Stochastic Projections and Rank-Aware Learning Rates for Federated LoRA

Saber Malekmohammadi
Stanford University
sabermm@stanford.edu

Kevin Kuo
Carnegie Mellon University
kkuo2@andrew.cmu.edu

Virginia Smith
Carnegie Mellon University
smithv@cmu.edu

Golnoosh Farnadi
McGill University, MILA
farnadig@mila.quebec

Abstract

Fine-tuning large-scale foundation models across resource-constrained, distributed clients presents significant computational and communication challenges. While merging Low-Rank Adaptation (LoRA) with Federated Learning (FL) enables rank-based flexibility, the interdependence of the adaptation matrices ($\mathbf{A}$ and $\mathbf{B}$) often complicates optimization. In this work, we theoretically demonstrate that LoRA fine-tuning with Gaussian initialization and a frozen $\mathbf{A}$ matrix acts as a stochastic approximation of full fine-tuning, where batch gradients are subject to a random, rank-dependent perturbation. Motivated by this insight, we propose freezing and periodically resampling $\mathbf{A}$ matrices at the client level. This strategy decouples the $\mathbf{A}$ and $\mathbf{B}$ gradients, diminishes LoRA low-rank constraints and reduces both computational and uplink communication overheads by 50%, allowing for the reallocation of resources toward higher adaptation ranks. Furthermore, we show that in this frozen-$\mathbf{A}$ regime, the adaptation rank r directly governs the signal-to-noise ratio (SNR) of the perturbed gradients. Specifically, we find that the SNR increases linearly with r, implying that clients' learning rates must be dynamically scaled with their adaptation ranks for higher utility. Building on these foundations, we introduce SPRA-LoRA (**S**tochastic **P**rojection and **R**ank-**A**ware LoRA), a federated rank-aware fine-tuning strategy that utilizes frozen, resampled $\mathbf{A}$ matrices and rank-dependent learning rates for clients. We also prove its convergence with heterogeneous client resources. Our experiments across benchmark datasets demonstrate that SPRA-LoRA consistently outperforms state-of-the-art baselines with half the overhead on clients. The implementation codes for the project are available at this link: https://github.com/Saber-mm/SPRA-LoRA.

1 Introduction

Adapting foundation models (e.g., GPT [1], BERT [2] and ViT [3]) to downstream tasks requires the use of fine-tuning datasets that are often unavailable to or represented by just a single client. Federated learning (FL) [4] addresses this by enabling collaborative fine-tuning across multiple clients in a privacy-preserving manner. Despite the promise of FL, the large number of parameters in foundation models can make federated fine-tuning expensive.

To reduce the computational and communication costs of federated fine-tuning, Parameter Efficient Fine-Tuning (PEFT) methods [5–7] freeze most pre-trained parameters, updating only a small subset. Specifically, LoRA [7] is a popular PEFT technique that has been integrated with FedAvg [4] for federated fine-tuning of LLMs [8]. LoRA formulates the update to an *adapter* $\mathbf{W}^0 \in \mathbb{R}^{n\times m}$ of a pre-

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

trained model as $\mathbf{W} = \mathbf{W}^0 + s\mathbf{BA}$, where $\mathbf{A} \in \mathbb{R}^{r\times m}$ are two low-rank matrices of rank r, where $r \ll \min\{m, n\}$. Also, $s = \alpha/r^\nu$ is a scaling factor, where α is a hyperparameter. This reduces the number of trainable parameters from mn to $r(m + n)$, making the rank r a key hyperparameter for controlling client-side computational and communication costs in federated fine-tuning.

Various strategies have been adopted in the literature to combine LoRA with FL. Some works directly apply the FedAvg algorithm [4] to the LoRA components $\mathbf{A}$ and $\mathbf{B}$ and average them on the server as $\mathbf{A} = \sum_i w_i\mathbf{A}_i$ and $\mathbf{B} = \sum_i w_i\mathbf{B}_i$, where w_i is the aggregation weight assigned to client i [8]. However, noting that LoRA models the adapter update as $\Delta\mathbf{W} = \mathbf{sBA}$, this aggregation strategy suffers from an ***aggregation noise*** [9, 10] rooted in the fact that:

$$\big(\sum_i w_i\mathbf{B}_i\big)\big(\sum_i w_i\mathbf{A}_i\big) \neq \sum_i w_i\Delta\mathbf{W}_i = \sum_i w_i(s_i\mathbf{B}_i\mathbf{A}_i). \tag{1}$$

To address the aggregation noise, existing methods either 1) alternatingly update $\mathbf{A}_i$ and $\mathbf{B}_i$ (between clients' local optimization steps or between communication rounds) and aggregate them separately [11, 12] (this requires uniform ranks across clients), or 2) simultaneously update both and aggregate the resulting adapter updates $\{\Delta\mathbf{W}_i\} = \{s_i\mathbf{B}_i\mathbf{A}_i\}$ [9, 10].

We demonstrate that both the above methods are sub-optimal. The first approach wastes significant client computation and communication on fine-tuning/transmitting the $\mathbf{A}_i$ matrices, which we show, for Gaussian-initialized LoRA, function more as random gradient perturbations than as essential knowledge-storing parameters. The second approach, while solving the aggregation noise problem, suffers from a ***gradient coupling*** issue between the $\mathbf{A}$ and $\mathbf{B}$ matrices: the current gradient update of $\mathbf{A}$ depends on the current $\mathbf{B}$, and vice versa. Hence, when fine-tuned *simultaneously*, one matrix adapts to a basis that the other has already shifted away from and this induces the ***gradient coupling noise*** [12]. While the aforementioned alternate optimization of the two matrices can avoid this noise [12, 11], it spends half of the computation/communication resources of each client i on fine-tuning/transmitting its $\mathbf{A}_i$ matrix. Avoiding the noise via freezing matrix $\mathbf{A}_i$, our proposed method spends all computation and communication resources of the client on matrix $\mathbf{B}_i$.

Freezing client matrices $\mathbf{A}_i$ to their initial value saves computational and communication resources at the cost of a lower model utility [11, 12]. To balance efficiency and utility, we propose resampling and freezing $\mathbf{A}_i$ matrices at each round. This approach avoids the gradient coupling noise found in existing resampling strategies—like FLoRA [10]—where both matrices are fine-tuned simultaneously. Consequently, our method cuts clients' overhead by half, eliminates gradient interference between $\mathbf{A}_i$ and $\mathbf{B}_i$, and mitigates the low-rank limitations of LoRA.

Finally, we demonstrate that per-round resampling and freezing of $\mathbf{A}_i$ approximates full fine-tuning under rank-dependent gradient perturbations, characterized by gradient scaling and additive noise. The signal-to-noise ratio (SNR) of perturbed gradients increases with the adaptation rank r. Therefore, clients with heterogeneous ranks experience varying levels of gradient perturbation. This variance renders a uniform learning rate suboptimal. We thus also propose rank-dependent learning rate adjustments to compensate for this heterogeneity and achieve further performance improvement.

Our main contributions are summarized below:

- We introduce SPRA-LoRA, a federated fine-tuning framework for rank-heterogeneous environments. By resampling and freezing client $\mathbf{A}_i$ matrices, it cuts computation and uplink communication costs by 50% while mitigating LoRA's low-rank constraints.
- SPRA-LoRA employs rank-aware learning rates for clients to account for their different levels of gradient perturbation. This tailored approach stabilizes training, leading to faster convergence and superior performance compared to the existing methods using uniform learning rates.
- We provide a convergence proof for SPRA-LoRA in non-convex settings, accounting for variations in rank, data and system resources. Our analysis demonstrates that SPRA-LoRA is fundamentally equivalent to performing Fed-Avg on adapters with noisier batch gradients.
- Experimental results across multiple models and language tasks show that SPRA-LoRA improves utility and convergence speed with 50% less computation and uplink communication overhead on clients compared to state-of-the-art baselines.

Table 1: Comparison of Fed-LoRA algorithms. ✕: NO. N_i is dataset size of client i; n^e is the total clients in round e. Red rows indicate algorithms for uniform ranks; blue for heterogeneous ranks.

algorithm	aggregation strategy	freezing $\{\mathbf{A}_i\}$ or $\{\mathbf{B}_i\}$	resampling $\{\mathbf{A}_i\}$	non-uniform learning rate
SLoRA [2023]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately	✕	✕	✕
HetLoRA [2024]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately	✕	✕	✕
FeDeRA [2024]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately	✕	✕	✕
FedIT [2024]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately	✕	✕	✕
FlexLoRA [2024]	$\{s_i\mathbf{B}_i\mathbf{A}_i\}$	✕	✕	✕
FLoRA [2024]	$\{s_i\mathbf{B}_i\mathbf{A}_i\}$	✕	every round	✕
FLASC [2024]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately	✕	✕	✕
FFA-LoRA [2024]	only $\{\mathbf{B}_i\}$	only $\{\mathbf{A}_i\}$	✕	✕
RoLoRA [2025]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ alternately	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ alternately	✕	✕
Replicate [2025]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately	✕	✕	✕
FedSA [2025]	only $\{\mathbf{A}_i\}$	✕	✕	✕
LoRA-A^2 [2025]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ alternately	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ alternately	✕	✕
EcoLoRA [2025]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately	✕	✕	✕
FSLoRA [2025]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately	✕	✕	✕
LA-LoRA [2026]	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately	$\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ alternately	✕	✕
SPRA-LoRA	$\{s_i\mathbf{B}_i\mathbf{A}_i\}$	only $\{\mathbf{A}_i\}$	every round	non-uniform

2 Related Works

LoRA is a prominent PEFT method for federated fine-tuning [13]. Recent work highlights the asymmetry between its component matrices $\mathbf{A}$ and $\mathbf{B}$: Zhu et al. [14] evaluates freezing $\mathbf{A}$ versus $\mathbf{B}$, while Hayou et al. [15] improves performance using matrix-specific learning rates. Recent work by Lee et al. [16] demonstrates that optimizing the learning rate significantly boosts the performance of LoRA. Kalajdzievski [17] studies the impact of the scaling factor s on the learning process of LoRA.

Existing Fed-LoRA research follows two categories, and Table 1 summarizes their key features:

2.1 Federated LoRA with homogeneous client ranks

The first category of research assumes homogeneous resources where all n clients use a uniform rank r, with matrices $\{\mathbf{A}_i\} \in \mathbb{R}^{r\times m}$ and $\{\mathbf{B}_i\} \in \mathbb{R}^{n\times r}$. Early methods like SLoRA [18], FedIT [8], and FeDeRA [20] independently apply FedAvg to these components. However, this causes client-side gradient coupling noise from simultaneous update of the components [12] and server-side aggregation noise [9, 10]. To reduce coupling, LA-LoRA [12] and RoLoRA [11] use alternating optimization, though separate component aggregation still incurs noise. LoRA-A^2 [25] extends this with mask-based adaptive rank selection. Conversely, FFA [22] and LoRA-FA [28] freeze $\{\mathbf{A}_i\}$ to aggregate only $\{\mathbf{B}_i\}$ for improved stability at the cost of model utility. While FedSA-LoRA [24] aggregates only $\{\mathbf{A}_i\}$—claiming it learns general global knowledge—we show $\mathbf{A}_i$ acts primarily as a random projection for gradients rather than a knowledge-capturing parameter. Finally, EcoLoRA [26] and FLASC [21] focus on communication bandwidth efficiency via segment sharing and sparsity.

The aforementioned algorithms aggregate $\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ separately, making them limited to homogeneous ranks and vulnerable to the aggregation noise [9, 10].

2.2 Federated LoRA with heterogeneous client ranks

Varied client resources often require heterogeneous ranks r_i, causing dimension mismatches in $\{\mathbf{A}_i\} \in \mathbb{R}^{r_i\times m}$ and $\{\mathbf{B}_i\} \in \mathbb{R}^{n\times r_i}$ that prevent direct component aggregation. To resolve this, HetLoRA [19] uses zero-padding with sparsity-weighted aggregation, while [23] employ replication-based padding; both require matrix truncation for redistribution. FSLoRA [27] uses a fixed global rank and separate aggregation of components on the server side, and each client uses sketching to update submatrices of the global LoRA modules. While addressing rank heterogeneity, the fixed global rank limits FSLoRA's rank heterogeneity. Aggregating components separately and fine-tuning them simultaneously, all the three methods suffer from the aggregation and coupling noises. FlexLoRA [9] and FLoRA [10] avoid the aggregation noise by aggregating reconstructed updates $\{\Delta\mathbf{W}_i = s_i\mathbf{B}_i\mathbf{A}_i\} \in \mathbb{R}^{n\times m}$. FlexLoRA then redistributes components through computationally expensive truncated SVD of the aggregated update. FLoRA implements the update aggregation by stacking clients' LoRA components and avoids the SVD by merging the aggregated update into clients'

local adapters and reinitializing both the LoRA matrices for the next round. Yet, both FLoRA and FlexLoRA still suffer from the coupling noise, due to simultaneous fine-tuning of LoRA components.

In contrast, by treating $\{\mathbf{A}_i\}$ as periodically resampled, frozen random projection matrices and fine-tuning only $\{\mathbf{B}_i\}$, SPRA-LoRA totally eliminates the gradient coupling noise as well as the aggregation noise, while halving clients' overhead. This random-$\mathbf{A}$ framework paves the way to precisely quantify a rank-dependent gradient perturbation existing in LoRA dynamics, leading to a theoretically grounded strategy where clients adapt their learning rates based on their specific ranks. Unlike heuristic alternatives, our approach accelerates convergence and improves model utility.

3 Notations

Bold uppercase and lowercase letters denote matrices and vectors. For $\mathbf{A} \in \mathbb{R}^{r\times m}$, $\|\mathbf{A}\|_F$ is the Frobenius norm. $\mathbb{I}_k$ is the $k \times k$ identity, and $\mathbf{0}_{k_1\times k_2}$ is the zero matrix. $\mathcal{N}(\mathbf{0}, \mathbb{I}_{k_1\times k_2})$ denotes a $k_1 \times k_2$ matrix with i.i.d. standard normal entries.

In a FL system of n clients (dataset $\mathcal{D}_i$, size N_i), $\mathcal{S}^e$ is the set of n^e clients participating in round $e \in \{1, \dots, E\}$. Local objective function of client i is $f_i(\mathbf{W}) = \frac{1}{N_i}\sum_{x\in\mathcal{D}_i} l(x; \mathbf{W})$, with minimum value f_i^*. Starting from $\mathbf{W}_i^{e,0} = \mathbf{W}^{e-1}$, client $i \in \mathcal{S}^e$ performs T local optimization steps (indexed by t). We denote adapter batch gradients with $\nabla_{\mathbf{W}_i^{e,t}}\mathcal{L}_i^t(\mathbf{W}_i^{e,t}) = \frac{1}{b_i}\sum_{x\in\mathcal{B}_i^{e,t}} \nabla_{\mathbf{W}_i^{e,t}} l(x; \mathbf{W}_i^{e,t})$, where l is the used loss function and b_i is the batch size. Server aggregates the resulting adapter updates and updates the global adapter as $\mathbf{W}^e = \mathbf{W}^{e-1} + \sum_{i\in\mathcal{S}^e} \Delta\mathbf{W}_i^e/n^e$.

4 LoRA randomly perturbs adapter batch gradients

To motivate the proposed SPRA-LoRA algorithm, we analyze the structural mechanism of LoRA. This section is summarized in Figure 1. Consider a pretrained model with layers $\ell \in \{1, \dots, L\}$ and weights $\mathbf{W}_\ell^0 \in \mathbb{R}^{n\times m}$. For a low rank r, LoRA introduces component matrices $\mathbf{B}_\ell \in \mathbb{R}^{n\times r}$ and $\mathbf{A}_\ell \in \mathbb{R}^{r\times m}$ in the ℓ-th adaptation layer of the model, defining the forward pass as:

$$y = (\mathbf{W}_\ell^0 + s\mathbf{B}_\ell^t\mathbf{A}_\ell^t)x = \mathbf{W}_\ell^0 x + s\mathbf{B}_\ell^t\mathbf{A}_\ell^t x. \tag{2}$$

Standard LoRA implementations, e.g., Hugging Face [29], initialize $\mathbf{B}_\ell^0$ with an all-zero matrix and sample elements of $\mathbf{A}_\ell^0$ from $\mathcal{N}(0, \sigma_{\mathbf{A}}^2)$, with $\sigma_{\mathbf{A}}^2 = 1/r^2$. Also, $s = \alpha/r^\nu$, with $\nu = 1$ for regular LoRA and $\nu = 0.5$ for rank-stabilized LoRA [17].

Prior work suggests that while $\mathbf{B}_\ell$ is updated significantly, the randomly initialized $\mathbf{A}_\ell^0$ remains nearly frozen during fine-tuning [30]. On the other hand, at every optimization step, the gradient of the batch loss $\mathcal{L}^t$ with respect to the current $\mathbf{B}_\ell^t$ becomes a projection of the full-weight gradient:

$$\nabla_{\mathbf{B}_\ell^t}\mathcal{L}^t = \frac{\partial\mathcal{L}^t}{\partial s\mathbf{B}_\ell^t\mathbf{A}_\ell^t}\cdot\frac{\partial s\mathbf{B}_\ell^t\mathbf{A}_\ell^t}{\partial\mathbf{B}_\ell^t} = \frac{\partial\mathcal{L}^t}{\partial y}\cdot\frac{\partial y}{\partial s\mathbf{B}_\ell^t\mathbf{A}_\ell^t}\cdot s\mathbf{A}_\ell^{t\top} = \frac{\partial\mathcal{L}^t}{\partial y}\cdot x^\top\cdot s\mathbf{A}_\ell^{t\top} = (\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t)(s\mathbf{A}_\ell^t)^\top. \tag{3}$$

In other words, every batch gradient computation w.r.t. the component matrix $\mathbf{B}_\ell^t$ at time step t is equal to projection of batch gradient $\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t$, w.r.t. the equivalent forward-pass adapter $\mathbf{W}_\ell^t = \mathbf{W}_\ell^0 + s\mathbf{B}_\ell^t\mathbf{A}_\ell^t$, with $(s\mathbf{A}_\ell^t)^\top$. Similarly, $\nabla_{\mathbf{A}_\ell^t}\mathcal{L}^t = (s\mathbf{B}_\ell^t)^\top(\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t)$. Therefore, assuming freezing $\mathbf{A}_\ell^0$, after T local SGD steps with learning rate η for fine-tuning $\mathbf{B}_\ell^0$, the resulting adapter weight is:

$$\mathbf{W}_\ell^T = \mathbf{W}_\ell^0 + s\mathbf{B}_\ell^T\mathbf{A}_\ell^0 = \mathbf{W}_\ell^0 - s\eta\sum_{t=0}^{T-1}\nabla_{\mathbf{B}_\ell^t}\mathcal{L}^t\mathbf{A}_\ell^0 = \mathbf{W}_\ell^0 - \eta\sum_{t=0}^{T-1}\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t(s\mathbf{A}_\ell^0)^\top(s\mathbf{A}_\ell^0). \tag{4}$$

This formulation reveals that LoRA with frozen $\{\mathbf{A}_\ell^0\}_{\ell=1}^L$ is equivalent to full fine-tuning $\{\mathbf{W}_\ell^0\}_{\ell=1}^L$ with *randomly perturbed* batch gradients, where $s^2\mathbf{A}_\ell^{0^\top}\mathbf{A}_\ell^0$ acts as a structural perturbation. While derived under a frozen-$\mathbf{A}$ assumption, we show in Lemma B.1 (Appendix) that this interpretation holds within $\mathcal{O}(\eta^3)$ for the original LoRA. Along with the lemma, this shows that $\mathbf{A}_\ell$ *matrices function more as random low-rank perturbation matrices than essential knowledge-capturing parameters.*

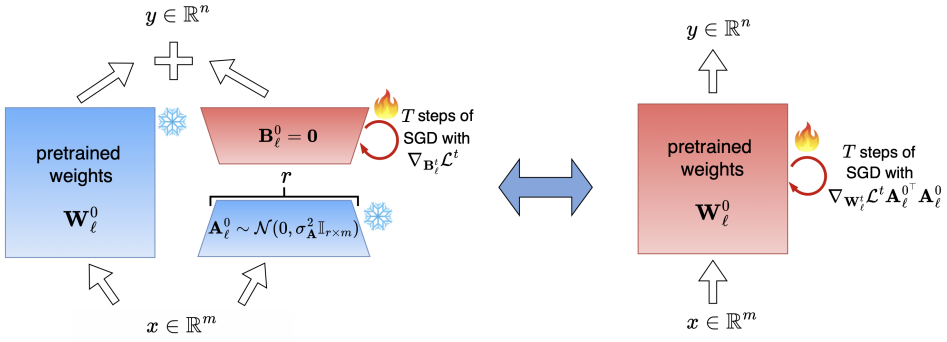

Figure 1: LoRA with frozen component $\mathbf{A}_\ell$. **Left:** Standard LoRA with a frozen $\mathbf{A}_\ell$ and fine-tuned $\mathbf{B}_\ell$. **Right:** Equivalent full-weight fine-tuning of $\mathbf{W}_\ell^0$ using structured noisy gradients $\nabla_{\mathbf{W}_\ell^t} \mathcal{L}^t \mathbf{A}_\ell^{0\top} \mathbf{A}_\ell^0$. Scaling factor s is not shown in the figure for simplicity.

4.1 Analysis of the perturbed batch gradients in eq. (4)

The columns of $\mathbf{A}_\ell^{0\top}$ are m-dimensional Gaussian variables sampled from $\mathcal{N}(\mathbf{0}, \sigma_{\mathbf{A}}^2 \mathbb{I}_{m\times m})$, and the product $\mathbf{A}_\ell^{0\top} \mathbf{A}_\ell^0$ follows a Wishart distribution with r degrees of freedom. For any statistically independent vector $\mathbf{q} \in \mathbb{R}^{1\times m}$, the term $\mathbf{q}(s^2 \mathbf{A}_\ell^{0\top} \mathbf{A}_\ell^0)$ represents a weighted sum of chi-squared variables. Based on the Lindeberg's Central Limit Theorem (Theorem C.4), this sum approximately follows a Gaussian distribution, a property we leverage to characterize the gradient perturbation.

