FRAQ: Efficient Core-Space Recompression for Federated Low-Rank Adaptation

Shenghui Li[1] **and Thiemo Voigt**[1,2]
[1]Uppsala University, Uppsala, Sweden
[2]Research Institutes of Sweden, Stockholm, Sweden
{shenghui.li,thiemo.voigt}@angstrom.uu.se

Abstract

Federated Low-Rank Adaptation (LoRA) enables parameter-efficient fine-tuning of large language models without sharing local data, but aggregating LoRA adapters is fundamentally different from aggregating full model updates. Directly averaging LoRA factors introduces cross-client approximation errors, while recovering the exact averaged update and recompressing it with truncated SVD incurs substantial server-side cost. We propose FRAQ, an efficient core-space recompression method for federated LoRA aggregation. Instead of materializing the dense weight update, FRAQ represents the exact aggregate through weighted stacked LoRA factors, computes a compact orthonormal basis via thin QR factorization, and recovers the singular spectrum from a small Gram matrix. An energy threshold then selects a compact merge patch, allowing the server to control the trade-off between accuracy and downlink communication. Across federated fine-tuning benchmarks with homogeneous and heterogeneous client ranks, FRAQ achieves superior accuracy compared with existing aggregation methods, while substantially reducing downlink communication and server-side recompression cost.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks (Brown et al., 2020; Touvron et al., 2023; Grattafiori et al., 2024; Qwen et al., 2025). While fine-tuning further improves the adaptability of LLMs to downstream domains (Howard and Ruder, 2018), fully updating their massive parameter sets remains computationally expensive and susceptible to overfitting. Parameter-efficient fine-tuning (PEFT) methods alleviate these limitations by adapting LLMs with only a small number of trainable parameters, with Low-Rank Adaptation (LoRA) (Hu et al., 2022) being a widely adopted approach that introduces trainable low-rank updates to existing weight matrices.

When the data required for fine-tuning LLMs are distributed across edge devices or institutions, directly centralizing them is often infeasible due to privacy or regulatory constraints. Federated Learning (FL) (McMahan et al., 2017) provides a natural paradigm for collaborative adaptation without sharing local data. Combined with LoRA, clients only train and communicate compact low-rank adapters rather than full model updates, substantially reducing communication overhead. However, LoRA also makes aggregation more subtle: conventional FedAvg-style aggregation (Zhang et al., 2024) averages the down- and up-projection matrices independently, although the product of averaged factors generally differs from the average of the corresponding low-rank updates. This mismatch has been shown to introduce noisy or biased global updates and degrade fine-tuning performance (Sun et al., 2024; Singhal et al., 2025).

To address this mismatch, existing methods have moved beyond direct factor averaging. Some modify the local adaptation procedure by restricting which LoRA factor is updated or aggregated: FFA-LoRA (Sun et al., 2024) freezes one factor and trains only the other, while Fed-A^2 alternates the trainable factor across communication rounds. These strategies improve aggregation stability by reducing the degrees of freedom that vary across clients, but they also constrain local adaptation and do not directly recompress the exact update-level aggregate. Other methods pursue update-level aggregation more directly, either by merging residual corrections into the backbone (Singhal et al., 2025) or by stacking client-side LoRA factors to exactly represent the averaged update (Wang et al., 2024). While these approaches better preserve the aggregated update, they do not provide a compact rank-adaptive consolidation: residual merging lacks an explicit optimal recompression step,

whereas stacking increases the rank of the transmitted update and hence the downlink and local computation costs. A more principled solution is to recover the exact aggregated update and recompress it into a compact low-rank form (Bai et al., 2024). However, full-update SVD requires materializing and decomposing a dense weight matrix, and even recent compact-space variants still rely on repeated truncated SVD operations during aggregation (Ramesh and Dass, 2026). This leaves efficient rank-adaptive recompression as a key bottleneck for federated LoRA consolidation.

To fill this research gap, we propose FRAQ, a framework for Fast Rank Aggregation with QR factorization. FRAQ aggregates client LoRA residuals at the update level, represents the exact aggregate in a compact core space through thin QR factorization, and recovers its singular spectrum from a small Gram matrix. An energy threshold then selects a compact consolidation subspace, from which FRAQ forms a projection-based merge patch without reconstructing the dense update or explicitly recovering the right singular factors. Clients bake this compact patch into their cached global model and reinitialize fresh residual adapters for the next round, enabling communication-efficient global consolidation across training rounds.

Our main contributions are summarized as follows:

- We propose a core-space recompression pipeline for federated LoRA aggregation. By applying thin QR factorization to the weighted stacked adapters, FRAQ preserves the exact update-level aggregate before compression and recovers its spectrum from a compact Gram matrix, avoiding dense update construction and full-matrix SVD.
- We introduce a projection-form merge patch for rank-adaptive global consolidation. Instead of explicitly reconstructing right singular factors, FRAQ projects the exact aggregate onto an energy-selected core subspace, producing an optimal low-rank merge update with a tunable accuracy–communication trade-off.
- We conduct extensive experiments across datasets and models, showing that FRAQ achieves a favorable accuracy–communication–computation trade-off. Our layer-wise rank analysis further reveals substantial variation in intrinsic ranks across layers and attention projections, motivating adaptive low-rank consolidation.

