of LoRA Rank and Data Distribution

In Figure 8(a), the proposed method consistently achieves the best performance across all α values and shows larger gains at smaller α. This suggests stronger robustness under highly heterogeneous settings. The advantage of the proposed method is maintained across all heterogeneity levels. In Figure 8(b), performance generally improves as the LoRA rank increases. The proposed method consistently outperforms the baselines across all rank settings and maintains its advantage even at low ranks. Notably, the performance improvement of FFA-LoRA is more pronounced than that of RoLoRA as the rank increases, suggesting that FFA-LoRA benefits more from the increased expressive capacity provided by larger ranks.

J.5 Communication and Memory Overhead Analysis

We analyze the additional overhead introduced by AS-LoRA. Let $\mathcal{M}_n$ denote the set of LoRA modules in Transformer layer n. We define

$$P_A = \sum_{n=1}^{N} \sum_{m \in \mathcal{M}_n} rd_{\text{in},m}(n), \qquad P_B = \sum_{n=1}^{N} \sum_{m \in \mathcal{M}_n} rd_{\text{out},m}(n),$$

where P_A and P_B denote the number of trainable parameters in the A and B LoRA components, respectively. FedLoRA communicates $P_A + P_B$ parameters per client per round, whereas FFA-LoRA

communicates P_B, and RoLoRA communicates either P_A or P_B. AS-LoRA also communicates only one active component per layer and therefore has the same order of parameter communication as alternating methods.

The additional communication of AS-LoRA comes only from the score and mode information. In our implementation, all LoRA modules within the same Transformer layer share one layer-level mode decision. Thus, each client uploads one scalar score per Transformer layer, and the server broadcasts one binary mode indicator per layer. The additional uplink and downlink costs are therefore N floating-point values and N bits per round, respectively. For a 24-layer backbone, this corresponds to only 24 FP32 values, or 96 bytes, per client per round, which is less than 0.02% of the active LoRA parameter communication when the hidden dimension is 1024.

J.6 Effect of gradient smoothing under DP-SGD

We investigate whether applying smoothing to DP-SGD gradients can further improve utility, motivated by the observation that injected Gaussian noise in DP-SGD is high-frequency in nature and may be partially mitigated by frequency-selective filtering. Specifically, we compare three smoothing strategies applied to the aggregated gradients: (i) a standard 5-tap Gaussian filter $[1/16,\ 4/16,\ 6/16,\ 4/16,\ 1/16]$ as proposed in (Liu et al., 2026), (ii) Laplacian smoothing (Liang et al., 2024), and (iii) EMA across communication rounds.

Table 12: Effect of gradient smoothing methods under DP-SGD ($\epsilon = 3$).

Method	MNLI (m)	MNLI (mm)
Proposed (w/o smoothing)	78.83 ± 0.27	79.97 ± 0.55
Gaussian	78.85 ± 0.14	79.74 ± 0.21
Laplacian	**79.48** ± 0.16	**80.31** ± 0.16
EMA	77.70 ± 0.19	78.65 ± 0.14

The Gaussian noise injected by DP-SGD is i.i.d. across parameters and thus behaves as white noise in the parameter space, concentrating its energy uniformly across all frequencies. In contrast, the true gradient signal is typically smooth and sparse in the frequency domain, with energy concentrated at low frequencies. In the frequency domain, Laplacian smoothing acts as:

$$\hat{g}_{\mathrm{smooth}}(\omega) = \frac{\hat{g}(\omega)}{1 + \sigma\lambda(\omega)}, \tag{61}$$

where $\lambda(\omega)$ denotes the eigenvalue of the graph Laplacian at frequency ω and $\sigma > 0$ is a smoothing coefficient. Since $\lambda(\omega) \approx 0$ for low frequencies, the true gradient signal passes through largely intact, whereas high-frequency noise components are strongly suppressed as $\lambda(\omega)$ grows. By contrast, the 5-tap Gaussian filter is a weighted average with fixed kernel $[1/16, 4/16, 6/16, 4/16, 1/16]$, which attenuates *all* frequency components including low-frequency signal, leading to signal distortion. EMA operates along the time axis rather than the parameter space, and therefore cannot exploit the spatial structure of DP noise within a single round, making it the least compatible with the spatial nature of DP-SGD noise.

Table 12 reports the effect of each smoothing method under $\epsilon = 3$ on the MNLI matched (m) and mismatched (mm) benchmarks. The results are consistent with our frequency-domain analysis. Laplacian smoothing is the only method that improves over the no-smoothing baseline, gaining $+0.33$ and $+0.28$ pp on MNLI (m) and MNLI (mm), respectively. Gaussian filtering slightly degrades performance relative to the baseline (-0.30 pp on MNLI (m)), confirming that indiscriminate attenuation of all frequency components distorts the gradient signal. EMA yields the largest degradation (-1.45 pp on MNLI (m)), as temporal averaging across rounds is ill-suited for suppressing within-round spatial noise introduced by DP-SGD. These findings suggest that frequency-selective smoothing, specifically Laplacian smoothing, is the most compatible post-processing strategy for DP-SGD gradients.