Lemma 4.1. *Consider a fixed* $\mathbf{q} \in \mathbb{R}^{1\times m}$ *and* $\mathbf{A}_\ell^0 \in \mathbb{R}^{r\times m} \sim \mathcal{N}(0, \sigma_{\mathbf{A}}^2 \mathbb{I}_{r\times m})$*, with* $\sigma_{\mathbf{A}}^2 = 1/r^\mu$ $(2\nu + \mu > 0.5)$*. Random perturbation* $\mathbf{q}(s^2 \mathbf{A}_\ell^{0\top} \mathbf{A}_\ell^0)$ *approaches the* m*-dimensional Gaussian* $\mathcal{N}\big(\alpha^2 \mathbf{q}/r^{2\nu+\mu-1}, \alpha^4(\|\mathbf{q}\|^2 \mathbb{I}_m + \mathbf{D})/r^{4\nu+2\mu-1}\big)$ *as* r *increases, where* $\mathbf{D} = diag(\mathbf{q}^2)$.

The above lemma shows that, for a fixed $\mathbf{q}$, the term $\mathbf{q}(s^2 \mathbf{A}_\ell^{0\top} \mathbf{A}_\ell^0)$ is a *biased* perturbation of $\mathbf{q}$ with a distribution centered at mean $\alpha^2 \mathbf{q}/r^{2\nu+\mu-1}$. Also, it has the following signal-to-noise ratio (SNR):

$$\forall\, \alpha, \nu, \mu : \mathtt{SNR}\big(\mathbf{q}(s^2 \mathbf{A}_\ell^{0\top} \mathbf{A}_\ell^0)\big) = \frac{\|\frac{\alpha^2 \mathbf{q}}{r^{2\nu+\mu-1}}\|^2}{Tr\big(\frac{\alpha^4}{r^{4\nu+2\mu-1}}(\|\mathbf{q}\|^2 \mathbb{I}_m + \mathbf{D})\big)} = \frac{\|\frac{\alpha^2 \mathbf{q}}{r^{2\nu+\mu-1}}\|^2}{\frac{\alpha^4 (m+1)\|\mathbf{q}\|^2}{r^{4\nu+2\mu-1}}} = \frac{r}{m+1}. \quad (5)$$

As ν decreases, the scaling effect decreases, while the additive noise magnitude increases. For example, at $\nu = -0.5$, the perturbation is *unbiased* with asymptotic distribution $\mathcal{N}\big(\mathbf{q}, (\|\mathbf{q}\|^2 \mathbb{I}_m + \mathbf{D})/r\big)$ (see Figure 7 in the Appendix). Previous work [17] showed that rank-stabilized LoRA with $\nu = 0.5$, i.e., $s = \alpha/r^{0.5}$, enhances performance in high-rank scenarios compared to regular LoRA with $\nu = 1$. Accordingly, we adopt rank-stabilized LoRA with $\alpha = 1$. Also, let us use $\mu = 2$, following the Gaussian initialization of Hugging Face [29], resulting in distribution $\mathcal{N}\big(\mathbf{q}/r^2, (\|\mathbf{q}\|^2 \mathbb{I}_m + \mathbf{D})/r^5\big)$. We have visualized this in Figure 2 for an example $\mathbf{q} = (10, 10)$. In Appendix E, we show that the above rank-dependent perturbation of gradients similarly exists for Kaiming-Uniform initialization [31] of $\{\mathbf{A}_\ell^0\}_{\ell=1}^L$ (PEFT default), with a rank-linear SNR.

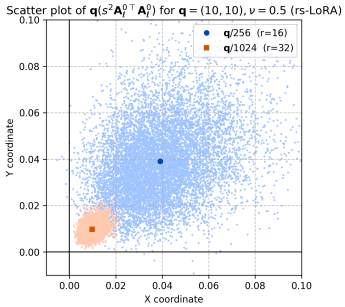

Figure 2: Scatter plot of $\mathbf{q}(s^2 \mathbf{A}_\ell^{0\top} \mathbf{A}_\ell^0)$, for example $\mathbf{q} = (10, 10)$, 10^4 samples $\mathbf{A}_\ell^0 \sim \mathcal{N}(0, \mathbb{I}_{r\times 2}/r^2)$ and $\nu = 0.5$.

While the previous lemma assumes independence between $\mathbf{q}$ and $\mathbf{A}_\ell^0$, in eq. (4), the gradient $\nabla_{\mathbf{W}_\ell^t} \mathcal{L}^t$ $(t \geq 1)$ depends on $\mathbf{A}_\ell^0$ (weakly). Given the *few local steps per round*, we use the lemma to approximate the distribution of the perturbed gradients in eq. (4)[1]. Consequently, the distribution of the perturbed batch gradient $\mathbf{Q} = \nabla_{\mathbf{W}_\ell^t} \mathcal{L}^t(s^2 \mathbf{A}_\ell^{0\top} \mathbf{A}_\ell^0)$

[1]For completeness, we show in Appendix F that the true distribution of the perturbed gradients introduces bounded mean and covariance shifts to the distribution returned by Lemma 4.1, where the bounds decrease with fewer local steps per round (T) and larger rank (r). We also use this bound in our convergence proof.

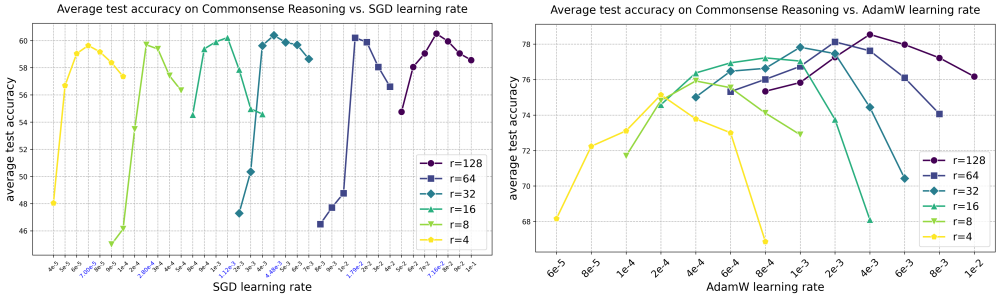

Figure 3: Average test accuracy for `LLaMA-3.2-3B` on Commonsense Reasoning obtained with `SGD` optimizer (left) and `AdamW` optimizer (right) with frozen and resampled $\mathbf{A}$.

approaches a Gaussian where row i converges to $\mathcal{N}\big(\mathbf{g}_{i,:}/r^2,\ (\|\mathbf{g}_{i,:}\|^2\mathbb{I}_m+\mathbf{D})/r^5\big)$. Thus, all gradient elements share a $1/r^2$ scaling factor but have distinct noise levels. Varying rank r causes inconsistent gradient scales and perturbation levels. As Figure 3 (left) confirms, optimal `SGD` learning rates follow the square law $\eta_r^* = \eta_c r^2$ and increase with adaptation rank r.

4.2 Extension of Equation (4) to `Adam` optimizer

The gradient perturbation persists in optimizers like `Adam` [32] and `AdamW` [33], as it affects the entire optimization trajectory. We now extend our previous analysis to `Adam` (and `AdamW`) optimizer. Under the frozen-$\mathbf{A}$ assumption ($\mathbf{B}_\ell^0 = 0$), `Adam` w.r.t $\mathbf{B}_\ell^t$ computes gradient moments as:

$$\begin{aligned}
&\textit{first moments:}\ \ m_{-1}=0 \to m_t=\beta_1 m_{t-1}+(1-\beta_1)\nabla_{\mathbf{B}_\ell^t}\mathcal{L}^t \to \hat{m}_t = m_t/(1-\beta_1)\\
&\textit{second moments:}\ \ v_{-1}=0 \to v_t=\beta_2 v_{t-1}+(1-\beta_2)(\nabla_{\mathbf{B}_\ell^t}\mathcal{L}^t \odot \nabla_{\mathbf{B}_\ell^t}\mathcal{L}^t) \to \hat{v}_t = v_t/(1-\beta_2)\\
&\textit{update:}\ \ \mathbf{B}_\ell^{t+1} = \mathbf{B}_\ell^t - \frac{\eta}{\sqrt{\hat{v}}+\epsilon}\odot \hat{m}_t, \qquad (6)
\end{aligned}$$

where $\odot$ denotes the Hadamard product. As $\nabla_{\mathbf{B}_\ell^t}\mathcal{L}^t = (\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t)(s\mathbf{A}_\ell^0)^\top$, we have $\hat{m}_t = \bar{m}_t(s\mathbf{A}_\ell^0)^\top$ as well ($\bar{m}_t$ being the first moment of $\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t$). Therefore, the final parameter update becomes:

$$\mathbf{W}_\ell^T = \mathbf{W}_\ell^0 + s\mathbf{B}_\ell^T\mathbf{A}_\ell^0 = \mathbf{W}_\ell^0 - \sum_{t=0}^{T-1}\left(\frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\odot \bar{m}_t(s\mathbf{A}_\ell^0)^\top\right)(s\mathbf{A}_\ell^0).$$

This shows that `Adam` w.r.t $\mathbf{B}_\ell^t$ is equivalent to `Adam` w.r.t $\mathbf{W}_\ell^t$, with the first and second moments computed using perturbed/projected adapter gradients. Unlike the square rule $\eta_r^* = \eta_c r^2$ for `SGD`, `Adam`'s element-wise normalization via $\sqrt{\hat{v}_t}$ complicates the rank dependence (otherwise there would be only the first moments being computed using the perturbed batch gradients $\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t(s\mathbf{A}_\ell^0)^\top(s\mathbf{A}_\ell^0)$), justifying a grid search. Yet, the optimum learning rate still increases with rank r (Figure 3, right).

Our derivations above were for frozen-$\mathbf{A}$. It is noteworthy that when both $\mathbf{A}$ and $\mathbf{B}$ are trainable, the optimal learning rate behaves differently and *decreases slowly* with rank r [34] (see Figure 4).

We propose freezing and resampling the $\mathbf{A}$ matrices with rank-aware client learning rates.

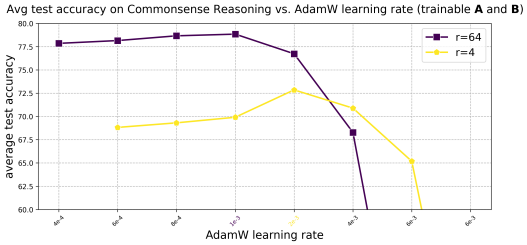

Figure 4: Average test accuracy for `LLaMA-3.2-3B` on Commonsense Reasoning obtained with `AdamW` optimizer and rank-stabilized `LoRA` ($\nu = 0.5$) and both $\mathbf{A}$ and $\mathbf{B}$ being trainable. Unlike when $\mathbf{A}$ is frozen (Figure 3, right), optimum learning rate *decreases slowly* (at rate $r^{-1/4}$) with r. When $\nu = 1$ (not shown), optimum learning rate does not vary (see [34] for a detailed analysis).

5 SPRA-LoRA algorithm

Our previous findings suggest that rather than applying a uniform learning rate—a standard practice in all previous Fed-LoRA methods—clients should employ learning rates tailored to their specific ranks. Applying a single, global learning rate forces some clients to operate outside their ideal operation zone, leading to diminished performance compared to their rank-specific optimum.

5.1 SPRA-LoRA: Per-round sampling and freezing of $\mathbf{A}_i^{e,0}$ with rank-aware learning rates

For simplicity, we drop the layer subscript ℓ in the next sections. Assuming that client i with rank r_i uses a rank-aware learning rate η_{r_i}, we describe the proposed SPRA-LoRA in the following. In round e, a participating client $i \in \mathcal{S}^e$ initializes with the last global model as $\mathbf{W}_i^{e,0} = \mathbf{W}^{e-1}$, resamples $\mathbf{A}_i^{e,0} \sim \mathcal{N}(\mathbf{0}, \frac{1}{r_i^2}\mathbb{I}_{r_i \times m})$ and resets $\mathbf{B}_i^{e,0} = \mathbf{0}_{n \times r_i}$ (participating clients can be synchronized at this stage by using the same random seed). Freezing $\mathbf{A}_i^{e,0}$ during the round, it uploads only the updated $\mathbf{B}_i^{e,T}$ after T local steps using `SGD` (or any other optimizer, e.g., `AdamW`):

$$\mathbf{W}_i^{e,T} = \mathbf{W}_i^{e,0} + s_i \mathbf{B}_i^{e,T} \mathbf{A}_i^{e,0} = \mathbf{W}_i^{e,0} - \eta_{r_i} \sum_{t=0}^{T-1} \nabla_{\mathbf{W}_i^{e,t}} \mathcal{L}_i^{e,t} (s_i \mathbf{A}_i^{e,0})^\top (s_i \mathbf{A}_i^{e,0}) \tag{7}$$

So far, by freezing $\mathbf{A}_i^{e,0}$, the client's computation and uplink communication overheads (measured by the number of trainable parameters) have been cut by 50%. Also, the gradient coupling noise has been completely removed. In order to eliminate the aggregation noise, the server aggregates $\{\Delta \mathbf{W}_i^e\} = \{s_i \mathbf{B}_i^{e,T} \mathbf{A}_i^{e,0}\}$ into $\Delta \mathbf{W}^e = \sum_{i \in \mathcal{S}^e} \Delta \mathbf{W}_i^e / n^e$ and updates the global adapter parameter as $\mathbf{W}^e = \mathbf{W}^{e-1} + \Delta \mathbf{W}^e$, which will be downloaded by clients for the next round.

Freezing matrices $\mathbf{A}_i^{e,0}$ resembles the FFA-LoRA algorithm[2], but SPRA-LoRA extends it by resampling the matrices $\mathbf{A}_i^{e,0}$ at every round e during fine-tuning, which diminishes the LoRA low-rank constraints. This approach also improves FLoRA [10] and FlexLoRA [9], which update both $\mathbf{A}_i^{e,0}$ and $\mathbf{B}_i^{e,0}$ simultaneously, suffering from the gradient coupling between them and the coupling noise [12].

Assuming that there are R different values of adaptation ranks in the system and that C_j^e denotes the set of clients $i \in \mathcal{S}^e$ using the j-th rank value, the aggregation on the server can be re-written as:

$$\Delta \mathbf{W}^e = \sum_{i \in \mathcal{S}^e} \Delta \mathbf{W}_i^e / n^e = \sum_{j=1}^{R} \frac{|C_j^e|}{n^e} \Big(\sum_{i \in C_j^e} \Delta \mathbf{W}_i^e / |C_j^e| \Big). \tag{8}$$

Specifically, participating clients $i \in \mathcal{S}^e$ are partitioned into R distinct clusters according to their respective adaptation ranks. Rather than applying a uniform learning rate across the entire clusters, our rank-aware SPRA-LoRA framework ensures that clients within each cluster j utilize a specific learning rate optimized to maximize fine-tuned model utility for that specific rank. The server subsequently aggregates these R cluster-specific models via the averaging process defined above.

5.2 Rank-aware learning rate selection for SPRA-LoRA

To determine the optimal learning rate η_{r_i} for individual ranks, we propose the following simple strategy: Each cluster of clients with a common rank performs federated fine-tuning and an *intra-cluster* evaluation over a grid of *cluster-specific* learning rates. Note that this method has ***no additional*** communication overhead compared to existing baselines. Additionally, when using `SGD` optimizer, the square law $\eta_r^* = \eta_c r^2$ could be used to further diminish the overhead of learning rate tuning.

[2]While FFA [22] was originally proposed for settings with homogeneous client ranks, it can be extended to heterogeneous settings by aggregating $\{\Delta \mathbf{W}_i^e = s_i \mathbf{B}_i^{e,T} \mathbf{A}_i^0\}$, instead of aggregating $\{\mathbf{B}_i^{e,T}\}$.

5.3 Convergence analysis of SPRA-LoRA

In this section, we prove the convergence of SPRA-LoRA in non-convex settings under mild assumptions. Let us define global objective function $f(\mathbf{W}) := \sum_{i=1}^{n} f_i(\mathbf{W})/n$, with minimum value $f^\star$ and minimizer $\mathbf{W}^*$. We also make some mild assumptions about the loss functions f_i, including their Lipschitz continuity and smoothness (see Assumptions G.1 in the Appendix).

Theorem 5.1 (**Convergence of** SPRA-LoRA **with** `SGD`). *Assume using rank-stabilized* LoRA *($\nu = 0.5$), assumptions G.1, and full participation with client i using learning rate $\eta_{r_i} = \eta_c r_i^2$. If the common learning rate satisfies $\eta_c \leq \dfrac{1}{4\beta T}$, then for any number of communication rounds $E \geq 1$,*

$$\boxed{\min_e \mathbb{E}\big[\, \|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2 \,\big] \;\leq\; \frac{2\big(f(\mathbf{W}^0) - f^\star\big)}{\eta_c T E} \;+\; \Psi}$$

where $\{\mathbf{W}^e\}_{e=0}^{E}$ are global iterates, and $\Psi = \mathcal{O}(\eta_c)$ is a constant. Choosing $\eta_c = \min\left(\frac{1}{4\beta T}, \frac{1}{\sqrt{E}}\right)$

$$\min_e \mathbb{E}\big[\, \|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2 \,\big] = \mathcal{O}\left(\frac{1}{\sqrt{E}}\right),$$

so the squared gradient norm of the global iterates converges, as $E \to \infty$, to a stationary point of f, recovering the classical $O(1/\sqrt{E})$ non-convex FedAvg rate [35, 36].

Discussion. The term Ψ decreases with smaller T, more i.i.d. client data and larger batch sizes. Unlike prior work, e.g. [10] (FLoRA), which assumes strong convexity of $\{f_i\}$, our results hold under more realistic assumptions. Under i.i.d. data, our result reduces to `SGD` [37] on $\mathbf{W}^e$, with rate $\frac{1}{\sqrt{E}}$.

6 Experimental results

We evaluate our approach on language generation and understanding tasks using `LLaMA-3.2-3B` [38] (on Commonsense Reasoning dataset [39]) and `RoBERTa-base` [40] (on GLUE benchmark [41]). Considering our *heterogeneous-rank* setting, only the baselines highlighted in blue in Table 1 are the main baselines comparable to SPRA-LoRA. Yet, we extend the aggregation strategies of the recent algorithms FedSA [24], RoLoRA [11] and LA-LoRA [12] with zero padding (similar to HetLoRA [19]) and use them as baselines as well. Also, we generalize the aggregation strategy of FFA to aggregate $\{\Delta\mathbf{W}_i^e = s_i \mathbf{B}_i^{e,T} \mathbf{A}_i^{e,0}\}$ at the end of each round e (instead of aggregating $\{\mathbf{B}_i^{e,T}\}$) to extend it to settings with heterogeneous client ranks. For Commonsense Reasoning (GLUE), data is partitioned across $n = 20$ ($n = 40$) clients with a Dirichlet distribution with $\alpha = 2$ ($\alpha = 0.8$) by default [10, 9, 27]. Client ranks are sampled uniformly and non-uniformly from $\{4, 8, \ldots, 128\}$, with identical configurations applied to all baselines to ensure a fair comparison. We use a cluster with 4 `NVIDIA H100 80 GB` GPUs for running our experiments. Each experiment is run on a single GPU. Further details about our experimental setup is explained in Appendix A.1.

The Commonsense benchmark results, detailed in Tables 2 and 3, compare uniform and log-normal client rank distributions across three SPRA-LoRA variants: a uniform learning rate version designed to isolate the effects of per-round resampling and weight freezing against the FFA baseline, the standard rank-aware learning rate configuration, and a "double-rank" alternative.

Despite 50% lower overhead, SPRA-LoRA with a uniform learning rate matches or exceeds most baselines. Its significant performance advantage over FFA stems from its per-round resampling strategy. Both the methods share the same number of trainable parameters, and the marginal increase in run time for SPRA-LoRA reflects the time for per-round parameter merging, saving and reloading in our experiments, rather than increased computational complexity.

HetLoRA suffers from both aggregation noise (due to aggregating $\mathbf{A}$ and $\mathbf{B}$ matrices separately) and coupling noise (due to simultaneous optimization of $\mathbf{A}$ and $\mathbf{B}$ matrices). While FLoRA shares SPRA-LoRA's per-round resampling and resetting strategy, it still suffers from the coupling noise, due to fine-tuning both the component matrices simultaneously with a uniform learning rate. In contrast, SPRA-LoRA demonstrates superior performance by resampling and freezing $\{\mathbf{A}_i^{e,0}\}$ and utilizing rank-aware learning rates for clients. Remarkably, SPRA-LoRA achieves these gains while reducing computational and communication overhead by 50% compared to FLoRA and other baselines.

Table 2: Average test accuracy (5 runs) with sample `std` error bars on Commonsense Reasoning, with Dirichlet distribution ($\alpha = 2$), `LLaMA-3.2-3B` model and `AdamW` optimizer. Clients' ranks follow ***uniform*** distribution over $\{4, 8, \ldots, 128\}$. SPRA-LoRA and its doubled-rank variant consistently outperform baselines while maintaining 50% less or equal client overhead, respectively.

