FED3DLORA: CROSS-LAYER TENSOR CONSENSUS FOR HETEROGENEOUS FEDERATED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated fine-tuning with low-rank adaptation (LoRA) lets resource-constrained clients adapt large language models collaboratively. When clients differ in both resources and data, the adapters they return can have different ranks and may come from different tasks, and the server has to combine them into one global adapter in every round. Existing methods make adapters of different rank compatible but aggregate each weight matrix separately, without an explicit shared representation of the updates that a client produces across the layers of a projection. We propose Fed3DLoRA, a server-side aggregation method that couples client updates across layers. Each client contributes a reference-corrected update, measured against the adapter it actually downloaded; the updates of a projection family are stacked into a third-order tensor with layer, output and input modes and projected onto a low-dimensional subspace shared across layers, where client-specific metrics derived from these projections determine a weighted consensus, while ordinary averaging is kept in the orthogonal complement. Client computation and communication are unchanged. We establish invariance to the LoRA factorisation, exact reduction to averaging under identical client metrics, a bounded deviation from it, and, under explicit assumptions, a stationarity bound that accounts for aggregation deviation and rank truncation. Extensive experiments on commonsense reasoning and natural language inference show that Fed3DLoRA outperforms existing federated fine-tuning approaches in most settings in terms of both accuracy and prediction confidence.

1 INTRODUCTION

Adapting large language models (LLMs) to private, decentralised data is a natural application of federated learning (McMahan et al., 2017; Kairouz et al., 2021). Because full fine-tuning exceeds the memory and bandwidth of most participants, low-rank adaptation (LoRA; Hu et al., 2022) has become a common vehicle for federated fine-tuning (Zhang et al., 2024; 2023; Ye et al., 2024; Babakniya et al., 2023; Sun et al., 2024): each client trains and exchanges a pair of small factor matrices per adapted layer while the backbone stays frozen. In practice, however, clients are heterogeneous in two respects that both affect server-side aggregation: their memory, compute and uplink differ, so the LoRA rank each can afford differs, and their data differ, so their updates move the adapter along different directions. In every round of *heterogeneous federated fine-tuning*, the server therefore combines adapters of different ranks, trained on different data, into one global adapter.

Existing approaches accommodate heterogeneous LoRA ranks through factor padding and stacking (Cho et al., 2024; Wang et al., 2024), product-space aggregation with spectral redistribution (Bai et al., 2024), spectral and gauge-aware server representations (Xiao et al., 2026; Chen et al., 2026), and module- or sketch-based local adaptation (Zhang et al., 2026; Fang et al., 2025). These mechanisms address differences in adapter rank and local parameterisation; we focus on a complementary aspect of server aggregation, the joint use of updates from corresponding projections across Transformer layers. A matrix-wise aggregation rule does not construct a shared cross-layer representation of these updates, which leaves open how information from several layers can enter update-dependent client contributions.

For update-dependent aggregation, the representation determines which features of a client's update can affect its contribution. A joint representation of a *projection family*, one projection type across all