2 Preliminaries and Motivation

Fine-tuning with LoRA. LoRA (Hu et al., 2022) represents the update of a pretrained weight matrix through a low-rank factorization. The fine-tuned weights $\boldsymbol{W}'$ are expressed as the sum of the frozen pretrained weights $\boldsymbol{W}_0$ and a low-rank update $\Delta\boldsymbol{W}$:

$$\boldsymbol{W}' = \boldsymbol{W}_0 + \Delta\boldsymbol{W} = \boldsymbol{W}_0 + \boldsymbol{B}\boldsymbol{A}, \qquad (1)$$

where $\boldsymbol{W}_0, \boldsymbol{W}' \in \mathbb{R}^{m\times n}$ are the pretrained and fine-tuned matrices, and $\boldsymbol{B} \in \mathbb{R}^{m\times r}$, $\boldsymbol{A} \in \mathbb{R}^{r\times n}$ are the low-rank factors of $\Delta\boldsymbol{W}$ with rank $r \ll \min(m, n)$. Instead of updating $\boldsymbol{W}_0$ directly, LoRA trains only the small factors $\boldsymbol{B}$ and $\boldsymbol{A}$, reducing the number of trainable parameters from mn to $r(m + n)$.

Federated fine-tuning. Federated learning (McMahan et al., 2017) trains a shared model across a set $\mathcal{C}$ of K clients holding private, potentially non-IID data, coordinated by a central server over T communication rounds. Under FedAvg, the server aggregates local updates by a data-weighted average; with n_k samples at client k, $N = \sum_k n_k$, and weight $\alpha_k = n_k/N$, the global update is

$$\overline{\Delta\boldsymbol{W}} = \sum_{k=1}^{K} \alpha_k \, \Delta\boldsymbol{W}_k, \qquad (2)$$

where $\Delta\boldsymbol{W}_k$ is the local update from client k. FedIT (Zhang et al., 2024) extends this scheme to LoRA: each client k fine-tunes local adapters $(\boldsymbol{B}_k, \boldsymbol{A}_k)$, and the server averages the factors independently,

$$\overline{\boldsymbol{B}} = \sum_{k=1}^{K} \alpha_k \, \boldsymbol{B}_k, \qquad \overline{\boldsymbol{A}} = \sum_{k=1}^{K} \alpha_k \, \boldsymbol{A}_k. \qquad (3)$$

Inexactness of LoRA aggregation. The difficulty is that aggregating adapters is not equivalent to aggregating their updates. Because each client's contribution is the *product* $\boldsymbol{B}_k\boldsymbol{A}_k$ rather than the individual factors, independently averaging $\boldsymbol{B}_k$ and

$\boldsymbol{A}_k$ does not recover the exact aggregated update:

$$\begin{aligned}\underbrace{\overline{\boldsymbol{B}}\,\overline{\boldsymbol{A}} = \sum_{i,j} \alpha_i \alpha_j \, \boldsymbol{B}_i \boldsymbol{A}_j}_{\text{FedIT (inexact)}} \\ \neq \underbrace{\sum_{k=1}^{K} \alpha_k \, \boldsymbol{B}_k \boldsymbol{A}_k = \overline{\Delta \boldsymbol{W}}}_{\text{exact aggregation}}.\end{aligned} \tag{4}$$

The discrepancy stems from the cross terms $\boldsymbol{B}_i \boldsymbol{A}_j$ with $i \neq j$, i.e., *the average of the products is not equal to the product of the averages*. A naive remedy is to materialize the exact aggregate $\overline{\Delta \boldsymbol{W}} \in \mathbb{R}^{m \times n}$ in Eq. (4) and re-decompose it with a truncated SVD, but this discards the low-rank structure that makes LoRA efficient and incurs a costly full-matrix decomposition on the server. Our method computes the rank-truncated form of the exact aggregate $\overline{\Delta \boldsymbol{W}}$ directly in a compact core space, avoiding both the cross-term noise of Eq. (4) and the cost of full-matrix SVD.

3 Proposed Method

We propose FRAQ, a core-space recompression method that turns the adapter-induced weight updates into a compact global merge patch. Rather than applying SVD to a dense weight update, FRAQ keeps the aggregate in factorized form, computes a QR-based core representation, and recovers its singular spectrum from a small Gram matrix. An energy threshold then selects the rank of the merge patch to control the trade-off between accuracy and communication.

Problem setup and exact weighted stacking. At the start of round t, the server holds a merged global model $\boldsymbol{W}_t$, obtained by accumulating all previous merge patches into the pretrained weights $\boldsymbol{W}_0$. Each selected client k fine-tunes a temporary low-rank residual adapter $(\boldsymbol{B}_k, \boldsymbol{A}_k)$ of rank r_k on top of $\boldsymbol{W}_t$, where

$$\boldsymbol{B}_k \in \mathbb{R}^{m \times r_k}, \qquad \boldsymbol{A}_k \in \mathbb{R}^{r_k \times n}, \tag{5}$$

and different clients may use heterogeneous ranks r_k. Let $\alpha_k = n_k / N$ be the aggregation weight. The server aggregates the weight updates represented by the adapters, i.e.,

$$\overline{\Delta \boldsymbol{W}} = \sum_{k=1}^{K} \alpha_k \boldsymbol{B}_k \boldsymbol{A}_k. \tag{6}$$