Method	Run time	BoolQ	WinoGrande	OBQA	ARC-e	ARC-c	SIQA	PIQA	HellaSwag	Avg
Base Pretrained Model	—	47.28	04.94	52.01	60.89	51.05	53.36	20.67	55.98	43.27
FFA [2024]	0.81h	$62.67_{\pm 0.67}$	$64.06_{\pm 0.31}$	$64.66_{\pm 1.81}$	$76.46_{\pm 0.52}$	$60.66_{\pm 2.15}$	$63.05_{\pm 2.23}$	$73.19_{\pm 2.18}$	$64.53_{\pm 1.62}$	$66.15_{\pm 0.61}$
HetLoRA [2024]	0.91h	$65.52_{\pm 0.27}$	$70.74_{\pm 0.84}$	$76.01_{\pm 0.35}$	$85.27_{\pm 0.22}$	$72.69_{\pm 0.41}$	$73.26_{\pm 0.63}$	$79.10_{\pm 0.46}$	$72.87_{\pm 0.40}$	$74.43_{\pm 0.32}$
FlexLoRA [2024]	5.11h	$65.87_{\pm 0.52}$	$71.04_{\pm 0.40}$	$78.01_{\pm 0.69}$	$86.56_{\pm 0.17}$	$73.04_{\pm 0.01}$	$72.62_{\pm 0.17}$	$80.81_{\pm 0.35}$	$71.51_{\pm 1.39}$	$74.93_{\pm 0.14}$
FLoRA [2024]	1.51h	$67.06_{\pm 0.88}$	$70.59_{\pm 1.67}$	$75.94_{\pm 0.42}$	$85.70_{\pm 0.58}$	$72.33_{\pm 1.05}$	$73.90_{\pm 0.49}$	$80.45_{\pm 0.63}$	$73.02_{\pm 0.82}$	$74.87_{\pm 0.40}$
FSLoRA [2025]	0.92h	$66.83_{\pm 0.33}$	$70.30_{\pm 0.50}$	$77.14_{\pm 0.42}$	$\underline{87.26}_{\pm 0.36}$	$72.87_{\pm 0.90}$	$73.90_{\pm 0.27}$	$81.06_{\pm 0.47}$	$71.52_{\pm 0.36}$	$75.11_{\pm 0.14}$
FedSA [2025]	1.08h	$59.45_{\pm 9.51}$	$50.18_{\pm 4.72}$	$62.44_{\pm 11.81}$	$79.53_{\pm 8.11}$	$63.48_{\pm 7.91}$	$60.57_{\pm 10.49}$	$67.64_{\pm 15.78}$	$48.12_{\pm 12.03}$	$61.42_{\pm 8.81}$
RoLoRA [2025]	0.91h	$66.21_{\pm 2.23}$	$73.48_{\pm 1.51}$	$76.94_{\pm 1.72}$	$87.12_{\pm 0.33}$	$73.94_{\pm 0.77}$	$75.18_{\pm 0.89}$	$80.68_{\pm 0.34}$	$75.01_{\pm 0.30}$	$76.07_{\pm 0.22}$
LA-LoRA [2026]	0.95h	$67.06_{\pm 2.07}$	$74.19_{\pm 0.67}$	$78.07_{\pm 0.83}$	$\mathbf{87.79}_{\pm \mathbf{0.19}}$	$\mathbf{74.81}_{\pm \mathbf{0.56}}$	$75.62_{\pm 0.49}$	$81.06_{\pm 0.91}$	$76.21_{\pm 0.07}$	$76.84_{\pm 0.23}$
SPRA-LoRA (uniform LR)	1.36h	$66.72_{\pm 0.99}$	$71.27_{\pm 1.06}$	$75.14_{\pm 1.53}$	$86.77_{\pm 0.27}$	$73.18_{\pm 0.18}$	$73.49_{\pm 0.67}$	$80.78_{\pm 0.36}$	$73.94_{\pm 0.26}$	$75.16_{\pm 0.30}$
SPRA-LoRA	1.38h	$\mathbf{68.54}_{\pm 2.17}$	$\underline{75.90}_{\pm 0.18}$	$\underline{78.34}_{\pm 1.40}$	$86.39_{\pm 0.54}$	$73.75_{\pm 0.81}$	$\underline{75.78}_{\pm 0.21}$	$\underline{81.85}_{\pm 0.93}$	$\underline{77.69}_{\pm 0.44}$	$\underline{77.28}_{\pm 0.24}$
SPRA-LoRA (doubled ranks)	1.52h	$\underline{67.59}_{\pm 0.63}$	$\mathbf{76.27}_{\pm 1.14}$	$\mathbf{79.01}_{\pm 0.60}$	$86.96_{\pm 0.21}$	$\underline{74.46}_{\pm 0.39}$	$\mathbf{76.12}_{\pm 0.72}$	$\mathbf{82.03}_{\pm 0.53}$	$\mathbf{77.80}_{\pm 0.21}$	$\mathbf{77.53}_{\pm 0.08}$

Table 3: Average test accuracy (5 runs) on Commonsense Reasoning dataset, with Dirichlet distribution ($\alpha = 2$), `LLaMA-3.2-3B` model and `AdamW` optimizer. Clients' ranks follow ***log-normal*** with heavy low tail distribution over $\{4, 8, \ldots, 128\}$. SPRA-LoRA and its doubled-rank variant consistently outperform baselines while maintaining 50% less or equal client overhead, respectively.

Method	Run time	BoolQ	WinoGrande	OBQA	ARC-e	ARC-c	SIQA	PIQA	HellaSwag	Avg
Base Pretrained Model	—	47.28	04.94	52.01	60.89	51.05	53.36	20.67	55.98	43.27
FFA [2024]	0.71h	$61.43_{\pm 0.40}$	$63.90_{\pm 0.28}$	$61.93_{\pm 2.04}$	$76.54_{\pm 2.04}$	$59.52_{\pm 1.35}$	$63.54_{\pm 0.51}$	$71.93_{\pm 3.10}$	$63.00_{\pm 0.93}$	$65.22_{\pm 0.88}$
HetLoRA [2024]	0.85h	$65.40_{\pm 1.19}$	$71.93_{\pm 1.15}$	$78.34_{\pm 0.64}$	$\mathbf{87.78}_{\pm 0.41}$	$\mathbf{74.52}_{\pm 0.52}$	$75.66_{\pm 0.91}$	$81.28_{\pm 0.38}$	$73.86_{\pm 2.18}$	$76.09_{\pm 0.25}$
FlexLoRA [2024]	5.01h	$66.37_{\pm 1.91}$	$72.09_{\pm 1.57}$	$77.01_{\pm 0.72}$	$87.07_{\pm 0.41}$	$72.87_{\pm 1.41}$	$73.88_{\pm 0.42}$	$\mathbf{81.52}_{\pm 0.71}$	$72.99_{\pm 0.69}$	$75.47_{\pm 0.51}$
FLoRA [2024]	1.42h	$67.32_{\pm 1.51}$	$72.56_{\pm 1.23}$	$77.14_{\pm 1.10}$	$86.34_{\pm 0.77}$	$72.10_{\pm 0.94}$	$74.90_{\pm 0.62}$	$80.20_{\pm 0.80}$	$73.26_{\pm 2.19}$	$75.47_{\pm 0.40}$
FSLoRA [2025]	0.93h	$67.83_{\pm 0.30}$	$70.33_{\pm 0.21}$	$77.54_{\pm 0.42}$	$87.25_{\pm 0.07}$	$73.14_{\pm 0.05}$	$73.30_{\pm 0.15}$	$80.78_{\pm 0.28}$	$71.10_{\pm 0.29}$	$74.87_{\pm 0.53}$
FedSA [2025]	1.11h	$60.79_{\pm 6.38}$	$53.56_{\pm 3.45}$	$63.04_{\pm 7.39}$	$78.76_{\pm 7.01}$	$63.79_{\pm 6.85}$	$60.13_{\pm 5.48}$	$63.96_{\pm 16.56}$	$46.23_{\pm 13.81}$	$61.28_{\pm 7.41}$
RoLoRA [2025]	0.92h	$67.77_{\pm 0.43}$	$71.21_{\pm 1.22}$	$77.81_{\pm 1.31}$	$\underline{87.54}_{\pm 0.14}$	$\underline{74.34}_{\pm 0.86}$	$74.58_{\pm 1.38}$	$80.26_{\pm 0.84}$	$73.18_{\pm 1.46}$	$75.83_{\pm 0.86}$
LA-LoRA [2026]	0.96h	$67.98_{\pm 0.86}$	$71.82_{\pm 0.13}$	$\underline{78.61}_{\pm 0.87}$	$87.50_{\pm 0.41}$	$74.08_{\pm 0.58}$	$75.71_{\pm 0.81}$	$81.04_{\pm 0.53}$	$73.97_{\pm 3.41}$	$76.34_{\pm 0.31}$
SPRA-LoRA (uniform LR)	1.26h	$68.10_{\pm 0.31}$	$72.22_{\pm 1.23}$	$76.01_{\pm 0.87}$	$86.24_{\pm 0.64}$	$71.96_{\pm 1.09}$	$74.91_{\pm 0.46}$	$80.36_{\pm 1.32}$	$73.66_{\pm 1.34}$	$75.24_{\pm 0.86}$
SPRA-LoRA	1.27h	$\underline{68.40}_{\pm 0.18}$	$\underline{75.88}_{\pm 0.92}$	$77.94_{\pm 0.92}$	$86.40_{\pm 0.15}$	$72.87_{\pm 0.38}$	$\mathbf{76.09}_{\pm 0.44}$	$80.56_{\pm 0.47}$	$\underline{76.82}_{\pm 0.05}$	$\underline{76.87}_{\pm 0.05}$
SPRA-LoRA (doubled ranks)	1.35h	$\mathbf{68.82}_{\pm 2.65}$	$\mathbf{76.26}_{\pm 0.29}$	$\mathbf{79.48}_{\pm 0.61}$	$86.49_{\pm 0.08}$	$73.15_{\pm 0.73}$	$\underline{75.99}_{\pm 0.54}$	$\underline{81.34}_{\pm 1.30}$	$\mathbf{76.94}_{\pm 0.51}$	$\mathbf{77.26}_{\pm 0.54}$

Table 4: Average test accuracy (5 runs) on GLUE language understanding benchmark with Dirichlet distribution ($\alpha = 0.8$), `RoBERTa` model and `AdamW` optimizer. Clients' ranks follow ***uniform*** distribution over $\{4, 8, \ldots, 128\}$. SPRA-LoRA and its doubled-rank variant outperform baselines while maintaining 50% less or equal client overhead, respectively.

Method	Run time	QNLI	MRPC	CoLA	MNLI	RTE	SST-2	QQP	Avg
Base Pretrained Model	—	49.46	31.62	30.87	35.45	52.71	49.08	63.18	44.62
FFA [2024]	0.90h	$83.23_{\pm 0.60}$	$80.15_{\pm 1.07}$	$77.66_{\pm 0.89}$	$73.22_{\pm 0.95}$	$66.64_{\pm 1.78}$	$91.01_{\pm 0.25}$	$79.62_{\pm 0.70}$	$78.79_{\pm 0.38}$
HetLoRA [2024]	0.62h	$81.33_{\pm 1.51}$	$85.29_{\pm 0.17}$	$79.94_{\pm 0.29}$	$77.25_{\pm 0.11}$	$64.40_{\pm 0.48}$	$92.61_{\pm 0.19}$	$81.25_{\pm 0.64}$	$80.30_{\pm 0.25}$
FlexLoRA [2024]	1.11h	$\mathbf{87.71}_{\pm 0.20}$	$84.61_{\pm 1.61}$	$\mathbf{81.36}_{\pm 0.41}$	$78.02_{\pm 0.76}$	$69.53_{\pm 1.38}$	$93.12_{\pm 0.19}$	$81.12_{\pm 2.18}$	$82.21_{\pm 0.45}$
FLoRA [2024]	0.92h	$84.47_{\pm 1.25}$	$85.05_{\pm 0.23}$	$79.23_{\pm 1.15}$	$76.20_{\pm 0.70}$	$67.87_{\pm 0.62}$	$92.36_{\pm 0.45}$	$81.60_{\pm 0.40}$	$80.97_{\pm 0.29}$
FSLoRA [2025]	0.84h	$76.02_{\pm 2.29}$	$83.43_{\pm 0.64}$	$79.31_{\pm 0.63}$	$60.40_{\pm 2.88}$	$65.06_{\pm 1.87}$	$92.00_{\pm 0.13}$	$79.13_{\pm 0.96}$	$76.48_{\pm 0.62}$
FedSA [2025]	1.28h	$69.65_{\pm 2.36}$	$68.53_{\pm 1.52}$	$60.86_{\pm 1.87}$	$46.46_{\pm 1.18}$	$51.02_{\pm 0.45}$	$87.28_{\pm 1.02}$	$65.29_{\pm 0.59}$	$64.16_{\pm 0.54}$
RoLoRA [2025]	0.60h	$74.65_{\pm 2.01}$	$82.99_{\pm 0.79}$	$76.91_{\pm 0.65}$	$52.92_{\pm 2.61}$	$62.53_{\pm 1.42}$	$90.78_{\pm 0.15}$	$77.97_{\pm 0.64}$	$74.11_{\pm 0.54}$
LA-LoRA [2026]	0.61h	$74.96_{\pm 2.54}$	$83.97_{\pm 0.46}$	$76.24_{\pm 0.84}$	$67.57_{\pm 1.78}$	$64.26_{\pm 0.16}$	$91.17_{\pm 0.06}$	$78.64_{\pm 0.60}$	$76.69_{\pm 0.47}$
SPRA-LoRA (uniform LR)	0.86h	$84.77_{\pm 0.55}$	$83.68_{\pm 0.46}$	$78.69_{\pm 1.19}$	$74.51_{\pm 0.97}$	$66.42_{\pm 0.56}$	$92.55_{\pm 0.21}$	$81.27_{\pm 0.46}$	$80.27_{\pm 0.26}$
SPRA-LoRA	0.86h	$\underline{87.52}_{\pm 1.20}$	$\mathbf{85.58}_{\pm 0.57}$	$\underline{80.88}_{\pm 0.50}$	$\underline{80.70}_{\pm 0.43}$	$\underline{70.54}_{\pm 1.27}$	$\mathbf{93.60}_{\pm 0.30}$	$\mathbf{84.26}_{\pm 0.32}$	$\underline{83.30}_{\pm 0.29}$
SPRA-LoRA (doubled ranks)	0.96h	$86.51_{\pm 2.03}$	$\underline{85.39}_{\pm 0.31}$	$80.63_{\pm 0.88}$	$\mathbf{81.80}_{\pm 0.21}$	$\mathbf{71.84}_{\pm 0.82}$	$\underline{93.14}_{\pm 0.30}$	$\underline{84.20}_{\pm 0.37}$	$\mathbf{83.36}_{\pm 0.35}$

To avoid coupling noise, LA-LoRA and RoLoRA alternately optimize $\mathbf{A}$ and $\mathbf{B}$ between local steps and rounds, respectively, matching the computational overhead of SPRA-LoRA. However, SPRA-LoRA outperforms them by treating $\mathbf{A}$ as a resampled random projection matrix and allocating the full computational budget to $\mathbf{B}$. This is motivated by the fact that with zero-initialized $\mathbf{B}$, matrix $\mathbf{A}$ remains nearly frozen during fine-tuning (Section 4, Lemma B.1 in Appendix, and [34]).

FedSA performs the weakest. By training both $\mathbf{A}$ and $\mathbf{B}$ locally while aggregating only $\mathbf{A}$ on the server, it leaves $\mathbf{B}$ local. Hence, clients never receive an aggregated $\mathbf{B}$ matrix. This necessitates strict full client participation [24]. Furthermore, because each client maintains a personalized model, FedSA's results are averaged across random seeds and clients. As shown, under partial participation, the overall performance for FedSA is low and there is a high variance across clients (large `stds`).

Finally, to ensure a fair comparison, the final row of results evaluates SPRA-LoRA with doubled client ranks to match the trainable parameters and overhead of the baselines. Under these equivalent resource constraints, SPRA-LoRA achieves its most significant performance gains, highlighting its superior efficiency and utility. Similar trends are observed for the GLUE benchmark in Table 4.

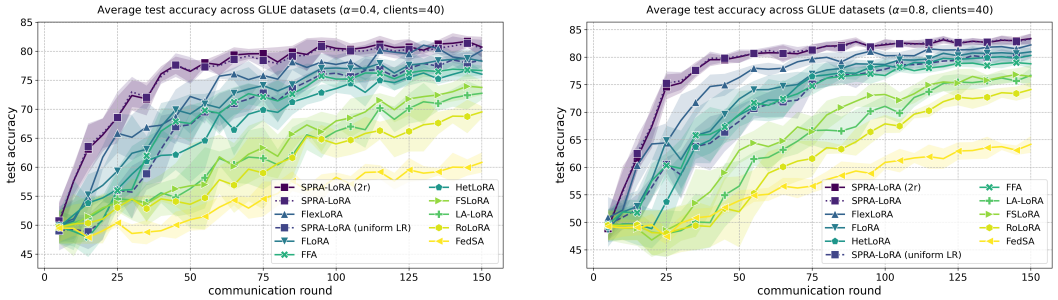

Figure 5: **Left:** Average test accuracy across 5 runs on GLUE with Dirichlet distribution ($\alpha = 0.4$), `RoBERTa` model and `AdamW` optimizer. **Right:** Same as left, with $\alpha = 0.8$.

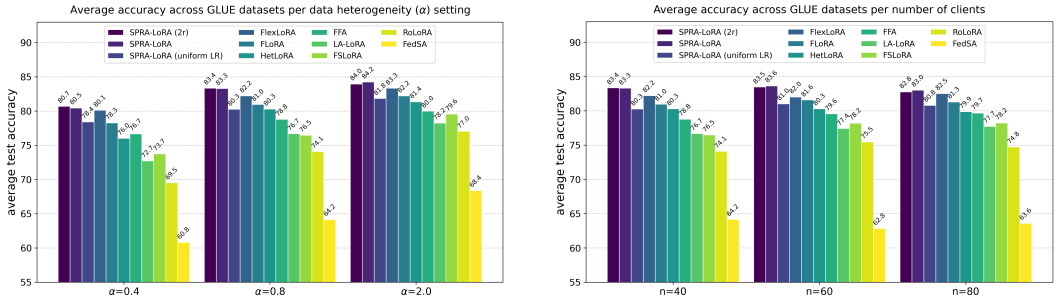

Figure 6: **Left:** Average test accuracy across 5 runs on GLUE language understanding benchmark with $n = 40$ and Dirichlet distribution with $\alpha \in \{0.4, 0.8, 2.0\}$, `RoBERTa` model and `AdamW` optimizer. **Right:** Average test accuracy across 5 runs on GLUE and Dirichlet distribution with $\alpha = 0.8$, number of clients $n \in \{40, 60, 80\}$, `RoBERTa` model and `AdamW` optimizer.

Table 5: Average test accuracy (10 runs) on Commonsense Reasoning dataset (`LLaMA-3.2-3B` model and `AdamW` optimizer) in a *centralized* setting with fixed adaptation rank $r = 64$. Total number of gradient steps is 1000, and the frozen matrix $\mathbf{A}$ is resampled every T steps.

Period	BoolQ	WinoGrande	OBQA	ARC-e	ARC-c	SIQA	PIQA	HellaSwag	Avg
$T = 20$	$66.11_{\pm 0.95}$	$67.45_{\pm 0.72}$	$73.12_{\pm 1.01}$	$85.24_{\pm 0.48}$	$70.26_{\pm 1.06}$	$71.52_{\pm 0.71}$	$79.54_{\pm 0.68}$	$70.34_{\pm 1.35}$	$72.94_{\pm 0.43}$
$T = 10$	$64.58_{\pm 1.06}$	$69.99_{\pm 1.19}$	$73.54_{\pm 1.28}$	$84.27_{\pm 0.55}$	$68.15_{\pm 1.14}$	$71.97_{\pm 0.72}$	$79.35_{\pm 1.06}$	$71.31_{\pm 1.14}$	$72.89_{\pm 0.23}$
$T = 5$	$65.65_{\pm 1.81}$	$68.08_{\pm 1.05}$	$72.81_{\pm 1.01}$	$85.61_{\pm 0.51}$	$69.99_{\pm 0.92}$	$72.06_{\pm 0.53}$	$79.24_{\pm 1.11}$	$69.84_{\pm 1.77}$	$72.91_{\pm 0.43}$
$T = 1$	$64.72_{\pm 2.23}$	$69.59_{\pm 2.41}$	$73.46_{\pm 0.41}$	$84.59_{\pm 0.64}$	$69.86_{\pm 1.31}$	$73.33_{\pm 0.68}$	$79.27_{\pm 0.47}$	$72.33_{\pm 1.62}$	$73.39_{\pm 0.31}$

SPRA-LoRA 's gains derive from per-round resampling of $\mathbf{A}$ matrices, focusing computational resources on optimizing $\mathbf{B}$ matrices and rank-aware heterogeneous learning rates. This approach avoids both aggregation and coupling noises. Convergence results in Figure 5 demonstrate that both variants of SPRA-LoRA consistently achieve higher average test accuracy across the GLUE benchmark throughout the communication rounds compared to the baselines.

Figure 6 shows an ablation study on data heterogeneity (α) and number of clients (n), with consistently superior performance of SPRA-LoRA variants. See Appendix A.2 for detailed results.

Finally, Table 5 examines T (local steps for SPRA-LoRA and matrix $\mathbf{A}$ resampling period). Using a centralized setting ($r = 64$, 1000 steps with `AdamW`) to isolate federated optimization effects, we find minimal difference between $T > 1$ and $T = 1$ for small T, showing negligible impact from the weak dependence between matrices $\mathbf{A}$ and batch gradients. Appendix F proves that per-T-steps resampling introduces a bounded mean shift to per-step sampling (Lemma 4.1), decaying with T.

7 Conclusion

We proposed SPRA-LoRA to address update scale imbalances across heterogeneous client ranks. By freezing and resampling matrices $\mathbf{A}$ and applying rank-scaled learning rates, SPRA-LoRA balances gradient perturbations for superior efficient performance. Future work will explore dynamic rank adaptation and continuous learning rate schedules.