J.7 Analysis of Layer-wise Mode Selection Patterns

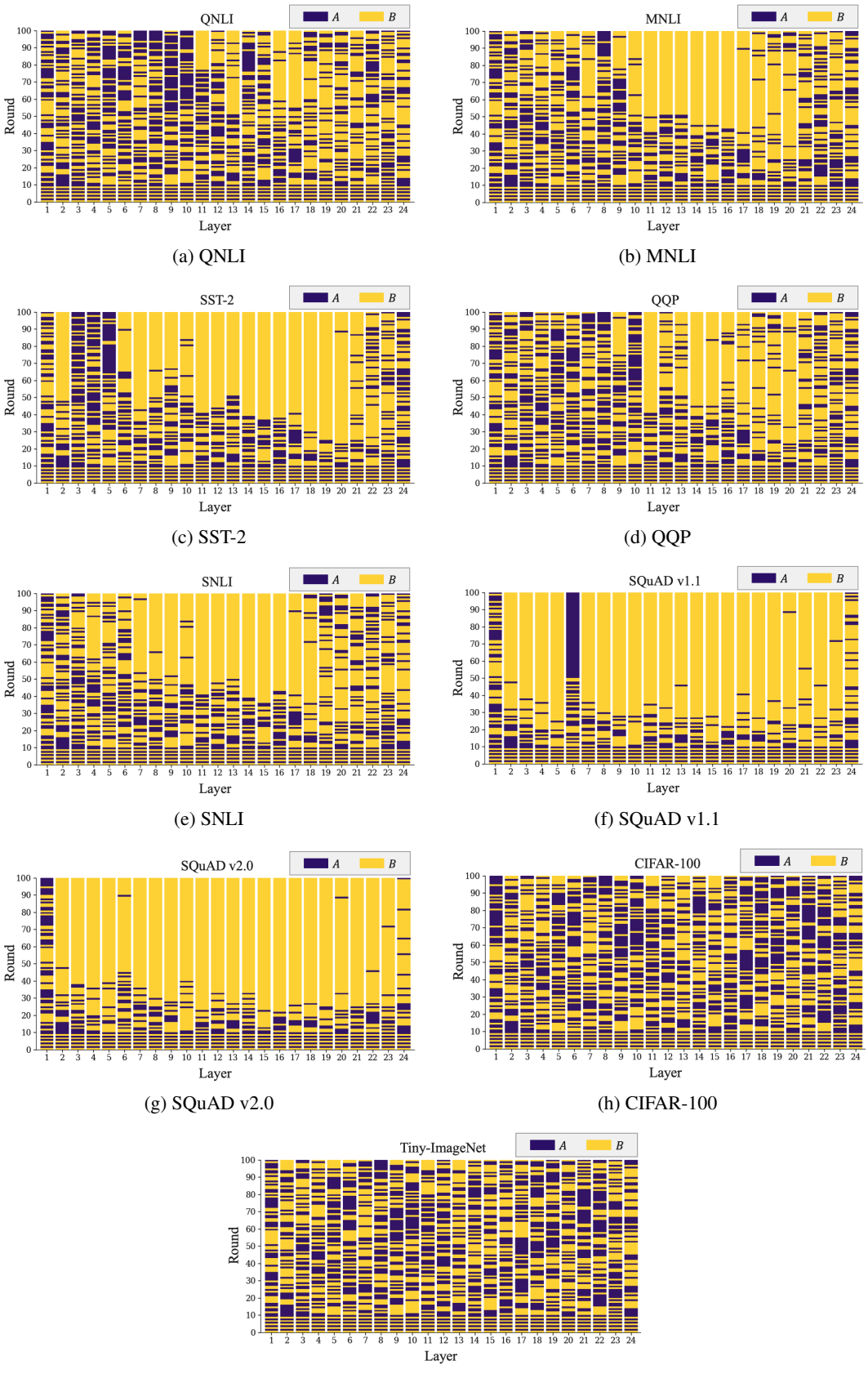
Figure 9 visualizes the layer-wise mode selection patterns over training rounds across different tasks, revealing several non-random and task-dependent behaviors. Overall, the proposed method does not produce uniformly alternating selections across layers; instead, each task exhibits its own characteristic layer-wise preference structure, indicating that the adaptive scores capture meaningful optimization differences between LoRA-A and LoRA-B components rather than inducing arbitrary switching behavior.

For language understanding tasks such as QNLI, MNLI, SST-2, QQP, and SNLI, mode selection patterns often exhibit partially coherent structures across neighboring layers. Certain intermediate or upper layers consistently favor the same component over many rounds, while other layers switch more frequently. This suggests that the relative utility of LoRA-A and LoRA-B varies across Transformer depth, with different layers playing distinct roles in private federated adaptation depending on both task and layer.

In contrast, vision tasks exhibit more mixed and fine-grained switching. On CIFAR-100 and Tiny-ImageNet, selected modes distribute more evenly across layers and rounds, with fewer large contiguous regions dominated by a single component. This implies that in vision tasks, the relative advantage between LoRA-A and LoRA-B varies more locally across rounds, leading to less stable layer-wise specialization than in language tasks.

These observations have two key implications. First, they support the need for adaptive layer-wise mode selection, as a fixed global schedule cannot capture such heterogeneous preferences across tasks and layers. Second, recurring structures in language tasks suggest cross-layer dependencies not explicitly modeled by the current greedy formulation. Although each AS-LoRA layer independently selects its active component based on its score, the dynamics imply that groups of layers may prefer similar modes or evolve in a correlated manner over training rounds.

This also highlights a limitation of the current framework. Our analysis and algorithm focus on layer-wise greedy selection, enabling tractable optimization and strong empirical performance, but do not exploit inter-layer correlations. Future work could extend AS-LoRA to a joint decision framework that models structured dependencies across layers, for example by encouraging coordinated mode assignments among related layers or learning higher-level scheduling policies over layer groups. Such extensions may improve global coordination and yield more efficient update schedules beyond the current independent layer-wise strategy.



(a) QNLI (b) MNLI

(c) SST-2 (d) QQP

(e) SNLI (f) SQuAD v1.1

(g) SQuAD v2.0 (h) CIFAR-100

(i) Tiny-ImageNet

Figure 9: Layer-wise mode selection patterns across different tasks.