Instead of averaging $\boldsymbol{B}_k$ and $\boldsymbol{A}_k$ separately, which introduces cross terms (Sun et al., 2024), the server constructs weighted stacked factors

$$\boldsymbol{B} = [\sqrt{\alpha_1}\boldsymbol{B}_1, \sqrt{\alpha_2}\boldsymbol{B}_2, \ldots, \sqrt{\alpha_K}\boldsymbol{B}_K] \in \mathbb{R}^{m \times r}, \tag{7}$$

$$\boldsymbol{A} = \begin{bmatrix} \sqrt{\alpha_1}\boldsymbol{A}_1 \\ \sqrt{\alpha_2}\boldsymbol{A}_2 \\ \vdots \\ \sqrt{\alpha_K}\boldsymbol{A}_K \end{bmatrix} \in \mathbb{R}^{r \times n}, \qquad r = \sum_{k=1}^{K} r_k. \tag{8}$$

Then the aggregated update is represented exactly as

$$\overline{\Delta \boldsymbol{W}} = \boldsymbol{B}\boldsymbol{A}. \tag{9}$$

Here r is the total client LoRA rank in the current round. In federated LoRA, r is typically much smaller than m and n, so the useful aggregation information lives in the union of the client low-rank subspaces rather than in the full $m \times n$ matrix.

Efficient recompression in the core space. A naive server can materialize $\overline{\Delta \boldsymbol{W}} \in \mathbb{R}^{m \times n}$ and compute a rank-p SVD of the full matrix. This is unnecessary. FRAQ first computes a thin QR factorization of the stacked left factor:

$$\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{R}, \qquad \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}, \tag{10}$$

where $\boldsymbol{Q} \in \mathbb{R}^{m \times q}$ spans the left client subspace, $\boldsymbol{R} \in \mathbb{R}^{q \times r}$, and $q = \operatorname{rank}(\boldsymbol{B}) \leq r$. The aggregated update becomes

$$\begin{aligned}\overline{\Delta \boldsymbol{W}} = \boldsymbol{B}\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{H}, \\ \boldsymbol{H} = \boldsymbol{R}\boldsymbol{A} \in \mathbb{R}^{q \times n}.\end{aligned} \tag{11}$$

Since $\boldsymbol{Q}$ has orthonormal columns, the non-zero singular values of $\overline{\Delta \boldsymbol{W}}$ are exactly the singular values of the compact core matrix $\boldsymbol{H}$. Rather than computing an SVD of $\boldsymbol{H}$ directly, FRAQ forms the small Gram matrix

$$\boldsymbol{G} = \boldsymbol{H}\boldsymbol{H}^\top \in \mathbb{R}^{q \times q}. \tag{12}$$

Let the full eigendecomposition of $\boldsymbol{G}$ be

$$\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^\top, \qquad \boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_q), \tag{13}$$

whose eigenvalues σ_i^2 are exactly the squared singular values of $\boldsymbol{H}$, and hence of the exact aggregated update $\overline{\Delta \boldsymbol{W}}$. The eigenvectors $\boldsymbol{U}$ are the left singular vectors of $\boldsymbol{H}$ in the core space, and the corresponding right factors are recovered implicitly through $\boldsymbol{U}^\top \boldsymbol{H} = \boldsymbol{\Sigma}\boldsymbol{V}^\top$, so $\overline{\Delta \boldsymbol{W}} = \boldsymbol{Q}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is an SVD of the exact aggregated update up to

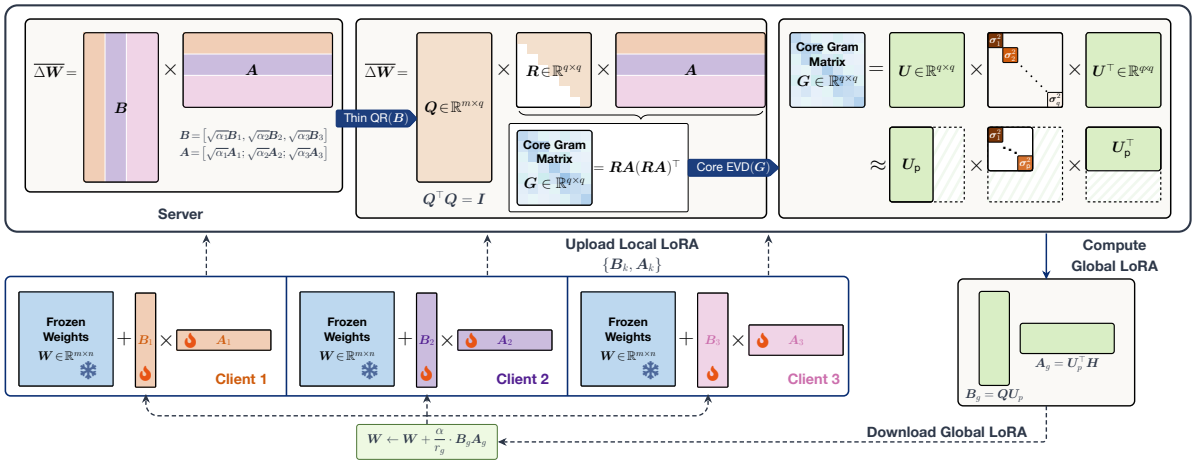

Figure 1: Workflow of FRAQ. The server aggregates client residual adapters in a compact core space without reconstructing the dense update matrix. It forms a weighted stacked representation, computes a thin QR basis $\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{R}$ for the left client subspace, obtains the core matrix $\boldsymbol{H} = \boldsymbol{R}\boldsymbol{A}$, and recovers the exact singular spectrum from the small Gram matrix $\boldsymbol{G} = \boldsymbol{H}\boldsymbol{H}^\top$. An energy retention threshold τ selects a rank-p core subspace $\boldsymbol{U}_p$, and FRAQ projects the update onto it to form the compact merge patch $(\boldsymbol{B}_g, \boldsymbol{A}_g) = (\boldsymbol{Q}\boldsymbol{U}_p,\ \boldsymbol{U}_p^\top \boldsymbol{H})$, which clients bake into the global model before reinitializing fresh local adapters.

zero components. Crucially, the entire singular spectrum is obtained from a $q \times q$ eigenproblem, without ever forming $\overline{\Delta \boldsymbol{W}}$ or running an SVD on $\boldsymbol{H}$.