Impact Statement

This goal of this paper is to advance the field of Machine Learning. Our paper is foundational research and not tied to any particular application with negative impacts. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgement and Disclosure of Funding

Funding support for project activities has been partially provided by NSERC Canada, Canada CIFAR AI Chair, Facebook award, and Google award.

References

[1] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, June 2019. URL https://aclanthology.org/N19-1423/.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

[5] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://aclanthology.org/2021.emnlp-main.243/.

[6] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021. URL https://aclanthology.org/2021.acl-long.353/.

[7] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[8] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, et al. Towards building the federat-edgpt: Federated instruction tuning. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024. URL https://ieeexplore.ieee.org/document/10447454.

[9] Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1a134b50202088aa8c595cc99b310e5a-Paper-Conference.pdf.

[10] 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. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/pdf?id=TcCorXxNJQ.

[11] Shuangyi Chen, Yuanxin Guo, Yue Ju, Hardik Dalal, Zhongwen Zhu, and Ashish J Khisti. Robust federated finetuning of LLMs via alternating optimization of loRA. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/pdf?id=e8DrPuJekZ.

[12] Jin Liu, Ning Xi, Yinbin Miao, and Junkang Liu. Rethinking loRA for privacy-preserving federated learning in large models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=BPzSV4uw0x.

[13] Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023. URL https://aclanthology.org/2023.findings-acl.632/.

[14] Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez De Ocáriz Borde, et al. Asymmetry in low-rank adapters of foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/zhu24c.html.

[15] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/hayou24a.html.

[16] Yu-Ang Lee, Ching-Yun Ko, Pin-Yu Chen, and Mi-Yen Yeh. Learning rate matters: Vanilla lora may suffice for llm fine-tuning, 2026. URL https://arxiv.org/abs/2602.04998.

[17] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora, 2023. URL https://arxiv.org/abs/2312.03732.

[18] Sara Babakniya, Ahmed Roushdy Elkordy, Yahya H. Ezzeldin, Qingfeng Liu, Kee-Bong Song, Mostafa El-Khamy, and Salman Avestimehr. Slora: Federated parameter efficient fine-tuning of language models, 2023. URL https://arxiv.org/abs/2308.06522.

[19] Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. Heterogeneous LoRA for federated fine-tuning of on-device foundation models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://aclanthology.org/2024.emnlp-main.717/.

[20] Yuxuan Yan, Qianqian Yang, Shunpu Tang, and Zhiguo Shi. Federa:efficient fine-tuning of language models in federated learning leveraging weight decomposition, 2024. URL https://arxiv.org/abs/2404.18848.

[21] Kevin Kuo, Arian Raje, Kousik Rajesh, and Virginia Smith. Federated lora with sparse communication, 2024. URL https://arxiv.org/abs/2406.05233.

[22] Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving loRA in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/pdf?id=NLPzL6HWNl.

[23] Yuji Byun and Jaeho Lee. Towards federated low-rank adaptation of language models with rank heterogeneity. 2025. URL https://aclanthology.org/2025.naacl-short.30/.

[24] Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=iX3uESGdsO.

[25] 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)*, 2025. URL https://aclanthology.org/2025.acl-long.19/.

[26] Han Liu, Ruoyao Wen, Srijith Nair, Jia Liu, Wenjing Lou, et al. EcoLoRA: Communication-efficient federated fine-tuning of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.emnlp-main.1046/.

[27] Wenzhi Fang, Dong-Jun Han, Liangqi Yuan, Seyyedali Hosseinalipour, and Christopher G. Brinton. Federated sketching lora: A flexible framework for heterogeneous collaborative fine-tuning of llms, 2025. URL https://arxiv.org/abs/2501.19389.

[28] Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning, 2023. URL https://arxiv.org/abs/2308.03303.

[29] HuggingFace. Hugging face documentation, 2026. URL https://huggingface.co/docs/peft/package_reference/lora.

[30] Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. URL https://proceedings.mlr.press/v235/hao24a.html.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. URL https://ieeexplore.ieee.org/document/7410480.

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

[34] Nan Chen, Soledad Villar, and Soufiane Hayou. Learning rate scaling across loRA ranks and transfer to full finetuning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=vCAIwEJZQ6.

[35] Saber Malekmohammadi, Yaoliang Yu, and Yang Cao. Noise-aware algorithm for heterogeneous differentially private federated learning, 2025. URL https://arxiv.org/abs/2406.03519.

[36] Guojun Zhang, Saber Malekmohammadi, Xi Chen, and Yaoliang Yu. Proportional fairness in federated learning. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=ryUHgEdWCQ.

[37] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 2013. URL https://doi.org/10.1137/120880811.

[38] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[39] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, et al. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.319/.

[40] 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, 2019. URL https://arxiv.org/abs/1907.11692.

[41] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL https://arxiv.org/abs/1804.07461.

[42] A. Mood, F. Graybill, and D. Boes. *Introduction to the Theory of Statistics*. McGraw-Hill, Inc., 1974. ISBN 0070428646. URL https://sistemas.fciencias.unam.mx/~misraim/Mood.pdf.

[43] Patrick Billingsley. *Probability and Measure*. John Wiley & Sons, Inc., 1995. ISBN 0471007102. URL https://www.colorado.edu/amath/sites/default/files/attached-files/billingsley.pdf.

Appendix: *"Stochastic Projections and Rank-Aware Learning Rates for Federated LoRA"*

A Further experimental details

We provide more details about our experiments in this section.

A.1 Experimental setup

In all experiments, we sample client ranks from $\{4, 8, 16, 32, 64, 128\}$. For each training experiment, we run 5 seeds and report the mean and sample standard deviation (`std`) across them. We use $p = 0.2$ participation rate, uniform batch size of $b_i = 16$, and `AdamW` optimizer. Next, we describe specific configuration choices for the GLUE and Commonsense reasoning benchmarks.

GLUE. The GLUE benchmark is a collection of natural language understanding tasks, formatted as classification tasks with 2 or 3 classes [41]. We randomly initialize and freeze the classification layer of a `RoBERTa-base` model. We then apply rank-stabilized LoRA (rs-LoRA) with $\alpha = 1$ to the `["query", "value", "key", "dense"]` parameters, where the first 3 refer to the attention input projections and "dense" is the second-to-last fully connected layer (before the final classifier layer). We use a maximum sequence length of 128 tokens. We tune learning rate over $\{1e{-}4, 2e{-}4, 4e{-}4, 1e{-}3, 2e{-}3, 4e{-}3\}$ and select the best performing learning rate based on the average across all 7 benchmarks. We use the same grid of learning rates when searching for the client-specific ranks for SPRA-LoRA.

We do an ablation study on the GLUE benchmark with varying data heterogeneity (Dirichlet parameter α) and number of clients (n), as shown in Table 6. For all experiments with GLUE, we use a fixed number of communication rounds $E = 150$, with $T = 20$ gradient steps during each round.

Table 6: Studied values of client population size and data heterogeneity (n, α) on GLUE.

n (clients)	40			60	80
α (heterogeneity)	`0.4`	`0.8`	`2.0`	`0.8`	

Commonsense Reasoning. The training set of the commonsense reasoning benchmark is a mixture of 8 datasets including about 170K training samples [39]. For fine-tuning, we use rs-LoRA with $\alpha = 1$. We apply LoRA to the `["q proj", "k proj", "v proj", "up proj", "down proj"]` parameters of a pretrained `LLaMA-3.2-3B` model. We use a maximum sequence length of 128 tokens. We tune learning rate over $\{1e{-}4, 2e{-}4, 4e{-}4, 6e{-}4, 8e{-}4, 1e{-}3, 2e{-}3, 4e{-}3, 6e{-}3, 8e{-}3\}$ and select the best performing learning rate based on the average across all the 8 datasets. We split data between $n = 20$ clients with Dirichlet distribution with $\alpha = 2$ and use a fixed number of communication rounds $E = 100$, with $T = 20$ gradient steps during each round.

A.2 Further detailed results

Table 7: Test accuracy averaged over 5 runs on GLUE benchmark with Dirichlet distribution ($\alpha = \mathbf{0.4}$), `RoBERTa` model, `AdamW` optimizer and $\mathbf{n = 40}$ clients. Clients' ranks follow ***uniform*** distribution over $\{4, 8, \ldots, 128\}$.

Method	Run time	QNLI	MRPC	CoLA	MNLI	RTE	SST-2	QQP	Avg
Base Pretrained Model	—	49.46	31.62	30.87	35.45	52.71	49.08	63.18	44.62
FFA [2024]	0.82h	$81.28_{\pm 0.70}$	$77.80_{\pm 1.70}$	$77.66_{\pm 0.33}$	$70.19_{\pm 1.15}$	$66.14_{\pm 0.82}$	$90.48_{\pm 0.19}$	$73.14_{\pm 3.35}$	$76.67_{\pm 0.58}$
HetLoRA [2024]	0.61h	$83.61_{\pm 0.78}$	$81.42_{\pm 1.82}$	$78.01_{\pm 0.95}$	$65.98_{\pm 1.89}$	$53.93_{\pm 2.70}$	$91.42_{\pm 0.29}$	$\underline{77.78}_{\pm 2.37}$	$76.02_{\pm 0.66}$
FlexLoRA [2024]	1.05h	$83.68_{\pm 3.44}$	$83.67_{\pm 0.59}$	$\underline{80.81}_{\pm 0.44}$	$77.05_{\pm 0.82}$	$65.34_{\pm 3.48}$	$90.83_{\pm 0.84}$	$\mathbf{79.42}_{\pm 1.49}$	$80.11_{\pm 0.76}$
FLoRA [2024]	0.84h	$82.15_{\pm 2.16}$	$80.78_{\pm 1.97}$	$79.64_{\pm 0.93}$	$73.24_{\pm 1.20}$	$66.50_{\pm 1.66}$	$91.77_{\pm 0.35}$	$73.85_{\pm 3.96}$	$78.28_{\pm 0.77}$
FSLoRA [2025]	0.78h	$79.48_{\pm 1.47}$	$78.72_{\pm 1.32}$	$74.27_{\pm 0.89}$	$58.83_{\pm 2.35}$	$56.17_{\pm 3.95}$	$91.88_{\pm 0.47}$	$76.80_{\pm 1.02}$	$73.74_{\pm 0.74}$
FedSA [2025]	1.27h	$64.43_{\pm 1.17}$	$62.03_{\pm 4.38}$	$65.54_{\pm 1.19}$	$43.35_{\pm 0.99}$	$51.28_{\pm 0.85}$	$73.45_{\pm 3.53}$	$65.60_{\pm 1.61}$	$60.81_{\pm 0.89}$
RoLoRA [2025]	0.58h	$79.49_{\pm 0.90}$	$75.49_{\pm 1.39}$	$74.50_{\pm 1.59}$	$39.96_{\pm 2.27}$	$53.72_{\pm 3.18}$	$90.00_{\pm 0.33}$	$73.46_{\pm 2.48}$	$69.52_{\pm 0.74}$
LA-LoRA [2026]	0.60h	$80.52_{\pm 0.97}$	$77.16_{\pm 2.27}$	$76.55_{\pm 1.14}$	$56.71_{\pm 2.45}$	$54.94_{\pm 3.29}$	$90.50_{\pm 0.44}$	$72.74_{\pm 3.50}$	$72.73_{\pm 0.86}$
SPRA-LoRA (uniform LR)	0.85h	$83.30_{\pm 1.39}$	$80.39_{\pm 1.66}$	$79.10_{\pm 0.45}$	$71.05_{\pm 1.95}$	$\mathbf{68.23}_{\pm 1.36}$	$91.42_{\pm 0.23}$	$75.48_{\pm 2.77}$	$78.42_{\pm 0.61}$
SPRA-LoRA	0.79h	$\mathbf{85.59}_{\pm 2.09}$	$\mathbf{84.66}_{\pm 0.75}$	$\underline{80.81}_{\pm 0.89}$	$\underline{78.28}_{\pm 0.86}$	$64.47_{\pm 2.36}$	$\mathbf{92.32}_{\pm 0.58}$	$77.10_{\pm 3.86}$	$\underline{80.46}_{\pm 0.75}$
SPRA-LoRA (doubled ranks)	0.83h	$\underline{85.53}_{\pm 2.36}$	$\underline{83.82}_{\pm 2.18}$	$\mathbf{81.07}_{\pm 0.79}$	$\mathbf{78.62}_{\pm 0.86}$	$\underline{67.00}_{\pm 1.66}$	$\underline{92.29}_{\pm 0.38}$	$76.54_{\pm 4.22}$	$\mathbf{80.70}_{\pm 0.81}$

Table 8: Test accuracy averaged over 5 runs on GLUE benchmark with Dirichlet distribution ($\alpha = \mathbf{2.0}$), `RoBERTa` model, `AdamW` optimizer, and $\mathbf{n = 40}$ clients. Clients' ranks follow ***uniform*** distribution over $\{4, 8, \ldots, 128\}$.

Method	Run time	QNLI	MRPC	CoLA	MNLI	RTE	SST-2	QQP	Avg
Base Pretrained Model	—	49.46	31.62	30.87	35.45	52.71	49.08	63.18	44.62
FFA [2024]	0.76h	$84.94_{\pm 0.19}$	$84.51_{\pm 0.80}$	$79.77_{\pm 0.77}$	$74.95_{\pm 0.37}$	$67.51_{\pm 0.52}$	$91.22_{\pm 0.11}$	$77.11_{\pm 0.93}$	$80.00_{\pm 0.23}$
HetLoRA [2024]	0.59h	$86.18_{\pm 0.48}$	$83.53_{\pm 1.44}$	$81.72_{\pm 0.29}$	$78.44_{\pm 0.12}$	$70.18_{\pm 0.54}$	$92.39_{\pm 0.15}$	$77.09_{\pm 0.46}$	$81.36_{\pm 0.24}$
FlexLoRA [2024]	1.02h	$87.31_{\pm 0.34}$	$\mathbf{86.42}_{\pm 0.33}$	$82.26_{\pm 0.21}$	$80.20_{\pm 0.09}$	$\underline{73.86}_{\pm 0.80}$	$92.98_{\pm 0.11}$	$80.35_{\pm 0.70}$	$83.34_{\pm 0.17}$
FLoRA [2024]	0.83h	$86.20_{\pm 0.60}$	$84.95_{\pm 0.80}$	$81.84_{\pm 0.30}$	$78.50_{\pm 0.23}$	$71.48_{\pm 0.59}$	$92.75_{\pm 0.23}$	$79.78_{\pm 0.76}$	$82.21_{\pm 0.21}$
FSLoRA [2025]	0.78h	$85.19_{\pm 0.04}$	$83.28_{\pm 1.56}$	$81.05_{\pm 0.25}$	$74.15_{\pm 0.61}$	$65.34_{\pm 1.67}$	$92.00_{\pm 0.17}$	$76.01_{\pm 0.75}$	$79.57_{\pm 0.36}$
FedSA [2025]	1.28h	$72.01_{\pm 2.13}$	$67.55_{\pm 1.60}$	$72.39_{\pm 0.99}$	$52.24_{\pm 1.90}$	$52.37_{\pm 0.41}$	$90.46_{\pm 0.24}$	$71.63_{\pm 1.21}$	$68.38_{\pm 0.52}$
RoLoRA [2025]	0.62h	$82.42_{\pm 0.54}$	$80.98_{\pm 2.29}$	$80.86_{\pm 0.37}$	$64.38_{\pm 2.24}$	$66.64_{\pm 0.57}$	$91.21_{\pm 0.12}$	$72.80_{\pm 1.25}$	$77.04_{\pm 0.51}$
LA-LoRA [2026]	0.60h	$82.25_{\pm 0.70}$	$80.78_{\pm 2.62}$	$80.54_{\pm 0.30}$	$71.18_{\pm 0.56}$	$68.30_{\pm 0.97}$	$91.72_{\pm 0.10}$	$72.89_{\pm 1.02}$	$78.24_{\pm 0.45}$
SPRA-LoRA (uniform LR)	0.77h	$86.12_{\pm 0.53}$	$84.66_{\pm 0.58}$	$81.65_{\pm 0.31}$	$77.75_{\pm 0.19}$	$70.25_{\pm 0.90}$	$92.73_{\pm 0.19}$	$79.78_{\pm 0.90}$	$81.85_{\pm 0.22}$
SPRA-LoRA	0.81h	$\underline{88.57}_{\pm 0.60}$	$\underline{86.03}_{\pm 0.64}$	$\underline{82.38}_{\pm 0.35}$	$\mathbf{82.73}_{\pm 0.12}$	$\mathbf{73.87}_{\pm 0.18}$	$\underline{93.37}_{\pm 0.09}$	$\mathbf{82.77}_{\pm 0.49}$	$\mathbf{84.25}_{\pm 0.16}$
SPRA-LoRA (doubled ranks)	0.95h	$\mathbf{89.45}_{\pm 0.48}$	$85.93_{\pm 1.15}$	$\mathbf{82.72}_{\pm 0.19}$	$\underline{82.71}_{\pm 0.11}$	$71.91_{\pm 1.02}$	$\mathbf{93.56}_{\pm 0.10}$	$\underline{81.43}_{\pm 0.93}$	$\underline{83.96}_{\pm 0.27}$

Table 9: Test accuracy averaged over 5 runs on GLUE benchmark with Dirichlet distribution ($\alpha = \mathbf{0.8}$), `RoBERTa` model, `AdamW` optimizer and $\mathbf{n = 60}$ clients. Clients' ranks follow ***uniform*** distribution over $\{4, 8, \ldots, 128\}$.

Method	Run time	QNLI	MRPC	CoLA	MNLI	RTE	SST-2	QQP	Avg
Base Pretrained Model	—	49.46	31.62	30.87	35.45	52.71	49.08	63.18	44.62
FFA [2024]	1.26h	$85.15_{\pm 0.43}$	$83.48_{\pm 0.28}$	$79.06_{\pm 0.37}$	$74.04_{\pm 0.58}$	$63.83_{\pm 1.54}$	$91.49_{\pm 0.28}$	$79.95_{\pm 0.79}$	$79.57_{\pm 0.28}$
HetLoRA [2024]	0.85h	$84.75_{\pm 0.92}$	$85.05_{\pm 0.28}$	$81.15_{\pm 0.31}$	$77.41_{\pm 0.84}$	$61.45_{\pm 0.91}$	$92.09_{\pm 0.25}$	$80.21_{\pm 0.94}$	$80.30_{\pm 0.27}$
FlexLoRA [2024]	1.45h	$86.22_{\pm 0.85}$	$84.75_{\pm 0.46}$	$80.79_{\pm 0.33}$	$78.30_{\pm 0.69}$	$\underline{69.03}_{\pm 0.79}$	$92.06_{\pm 0.38}$	$83.02_{\pm 0.20}$	$82.02_{\pm 0.22}$
FLoRA [2024]	1.31h	$85.48_{\pm 0.85}$	$\underline{86.13}_{\pm 0.28}$	$79.79_{\pm 0.61}$	$77.06_{\pm 0.32}$	$68.45_{\pm 0.67}$	$93.09_{\pm 0.07}$	$81.17_{\pm 0.87}$	$81.60_{\pm 0.23}$
FSLoRA [2025]	1.23h	$81.31_{\pm 2.57}$	$84.31_{\pm 0.77}$	$80.04_{\pm 0.25}$	$67.99_{\pm 2.77}$	$62.60_{\pm 2.12}$	$91.56_{\pm 0.24}$	$79.56_{\pm 0.75}$	$78.20_{\pm 0.64}$
FedSA [2025]	1.86h	$69.12_{\pm 1.35}$	$59.31_{\pm 0.86}$	$61.77_{\pm 2.35}$	$46.83_{\pm 1.89}$	$51.00_{\pm 0.52}$	$84.87_{\pm 2.20}$	$66.97_{\pm 1.52}$	$62.84_{\pm 0.62}$
RoLoRA [2025]	0.89h	$80.48_{\pm 1.31}$	$82.35_{\pm 0.52}$	$79.44_{\pm 0.49}$	$59.10_{\pm 4.99}$	$60.00_{\pm 1.56}$	$91.12_{\pm 0.25}$	$75.85_{\pm 1.60}$	$75.48_{\pm 0.81}$
LA-LoRA [2026]	0.87h	$80.85_{\pm 1.66}$	$83.92_{\pm 0.84}$	$79.06_{\pm 0.45}$	$69.49_{\pm 3.42}$	$60.65_{\pm 1.42}$	$91.77_{\pm 0.22}$	$76.24_{\pm 1.87}$	$77.42_{\pm 0.65}$
SPRA-LoRA (uniform LR)	1.18h	$85.82_{\pm 0.66}$	$85.44_{\pm 0.72}$	$79.52_{\pm 0.75}$	$75.46_{\pm 0.40}$	$66.86_{\pm 1.93}$	$92.91_{\pm 0.08}$	$81.21_{\pm 0.62}$	$81.03_{\pm 0.34}$
SPRA-LoRA	1.17h	$\mathbf{89.32}_{\pm 0.58}$	$85.98_{\pm 0.38}$	$\mathbf{81.32}_{\pm 0.32}$	$\underline{81.31}_{\pm 0.28}$	$\mathbf{69.89}_{\pm 0.72}$	$\mathbf{93.33}_{\pm 0.13}$	$\mathbf{84.32}_{\pm 0.67}$	$\mathbf{83.64}_{\pm 0.18}$
SPRA-LoRA (doubled ranks)	1.32h	$\underline{89.02}_{\pm 0.57}$	$\mathbf{86.91}_{\pm 0.21}$	$\underline{81.23}_{\pm 0.22}$	$\mathbf{81.66}_{\pm 0.28}$	$68.09_{\pm 1.47}$	$\underline{93.16}_{\pm 0.09}$	$\underline{84.30}_{\pm 0.85}$	$\underline{83.48}_{\pm 0.26}$

Table 10: Test accuracy averaged over 5 runs on GLUE benchmark with Dirichlet distribution ($\alpha = \mathbf{0.8}$), `RoBERTa` model, `AdamW` optimizer and $\mathbf{n = 80}$ clients. Clients' ranks follow ***uniform*** distribution over $\{4, 8, \ldots, 128\}$.