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

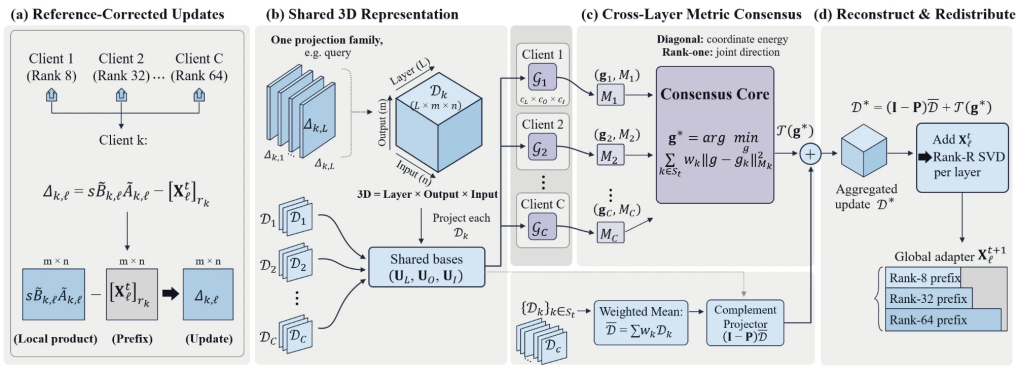

Figure 1: Overview of Fed3DLoRA. (a) The server forms each client's reference-corrected update relative to the prefix that client downloaded. (b) For each projection family, the layer-wise updates of a client are stacked into a third-order tensor with layer, output and input modes and projected onto shared mode bases, giving client cores $\mathcal{G}_k$ in common coordinates. (c) Metrics derived from $g_k = \mathrm{vec}(\mathcal{G}_k)$ determine the cross-layer consensus core $g^\star$. (d) The lifted consensus is combined with the orthogonal-complement component of the product-space mean, and the resulting update is added to the global adapter before a per-layer rank-R spectral write-back that serves nested prefixes to clients of every rank. Client training and per-round communication are unchanged. Of the K clients, the $C = |S_t|$ participating in round t are labelled $1, \ldots, C$ for illustration; sums run over $k \in S_t$.

layers, makes a client's updates at every layer available to a single aggregation decision, but averaging does not use it: stacking the updates and taking a common weighted mean still averages each layer separately. We therefore construct client-specific metrics from a shared tensor representation, so that the adaptive correction at one layer can depend on update information from other layers.

We propose **Fed3DLoRA**, a server-side method for metric-weighted aggregation in a shared cross-layer tensor subspace (Figure 1). The server first separates local adaptation from omitted reference components by measuring each update against the adapter the client actually downloaded (§3.1). The resulting updates of a projection family form a layer × output × input tensor, each client is represented by its core in a subspace shared by the cohort (§3.2), and metrics derived from these cores determine a consensus (§3.3). Because the consensus covers only the shared subspace, the product-space mean is retained in the orthogonal complement, and a spectral write-back returns nested factors for clients of every rank (§3.4). Client computation and per-round communication remain unchanged. Our contributions are threefold.

(1) We propose an update-dependent aggregation operator for heterogeneous-rank LoRA that couples corresponding projections across layers through shared tensor coordinates, while preserving the product-space mean outside the shared subspace before rank truncation.

(2) We prove invariance to the LoRA factorisation, exact recovery of product-space averaging under identical client metrics and a bound on the deviation from it, and we derive a conditional stationarity bound with explicit residual terms for aggregation deviation and rank truncation.

(3) We evaluate Fed3DLoRA on Commonsense8 and MultiNLI with four language models and nine baselines: it attains the highest mean final-round accuracy in most settings, and its cross-layer consensus improves mean accuracy over a layer-wise control by 0.23–0.95 points in all four comparisons with matched total core dimensionality and over the product-space mean in all four settings.

2 Federated Fine-Tuning System

Objective. We consider K clients, where client k holds a private dataset D_k, and a frozen pre-trained model with weights W^0. LoRA is attached to a set of projection families Φ (the query and value projections in our experiments) in each of the L transformer layers. For one family, the adapted weight of layer ℓ is $W_\ell^0 + X_\ell$ with the *represented adapter*

$$X_\ell \;=\; s\, B_\ell A_\ell, \qquad B_\ell \in \mathbb{R}^{m\times R},\; A_\ell \in \mathbb{R}^{R\times n},\; s = \alpha/R, \tag{1}$$

and we write $X = \{X_\ell\}$ for the collection over layers and families. The goal is

$$\min_X \; f(X) \;=\; \sum_{k=1}^{K} p_k \, f_k(X), \qquad f_k(X) = \mathbb{E}_{\xi \sim D_k} \, \ell\big(W^0 + X; \xi\big)\,, \tag{2}$$

with client weights $p_k \geq 0$, $\sum_k p_k = 1$. The loss depends on the factors only through the product $B_\ell A_\ell$; the factors themselves are determined only up to a gauge, since $B_\ell A_\ell = (B_\ell Q)(Q^{-1} A_\ell)$ for any invertible Q.

Heterogeneous ranks and nested prefixes. Client k can afford rank $r_k \leq R$. The server keeps the global adapter in spectral order (columns of B_ℓ and rows of A_ℓ sorted by the singular values of X_ℓ), and we denote the leading rank-r part by $[X_\ell]_r = s\, B_{\ell,[:,\,:r]} A_{\ell,[:r,\,:]}$. Round t proceeds in three steps: (i) the server samples a cohort S_t of C clients and sends client k the leading r_k components of X^t; (ii) client k trains this prefix for τ local steps on D_k and uploads the resulting factors $(\tilde{B}_{k,\ell}, \tilde{A}_{k,\ell}) \in \mathbb{R}^{m \times r_k} \times \mathbb{R}^{r_k \times n}$; (iii) the server aggregates the uploads into X^{t+1}. Both directions cost $r_k(m+n)$ numbers per adapted matrix; this paper is about step (iii).