Energy-based rank selection and projection-form merge patch. The core spectrum exposes how much of the aggregated update each direction captures, so FRAQ selects the smallest rank p that preserves at least a τ fraction of the spectral energy:

$$\frac{\sum_{i=1}^{p} \sigma_i^2}{\sum_{i=1}^{q} \sigma_i^2} \geq \tau. \tag{14}$$

Let $\boldsymbol{U}_p$ collect the top-p eigenvectors of $\boldsymbol{G}$. Instead of explicitly reconstructing the right factors $\boldsymbol{V}_p^\top = \boldsymbol{\Sigma}_p^{-1} \boldsymbol{U}_p^\top \boldsymbol{H}$, FRAQ writes the rank-$p$ truncation of the exact update as the orthogonal projection of $\overline{\Delta \boldsymbol{W}}$ onto the retained core subspace,

$$\widehat{\Delta \boldsymbol{W}} = \boldsymbol{Q}\, \boldsymbol{U}_p \boldsymbol{U}_p^\top\, \boldsymbol{H}, \tag{15}$$

and broadcasts the corresponding compact merge patch

$$\begin{aligned} \boldsymbol{B}_g &= \boldsymbol{Q}\boldsymbol{U}_p \in \mathbb{R}^{m \times p}, \\ \boldsymbol{A}_g &= \boldsymbol{U}_p^\top \boldsymbol{H} \in \mathbb{R}^{p \times n}. \end{aligned} \tag{16}$$

Since $\boldsymbol{U}_p^\top \boldsymbol{H} = \boldsymbol{\Sigma}_p \boldsymbol{V}_p^\top$, the patch satisfies $\boldsymbol{B}_g \boldsymbol{A}_g = \boldsymbol{Q}\boldsymbol{U}_p \boldsymbol{\Sigma}_p \boldsymbol{V}_p^\top$, i.e., it equals the rank-p SVD truncation of the exact aggregated update, yet it is obtained without inverting $\boldsymbol{\Sigma}_p$ or forming $\boldsymbol{V}_p$. This projection form is both numerically more stable, as it never divides by small singular values, and simpler to implement, as it skips the right-factor reconstruction.

Merge and reinitialize. The energy retention threshold τ turns rank selection into a server-side orchestration knob: clients train with heterogeneous local ranks r_k, while the consolidation rank p is determined by the observed spectrum of the exact aggregate. Each client merges the broadcast patch into its cached global model,

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \boldsymbol{B}_g \boldsymbol{A}_g, \tag{17}$$

discards the patch, and reinitializes a fresh residual LoRA adapter at its own rank r_k for the next round. FRAQ is therefore a merge-and-reinitialize scheme, like FLoRA, but it broadcasts a single compact rank-p patch instead of stacked high-rank adapters. This decouples the global consolidation rank p from the local adaptation ranks $\{r_k\}$, so neither truncation nor zero-padding of a shared global adapter is needed to reconcile mismatched ranks.

Theoretical analysis. The core-space reduction is lossless before rank truncation. From $\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{R}$ and $\boldsymbol{H} = \boldsymbol{R}\boldsymbol{A}$, we have $\overline{\Delta \boldsymbol{W}} = \boldsymbol{Q}\boldsymbol{H}$. Because $\boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}$, left multiplication by $\boldsymbol{Q}$ preserves singular values and Frobenius norms on the column space of $\boldsymbol{Q}$. Therefore, if $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, then

$$\overline{\Delta \boldsymbol{W}} = \boldsymbol{Q}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \tag{18}$$

is an SVD of the exact aggregated update up to zero singular components, and $\boldsymbol{G} = \boldsymbol{H}\boldsymbol{H}^\top =$

Algorithm 1: FRAQ: Core-Space Aggregation with Projection-Form Merge

Input: Pretrained weights $\boldsymbol{W}_0$, rounds T, clients $\mathcal{C}$ with ranks $\{r_k\}$, dataset sizes $\{n_k\}$, energy retention threshold τ

Output: Merged global model $\boldsymbol{W}_T$

$\boldsymbol{W}_1 \leftarrow \boldsymbol{W}_0$

for $t = 1$ **to** T **do**

  Server: sample clients $\mathcal{C}^t \subset \mathcal{C}$ and broadcast the latest merge patch (empty when $t = 1$)

  foreach $k \in \mathcal{C}^t$ **do in parallel**

    Bake the broadcast patch into the cached global model $\boldsymbol{W}_t$

    Initialize a fresh residual adapter $(\boldsymbol{B}_k, \boldsymbol{A}_k)$ of rank r_k

    $(\boldsymbol{B}_k, \boldsymbol{A}_k) \leftarrow$ `LocalUpdate`$(\boldsymbol{W}_t, \boldsymbol{B}_k, \boldsymbol{A}_k)$