Method	Run time	QNLI	MRPC	CoLA	MNLI	RTE	SST-2	QQP	Avg
Base Pretrained Model	—	49.46	31.62	30.87	35.45	52.71	49.08	63.18	44.62
FFA [2024]	1.51h	$84.68_{\pm 0.55}$	$82.45_{\pm 0.85}$	$79.46_{\pm 0.50}$	$74.76_{\pm 0.80}$	$66.57_{\pm 1.59}$	$91.72_{\pm 0.32}$	$78.12_{\pm 1.44}$	$79.68_{\pm 0.37}$
HetLoRA [2024]	1.08h	$86.10_{\pm 0.33}$	$84.07_{\pm 0.58}$	$\underline{80.71}_{\pm 0.35}$	$77.74_{\pm 0.48}$	$58.77_{\pm 2.02}$	$92.27_{\pm 0.11}$	$79.46_{\pm 1.06}$	$79.87_{\pm 0.35}$
FlexLoRA [2024]	1.54h	$87.36_{\pm 0.20}$	$85.73_{\pm 0.40}$	$80.59_{\pm 1.01}$	$78.88_{\pm 0.23}$	$\mathbf{70.47}_{\pm 1.99}$	$92.98_{\pm 0.25}$	$81.66_{\pm 1.15}$	$82.52_{\pm 0.37}$
FLoRA [2024]	1.48h	$85.87_{\pm 0.63}$	$84.85_{\pm 0.66}$	$80.36_{\pm 0.23}$	$76.45_{\pm 0.39}$	$\underline{68.52}_{\pm 1.30}$	$92.61_{\pm 0.23}$	$80.30_{\pm 1.23}$	$81.28_{\pm 0.30}$
FSLoRA [2025]	1.40h	$84.50_{\pm 0.35}$	$82.69_{\pm 0.91}$	$80.50_{\pm 0.41}$	$71.83_{\pm 1.22}$	$58.70_{\pm 2.63}$	$91.79_{\pm 0.18}$	$77.55_{\pm 0.72}$	$78.22_{\pm 0.45}$
FedSA [2025]	2.64h	$68.73_{\pm 2.79}$	$61.69_{\pm 1.12}$	$65.24_{\pm 1.32}$	$47.62_{\pm 0.67}$	$50.85_{\pm 0.24}$	$84.40_{\pm 0.54}$	$66.58_{\pm 1.62}$	$63.59_{\pm 0.54}$
RoLoRA [2025]	1.09h	$82.10_{\pm 0.49}$	$81.62_{\pm 0.63}$	$78.18_{\pm 0.47}$	$56.60_{\pm 2.51}$	$58.85_{\pm 1.39}$	$91.17_{\pm 0.17}$	$74.79_{\pm 1.33}$	$74.76_{\pm 0.47}$
LA-LoRA [2026]	1.07h	$82.60_{\pm 0.74}$	$83.18_{\pm 0.99}$	$78.62_{\pm 0.59}$	$69.94_{\pm 1.56}$	$61.95_{\pm 1.50}$	$91.42_{\pm 0.17}$	$76.43_{\pm 1.18}$	$77.73_{\pm 0.40}$
SPRA-LoRA (uniform LR)	1.50h	$85.62_{\pm 0.55}$	$84.95_{\pm 0.24}$	$80.04_{\pm 0.37}$	$75.03_{\pm 0.61}$	$68.09_{\pm 1.96}$	$92.59_{\pm 0.21}$	$79.31_{\pm 1.51}$	$80.80_{\pm 0.38}$
SPRA-LoRA	1.53h	$\mathbf{89.03}_{\pm 0.84}$	$\underline{86.03}_{\pm 0.42}$	$\mathbf{82.09}_{\pm 0.30}$	$\mathbf{81.54}_{\pm 0.26}$	$65.63_{\pm 1.80}$	$\underline{93.19}_{\pm 0.10}$	$\mathbf{83.65}_{\pm 0.58}$	$\mathbf{83.02}_{\pm 0.31}$
SPRA-LoRA (doubled ranks)	1.66h	$\underline{88.94}_{\pm 0.67}$	$\mathbf{86.22}_{\pm 0.28}$	$82.09_{\pm 0.20}$	$\underline{81.50}_{\pm 0.41}$	$64.69_{\pm 2.41}$	$\mathbf{93.26}_{\pm 0.14}$	$\underline{82.73}_{\pm 0.86}$	$\underline{82.78}_{\pm 0.39}$

A.3 Visualization of Lemma 4.1 for $\nu = -0.5$ and $\mu = 2$

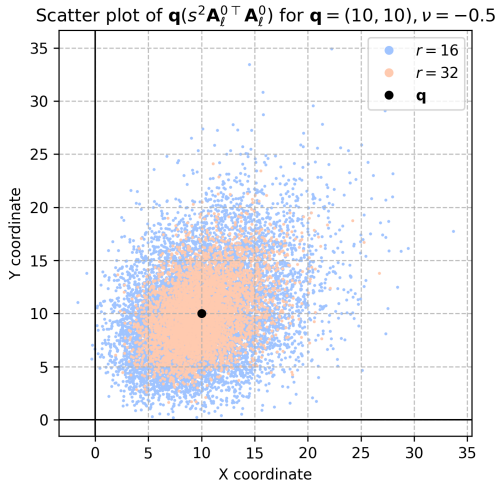

Figure 7: Scatter plot of $\mathbf{q}(s^2\mathbf{A}_\ell^{0\top}\mathbf{A}_\ell^0)$ plotted for $\mathbf{q} = (10, 10)$, 10^4 samples $\mathbf{A}_\ell^0 \sim \mathcal{N}(0, \mathbb{I}_{r\times 2}/r^2)$, $s = \alpha/r^\nu = r^{0.5}$, and two values of adaptation rank $r \in \{16, 32\}$. As shown, unlike when $\nu > -0.5$, the perturbation is unbiased for every value of r. As r increases, the additive noise magnitude decreases inversely proportional to r and `SNR` increases linearly with r.

B Component A is almost frozen when fine-tuning with LoRA

The following lemma shows that when fine-tuning with LoRA, the component matrix $\mathbf{A}$ is almost frozen, compared to the component matrix $\mathbf{B}$.

Lemma B.1. *Initializing $\mathbf{B}_\ell$ with a zero matrix, let LoRA update matrices $\mathbf{A}_\ell$ and $\mathbf{B}_\ell$ with SGD and learning rate η for T steps. Also, let us assume $s = 1$ for simplicity. Let $\mathbf{W}_\ell^t = \mathbf{W}_\ell^0 + \mathbf{B}_\ell^t\mathbf{A}_\ell^t$ be the **equivalent forward pass parameter** at the end of step t. The final equivalent parameter $\mathbf{W}_\ell^T$ can be written as:*

$$\mathbf{W}_\ell^T = \mathbf{W}_\ell^0 + \mathbf{B}_\ell^T\mathbf{A}_\ell^T = \mathbf{W}_\ell^0 - \eta\sum_{t=0}^{T-1}\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t\mathbf{A}_\ell^{0\top}\mathbf{A}_\ell^0 + \mathcal{O}(\eta^3), \tag{9}$$

where the term $\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t$ in the sum is the full batch gradient at time step t w.r.t. $\mathbf{W}_\ell^t = \mathbf{W}_\ell^0 + \mathbf{B}_\ell^t\mathbf{A}_\ell^t$.

Proof. Our proof is directly based on the derivations by [30]. We first restate the following theorem from the work without restating its proof:

Theorem B.2 (Hao et al. [30])**.** *Let LoRA update matrices $\mathbf{A}_\ell$ and $\mathbf{B}_\ell$ with SGD as:*

$$\mathbf{A}_\ell^{t+1} \leftarrow \mathbf{A}_\ell^t - \eta\frac{\partial\mathcal{L}}{\partial\mathbf{A}_\ell^t} = \mathbf{A}_\ell^t - \eta\mathbf{B}_\ell^{t\top}(\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t),$$

$$\mathbf{B}_\ell^{t+1} \leftarrow \mathbf{B}_\ell^t - \eta\frac{\partial\mathcal{L}}{\partial\mathbf{B}_\ell^t} = \mathbf{B}_\ell^t - \eta(\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t)\mathbf{A}_\ell^{t\top},$$

where η is the learning rate. We assume $\|\sum_{t=0}^{\tau}\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t\|_F \le L$ for every $\tau \le T-1$ during training, implying that the model stays within a finite Euclidean ball from $\mathbf{W}_\ell^0$. In this case, the dynamics of $\mathbf{A}_\ell^t$ and $\mathbf{B}_\ell^t$ are given by

$$\mathbf{A}_\ell^t = \mathbf{A}_\ell^0 + \eta\mathbf{A}_\ell^0 f_{\mathbf{A}_\ell}(t), \quad \mathbf{B}_\ell^t = \eta f_{\mathbf{B}_\ell}(t)\mathbf{A}_\ell^{0\top}, \tag{10}$$

where the forms of $f_{\mathbf{A}_\ell}(t) \in \mathbb{R}^{m\times m}$ and $f_{\mathbf{B}_\ell}(t) \in \mathbb{R}^{n\times m}$ are as follows for $t \ge 1$:

$$f_{\mathbf{A}_\ell}(t) = -\eta\sum_{i=0}^{t-1} f_{\mathbf{B}_\ell}^\top(i)(\nabla_{W^i}\mathcal{L}^i),$$
$$f_{\mathbf{B}_\ell}(t) = -\sum_{i=0}^{t-1}(\nabla_{W^i}\mathcal{L}^i)(\eta f_{\mathbf{A}_\ell}^\top(i) + I),$$

and $f_{\mathbf{A}_\ell}(0) = f_{\mathbf{B}_\ell}(0) = 0$. In particular, $\|f_{\mathbf{A}_\ell}(t)\| \le \frac{\eta L^2\left(1-(\eta^2L^2)^t\right)}{1-\eta^2L^2}$ for every t.

Therefore, we have:

$$\begin{aligned}\mathbf{W}_\ell^T = \mathbf{W}_\ell^0 + \mathbf{B}_\ell^T\mathbf{A}_\ell^T &= \mathbf{W}_\ell^0 + (\eta f_B(T)\mathbf{A}_\ell^{0\top})(\mathbf{A}_\ell^0 + \eta\mathbf{A}_\ell^0 f_A(T)) \\ &= \mathbf{W}_\ell^0 + \eta f_B(T)\mathbf{A}_\ell^{0\top}\mathbf{A}_\ell^0 + \eta^2 f_B(T)\mathbf{A}_\ell^{0\top}\mathbf{A}_\ell^0 f_A(T).\end{aligned}$$

After substituting $f_B(T)$ into the last term of the above equation and noting that $f_A(T) \in \mathcal{O}(\eta)$, we get to:

$$\mathbf{W}_\ell^T = \mathbf{W}_\ell^0 + \mathbf{B}_\ell^T\mathbf{A}_\ell^T = \mathbf{W}_\ell^0 - \eta\left(\sum_{t=0}^{T-1}\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t\mathbf{A}_\ell^{0\top}\mathbf{A}_\ell^0\right) + \mathcal{O}(\eta^3),$$

which completes the proof. □

C Useful lemmas and results

Theorem C.1 (Chi-Squared distribution: [42], Section 4.3, Theorem 7)**.** *If the random variables X_i, $i = 1, \ldots, k$, are normally and independently distributed with means μ_i and variances σ_i^2, then*

$$U = \sum_{i=1}^{k} \left(\frac{X_i - \mu_i}{\sigma_i}\right)^2$$

has a chi-squared distribution with k degrees of freedom: $U \sim \mathcal{X}_k^2$. Also, $\mathbb{E}[U] = k$ and $\texttt{Var}[U] = 2k$.

The theorem above states that sum of the squares of k standard normal random variables is a chi-squared distribution with k degrees of freedom.

Lemma C.2 (Raw moments of Chi-Squared distribution)**.** *Suppose $X \sim \mathcal{X}_k^2$. Then, the m-th raw moment of X can be found as follows;*

$$\mathbb{E}[X^m] = \prod_{i=0}^{m-1} (k + 2i)$$

Proof. From the definition of Chi-Squared distribution with r degrees of reddom, X has the following probability density function:

$$f_X(x) = \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})} x^{\frac{k}{2}-1} e^{-\frac{x}{2}}$$

Therefore, we have:

$$\mathbb{E}[X^m] = \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})} \int_0^{+\infty} x^{\frac{k}{2}+m-1} e^{-\frac{x}{2}} dx = \frac{2}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})} \int_0^{+\infty} (2u)^{\frac{k}{2}+m-1} e^{-u} du$$

$$= \frac{2^{\frac{k}{2}+m-1+1}}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})} \int_0^{+\infty} u^{\frac{k}{2}+m-1} e^{-u} du = \frac{2^m}{\Gamma(\frac{k}{2})}\Gamma(\frac{k}{2}+m) = \frac{2^m\Gamma(\frac{k}{2})}{\Gamma(\frac{k}{2})} \prod_{i=0}^{m-1} (\frac{k}{2}+i) = \prod_{i=0}^{m-1} (k+2i).$$

Note that the fifth equality directly results from the property of gamma function that for $z > 0$, $\Gamma(1+z) = z\Gamma(z)$. □

Theorem C.3 (Classical Central Limit Theorem: [43], Theorem 27.1)**.** *Suppose that $\{X_i\}_{i=1}^n$, is an independent sequence of random variables having the same distribution with mean μ and positive variance σ^2. Define $S_n = \sum_{i=1}^n X_i$ as their sum. Let Z_n be defined by*

$$Z_n = \frac{S_n - n\mu}{\sqrt{n}\sigma}.$$

Then, the distribution of Z_n approaches standard normal distribution as n approaches infinity.

The theorem above states that S_n is approximately, or asymptotically, distributed as a normal distribution with mean $n\mu$ and variance $n\sigma^2$.

The next theorem is about the Lindeberg's condition, which is a sufficient (and under certain conditions also a necessary condition) for the Central Limit Theorem (CLT) to hold for a sequence of independent random variables $\{X_i\}_{i=1}^n$. Unlike the classical CLT stated above, which requires the sequence of random variables to have a finite variance and be both independent and identically distributed (*i.i.d*), Lindeberg's CLT only requires the sequence of random variables to have finite variance, be independent and also satisfy the Lindeberg's condition. The following states the theorem.

Theorem C.4 (Lindeberg and Lyapounov Theorem: [43], Theorem 27.2)**.** *Suppose $X_1, \ldots, X_n$ are n independent random variables with $\mathbb{E}[X_i] = \mu_i$ and $\texttt{Var}[X_i] = \sigma_i^2 > 0$. Define $S_n = \sum_{i=1}^n X_i$ and let $s_n^2 = \sum_{i=1}^n \sigma_i^2$. Also assume the following condition holds for all $\epsilon > 0$:*

$$\textit{Lindeberg's condition:} \quad \lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{s_n^2} \int_{|x-\mu_i|\geq \epsilon s_n} (x-\mu_i)^2 P_{X_i}(x)dx = 0,$$

where P_{X_i} is the pdf of variable X_i. Assuming $Z_n = \frac{S_n - \sum_{i=1}^n \mu_i}{s_n}$, the distribution of Z_n approaches standard normal distribution as n approaches infinity.

The theorem above states that, given that Lindeberg's condition is satisfied, S_n is approximately, or asymptotically, distributed as a normal distribution with mean $\sum_{i=1}^n \mu_i$ and variance s_n^2, even if the sequence of variables are not identically distributed.

D Proof of Lemma 4.1, with Gaussian initialization of $\{\mathbf{A}_\ell\}_{\ell=1}^L$

Lemma 4.1. *Consider a fixed $\mathbf{q} \in \mathbb{R}^{1\times m}$ and $\mathbf{A}_\ell^0 \in \mathbb{R}^{r\times m} \sim \mathcal{N}(0, \sigma_{\mathbf{A}}^2 \mathbb{I}_{r\times m})$, with $\sigma_{\mathbf{A}}^2 = 1/r^\mu$ $(2\nu + \mu > 0.5)$. Random perturbation $\mathbf{q}(s^2 \mathbf{A}_\ell^{0\top} \mathbf{A}_\ell^0)$ approaches the m-dimensional Gaussian $\mathcal{N}\big(\alpha^2 \mathbf{q}/r^{2\nu+\mu-1}, \alpha^4(\|\mathbf{q}\|^2 \mathbb{I}_m + \mathbf{D})/r^{4\nu+2\mu-1}\big)$ as r increases, where $\mathbf{D} = diag(\mathbf{q}^2)$.*

Proof. Let us denote $\mathbf{A}_\ell^0$ with $\mathbf{A}_\ell$ for simplicity. We first derive the distribution of $\mathbf{q} \cdot (\mathbf{A}_\ell^\top \mathbf{A}_\ell - \mathbb{I}_m)$ for simplicity of notations. Then, we translate it to the distribution of $\mathbf{q}\cdot((s\mathbf{A}_\ell^\top)(s\mathbf{A}_\ell) - \mathbb{I}_m) \in \mathbb{R}^{1\times m}$. Finally, distribution of $\mathbf{q}(s^2 \mathbf{A}_\ell^\top \mathbf{A}_\ell) \in \mathbb{R}^{1\times m}$ will be obtained after a shift in the mean with $\mathbf{q}$.

From the theorem's assumption, we know that elements of $\mathbf{A}_\ell$ are from $\mathcal{N}(0, \sigma_{\mathbf{A}}^2 \mathbb{I}_{r\times m})$. Therefore, we can rewrite the product $\mathbf{q} \cdot (\mathbf{A}_\ell^\top \mathbf{A}_\ell - \mathbb{I}_m) \in \mathbb{R}^{1\times m}$ as the following product:

$$\mathbf{q} \cdot (\sigma_{\mathbf{A}}^2 \mathbf{A}_\ell^\top \mathbf{A}_\ell - \mathbb{I}_m) \in \mathbb{R}^{1\times m} \tag{11}$$

where *the elements of $\mathbf{A}_\ell$ are now from standard normal distribution*. Let $a_{i,j}$ denote the element in i-th row and j-th column of this new standard normal $\mathbf{A}_\ell$. Therefore, for all i and j, $a_{i,j}$ has distribution $\mathcal{N}(0,1)$. Let $\mathbf{B} = \sigma_{\mathbf{A}}^2 \mathbf{A}_\ell^\top \mathbf{A}_\ell - \mathbb{I}_m$. Also, let $\mathbf{A}_{i,:}$ and $\mathbf{A}_{:,j}$ denote the i-th row and j-th column of the new $\mathbf{A}_\ell$, respectively. We have:

$$\mathbf{B}_{i,i} = \sigma_{\mathbf{A}}^2 [\mathbf{A}_\ell^\top \mathbf{A}_\ell]_{i,i} - 1 = \sigma_{\mathbf{A}}^2 \mathbf{A}_{:,i}^\top \mathbf{A}_{:,i} - 1 = \sigma_{\mathbf{A}}^2 \|\mathbf{A}_{:,i}\|_2^2 - 1 = (\sigma_{\mathbf{A}}^2 \sum_{l=1}^{r} a_{l,i}^2) - 1$$

From eq. (11), we know that $a_{l,i}$ is from standard normal distribution. Hence, $a_{l,i}^2$ is a chi-squared with 1 degree of freedom: $a_{l,i}^2 \sim \mathcal{X}_1^2$. Therefore, $\sum_{l=1}^r a_{l,i}^2$, which is the sum of r independent chi-squared variables with 1 degree of freedom, is a chi-squared with r degrees of freedom: $\sum_{l=1}^r a_{l,i}^2 \sim \mathcal{X}_r^2$ (see theorem C.1). Therefore, for $i \in \{1, \ldots, m\}$, we have:

$$\begin{aligned}
\mathbb{E}[\mathbf{B}_{i,i}] &= \mathbb{E}\big[\sigma_{\mathbf{A}}^2 \sum_{l=1}^{r} a_{l,i}^2\big] - 1 = r\sigma_{\mathbf{A}}^2 - 1, \\
\texttt{Var}[\mathbf{B}_{i,i}] &= \texttt{Var}[\sigma_{\mathbf{A}}^2 \sum_{l=1}^{r} a_{l,i}^2] = \sigma_{\mathbf{A}}^4 \texttt{Var}(\mathcal{X}_r^2) = 2r\sigma_{\mathbf{A}}^4.
\end{aligned} \tag{12}$$

Similarly, we find the mean and variance of the non-diagonal elements $\mathbf{B}_{i,j} (i \neq j)$ of $\mathbf{B}$. We have:

$$\mathbf{B}_{i,j} = \sigma_{\mathbf{A}}^2 [\mathbf{A}_\ell^\top \mathbf{A}_\ell]_{i,j} = \sigma_{\mathbf{A}}^2 \mathbf{A}_{:,i}^\top \mathbf{A}_{:,j} = \sigma_{\mathbf{A}}^2 \sum_{l=1}^{r} a_{l,i} a_{l,j}, \tag{13}$$

where $a_{l,i}$ and $a_{l,j}$ are independent and standard normal. Therefore, $a_{l,i} + a_{l,j} \sim \mathcal{N}(0, 2)$. Similarly, $a_{l,i} - a_{l,j} \sim \mathcal{N}(0, 2)$. So we can rewrite $a_{l,i}a_{l,j}$ as:

$$a_{l,i}a_{l,j} = \frac{1}{4}(a_{l,i} + a_{l,j})^2 - \frac{1}{4}(a_{l,i} - a_{l,j})^2 = \frac{1}{2}z_1^2 - \frac{1}{2}z_2^2,$$

where z_1 and z_2 are from standard normal. Therefore, $a_{l,i}a_{l,j} = \frac{\nu_1 - \nu_2}{2}$, where $\nu_1, \nu_2 \sim \mathcal{X}_1^2$. Also, $a_{l,i} + a_{l,j}$ and $a_{l,i} - a_{l,j}$ are independent variables (because both $a_{l,i}$ and $a_{l,j}$ are Gaussian, independent, and have equal variance). Hence, z_1 and z_2 are independent, and likewise ν_1 and ν_2 are independent. We conclude that:

$$a_{l,i}a_{l,j} = \frac{1}{2}(\nu_1 - \nu_2), \tag{14}$$

where $\nu_1, \nu_2 \sim \mathcal{X}_1^2$, and are independent.