3 FED3DLORA

Fed3DLoRA addresses how a client's updates across the layers of a Transformer can jointly determine its aggregation contribution when client ranks differ. Since clients start from different spectral prefixes, local adaptation must first be separated from the reference components a client never received (§3.1). To make the updates jointly available across layers while keeping their matrix structure, we stack each projection family into a third-order tensor with layer, output and input modes and build shared compact coordinates along these modes (§3.2). This representation alone leaves averaging unchanged, so client-specific metrics derived from the joint cores let aggregation depend on coordinate-wise energy and joint update directions (§3.3). Because the consensus covers only the shared subspace, the mean is retained outside it, and a spectral reconstruction yields nested adapters for every client rank (§3.4; Algorithm 1 in Appendix A).

3.1 REFERENCE-CORRECTED CLIENT UPDATES

Because clients start from different truncations of the same adapter, the change produced by local training has to be isolated from the reference each client starts from. A rank-r_k client receives $[X_\ell^t]_{r_k}$, while the *reference tail* $T_{k,\ell} = X_\ell^t - [X_\ell^t]_{r_k}$ stays on the server. The difference between the returned product $s\tilde{B}_{k,\ell}\tilde{A}_{k,\ell}$ and the full adapter therefore contains $-T_{k,\ell}$ even when the client does not train: with aggregation weights $w_k \geq 0$, $\sum_{k \in S_t} w_k = 1$, and no local training, adding the weighted mean of these uncorrected differences back to the global adapter retains spectral component j only with coefficient $\sum_{k: r_k > j} w_k$ (Proposition 2). We therefore define the *reference-corrected client update* relative to the adapter that was actually available for local adaptation,

$$\Delta_{k,\ell} = s\tilde{B}_{k,\ell}\tilde{A}_{k,\ell} - [X_\ell^t]_{r_k} = s\big(\tilde{B}_{k,\ell}\tilde{A}_{k,\ell} - B^t_{\ell,[:,\,:r_k]} A^t_{\ell,[:r_k,\,:]}\big). \tag{3}$$

Equivalently, the server completes the returned prefix with the omitted tail (*tail completion*), $X_{k,\ell} = s\tilde{B}_{k,\ell}\tilde{A}_{k,\ell} + T_{k,\ell} = X_\ell^t + \Delta_{k,\ell}$: the omitted tail induces no update, an unchanged return gives $\Delta_{k,\ell} = 0$, and since $\Delta_{k,\ell}$ depends on the uploaded factors only through their product, it is unaffected by a change of their gauge (Propositions 1 and 2). Reference correction thus keeps rank-truncated initialisation from introducing an artificial update before any cross-layer statistics are formed. Whatever their factorisation rank, the resulting updates lie in the same $m \times n$ space, which makes them comparable across clients but does not yet let aggregation use a client's update pattern across layers.

3.2 SHARED CROSS-LAYER TENSOR REPRESENTATION

Letting aggregation depend on how a client adapts a projection family across depth requires a joint representation of its layer-wise updates. Keeping the output and input axes of each update and adding the layer as a third mode gives the client's cross-layer update tensor $\mathcal{D}_k \in \mathbb{R}^{L \times m \times n}$, whose layer slices are $\mathcal{D}_k[\ell, :, :] = \Delta_{k,\ell}$; each family is processed separately. These tensors already share ambient

coordinates but are large, and compact bases fitted to each client separately would place the clients in different coordinates, so we estimate one set of layer, output and input bases from the cohort and use it for every client. To keep opposing client updates from cancelling before the bases are formed, the mode-wise second moments are accumulated client by client; with mode-j unfoldings $\mathcal{D}_{k(j)}$,

$$H_L = \sum_{k\in S_t} w_k\, \mathcal{D}_{k(1)}\mathcal{D}_{k(1)}^\top, \qquad H_O = \sum_{k\in S_t} w_k\, \mathcal{D}_{k(2)}\mathcal{D}_{k(2)}^\top, \qquad H_I = \sum_{k\in S_t} w_k\, \mathcal{D}_{k(3)}\mathcal{D}_{k(3)}^\top, \quad (4)$$