    `Upload`$(\boldsymbol{B}_k, \boldsymbol{A}_k)$ to server

  Server: form weighted stacks $\boldsymbol{B} \leftarrow [\sqrt{\alpha_1}\boldsymbol{B}_1, \ldots, \sqrt{\alpha_K}\boldsymbol{B}_K]$ and $\boldsymbol{A} \leftarrow [\sqrt{\alpha_1}\boldsymbol{A}_1; \ldots; \sqrt{\alpha_K}\boldsymbol{A}_K]$

  Compute thin QR: $\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{R}$

  Form the core matrix $\boldsymbol{H} \leftarrow \boldsymbol{R}\boldsymbol{A}$

  Form the Gram matrix $\boldsymbol{G} \leftarrow \boldsymbol{H}\boldsymbol{H}^\top$

  Eigendecompose $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^\top$

  Select smallest p with $\sum_{i=1}^{p} \sigma_i^2 / \sum_i \sigma_i^2 \geq \tau$; take top-$p$ vectors $\boldsymbol{U}_p$

  $\boldsymbol{B}_g \leftarrow \boldsymbol{Q}\boldsymbol{U}_p, \quad \boldsymbol{A}_g \leftarrow \boldsymbol{U}_p^\top \boldsymbol{H}$

  $\boldsymbol{W}_{t+1} \leftarrow \boldsymbol{W}_t + \boldsymbol{B}_g\boldsymbol{A}_g$

return $\boldsymbol{W}_T$

$\boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^\top$ recovers the same left singular vectors and singular values in the core space.

The projection-form patch is exactly the rank-p SVD truncation. Since $\boldsymbol{U}_p^\top \boldsymbol{H} = \boldsymbol{\Sigma}_p \boldsymbol{V}_p^\top$, the broadcast patch satisfies $\boldsymbol{B}_g\boldsymbol{A}_g = \boldsymbol{Q}\boldsymbol{U}_p\boldsymbol{U}_p^\top \boldsymbol{H} = \boldsymbol{Q}\boldsymbol{U}_p\boldsymbol{\Sigma}_p\boldsymbol{V}_p^\top$, so by the Eckart–Young–Mirsky theorem (Golub et al., 1987),

$$\|\overline{\Delta \boldsymbol{W}} - \boldsymbol{B}_g\boldsymbol{A}_g\|_F = \left(\sum_{i=p+1}^{q} \sigma_i^2 \right)^{1/2}, \quad (19)$$

where σ_i are the singular values obtained from $\boldsymbol{G}$. This is the smallest achievable rank-p error: FRAQ is exact when all non-zero core components are retained, and gives the optimal rank-p approximation of the exact aggregated update under the selected energy retention threshold τ.

Complexity analysis. Let $r = \sum_k r_k$ and $q = \operatorname{rank}(\boldsymbol{B}) \leq r$, with $r \ll \min(m, n)$ in typical LoRA settings. For each layer, FRAQ computes the thin QR factorization of $\boldsymbol{B} \in \mathbb{R}^{m \times r}$, forms $\boldsymbol{H} = \boldsymbol{R}\boldsymbol{A} \in \mathbb{R}^{q \times n}$, builds $\boldsymbol{G} = \boldsymbol{H}\boldsymbol{H}^\top \in \mathbb{R}^{q \times q}$, and decomposes $\boldsymbol{G}$. The dominant server-side cost is

$$\mathcal{O}\left(mr^2 + nqr + nq^2 + q^3\right), \quad (20)$$

which reduces to $\mathcal{O}((m+n)r^2 + r^3)$ when $q \approx r$. Crucially, the decomposition is performed in a $q \times q$ core space rather than on the full $\overline{\Delta \boldsymbol{W}} \in \mathbb{R}^{m \times n}$. Forming the projection-form patch $\boldsymbol{B}_g = \boldsymbol{Q}\boldsymbol{U}_p$, $\boldsymbol{A}_g = \boldsymbol{U}_p^\top \boldsymbol{H}$ adds only two thin matrix products and avoids the singular-value inversion $\boldsymbol{\Sigma}_p^{-1}\boldsymbol{U}_p^\top \boldsymbol{H}$ otherwise needed to reconstruct the right factors, improving numerical stability at no extra asymptotic cost. Across L layers, the cost scales linearly with the number of LoRA layers, while the broadcast cost depends on the selected ranks $\{p_l\}_{l=1}^{L}$ rather than the sum of all client ranks.

4 Experiments

In this section, we evaluate the performance of our algorithm against existing FL methods combined with LoRA across various heterogeneity settings and datasets. We assess performance based on accuracy and the total number of downloaded parameters.

4.1 Experimental Setup

Datasets and configurations. We mainly adopt pre-trained RoBERTa-base (Liu et al., 2019b) as the base model for fine-tuning, following the federated LoRA evaluation setting with heterogeneous clients (Koo et al., 2025). RoBERTa-base has approximately 125M parameters, which remain frozen during LoRA fine-tuning. For fine-tuning, we use four text classification datasets: BANKING77 (Casanueva et al., 2020), 20 Newsgroups (Lang, 1995), CLINC150 (Larson et al., 2019), and HWU64 (Liu et al., 2019a). BANKING77 and HWU64 are fine-grained intent classification datasets, CLINC150 is a multi-domain intent classification benchmark with out-of-scope queries, and 20 Newsgroups is a topic classification benchmark. These datasets allow us to simulate controlled data heterogeneity across clients using Dirichlet distribution-based partitions. Our federated setup consists of 100 clients, with 10 clients

Table 1: Performance of federated LoRA aggregation methods. **(a)** Test accuracy (%) on four intent-classification datasets under two label-heterogeneity levels, $\mathrm{Dir}(0.1)$ and $\mathrm{Dir}(0.02)$, together with downlink (normalized by FLoRA % and averaged across the two heterogeneity levels) and server aggregation time. **(b)** Accuracy (%) on eight reasoning benchmarks under the additional evaluation setting, together with downlink and server aggregation time. Higher accuracy and lower downlink or aggregation cost are better; best and second-best accuracy results among methods with compressed downlink are shown in **bold** and underline, respectively, while FLoRA and FedEx-LoRA are excluded from ranking.