Now, let us assume $\nu_1, \nu_2 \sim \mathcal{X}_k^2$ (a more general case), and let $M_{\nu_1}(t) = \mathbb{E}[e^{t\nu_1}]$ be the moment generating function (MGF) of ν_1. In this case, we know that $M_{\nu_1}(t) = M_{\nu_2}(t) = (1 - 2t)^{-\frac{k}{2}}$ (MGF of $\mathcal{X}_k^2$). Hence, $M_{\nu_1 - \nu_2}(t) = M_{\nu_1}(t) \cdot M_{\nu_2}(-t) = (1 - 4t^2)^{-\frac{k}{2}} = \left(\frac{\frac{1}{4}}{\frac{1}{4} - t^2}\right)^{\frac{k}{2}}$, which is the MGF of a symmetric about origin `Variance-Gamma` distribution with parameters $\lambda = \frac{k}{2}, \alpha = \frac{1}{2}, \beta = 0, \mu = 0, \gamma = \frac{1}{2}$. Therefore, when $\nu_1, \nu_2 \sim \mathcal{X}_k^2$, then $\nu_1 - \nu_2$ has this distribution, which has mean $\mu + 2\beta\lambda/\gamma^2 = 0$ and variance $2\lambda(1 + 2\beta^2/\gamma^2)/\gamma^2 = 4k$.

In eq. (14), we had $k = 1$, as we had $\nu_1, \nu_2 \sim \mathcal{X}_1^2$. Hence, based on the discussion above, we have:

$$\mathbb{E}[a_{l,i}a_{l,j}] = 0$$

$$\texttt{Var}[a_{l,i}a_{l,j}] = \frac{1}{4}\texttt{Var}[\nu_1 - \nu_2] = \frac{4k}{4}|_{k=1} = 1$$

Consequently, based on eq. (13) and from the results above, we can compute the mean and variance of the non-diagonal elements of $\mathbf{B}$ ($i \neq j$):

$$\begin{aligned}
\mathbb{E}[\mathbf{B}_{i,j}] &= \mathbb{E}\big[\sigma_{\mathbf{A}}^2 \sum_{l=1}^{r} a_{l,i}a_{l,j}\big] = \sigma_{\mathbf{A}}^2 \sum_{l=1}^{r} \mathbb{E}[a_{l,i}a_{l,j}] = 0, \\
\texttt{Var}[\mathbf{B}_{i,j}] &= \texttt{Var}[\sigma_{\mathbf{A}}^2 \sum_{l=1}^{r} a_{l,i}a_{l,j}] = \sigma_{\mathbf{A}}^4 \sum_{l=1}^{r} \texttt{Var}[a_{l,i}a_{l,j}] = r\sigma_{\mathbf{A}}^4.
\end{aligned} \tag{15}$$

So far, we have computed the mean and variance of each entry in $\mathbf{B} = \sigma_{\mathbf{A}}^2 \mathbf{A}_\ell^\top \mathbf{A}_\ell - \mathbb{I}_m \in \mathbb{R}^{m \times m}$ in eq. (12) and eq. (15). Now, for a given $\mathbf{q} \in \mathbb{R}^{1 \times m}$, we have:

$$\mathbf{q} \cdot \mathbf{B} = \sum_{l=1}^{m} \mathbf{q}_l \mathbf{B}_{l,:}, \tag{16}$$

where $\mathbf{B}_{l,:}$ is row l of $\mathbf{B}$. Let u_i denote the i-th element of $\mathbf{q} \cdot \mathbf{B}$. Hence, for each element u_i ($i \in \{1, \ldots, m\}$), we have:

$$
\begin{aligned}
\mathbb{E}[u_i] &= \mathbb{E}\Big[\sum_{l=1}^{m} \mathbf{q}_l \mathbf{B}_{l,i}\Big] = \mathbf{q}_i \mathbb{E}[\mathbf{B}_{i,i}] + \sum_{l\neq i} \mathbf{q}_l \mathbb{E}[\mathbf{B}_{l,i}] = (r\sigma_{\mathbf{A}}^2 - 1)\mathbf{q}_i, \\
\texttt{Var}[u_i] &= \texttt{Var}\Big[\sum_{l=1}^{m} \mathbf{q}_l \mathbf{B}_{l,i}\Big] = \texttt{Var}\Big[\sum_{l=1}^{m} \mathbf{q}_l \sigma_{\mathbf{A}}^2 \sum_{t=1}^{r} a_{t,l} a_{t,i}\Big] \\
&= \sum_{l=1}^{m} \mathbf{q}_l^2 \sigma_{\mathbf{A}}^4 \sum_{t=1}^{r} \texttt{Var}[a_{t,l} a_{t,i}] = 2\mathbf{q}_i^2 r\sigma_{\mathbf{A}}^4 + \sum_{l\neq i} \mathbf{q}_l^2 r\sigma_{\mathbf{A}}^4 \\
&= \mathbf{q}_i^2 r\sigma_{\mathbf{A}}^4 + \sum_{l=1}^{m} \mathbf{q}_l^2 r\sigma_{\mathbf{A}}^4 = r\sigma_{\mathbf{A}}^4(\mathbf{q}_i^2 + \|\mathbf{q}\|_2^2).
\end{aligned} \tag{17}
$$

In the second last line we could take the $\texttt{Var}$ into the summations because every pair of the mr summands are uncorrelated (so variance of their sum is sum of their variances).

Finally, we check if the conditions of Theorem C.4 hold for each element u_i of $\mathbf{q} \cdot \mathbf{B}$. According to eq. (16), each element u_i of $\mathbf{q} \cdot \mathbf{B}$ can be rewritten as the following sum:

$$
u_i = \sum_{l=1}^{m} \mathbf{q}_l \mathbf{B}_{l,i} = \sum_{l=1}^{m} \mathbf{q}_l \sigma_{\mathbf{A}}^2 \sum_{t=1}^{r} a_{t,l} a_{t,i} = \sum_{t=1}^{r} \Big(\sigma_{\mathbf{A}}^2 \sum_{l=1}^{m} \mathbf{q}_l a_{t,l} a_{t,i}\Big). \tag{18}
$$

Note every pair of the mr summands are uncorrelated. However, not all of them are independent. Yet, every pair of the r summands $\sum_{l=1}^{m} \mathbf{q}_l a_{t,l} a_{t,i}$ and $\sum_{l=1}^{m} \mathbf{q}_l a_{t',l} a_{t',i}$ are independent (for $t \neq t'$). Therefore, each u_i is the sum of r *independent* random variables. Also, each of the r summands has variance $\sigma_{\mathbf{A}}^4(\mathbf{q}_i^2 + \|\mathbf{q}\|_2^2)$. Hence, the sum of their variances of the r independent variables constituting u_i is $s_r^2 = r\sigma_{\mathbf{A}}^4(\mathbf{q}_i^2 + \|\mathbf{q}\|_2^2)$, which is the same as the variance we found in Equation (17). Assuming a non-zero $\mathbf{q}$, $s_r^2 \to \infty$, as $r \to \infty$. As each of the r summands $\sum_{l=1}^{m} \mathbf{q}_l a_{t,l} a_{t,i}$, for $t \in \{1, \cdots, r\}$, has a constant variance $\sigma_{\mathbf{A}}^4(\mathbf{q}_i^2 + \|\mathbf{q}\|_2^2)$ and a vanishing distribution tail, the Lindeberg's condition is also satisfied for every element u_i. Therefore, from theorem C.4, we conclude that as $r \to \infty$, each of the m elements of $\mathbf{q} \cdot \mathbf{B}$ (i.e., each element in $\{u_i\}_{i=1}^{m}$) approaches a Gaussian with its specific mean and variance found in eq. (17).

However, the m elements of $\mathbf{q} \cdot \mathbf{B}$ are not completely independent and are positively correlated. So the covariance matrix of $\mathbf{q} \cdot \mathbf{B}$ is not diagonal. Yet, for $\sigma_{\mathbf{A}}^2 = 1/r^{\mu}$ with $\mu > 0.5$, the relative impact of the dependency between the elements vanishes as r grows: according to Equation (17), $\mathbf{q} \cdot \mathbf{B}$ becomes increasingly "deterministic" and aligned with $-\mathbf{q}$, making its m elements effectively independent of each other. Hence, as r grows, the covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{m \times m}$ of $\mathbf{q} \cdot \mathbf{B}$ approaches a diagonal matrix with off-diagonal elements approaching zero and the i-th diagonal element coming from Equation (17). Therefore, we conclude that for $\mathbf{A}_\ell \sim \mathcal{N}(0, \sigma_{\mathbf{A}}^2 \mathbb{I}_{r \times m})$ with $\sigma_{\mathbf{A}}^2 = 1/r^{\mu}$ ($\mu > 0.5$), $\mathbf{q} \cdot (\mathbf{A}_\ell^\top \mathbf{A}_\ell - \mathbb{I}_m) \in \mathbb{R}^{1 \times m}$ approaches the Gaussian distribution $\mathcal{N}\big((r\sigma_{\mathbf{A}}^2 - 1)\,\mathbf{q},\ r\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2\,\mathbb{I}_m + \mathbf{D})\big)$, with $\mathbf{D} = diag(\mathbf{q}^2)$, as r increases.

Having found the asymptotic distribution of $\mathbf{q} \cdot (\mathbf{A}_\ell^\top \mathbf{A}_\ell - \mathbb{I}_m)$ above, we can directly translate it to the distribution of $\mathbf{q} \cdot \big((s\mathbf{A}_\ell)^\top (s\mathbf{A}_\ell) - \mathbb{I}_m\big)$, by replacing $\sigma_{\mathbf{A}}^2$ with $s^2\sigma_{\mathbf{A}}^2$, resulting in the asymptotic Gaussian distribution $\mathcal{N}\big((rs^2\sigma_{\mathbf{A}}^2 - 1)\,\mathbf{q},\ rs^4\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2\,\mathbb{I}_m + \mathbf{D})\big)$.

By shifting the mean by $\mathbf{q}$, we get distribution $\mathcal{N}\big(rs^2\sigma_{\mathbf{A}}^2\,\mathbf{q},\ rs^4\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2\,\mathbb{I}_m + \mathbf{D})\big)$ for $\mathbf{q} \cdot s^2 \mathbf{A}_\ell^\top \mathbf{A}_\ell$. Finally, after replacing $s = \alpha/r^{\nu}$ for LoRA and $\sigma_{\mathbf{A}}^2 = 1/r^{\mu}$, we get to the following distribution

$$
\mathcal{N}\big(\alpha^2 \mathbf{q}/r^{2\nu+\mu-1}, (\|\mathbf{q}\|^2 \mathbb{I}_m + \mathbf{D})\alpha^4/r^{4\nu+2\mu-1}\big).
$$

Finally, note that $s^2\sigma_{\mathbf{A}}^2 = \alpha^2/r^{2\nu+\mu}$, which imposes the constraint $2\nu + \mu > 0.5$. This completes the proof.

□

E Extension of Lemma 4.1 to Kaiming-Uniform initialization of $\{\mathbf{A}_\ell\}_{\ell=1}^L$

We now extend the result derived in Lemma 4.1 to Kaiming-Uniform initialization of $\{\mathbf{A}_\ell\}_{\ell=1}^L$ to show that a similar rank-dependent behavior exists.

Kaiming uniform initialization [31], is primarily designed for neural networks employing `ReLU`-like activation functions. While it was originally derived for `ReLU`, its principles can be extended to other non-linear activations like `GELU`, which is a smoother variant of `ReLU`. The core idea of Kaiming initialization is to set the initial weights in a way that helps maintain a consistent variance of activations throughout the network, preventing vanishing or exploding gradients during training. For uniform distribution, the weights are drawn from a uniform distribution within a specific range $\mathcal{U}(-b, b)$, where $b = \sqrt{6/m}$. The value of b is found for each layer such that the variance of each subsequent pre-activation layer is equal, e.g. in Figure 1, the variance of each element in y is equal to that of each element in the pre-activation of input x. We have the following lemma about this initialization.

Lemma E.1. *Consider a fixed $\mathbf{q} \in \mathbb{R}^{1\times m}$ independent of $\mathbf{A}_\ell^0 \in \mathbb{R}^{r\times m} \sim \mathcal{U}(-b, b)$, and assume $r \ll m$ holds. Random perturbation $\mathbf{q}(s^2\mathbf{A}_\ell^{0\top}\mathbf{A}_\ell^0)$ approaches the m-dimensional distribution $\mathcal{N}\big(\alpha^2\sigma_{\mathbf{A}}^2\ \mathbf{q}/r^{2\nu-1},\ \alpha^4\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2\,\mathbb{I}_m + \mathbf{D})/r^{4\nu-1}\big)$, as r increases. Also, $\sigma_{\mathbf{A}}^2 = b^2/3$ is the variance of each element in $\mathbf{A}_\ell$ and $\mathbf{D} \in \mathbb{R}^{m\times m}$ is a diagonal matrix with $\mathbf{D}_{i,i} = -\mathbf{q}_i^2/5$.*

Proof. Let us denote $\mathbf{A}_\ell^0$ with $\mathbf{A}_\ell$ for simplicity. We first derive the distribution of $\mathbf{q}\cdot(\mathbf{A}_\ell^\top\mathbf{A}_\ell - \mathbb{I}_m)$ for simplicity of notations. Then, we translate it to the distribution of $\mathbf{q}\cdot((s\mathbf{A}_\ell^\top)(s\mathbf{A}_\ell) - \mathbb{I}_m) \in \mathbb{R}^{1\times m}$. Finally, distribution of $\mathbf{q}(s^2\mathbf{A}_\ell^\top\mathbf{A}_\ell) \in \mathbb{R}^{1\times m}$ will be obtained after a shift in the mean with $\mathbf{q}$.

The elements of $\mathbf{A}_\ell$ are from $\mathcal{U}(-b, b)$, which has variance $\sigma_{\mathbf{A}}^2 = b^2/3$. Let $a_{i,j}$ denote the element in i-th row and j-th column of $\mathbf{A}_\ell$. Therefore, for all i and j, $a_{i,j}$ has distribution $\mathcal{U}(-b, b)$. Let $\mathbf{B} = \mathbf{A}_\ell^\top\mathbf{A}_\ell - \mathbb{I}_m$. Also, let $A_{i,:}$ and $A_{:,j}$ denote the i-th row and j-th column of $\mathbf{A}_\ell$, respectively. We have:

$$\mathbf{B}_{i,i} = [\mathbf{A}_\ell^\top\mathbf{A}]_{i,i} - 1 = \mathbf{A}_{:,i}^\top\mathbf{A}_{:,i} - 1 = \|\mathbf{A}_{:,i}\|_2^2 - 1 = \big(\sum_{l=1}^r a_{l,i}^2\big) - 1$$

We know that $a_{l,i}$ is from $\mathcal{U}(-b, b)$ distribution. Hence, $a_{l,i}^2/b^2$ follows `Beta`(1/2, 1) distribution. Equivalently, $\mathbb{E}[a_{l,i}^2] = \texttt{Var}[a_{l,i}] = \sigma_{\mathbf{A}}^2 = b^2/3$ and $\texttt{Var}[a_{l,i}^2] = 4b^4/45$. Therefore, for $\mathbf{B}_{i,i} = \sum_{l=1}^r a_{l,i}^2 - 1$ ($i \in \{1, \dots, m\}$), which is the sum of r independent such variables, we have:

$$\begin{aligned}\mathbb{E}[\mathbf{B}_{i,i}] &= \mathbb{E}\big[\sum_{l=1}^r a_{l,i}^2\big] - 1 = \frac{rb^2}{3} - 1,\\ \texttt{Var}[\mathbf{B}_{i,i}] &= \texttt{Var}[\sum_{l=1}^r a_{l,i}^2] = \frac{4rb^4}{45}.\end{aligned} \tag{19}$$

Similarly, we find the mean and variance of the non-diagonal elements $\mathbf{B}_{i,j} (i \neq j)$ of $\mathbf{B}$. We have:

$$\mathbf{B}_{i,j} = [\mathbf{A}_\ell^\top A]_{i,j} = \mathbf{A}_{:,i}^\top\mathbf{A}_{:,j} = \sum_{l=1}^r a_{l,i}a_{l,j},$$

where $a_{l,i}$ and $a_{l,j}$ are independent. Therefore:

$$\begin{aligned}\mathbb{E}[a_{l,i}a_{l,j}] &= 0\\ \texttt{Var}[a_{l,i}a_{l,j}] &= \frac{5b^4}{45}.\end{aligned}$$

Consequently, we can compute the mean and variance of the non-diagonal elements of $\mathbf{B}$ ($i \neq j$):

$$\mathbb{E}[\mathbf{B}_{i,j}] = \mathbb{E}\big[\sum_{l=1}^{r} a_{l,i}a_{l,j}\big] = 0,$$
$$\texttt{Var}[\mathbf{B}_{i,j}] = \texttt{Var}[\sum_{l=1}^{r} a_{l,i}a_{l,j}] = \sum_{l=1}^{r} \texttt{Var}[a_{l,i}a_{l,j}] = \frac{5rb^4}{45}. \tag{20}$$

In the last line, we could take the $\texttt{Var}$ into the summation because the r summands are independent.

So far, we have computed the mean and variance of each entry in $\mathbf{B} = \mathbf{A}_\ell^\top \mathbf{A}_\ell - \mathbb{I}_m \in \mathbb{R}^{m\times m}$. Now, for a given $\mathbf{q} \in \mathbb{R}^{1\times m}$, we have:

$$\mathbf{q}\cdot\mathbf{B} = \sum_{l=1}^{m} \mathbf{q}_l \mathbf{B}_{l,:}, \tag{21}$$

where $\mathbf{B}_{l,:}$ is row l of $\mathbf{B}$. Let u_i denote the i-th element of $\mathbf{q}\cdot\mathbf{B}$. Hence, for each element u_i ($i \in \{1,\dots,m\}$), we have:

$$\mathbb{E}[u_i] = \mathbb{E}\bigg[\sum_{l=1}^{m} \mathbf{q}_l \mathbf{B}_{l,i}\bigg] = \mathbf{q}_i \mathbb{E}[\mathbf{B}_{i,i}] + \sum_{l\neq i} \mathbf{q}_l \mathbb{E}[\mathbf{B}_{l,i}] = (rb^2/3 - 1)\mathbf{q}_i = (r\sigma_{\mathbf{A}}^2 - 1)\mathbf{q}_i,$$
$$\begin{aligned}
\texttt{Var}[u_i] &= \texttt{Var}\bigg[\sum_{l=1}^{m} \mathbf{q}_l \mathbf{B}_{l,i}\bigg] = \texttt{Var}\bigg[\sum_{l=1}^{m} \mathbf{q}_l \sum_{t=1}^{r} a_{t,l}a_{t,i}\bigg] = \texttt{Var}\bigg[\mathbf{q}_i \sum_{t=1}^{r} a_{t,i}^2 + \sum_{l\neq i, l=1}^{m} \mathbf{q}_l \sum_{t=1}^{r} a_{t,l}a_{t,i}\bigg] \\
&= \sum_{t=1}^{r} \mathbf{q}_i^2 \texttt{Var}[a_{t,i}^2] + \sum_{l\neq i, l=1}^{m} \mathbf{q}_l^2 \sum_{t=1}^{r} \texttt{Var}[a_{t,l}a_{t,i}] \\
&= \frac{4rb^4}{45}\mathbf{q}_i^2 + \sum_{l\neq i} \frac{5rb^4}{45}\mathbf{q}_l^2 = \frac{5rb^4}{45}\|\mathbf{q}\|_2^2 - \frac{rb^4}{45}\mathbf{q}_i^2 = r\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2 - \frac{\mathbf{q}_i^2}{5}).
\end{aligned} \tag{22}$$

In the second last line we could take the $\texttt{Var}$ into the summations because every pair of the mr summands are uncorrelated (so variance of their sum is sum of their variances).

Finally, we check if the conditions of Theorem C.4 hold for each element u_i of $\mathbf{q}\cdot\mathbf{B}$. According to eq. (21), each element u_i of $\mathbf{q}\cdot\mathbf{B}$ can be rewritten as the following sum:

$$u_i = \sum_{l=1}^{m} \mathbf{q}_l \mathbf{B}_{l,i} = \sum_{l=1}^{m} \mathbf{q}_l \sigma_{\mathbf{A}}^2 \sum_{t=1}^{r} a_{t,l}a_{t,i} = \sum_{t=1}^{r} \bigg(\sigma_{\mathbf{A}}^2 \sum_{l=1}^{m} \mathbf{q}_l a_{t,l}a_{t,i}\bigg). \tag{23}$$

Note every pair of the mr summands are uncorrelated. However, not all of them are independent. Yet, every pair of the r summands $\sum_{l=1}^{m} \mathbf{q}_l a_{t,l}a_{t,i}$ and $\sum_{l=1}^{m} \mathbf{q}_l a_{t',l}a_{t',i}$ are independent (for $t \neq t'$). Therefore, each u_i is the sum of r *independent* random variables. Also, each of the r summands has variance $\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2 - \mathbf{q}_i^2/5)$. Hence, the sum of the variances of the r independent variables constituting u_i is $s_r^2 = r\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2 - \mathbf{q}_i^2/5)$, which is the same as the variance we found in Equation (22). Assuming a non-zero $\mathbf{q}$, $s_r^2 \to \infty$, as $r \to \infty$. As each of the r summands $\sum_{l=1}^{m} \mathbf{q}_l a_{t,l}a_{t,i}$, for $t \in \{1,\cdots,r\}$, has a constant variance $\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2 - \mathbf{q}_i^2/5)$ and a vanishing distribution tail, the Lindeberg's condition is also satisfied for every element u_i. Therefore, from theorem C.4, we conclude that as $r \to \infty$, each of the m elements of $\mathbf{q}\cdot\mathbf{B}$ (i.e., each element in $\{u_i\}_{i=1}^m$) approaches a Gaussian with its specific mean and variance found in eq. (22).