(a) Intent classification: test accuracy.

Method	Accuracy								Downlink (% FLoRA)	Avg. Agg. Time (ms)
	20 Newsgroups		Banking77		CLINC150		HWU64			
	Dir(0.1)	Dir(0.02)	Dir(0.1)	Dir(0.02)	Dir(0.1)	Dir(0.02)	Dir(0.1)	Dir(0.02)		
FLoRA	64.30	56.80	83.30	73.90	92.60	84.80	87.30	80.10	100.00	4.21
FedEx-LoRA	63.30	52.60	75.30	68.70	86.90	81.50	80.50	73.00	103.33	12.25
FedIT	63.80	48.90	81.00	63.10	90.00	70.00	84.50	70.80	3.33	1.46
LoRA-A^2	63.10	55.30	80.90	66.40	90.20	71.70	87.10	74.50	3.33	0.74
FlexLoRA	**65.20**	49.20	**84.00**	69.70	91.80	76.90	**87.60**	74.80	3.33	3940.75
FLoRIST	61.50	49.20	73.30	55.50	80.80	62.30	77.80	64.60	0.82	231.4
FraQ ($\tau = 0.80$)	63.90	**57.10**	80.60	70.70	91.40	81.10	84.20	76.40	15.35	67.2
FraQ ($\tau = 0.98$)	64.30	54.70	82.80	**73.40**	**92.30**	**84.40**	87.10	**79.60**	58.29	66.8

(b) Reasoning benchmarks.

Method	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-E	ARC-C	OBQA	Avg.	Downlink (% FLoRA)	Avg. Agg. Time (ms)
FLoRA	88.78	79.49	78.45	85.34	77.27	89.73	78.24	81.60	82.36	100.00	17.78
FedEx-LoRA	86.79	75.35	73.95	34.23	60.54	79.46	67.49	74.80	69.08	112.51	50.13
FedIT	87.31	78.67	76.82	79.64	73.64	89.86	77.73	80.40	80.51	12.51	9.58
LoRA-A^2	85.84	71.49	72.67	55.23	64.09	81.27	63.99	74.00	71.07	12.51	4.72
FlexLoRA	87.86	79.49	76.41	81.18	75.85	89.86	**78.41**	80.40	81.18	12.51	197270.37
FLoRG	86.97	77.42	75.13	74.50	71.19	87.37	74.32	79.60	78.31	4.01	6999.40
FLoRIST	85.66	77.48	73.13	70.25	71.03	88.43	75.51	76.40	77.24	12.51	730.01
FraQ ($\tau = 0.80$)	88.47	**80.03**	77.02	82.19	76.24	89.18	76.45	**82.00**	81.45	40.01	108.39
FraQ ($\tau = 0.98$)	**89.11**	79.65	**77.64**	**84.26**	**77.98**	**90.28**	77.82	81.80	**82.32**	88.13	118.84

randomly sampled in each communication round. Consistent with prior works (Zhang et al., 2024; He et al., 2020; Lai et al., 2022), we use Dirichlet distribution-based non-IID partitions of the dataset, with a concentration parameter of $\alpha = 0.5$, to create local data for clients. All experiments are run for a total of 75 communication rounds to ensure convergence. In the homogeneous configuration, all clients use LoRA rank 16. In the heterogeneous configuration, we adopt an extreme heavy-tail-light distribution of client ranks with $16\times$ rank disparity ($4 \rightarrow 64$) to rigorously evaluate the model under a regime of high heterogeneity: 40 clients use rank 4, 20 clients use rank 8, 20 clients use rank 16, 10 clients use rank 32, and 10 clients use rank 64. This distribution reflects realistic variations in client capacity, where the majority of participants operate at lower ranks and a small fraction contributes higher-rank updates. Each communication round is followed by local fine-tuning with a learning rate of 0.0003. Training is conducted on a large-scale cluster equipped with NVIDIA H100 GPUs under both homogeneous and heterogeneous settings. We additionally evaluate on the commonsense reasoning benchmark suite used by FedEx-LoRA (Singhal et al., 2025), following the same Commonsense-170K fine-tuning and eight-task multiple-choice evaluation protocol; dataset details are provided in Appendix A.2.

Baselines. We compare our proposed FraQ method against the following related works: FedIT (Zhang et al., 2024) integrates LoRA with FedAvg and only supports homogeneous LoRA ranks across clients. It relies on zero-padding (HetLoRA (Cho et al., 2023)) to handle heterogeneity up to the maximum client rank. Zero-padding is used by both FedIT and FFA-LoRA to accommodate rank differences. FLoRA (Wang et al., 2024) is a stacking-based aggregation strategy with heterogeneous LoRA support. FlexLoRA (Bai et al.,

2024) redistributes SVD of full-weight update to create multiple global adapters to match the client's local rank. FFA-LoRA (Sun et al., 2024) freezes one of the LoRA adapters during training and only transmits the other half of the adapters. We refer to Appendix E for detailed descriptions of datasets, and baseline methods used for our experimental results.

Rank-selection settings. We report FRAQ under two energy-retention settings. **FRAQ** (τ=0.80) favors stronger downlink compression by retaining a smaller core subspace, while **FRAQ** (τ=0.98) retains more spectral energy to prioritize accuracy. These two settings expose the communication–accuracy trade-off of the same core-space recompression procedure.

Communication cost definition. Throughout this section, *communication cost* refers exclusively to the *download* cost, i.e., the number of LoRA parameters transmitted from the server to the selected clients in each communication round. Table 1 reports this cost as DOWNLINK normalized by FLoRA, so 100% corresponds to broadcasting the full stacked FLoRA update and smaller values indicate lower server-to-client communication. Since all compared methods communicate LoRA matrices whose parameter count scales linearly with rank, we compute the normalized downlink using the corresponding total download rank.

4.2 Performance Analysis

Accuracy under moderate heterogeneity (Dir(0.1)). Under the milder non-IID setting, all adapter-based methods perform comparably, and FRAQ ranks among the strongest compressed-downlink methods (Table 1). FRAQ (τ=0.98) attains the best accuracy on CLINC150 (92.30%) and the second-best on 20 Newsgroups (64.30%), Banking77 (82.80%), and HWU64 (87.10%), trailing the leader FlexLoRA by only about 0.5–1.2 points, while FlexLoRA leads on 20 Newsgroups (65.20%), Banking77 (84.00%), and HWU64 (87.60%). Importantly, FRAQ matches this accuracy while cutting server aggregation time from FlexLoRA's 3940.75 ms to 92.16 ms (a 42.8× speedup), since it avoids full-weight SVD on the server.

Accuracy under extreme heterogeneity (Dir(0.02)). The advantage of FRAQ becomes decisive under severe label skew. Among methods with compressed downlink, FRAQ achieves the best accuracy on *every* dataset: FRAQ (τ=0.80) leads on 20 Newsgroups (57.10%), and FRAQ (τ=0.98) leads on Banking77 (73.40%), CLINC150 (84.40%), and HWU64 (79.60%), with the second-best result also held by a FRAQ variant on three of the four datasets. Remarkably, FRAQ nearly closes the gap to the *uncompressed* FLoRA upper bound, staying within 0.5 points on CLINC150 (84.40 vs. 84.80), HWU64 (79.60 vs. 80.10), and Banking77 (73.40 vs. 73.90), and even surpassing it on 20 Newsgroups (57.10 vs. 56.80). In contrast, the remaining compressed baselines degrade sharply, e.g., FedIT (48.90% on 20 Newsgroups, 70.00% on CLINC150) and FLoRIST (55.50% on Banking77, 62.30% on CLINC150).

Communication–computation trade-off. The two retention settings expose a controllable trade-off. FRAQ (τ=0.80) compresses the downlink to roughly 15% of FLoRA while retaining strong accuracy, whereas FRAQ (τ=0.98) spends more downlink (∼57–59%) to maximize accuracy. While FLoRIST attains the smallest downlink (0.82% of FLoRA), it does so at a steep accuracy cost, especially under Dir(0.02). On the server side, the two FRAQ variants aggregate in 92.16–93.64 ms, roughly 2.8× faster than FLoRIST (261.36 ms) and more than 42× faster than FlexLoRA (3940.75 ms). Overall, FRAQ delivers the best accuracy among compressed-downlink methods while keeping both communication and server aggregation cost low, and its robustness under extreme heterogeneity makes it a reliable choice for federated fine-tuning.

5 Conclusion

We presented FRAQ, a core-space recompression framework for federated LoRA aggregation. FRAQ recovers the exact spectrum of the aggregated update through weighted stacking, thin QR factorization, and a compact Gram eigenproblem, avoiding dense update materialization and expensive full-matrix SVD. Its projection-form merge patch enables rank-adaptive downlink compression while preserving the optimal rank-p approximation of the exact aggregate. Experiments across various federated fine-tuning settings show that FRAQ achieves a strong accuracy–communication–computation trade-off compared with factor-averaging, stacking-based, and SVD-based baselines. These results

suggest that core-space recompression is a practical direction for scalable federated adaptation of large models.

Limitations

We note two limitations of FRAQ. First, the cost of core-space aggregation scales with the total stacked rank of the sampled clients in each round, that is, the sum of their local LoRA ranks. When the server samples a large cohort of clients per round, this total rank grows accordingly, and the thin QR factorization, the Gram-matrix construction, and the eigendecomposition all become more expensive. Consequently, the server-side aggregation efficiency degrades as the number of sampled clients increases, and the advantage of FRAQ over full-weight SVD shrinks in the regime of very large per-round participation. A detailed cost breakdown is given in the appendix.

Second, like FLoRA, FRAQ follows a merge-and-reinitialize scheme: in every round the consolidated update is baked into the global model and clients reinitialize fresh adapters, so local adapters are not warm-started from the previous round as in PEFT-style methods that reuse the downloaded adapter. This can slow per-adapter convergence relative to methods that keep refining a persistent adapter. Importantly, however, the core-space recompression at the heart of FRAQ is not tied to the merge-and-reinitialize paradigm. The same QR-and-Gram trick can be dropped into non-merge-and-reinitialize methods such as `FLoRIST` and FLoRG as a faster replacement for their SVD-based recompression, recovering the same aggregated update while running faster than their original procedures (see the appendix for details).