However, the m elements of $\mathbf{q}\cdot\mathbf{B}$ are not independent and are positively correlated. So the covariance matrix of $\mathbf{q}\cdot\mathbf{B}$ is not completely diagonal. Yet, according to Equation (22) and as long as $r\sigma_{\mathbf{A}}^4 \ll 1$,

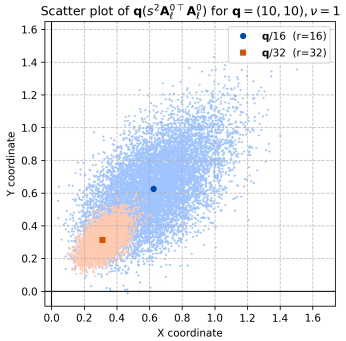

Figure 8: Scatter plot of $\mathbf{q}(s^2\mathbf{A}_\ell^{0\top}\mathbf{A}_\ell^0)$, for example $\mathbf{q}=(10,10)$, i.e., $m=2$, $b=\sqrt{6/m}$, 10^4 samples $\mathbf{A}_\ell^0 \sim \mathcal{U}(-b,b)$, $\alpha=1$, $\nu=1$ and $s=\alpha/r^\nu=1/r$.

$\mathbf{q}\cdot\mathbf{B}$ is almost "deterministic" and aligned with $-\mathbf{q}$, making its m elements effectively independent of each other. In other words, the relative impact of the dependency between the elements vanishes as long as $r\sigma_{\mathbf{A}}^4 \ll 1$, or equivalently $rb^4 \ll 1$ (remember that $\sigma_{\mathbf{A}}^2 = b^2/3$). Also, remember that for Kaiming-Uniform initialization with ReLU/GeLU activation, $b=\sqrt{6/m}$. Hence, the condition $rb^4 \ll 1$ reduces to condition $r/m^2 \ll 1$, which holds from the assumption $r/m \ll 1$. Under this condition, the covariance matrix $\mathbf{\Sigma}\in\mathbb{R}^{m\times m}$ of $\mathbf{q}\cdot\mathbf{B}$ is almost diagonal with off-diagonal elements approaching zero and the i-th diagonal element coming from the variance in Equation (22). Therefore, we conclude that, under $r \ll m$ and as r increases, $\mathbf{q}\cdot(\mathbf{A}_\ell^\top\mathbf{A}_\ell - \mathbb{I}_m) \in \mathbb{R}^{1\times m}$ approaches a Gaussian $\mathcal{N}\big((r\sigma_{\mathbf{A}}^2-1)\,\mathbf{q},\ r\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2\,\mathbb{I}_m+\mathbf{D})\big)$, where $\mathcal{D}$ is diagonal and $\mathcal{D}_{i,i}=-\mathbf{q}_i^2/5$.

Having found the asymptotic distribution of $\mathbf{q}\cdot(\mathbf{A}_\ell^\top\mathbf{A}_\ell-\mathbb{I}_m)$ above, we can directly translate it to the distribution of $\mathbf{q}\cdot\big((s\mathbf{A}_\ell)^\top(s\mathbf{A}_\ell)-\mathbb{I}_m\big)$, by replacing $\sigma_{\mathbf{A}}^2$ with $s^2\sigma_{\mathbf{A}}^2$, resulting in the asymptotic Gaussian distribution $\mathcal{N}\big((rs^2\sigma_{\mathbf{A}}^2-1)\,\mathbf{q},\ rs^4\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2\,\mathbb{I}_m+\mathbf{D})\big)$.

By shifting the mean of the distribution, we get distribution $\mathcal{N}\big(rs^2\sigma_{\mathbf{A}}^2\,\mathbf{q},\ rs^4\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2\,\mathbb{I}_m+\mathbf{D})\big)$ for $\mathbf{q}\cdot s^2\mathbf{A}_\ell^\top\mathbf{A}_\ell$. Finally, after replacing $s=\alpha/r^\nu$ for LoRA, we get to the following distribution:

$$\mathcal{N}\big(\alpha^2\sigma_{\mathbf{A}}^2\,\mathbf{q}/r^{2\nu-1},\ \alpha^4\sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2\,\mathbb{I}_m+\mathbf{D})/r^{4\nu-1}\big).$$

Finally, note that condition $rs^4\sigma_{\mathbf{A}}^4 \ll 1$ is satisfied if $r \ll m$. This completes the proof. □

The above lemma shows that, for a fixed $\mathbf{q}$, the term $\mathbf{q}(s^2\mathbf{A}_\ell^{0\top}\mathbf{A}_\ell^0)$ is a *biased* perturbation of $\mathbf{q}$ with a distribution centered at mean $\alpha^2\sigma_{\mathbf{A}}^2\,\mathbf{q}/r^{2\nu-1}$. Also, it has the following signal-to-noise ratio (SNR):

$$\forall\,\alpha,\nu:\ \mathrm{SNR}\big(\mathbf{q}(s^2\mathbf{A}_\ell^{0\top}\mathbf{A}_\ell^0)\big)=\frac{\|\alpha^2\sigma_{\mathbf{A}}^2\,\frac{\mathbf{q}}{r^{2\nu-1}}\|^2}{Tr\big(\frac{\alpha^4\sigma_{\mathbf{A}}^4}{r^{4\nu-1}}(\|\mathbf{q}\|_2^2\,\mathbb{I}_m+\mathbf{D})\big)}=\frac{\|\alpha^2\sigma_{\mathbf{A}}^2\,\frac{\mathbf{q}}{r^{2\nu-1}}\|^2}{\frac{\alpha^4\sigma_{\mathbf{A}}^4}{r^{4\nu-1}}(m-0.2)\|\mathbf{q}\|^2}=\frac{r}{m-0.2},\tag{24}$$

which linearly increases with adaptation rank r (similar to Equation (5) for Gaussian initialization). Choosing $\alpha=1$, $\nu=1$ yields to the following distribution:

$$\mathcal{N}\big(\sigma_{\mathbf{A}}^2\,\mathbf{q}/r,\ \sigma_{\mathbf{A}}^4(\|\mathbf{q}\|_2^2\,\mathbb{I}_m+\mathbf{D})/r^3\big),$$

with $\sigma_{\mathbf{A}}^2=2/m$. A scatter plot is shown in Figure 8 for an example $\mathbf{q}=(10,10)$. It is noteworthy that, due to the 2D constraint of the figure, the condition $r \ll m (m=2)$ is not satisfied in this figure. Hence, the scatter plots are not completely round-shaped like a Gaussian distribution.

F Effect of freezing and reusing $\mathbf{A}_\ell^0$ across time steps

We derived Lemma 4.1 under the assumption of independence between $\mathbf{q}$ and $\mathbf{A}_\ell^0$. However, in the paper we proposed to resample $\mathbf{A}_\ell^0$ for clients at the beginning of each round, i.e., after every T local steps. Therefore, in eq. (4) and eq. (7), the gradient $\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t$ weakly depends on $\mathbf{A}_\ell^0$ for $1 \leq t \leq T-1$, as $\mathbf{A}_\ell^0$ is resampled at step $t=0$ and frozen after that.

We now show that the true distribution of $\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t(s\mathbf{A}_\ell^0)^\top(s\mathbf{A}_\ell^0)$ and the distribution returned by Lemma 4.1, which assumes independence, exhibit a bounded mean and covariance shift, where the bound decreases linearly with T (and decreases with r).

Let us assume $s = 1/r^\nu$, i.e., $\alpha = 1$. Also, consistent with Lemma 4.1, let us assume that $\mathbf{A}_\ell^0 \in \mathbb{R}^{r\times m} \sim \mathcal{N}(0, \sigma_{\mathbf{A}}^2 \mathbb{I}_{r\times m})$, with $\sigma_{\mathbf{A}}^2 = 1/r^\mu$. Let us denote the scaled matrix $s\mathbf{A}_\ell^0$ with $\tilde{\mathbf{A}}$. Therefore, we have:

$$[\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}]_{i,j} = \tilde{\mathbf{A}}_{:,i}^\top\tilde{\mathbf{A}}_{:,j} = \sum_{l=1}^{r}\tilde{a}_{l,i}\tilde{a}_{l,j} = r^{-2\nu}\sum_{l=1}^{r}a_{l,i}a_{l,j}.$$

Therefore, we have:

$$\begin{aligned}\mathbb{E}\big[[\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}]_{i,j}\big] &= \mathbb{E}\big[\sum_{l=1}^{r}\tilde{a}_{l,i}\tilde{a}_{l,j}\big] = \mathbb{E}\big[r^{-2\nu}\sum_{l=1}^{r}a_{l,i}a_{l,j}\big] = r^{-2\nu}\sum_{l=1}^{r}\mathbb{E}\big[a_{l,i}a_{l,j}\big] \\ &= r^{-2\nu}\cdot r\cdot r^{-\mu}\cdot\delta_{i,j} = r^{1-2\nu-\mu}\delta_{i,j}.\end{aligned}$$

Therefore, we have:

$$\mathbb{E}\big[\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\big] = r^{1-2\nu-\mu}\mathbb{I}_m.$$

For simplicity of notations, let us denote the batch gradient $\nabla_{\mathbf{W}_\ell^t}\mathcal{L}^t$ at time step t with $\mathbf{g}^t$, where $1 \leq t \leq T-1$. The true mean of the perturbed batch gradient at time step t can be shown as the total expectation $\mathbb{E}[\mathbf{g}^t\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}]$ (over $\mathbf{g}^t$ and $\tilde{\mathbf{A}}$). In contrast, under the assumption of independence between $\tilde{\mathbf{A}}$ and $\mathbf{g}^t$, which resembles Lemma 4.1, the mean of the perturbed batch gradient at time step t is

$$\mathbb{E}[\mathbf{g}^t\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}] = \mathbb{E}[\mathbf{g}^t]\cdot\mathbb{E}[\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}] = r^{1-2\nu-\mu}\mathbb{E}[\mathbf{g}^t].$$

Therefore, the mean shift between the true distribution and the distribution returned by Lemma 4.1 under independence assumption can be found as follows:

$$\Delta_\mu^t = \mathbb{E}[\mathbf{g}^t\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}] - \big(r^{1-2\nu-\mu}\mathbb{E}[\mathbf{g}^t]\big) = \mathbb{E}\Big[(\mathbf{g}^t-\mathbf{g}^0)(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} - r^{1-2\nu-\mu}\mathbb{I}_m)\Big],$$

where in the last equality, we used the fact that $\mathbf{g}^0$ and $\tilde{\mathbf{A}}$ are independent. Therefore, we have:

$$\begin{aligned}\|\Delta_\mu^t\|_F &= \Big\|\mathbb{E}\Big[(\mathbf{g}^t-\mathbf{g}^0)(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} - r^{1-2\nu-\mu}\mathbb{I}_m)\Big]\Big\|_F \leq \mathbb{E}\Big[\Big\|(\mathbf{g}^t-\mathbf{g}^0)(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} - r^{1-2\nu-\mu}\mathbb{I}_m)\Big\|_F\Big] \\ &\leq \mathbb{E}\Big[\big\|(\mathbf{g}^t-\mathbf{g}^0)\big\|_F\big\|(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} - r^{1-2\nu-\mu}\mathbb{I}_m)\big\|_2\Big]. \qquad (25)\end{aligned}$$

Assuming `SGD`, we have:

$$\mathbf{W}_\ell^t = \mathbf{W}_\ell^0 - \eta\sum_{\tau=0}^{t-1}\mathbf{g}^\tau(s\mathbf{A}_\ell^0)^\top(s\mathbf{A}_\ell^0) = \mathbf{W}_\ell^0 - \eta\sum_{\tau=0}^{t-1}\mathbf{g}^\tau\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}},$$

where $1 \leq t \leq T-1$. Using β-smoothness of the loss function, we have:

$$\|\mathbf{g}^t - \mathbf{g}^0\|_F \leq \beta \|\mathbf{W}_\ell^t - \mathbf{W}_\ell^0\|_F = \beta\eta \left\| \sum_{\tau=0}^{t-1} \mathbf{g}^\tau \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \right\|_F \leq \beta\eta \sum_{\tau=0}^{t-1} \left\| \mathbf{g}^\tau \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \right\|_F$$

$$\leq \beta\eta \sum_{\tau=0}^{t-1} \left\| \mathbf{g}^\tau \right\|_F \left\| \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \right\|_2 = \beta\eta \cdot \left(\sum_{\tau=0}^{t-1} \left\| \mathbf{g}^\tau \right\|_F \right) \cdot \left\| \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \right\|_2.$$

Now, note that for random variables a, b and c, if $a \leq c$ and $b \geq 0$ almost surely, then pointwise $ab \leq cb$, which implies $\mathbb{E}[ab] \leq \mathbb{E}[cb]$ by the monotonicity of expected value. Therefore, by plugging the above bound into the right side of Equation (25), we get:

$$\|\Delta_\mu^t\|_F \leq \beta\eta \cdot \mathbb{E}\left[\left(\sum_{\tau=0}^{t-1} \left\| \mathbf{g}^\tau \right\|_F \right) \cdot \left\| \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \right\|_2 \cdot \left\| (\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} - r^{1-2\nu-\mu} \mathbb{I}_m) \right\|_2 \right]. \tag{26}$$

We can now use the L_0-Lipschitz continuity of the loss function to derive an upper bound for the term in parentheses:

$$\|\Delta_\mu^t\|_F \leq \beta\eta t L_0 \cdot \mathbb{E}\left[\left\| \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \right\|_2 \cdot \left\| (\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} - r^{1-2\nu-\mu} \mathbb{I}_m) \right\|_2 \right]$$

$$\leq \beta\eta T L_0 \cdot \mathbb{E}\left[\left\| \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \right\|_2 \cdot \left\| (\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} - r^{1-2\nu-\mu} \mathbb{I}_m) \right\|_2 \right],$$

where in the last line, we simply used the fact that $1 \leq t \leq T-1$. As observed, the above upper-bound on the mean shift decreases linearly as T (number of clients' local steps during each round or the *resampling period*) and η decrease. Also, it can be shown that the expectation term decreases as rank r increases. Therefore, the mean shift decreases as r increases.

A similar bound can be derived on the covariance shift norm $\|\Delta_\Sigma^t\|_F$, which decreases linearly as T and η decrease.

The above results show that freezing and reusing the LoRA matrix $\mathbf{A}_\ell^0$ results in a bounded shift in the mean and covariance derived in Lemma 4.1. More frequent resampling of the matrix reduces this shift. This validates the distribution approximation mentioned at the end of Section 4.1.

We can now get a more useful insight about eq. (26). Remember that $\mathbb{E}[\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}] = r^{1-2\nu-\mu} \mathbb{I}_m$. Therefore, the norm product term $\left\| \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \right\|_2 \cdot \left\| (\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} - r^{1-2\nu-\mu} \mathbb{I}_m) \right\|_2$ is almost surely bounded and has a high-probability bound ζ_r, which decreases with r. Conditioning on the event that $\left\| \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \right\|_2 \cdot \left\| (\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} - r^{1-2\nu-\mu} \mathbb{I}_m) \right\|_2 \leq \zeta_r$ holds, eq. (26) can be rewritten as:

$$\|\Delta_\mu^t\|_F \leq \beta\eta\zeta_r \mathbb{E}\left[\left(\sum_{\tau=0}^{t-1} \left\| \mathbf{g}^\tau \right\|_F \right) \right] = \beta\eta\zeta_r \sum_{\tau=0}^{t-1} \mathbb{E}\left[\left\| \mathbf{g}^\tau \right\|_F \right] \leq \gamma_r \sum_{\tau=0}^{t-1} \mathbb{E}\left[\left\| \mathbf{g}^\tau \right\|_F \right], \tag{27}$$

where constants ζ_r and $\gamma_r = \beta\eta\zeta_r$ depend on rank r. **Note that for any r, $\gamma_r \propto \eta$.** We will use this point in our convergence proof in Section G. The above inequality bounds the mean shift norm at step t ($1 \leq t \leq T-1$) with the sum of the expected batch gradient norms at the previous steps. Note that batch gradient norms generally decrease as training continues. Therefore, as SPRA-LoRA training continues, the upper-bound and consequently the mean shift decrease from one round to the next round. **Therefore, the per-T-steps resampling method of SPRA-LoRA becomes more similar to the per-step resampling gradually**. This observation provides an intuitive insight about convergence of SPRA-LoRA.

G Proof of Theorem 5.1

Theorem 5.1 (Convergence of SPRA-LoRA with SGD). *Assume using rank-stabilized LoRA ($\nu = 0.5$), assumptions G.1, and full participation with client i using learning rate $\eta_{r_i} = \eta_c r_i^2$. If the common learning rate satisfies $\eta_c \leq \frac{1}{4\beta T}$, then for any number of communication rounds $E \geq 1$,*

$$\boxed{\min_e \mathbb{E}\big[\, \|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2 \,\big] \;\leq\; \frac{2\big(f(\mathbf{W}^0) - f^\star\big)}{\eta_c T E} \;+\; \Psi}$$

where $\{\mathbf{W}^e\}_{e=0}^{E}$ are global iterates, and $\Psi = \mathcal{O}(\eta_c)$ is a constant. Choosing $\eta_c = \min\big(\frac{1}{4\beta T}, \frac{1}{\sqrt{E}}\big)$

$$\min_e \mathbb{E}\big[\, \|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2 \,\big] = \mathcal{O}\left(\frac{1}{\sqrt{E}}\right),$$

so the squared gradient norm of the global iterates converges, as $E \to \infty$, to a stationary point of f, recovering the classical $O(1/\sqrt{E})$ non-convex FedAvg rate [35, 36].

Proof. We first explain the notations and assumptions used in the proof in the following two subsections.

G.1 Notations used in the proof

We use the following notations for simplicity:

$$\mathbf{g}_i^{e,t} = \nabla_{\mathbf{W}_i^{e,t}} \mathcal{L}_i^{e,t}(\mathbf{W}_i^{e,t}) = \frac{1}{b_i} \sum_{x \in \mathcal{B}_i^{e,t}} \nabla_{\mathbf{W}_i^{e,t}} l(x; \mathbf{W}_i^{e,t}), \tag{28}$$

i.e., batch gradient of client i at adapter $\mathbf{W}_i^{e,t}$. We also use $\tilde{\mathbf{g}}_i^{e,t}$ to denote the perturbed batch gradient

$$\tilde{\mathbf{g}}_i^{e,t} = \mathbf{g}_i^{e,t}(s_i \mathbf{A}_i^{e,0})^\top (s_i \mathbf{A}_i^{e,0}).$$

We also use the following simplified notations:

$$\begin{aligned} f_i^{e,t} &= f_i(\mathbf{W}_i^{e,t}) = \frac{1}{N_i} \sum_{x \in \mathcal{D}_i} l(x; \mathbf{W}_i^{e,t}), \\ \nabla f_i^{e,t} &= \nabla_{\mathbf{W}_i^{e,t}} f_i(\mathbf{W}_i^{e,t}), \end{aligned} \tag{29}$$

i.e., the objective function of client i and its *full* gradient at the adapter $\mathbf{W}_i^{e,t}$. Finally, for matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$, define the Frobenius inner product and norm by:

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = Tr(\mathbf{A}, \mathbf{B}), \quad \|\mathbf{A}\|_F = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_F}.$$

G.2 Assumptions used in the proof

In this section, we formalize our assumptions used for the convergence analysis of SPRA-LoRA:

Assumption G.1 (Lipschitz continuity, β-smoothness, bounded gradient heterogeneity, and bounded gradient variance).

- Functions $\{f_i\}_{i=1}^n$ are L_0-Lipschitz continuous and β-smooth: $\forall\, \mathbf{W}, \mathbf{W}'$, $|f_i(\mathbf{W}) - f_i(\mathbf{W}')| \leq L_0 \|\mathbf{W} - \mathbf{W}'\|_F$ and $\|\nabla f_i(\mathbf{W}) - \nabla f_i(\mathbf{W}')\|_F \leq \beta \|\mathbf{W} - \mathbf{W}'\|_F$.
- For every i and j, $f_i - f_j$ is σ_{het}-Lipschitz continuous: $\|\nabla f_i(\mathbf{W}) - \nabla f_j(\mathbf{W})\|_F \leq \sigma_{het}$. In particular $\|\nabla f_i(\mathbf{W}) - \nabla f(\mathbf{W})\|_{\mathrm{F}} \leq \sigma_{het}$.
- The stochastic batch gradient $\mathbf{g}_i^{e,t}$ is an unbiased estimate of $\nabla f_i^{e,t}$ with a bounded variance: $\mathbb{E}_{\mathcal{B}_i^{e,t}}[\mathbf{g}_i^{e,t}] = \nabla f_i^{e,t}$, $\mathbb{E}_{\mathcal{B}_i^{e,t}}\left[\|\mathbf{g}_i^{e,t} - \nabla f_i^{e,t}\|_F^2\right] \leq \sigma_{i,g}^2$.