References

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. 2024. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Tom Brown and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Inigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45. Association for Computational Linguistics.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, Matt Barnes, and Gauri Joshi. 2023. Heterogeneous loRA for federated fine-tuning of on-device foundation models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Gene H Golub, Alan Hoffman, and Gilbert W Stewart. 1987. A generalization of the eckart-young-mirsky matrix approximation theorem. *Linear Algebra and its applications*, 88:317–327.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. 2020. Fedml: A research library and benchmark for federated machine learning. *CoRR*, abs/2007.13518.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023. 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 (EMNLP)*.

Jabin Koo, Minwoo Jang, and Jungseul Ok. 2025. 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)*, pages 416–429, Vienna, Austria. Association for Computational Linguistics.

Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. 2022. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International conference on machine learning*, pages 11814–11827. PMLR.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1311–1316. Association for Computational Linguistics.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019a. Benchmarking natural language understanding services for building conversational agents. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology*, Ortigia, Siracusa, Italy. Springer.

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

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Hariharan Ramesh and Jyotikrishna Dass. 2026. FLoRIST: Singular value thresholding for efficient and accurate federated fine-tuning of large language models. In *Ninth Conference on Machine Learning and Systems*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Raghav Singhal, Kaustubh Ponkshe, and Praneeth Vepakomma. 2025. FedEx-LoRA: Exact aggregation for federated and efficient fine-tuning of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1316–1336, Vienna, Austria. Association for Computational Linguistics.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. 2024. Improving lora in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Appendix

A Dataset Details

This appendix expands on the datasets used in our two evaluation settings: the federated intent/topic classification benchmark (Table 1a) and the commonsense reasoning benchmark used in the additional evaluation (Table 1b).

A.1 Intent and Topic Classification

For the federated RoBERTa-base experiments we use four English text classification datasets:

- **BANKING77** (Casanueva et al., 2020): a fine-grained single-domain intent detection dataset of online-banking customer queries spanning 77 intents.
- **CLINC150** (Larson et al., 2019): a multi-domain intent classification benchmark covering 150 intents across 10 domains, designed to also include out-of-scope queries.
- **HWU64** (Liu et al., 2019a): a multi-domain home-assistant intent dataset with 64 intents across 21 domains.
- **20 Newsgroups** (Lang, 1995): a classic topic classification benchmark of newsgroup posts grouped into 20 categories.

To emulate realistic client data heterogeneity, each dataset is partitioned across clients with a Dirichlet distribution $\mathrm{Dir}(\alpha)$ over the label space, where a smaller concentration parameter α yields a more skewed (non-IID) partition. We report results under a mild heterogeneity level $\mathrm{Dir}(0.1)$ and an extreme level $\mathrm{Dir}(0.02)$, and evaluate global test accuracy at communication round 49 (R49).

A.2 Commonsense Reasoning

For the additional evaluation in Table 1b, we adopt the commonsense reasoning protocol of Hu et al. (2023), which is *identical* to the setup used by FedEx-LoRA (Singhal et al., 2025): models are fine-tuned on the Commonsense-170K instruction-tuning corpus and evaluated on eight multiple-choice reasoning tasks. Each task is cast as a multiple-choice problem in which the model selects the correct answer from the provided options, and we report per-task accuracy together with the eight-task average. The benchmarks are:

- **BoolQ** (Clark et al., 2019): naturally occurring yes/no questions, each paired with a Wikipedia passage that provides the context.
- **PIQA** (Bisk et al., 2020): physical commonsense questions that require choosing the more sensible of two candidate solutions to an everyday physical goal.
- **SIQA** (Social IQa) (Sap et al., 2019): questions probing reasoning about people's social interactions and their likely motivations and consequences.
- **HellaSwag** (Zellers et al., 2019): sentence-completion questions in which the model selects the most plausible continuation of a given context.
- **WinoGrande** (Sakaguchi et al., 2021): a large-scale adversarial Winograd schema benchmark requiring binary pronoun/coreference resolution.
- **ARC-Easy (ARC-e)** and **ARC-Challenge (ARC-c)** (Clark et al., 2018): grade-school science questions split into an easier subset and a harder subset of questions that simple retrieval methods answer incorrectly.
- **OpenBookQA (OBQA)** (Mihaylov et al., 2018): elementary science questions that require combining a small "open book" of core science facts with broad commonsense knowledge.

Because our reasoning evaluation matches FedEx-LoRA exactly, the results in Table 1b are directly comparable to theirs.

B Hyperparameter Details

Table 2 lists the shared training and federated learning hyperparameters for the two decoder-only models used in our experiments. Unless otherwise noted, we use the same setting for all methods and datasets to ensure a controlled comparison.

Table 2: Hyperparameter details for the decoder-only language-model experiments. The heterogeneous rank setting assigns ranks 4, 8, 16, 32, and 64 to 40, 20, 20, 10, and 10 clients, respectively.

Hyperparameter	**TinyLlama-1.1B**	**LLaMA-3.2-1B**
Optimizer	AdamW	AdamW
Learning rate	3×10^{-4}	3×10^{-4}
Batch size (per client)	4	4
Local epochs	1	1
Global communication rounds	75	75
Total clients	100	100
Clients per round	10	10
Client sampling rate	0.1	0.1
Dirichlet concentration α	0.5	0.5
LoRA rank (homogeneous)	16	16
LoRA rank (heterogeneous)	4–64	4–64
LoRA scaling α	16	16
LoRA dropout	0.05	0.05
Target modules	`q_proj`, `v_proj`	`q_proj`, `v_proj`
Energy retention threshold τ	0.9	0.9