G.3 Proof of the Theorem

For the proof, we use the result of Lemma 4.1 and the findings in Section F. In SPRA-LoRA with $\nu = 0.5$, i.e., rank-stabilized LoRA, each client samples the frozen LoRA components $\mathbf{A}_i^{e,0}$ from $\mathcal{N}(0, \frac{1}{r_i^2}\mathbb{I}_{r_i \times n})$.

The adapter update of client i made at the $(t+1)$-th local SGD step ($t \in \{0, \dots, T-1\}$) in round e is obtained from the multiplication of the update $s_i \Delta \mathbf{B}_i^{e,t}$ and the resampled $\mathbf{A}_i^{e,0}$ of the round:

$$\begin{aligned}\Delta \mathbf{W}_i^{e,t} = s_i \Delta \mathbf{B}_i^{e,t} \mathbf{A}_i^{e,0} = -\eta_{r_i} (\nabla_{\mathbf{B}_i^{e,t}} \mathcal{L}_i^{e,t})(s_i \mathbf{A}_i^{e,0}) &= -\eta_{r_i} (\nabla_{\mathbf{W}_i^{e,t}} \mathcal{L}_i^{e,t})(s_i \mathbf{A}_i^{e,0})^\top (s_i \mathbf{A}_i^{e,0}) \\ &= -\eta_{r_i} \mathbf{g}_i^{e,t} (s_i \mathbf{A}_i^{e,0})^\top (s_i \mathbf{A}_i^{e,0}) \\ &= -\eta_{r_i} \widetilde{\mathbf{g}}_i^{e,t},\end{aligned}$$

where in the second line, we used the batch gradient notation introduced in Equation (28). This is equivalent to a SGD step with step size η_{r_i} and perturbed batch gradient $\widetilde{\mathbf{g}}_i^{e,t} = \mathbf{g}_i^{e,t} (s_i \mathbf{A}_i^{e,0})^\top (s_i \mathbf{A}_i^{e,0})$ w.r.t the adapter $\mathbf{W}_i^{e,t}$. Now, remember from Section F that, due to the dependence between $\mathbf{g}_i^{e,t}$ ($t \geq 1$) and $\mathbf{A}_i^{e,0}$, $\mathbb{E}[\widetilde{\mathbf{g}}_i^{e,t}]$ experiences a bounded shift from the mean $\mathbb{E}[\mathbf{g}_i^{e,t}]/r_i^2 = \nabla f_i^{e,t}/r_i^2$ returned by Lemma 4.1 (Remember, we have assumed rank-stabilized LoRA with $\nu = 0.5$ and $\mu = 2$). Therefore, based on the derivations from the same Section F, $\mathbb{E}[\widetilde{\mathbf{g}}_i^{e,t}]$ can be written as follows based on a bounded mean shift $\mathbf{\Delta}_{\mu_i}^{e,t}$:

$$\mathbb{E}[\widetilde{\mathbf{g}}_i^{e,t}] = \frac{\nabla f_i^{e,t}}{r_i^2} + \mathbf{\Delta}_i^{e,t},$$

where the constant mean shift $\mathbf{\Delta}_i^{e,t}$ satisfies $\|\mathbf{\Delta}_i^{e,t}\|_F \leq \gamma_{r_i} \sum_{\tau=0}^{t-1} \left\|\mathbf{g}_i^{e,\tau}\right\|_F$. Based on eq. (27), constant γ_{r_i} depends on the rank r_i of client i and is proportional to the step size $\eta_{r_i} = \eta_c r_i^2$, and hence proportional to η_c as well. Assuming a bounded variance around this shifted mean, we can rewrite the perturbed batch gradient $\widetilde{\mathbf{g}}_i^{e,t}$ as follows:

$$\widetilde{\mathbf{g}}_i^{e,t} = \frac{\nabla f_i^{e,t}}{r_i^2} + \mathbf{\Delta}_i^{e,t} + \mathbf{\Sigma}_i^{e,t},$$

where $\mathbf{\Sigma}_i^{e,t} \in \mathbb{R}^{n \times m}$ is random with zero mean and has a bounded variance $\mathbb{E}[\|\mathbf{\Sigma}_i^{e,t}\|_F^2] \leq \sigma_{i,\tilde{g}}^2$, and the random matrices $\{\mathbf{\Sigma}_i^{e,t}\}_{i,t}$ are mutually independent across clients i and steps t within a round.

Throughout, $\mathcal{F}^e := \mathcal{F}^{e,0}$ denotes the σ-algebra of all randomness up to (not including) round e, so that $\mathbf{W}^e$ is $\mathcal{F}^e$-measurable. At each local step, client i performs the following update:

$$\mathbf{W}_i^{e,t+1} = \mathbf{W}_i^{e,t} - \eta_c r_i^2 \widetilde{\mathbf{g}}_i^{e,t}.$$

After T local steps, the server averages the local models:

$$\mathbf{W}^{e+1} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{W}_i^{e,T}.$$

We first rewrite the local update. Since

$$\widetilde{\mathbf{g}}_i^{e,t} = \frac{1}{r_i^2} \nabla f_i^{e,t} + \mathbf{\Delta}_i^{e,t} + \mathbf{\Sigma}_i^{e,t},$$

the update

$$\mathbf{W}_i^{e,t+1} = \mathbf{W}_i^{e,t} - \eta_c r_i^2 \widetilde{\mathbf{g}}_i^{e,t}$$

can be written as

$$\boxed{\mathbf{W}_i^{e,t+1} = \mathbf{W}_i^{e,t} - \eta_c \nabla f_i^{e,t} - \eta_c r_i^2 \mathbf{\Delta}_i^{e,t} - \eta_c r_i^2 \mathbf{\Sigma}_i^{e,t}.} \tag{30}$$

Step 1: Round update. Averaging the equality ebove over i and telescoping over $t = 0, \dots, T-1$ (using $\mathbf{W}_i^{e,0} = \mathbf{W}^e$),

$$\mathbf{W}^{e+1} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{W}_i^{e,T} = \mathbf{W}^e - \eta_c \sum_{t=0}^{T-1} \Big(\bar{g}^{e,t} + \bar{b}^{e,t} + \bar{\mathbf{\Sigma}}^{e,t} \Big), \tag{31}$$

where

$$\bar{g}^{e,t} := \frac{1}{n}\sum_{i=1}^{n} \nabla f_i^{e,t}, \qquad \bar{b}^{e,t} := \frac{1}{n}\sum_{i=1}^{n} r_i^2 \mathbf{\Delta}_i^{e,t}, \qquad \bar{\mathbf{\Sigma}}^{e,t} := \frac{1}{n}\sum_{i=1}^{n} r_i^2 \mathbf{\Sigma}_i^{e,t}.$$

Define

$$\mathbf{E}^e := \sum_{t=0}^{T-1} \big(\bar{g}^{e,t} - \nabla f(\mathbf{W}^e) \big), \qquad \mathbf{B}^e := \sum_{t=0}^{T-1} \bar{b}^{e,t}, \qquad \mathbf{N}^e := \sum_{t=0}^{T-1} \bar{\mathbf{\Sigma}}^{e,t}.$$

Then eq. (31) reads

$$\boxed{\mathbf{W}^{e+1} - \mathbf{W}^e = -\eta_c T \nabla f(\mathbf{W}^e) - \eta_c \big(\mathbf{E}^e + \mathbf{B}^e + \mathbf{N}^e \big).} \tag{32}$$

Step 2: Bounding client drift. Telescoping eq. (30) for a single client from step 0 to $t-1$,

$$\mathbf{W}_i^{e,t} - \mathbf{W}^e = -\eta_c \Big(\underbrace{\sum_{s=0}^{t-1} \nabla f_i^{e,s}}_{A_t} + r_i^2 \underbrace{\sum_{s=0}^{t-1} \mathbf{\Delta}_i^{e,s}}_{B_t} + r_i^2 \underbrace{\sum_{s=0}^{t-1} \mathbf{\Sigma}_i^{e,s}}_{C_t} \Big).$$

Bound on A_t***:*** By our assumptions, $\|\nabla f_i^{e,s}\|_{\mathrm{F}} \le \|\nabla f(\mathbf{W}_i^{e,s})\|_{\mathrm{F}} + \sigma_{het} \le L_0 + \sigma_{het}$ a.s., hence $\|A_t\|_{\mathrm{F}} \le t(L_0 + \sigma_{het})$ a.s.

Bound on B_t***:*** By our bounded variance assumption,

$$\mathbb{E}\big[\, \|g_i^{e,s}\|_{\mathrm{F}} \,\big] \le \sqrt{\mathbb{E}\big[\, \|g_i^{e,s}\|_{\mathrm{F}}^2 \big]} = \sqrt{\|\nabla f_i^{e,s}\|_{\mathrm{F}}^2 + \mathbb{E}\big[\, \|g_i^{e,s} - \nabla f_i^{e,s}\|_{\mathrm{F}}^2 \,\big]} \le \sqrt{L_0^2 + \sigma_{i,g}^2} \le L_0 + \sigma_{i,g}.$$

By the above inequality and the bound we derived on $\|\mathbf{\Delta}_i^{e,s}\|_{\mathrm{F}}$, we have $\|\mathbf{\Delta}_i^{e,s}\|_{\mathrm{F}} \le \gamma_{r_i} \sum_{\tau<s} \mathbb{E}\big[\, \|g_i^{e,\tau}\|_{\mathrm{F}} \,\big] \le \gamma_{r_i}(L_0 + \sigma_{i,g}) s$ a.s., so

$$\|B_t\|_{\mathrm{F}} \le \sum_{s=0}^{t-1} \gamma_{r_i}(L_0 + \sigma_{i,g})\, s = \gamma_{r_i}(L_0 + \sigma_{i,g}) \frac{t(t-1)}{2} \quad \text{a.s.} \tag{33}$$

Bound on C_t***:*** By assumption of $\mathbf{\Sigma}_i^{e,s}$ having a zero-mean with bounded variance $\sigma_{i,\tilde{g}}^2$ and their independence across s, the cross terms vanish in expectation, so

$$\mathbb{E}\big[\, \|C_t\|_{\mathrm{F}}^2 \,\big] = \sum_{s=0}^{t-1} \mathbb{E}\big[\, \|\mathbf{\Sigma}_i^{e,s}\|_{\mathrm{F}}^2 \,\big] \le t\, \sigma_{i,\tilde{g}}^2.$$

Combining the bounds on A_t, B_t and C_t using $\|x+y+z\|_{\mathrm{F}}^2 \le 3(\|x\|_{\mathrm{F}}^2 + \|y\|_{\mathrm{F}}^2 + \|z\|_{\mathrm{F}}^2)$ together with $t \le T$,

$$\mathbb{E}\Big[\big\|\mathbf{W}_i^{e,t} - \mathbf{W}^e\big\|_{\mathrm{F}}^2 \Big] \le 3\eta_c^2 \Big[T^2(L_0 + \sigma_{het})^2 + \frac{r_i^4 \gamma_{r_i}^2 (L_0 + \sigma_{i,g})^2 T^2 (T-1)^2}{4} + r_i^4 T \sigma_{i,\tilde{g}}^2 \Big] =: \hat{D}_i^2.$$

Step 3: Bounding $\mathbf{E}^e$**.** By the β-smoothness assumption and Jensen's inequality,

$$\begin{aligned} \big\|\bar{g}^{e,t} - \nabla f(\mathbf{W}^e)\big\|_{\mathrm{F}}^2 &= \Big\| \frac{1}{n} \sum_i \big(\nabla f_i(\mathbf{W}_i^{e,t}) - \nabla f_i(\mathbf{W}^e) \big) \Big\|_{\mathrm{F}}^2 \\ &\le \frac{1}{n} \sum_i \big\| \nabla f_i(\mathbf{W}_i^{e,t}) - \nabla f_i(\mathbf{W}^e) \big\|_{\mathrm{F}}^2 \\ &\le \frac{\beta^2}{n} \sum_i \big\| \mathbf{W}_i^{e,t} - \mathbf{W}^e \big\|_{\mathrm{F}}^2 . \end{aligned} \tag{34}$$

Taking expectation of eq. (34) and using the previous upper bound $\hat{D}_i^2$:

$$\mathbb{E}\big[\,\big\|\bar{g}^{e,t} - \nabla f(\mathbf{W}^e)\big\|_{\mathrm{F}}^2\,\big] \le \beta^2 \hat{\bar{D}}^2, \qquad \hat{\bar{D}}^2 := \frac{1}{n}\sum_i \hat{D}_i^2.$$

Since $\mathbf{E}^e$ is a sum of T such terms, by Cauchy–Schwarz,

$$\mathbb{E}\big[\,\|\mathbf{E}^e\|_{\mathrm{F}}^2\,\big] \le T\sum_{t=0}^{T-1}\mathbb{E}\Big[\,\big\|\bar{g}^{e,t} - \nabla f(\mathbf{W}^e)\big\|_{\mathrm{F}}^2\Big] \le T^2\beta^2\hat{\bar{D}}^2 =: \epsilon_1^2.$$

Step 4: Bounding $\mathbf{B}^e$. By the bound derived before eq. (33), we have $\big\|\mathbf{\Delta}_i^{e,t}\big\|_{\mathrm{F}} \le \gamma_{r_i}(L_0 + \sigma_{i,g})\,t$ a.s. Hence, by the triangle inequality,

$$\|\mathbf{B}^e\|_{\mathrm{F}} \le \sum_{t=0}^{T-1}\frac{1}{n}\sum_i r_i^2\gamma_{r_i}(L_0+\sigma_{i,g})\,t = \Gamma\sum_{t=0}^{T-1} t = \Gamma\,\frac{T(T-1)}{2} =: \epsilon_2 \qquad \text{a.s.},$$

where $\Gamma = \frac{1}{n}\sum_i r_i^2\gamma_{r_i}(L_0+\sigma_{i,g})$.

Step 5: Bounding $\mathbf{N}^e$. By our assumptions on $\mathbf{\Sigma}_i^{e,t}$, $\mathbb{E}[\bar{\mathbf{\Sigma}}^{e,t} \mid \mathcal{F}^{e,t}] = 0$, so $\mathbb{E}[\mathbf{N}^e \mid \mathcal{F}^e] = 0$. By the assumption of independence of $\mathbf{\Sigma}_i^{e,t}$ across i and t), cross terms vanish and

$$\mathbb{E}\big[\,\|\mathbf{N}^e\|_{\mathrm{F}}^2\,\big] = \sum_{t=0}^{T-1}\mathbb{E}\big[\,\big\|\bar{\mathbf{\Sigma}}^{e,t}\big\|_{\mathrm{F}}^2\,\big] = \sum_{t=0}^{T-1}\frac{1}{n^2}\sum_i r_i^4\,\mathbb{E}\big[\,\big\|\mathbf{\Sigma}_i^{e,t}\big\|_{\mathrm{F}}^2\,\big] \le T\cdot\frac{1}{n^2}\sum_i r_i^4\sigma_{i,\tilde{g}}^2 = TS^2 =: \epsilon_3^2,$$

where $S^2 := \frac{1}{n^2}\sum_{i=1}^n r_i^4\sigma_{i,\tilde{g}}^2$.

Step 6: One-round descent. Using the smoothness inequality with $\mathbf{W}^e$ and $\mathbf{W}^{e+1}$, and eq. (32):

$$f(\mathbf{W}^{e+1}) \le f(\mathbf{W}^e) - \eta_c T\langle\nabla f(\mathbf{W}^e),\,\nabla f(\mathbf{W}^e)\rangle_{\mathrm{F}} - \eta_c\langle\nabla f(\mathbf{W}^e),\,\mathbf{E}^e + \mathbf{B}^e + \mathbf{N}^e\rangle_{\mathrm{F}}$$
$$+\frac{\beta}{2}\eta_c^2\,\|T\nabla f(\mathbf{W}^e) + \mathbf{E}^e + \mathbf{B}^e + \mathbf{N}^e\|_{\mathrm{F}}^2\,.$$

Take $\mathbb{E}[\cdot \mid \mathcal{F}^e]$. For the linear term, since $\mathbb{E}[\mathbf{N}^e \mid \mathcal{F}^e] = 0$,

$$-\eta_c\,\mathbb{E}\big[\langle\nabla f(\mathbf{W}^e),\,\mathbf{E}^e + \mathbf{B}^e + \mathbf{N}^e\rangle_{\mathrm{F}}\,\big|\,\mathcal{F}^e\big] = -\eta_c\langle\nabla f(\mathbf{W}^e),\,\mathbb{E}[\mathbf{E}^e \mid \mathcal{F}^e] + \mathbf{B}^e\rangle_{\mathrm{F}}$$
$$\le \eta_c L_0\big(\mathbb{E}[\|\mathbf{E}^e\|_{\mathrm{F}} \mid \mathcal{F}^e] + \epsilon_2\big),$$

using Lipschitz continuity assumption, Cauchy–Schwarz, and the bound derived in Step 4. By Jensen and the bound derived in Step 3, $\mathbb{E}[\|\mathbf{E}^e\|_{\mathrm{F}} \mid \mathcal{F}^e] \le \epsilon_1$ (taking full expectation later). For the quadratic term, using $\|a+b+c+d\|_{\mathrm{F}}^2 \le 4(\|a\|_{\mathrm{F}}^2 + \|b\|_{\mathrm{F}}^2 + \|c\|_{\mathrm{F}}^2 + \|d\|_{\mathrm{F}}^2)$,

$$\mathbb{E}\big[\,\|T\nabla f(\mathbf{W}^e) + \mathbf{E}^e + \mathbf{B}^e + \mathbf{N}^e\|_{\mathrm{F}}^2\,\big|\,\mathcal{F}^e\big] \le 4T^2\,\|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2 + 4\big(\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2\big).$$

Combining and taking full expectation,

$$\mathbb{E}\big[f(\mathbf{W}^{e+1})\big] \le \mathbb{E}\big[f(\mathbf{W}^e)\big] - \eta_c T\,\mathbb{E}\big[\,\|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2\,\big] + \eta_c L_0(\epsilon_1+\epsilon_2)$$
$$+2\beta\eta_c^2T^2\,\mathbb{E}\big[\,\|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2\,\big] + 2\beta\eta_c^2(\epsilon_1^2+\epsilon_2^2+\epsilon_3^2).$$

Rearranging the above, we get

$$\eta_c T\big(1 - 2\beta\eta_c T\big)\,\mathbb{E}\big[\,\|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2\,\big] \le \mathbb{E}\big[f(\mathbf{W}^e)\big] - \mathbb{E}\big[f(\mathbf{W}^{e+1})\big] + \eta_c L_0(\epsilon_1+\epsilon_2)$$
$$+2\beta\eta_c^2(\epsilon_1^2+\epsilon_2^2+\epsilon_3^2).$$

Step 7: Telescoping. Since $\eta_c \leq \frac{1}{4\beta T}$, we have $1 - 2\beta\eta_c T \geq \frac{1}{2}$, so from above,

$$\frac{\eta_c T}{2}\,\mathbb{E}\big[\,\|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2\,\big] \leq \mathbb{E}\big[f(\mathbf{W}^e)\big] - \mathbb{E}\big[f(\mathbf{W}^{e+1})\big] + \eta_c L_0(\epsilon_1+\epsilon_2) + 2\beta\eta_c^2(\epsilon_1^2+\epsilon_2^2+\epsilon_3^2).$$

Summing over $e = 0, \ldots, E-1$, the right-hand side telescopes:

$$\frac{\eta_c T}{2}\sum_{e=0}^{E-1}\mathbb{E}\big[\,\|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2\,\big] \leq f(\mathbf{W}^0) - \mathbb{E}\big[f(\mathbf{W}^E)\big] + E\eta_c L_0(\epsilon_1+\epsilon_2) + 2E\beta\eta_c^2(\epsilon_1^2+\epsilon_2^2+\epsilon_3^2).$$

Using $\mathbb{E}\big[f(\mathbf{W}^E)\big] \geq f^\star$ and dividing by $\eta_c TE/2$,

$$\frac{1}{E}\sum_{e=0}^{E-1}\mathbb{E}\big[\,\|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2\,\big] \leq \frac{2\big(f(\mathbf{W}^0)-f^\star\big)}{\eta_c TE} + \frac{2L_0(\epsilon_1+\epsilon_2)}{T} + \frac{4\beta\eta_c(\epsilon_1^2+\epsilon_2^2+\epsilon_3^2)}{T}.$$

We observe that $\epsilon_1 = O(\eta_c)$ (each $\hat{D}_i = O(\eta_c)$). Also, substituting $\epsilon_2 = \frac{T(T-1)}{2}\Gamma$ so that $\frac{2L_0\epsilon_2}{T} = L_0(T-1)\Gamma$, we observe that $\epsilon_2 = O(\eta_c)$ ($\Gamma \propto \sum_i r_i^2\gamma_{r_i}$, where $\gamma_{r_i} \propto \eta_{r_i}$ (from eq. (27)), and $\eta_{r_i} = \eta_c r_i^2$) and $\epsilon_3^2 = O(1)$ independent of η_c. Therefore,

$$\boxed{\min_e \mathbb{E}\big[\,\|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2\,\big] \leq \frac{2\big(f(\mathbf{W}^0)-f^\star\big)}{\eta_c TE} + \Psi}$$

where $\Psi = \mathcal{O}(\eta_c)$.

Corollary (rate). Choosing $\eta_c = \min\big(\frac{1}{4\beta T}, \frac{1}{\sqrt{E}}\big)$ makes $\frac{2(f(\mathbf{W}^0)-f^\star)}{\eta_c TE} = \mathcal{O}(1/\sqrt{E})$, and $\Psi = \mathcal{O}(\eta_c) = \mathcal{O}(1/\sqrt{E})$. Hence

$$\min_e \mathbb{E}\big[\,\|\nabla f(\mathbf{W}^e)\|_{\mathrm{F}}^2\,\big] = \mathcal{O}\left(\frac{1}{\sqrt{E}}\right),$$

proving the bound reduces to the classical $\mathcal{O}(1/\sqrt{E})$ non-convex FedAvg convergence rate.

□