and $U_L \in \mathbb{R}^{L\times c_L}$, $U_O \in \mathbb{R}^{m\times c_O}$, $U_I \in \mathbb{R}^{n\times c_I}$ collect their leading eigenvectors, as in a Tucker decomposition (Tucker, 1966; De Lathauwer et al., 2000; Kolda & Bader, 2009). Since, e.g., $\sum_k w_k \|U_L^\top \mathcal{D}_{k(1)}\|_F^2 = \mathrm{tr}(U_L^\top H_L U_L)$, the leading eigenspace of each mode maximises the weighted projected energy of that mode for the prescribed dimension. Every client is then represented by its own core in the same subspace,

$$\mathcal{G}_k = \mathcal{D}_k \times_1 U_L^\top \times_2 U_O^\top \times_3 U_I^\top \in \mathbb{R}^{c_L\times c_O\times c_I}, \qquad g_k = \mathrm{vec}(\mathcal{G}_k) \in \mathbb{R}^p, \;\; p = c_L c_O c_I. \quad (5)$$

We write $\mathcal{T}(g) = \mathcal{G} \times_1 U_L \times_2 U_O \times_3 U_I$ for the lift back to the represented space, whose layer-ℓ slice $U_O\big(\sum_j u_\ell[j]\,\mathcal{G}[j,:,:]\big)U_I^\top$ combines the core slices through the layer code u_ℓ (the ℓ-th row of U_L), and $P = \mathcal{T}\mathcal{T}^*$ for the orthogonal projector onto the core subspace. The core summarises a client's updates across all layers in coordinates common to the cohort and is invariant to the gauge of the uploaded factors. It supplies the information needed for cross-layer aggregation but no rule for combining clients: averaging the cores and keeping the complementary mean recovers product-space averaging (§3.4). To let the joint update pattern affect client contributions, we next derive an aggregation metric from each client core.

3.3 Update-dependent metric consensus

The shared representation makes client updates comparable, yet averaging still gives each client the same relative weight in every direction: the core mean $\bar g = \sum_k w_k g_k$ minimises $\sum_k w_k \|g - g_k\|_2^2$, a Euclidean objective that treats all directions alike. We let the shape of each client's joint update enter this objective by replacing the Euclidean penalty with a client-specific quadratic penalty $\|g - g_k\|_{M_k}^2 = (g-g_k)^\top M_k (g-g_k)$, so that deviations from different clients are weighted differently across coordinates and directions. The metric combines two characteristics of the joint core, its relative squared coordinates q_k (how the update energy is distributed) and its unit direction v_k (how the update is oriented across these coordinates):

$$M_k = \lambda I + \mathrm{Diag}(q_k) + \mu\, v_k v_k^\top, \qquad q_k[i] = \min\Big\{\tfrac{g_k[i]^2}{\|g_k\|^2/p},\, c\Big\}, \qquad v_k = \tfrac{g_k}{\|g_k\|}. \quad (6)$$

In the expanded quadratic form

$$\|g - g_k\|_{M_k}^2 = \lambda\|g-g_k\|_2^2 + \sum_i q_k[i]\big(g[i]-g_k[i]\big)^2 + \mu\big(v_k^\top(g-g_k)\big)^2, \quad (7)$$

the isotropic damping provides a base distance and makes the metric positive definite, the diagonal term penalises deviations on the coordinates where the client's update energy is concentrated, and the rank-one term penalises deviations along its joint update direction; clipping at c bounds the coordinate-dependent penalties, so that $\lambda I \preceq M_k \preceq (\lambda + c + \mu)I$. The consensus core minimises the weighted sum of these penalties,

$$g^\star = \arg\min_g \sum_{k\in S_t} w_k\,(g-g_k)^\top M_k\,(g-g_k) = \Big(\sum_{k\in S_t} w_k M_k\Big)^{-1} \sum_{k\in S_t} w_k M_k g_k. \quad (8)$$

Since each M_k is diagonal plus rank one, the system matrix is diagonal plus a term of rank at most C, and the Woodbury identity (Golub & Van Loan, 2013) reduces (8) to a $C\times C$ solve without forming a dense $p\times p$ metric (Appendix A.2). Because every metric is derived from a core that represents the whole projection family, the aggregation decision for one layer can depend on update information from other layers. With the diagonal term alone, coordinate i of the consensus is the average $\sum_k a_k[i] g_k[i]$ with $a_k[i] \propto w_k(\lambda + q_k[i])$; as $\lambda \to 0$ it tends to a weighted average restricted to the clients carrying that coordinate, with weights proportional to $w_k q_k[i]$ (Appendix C). The rank-one term is characterised in Proposition 5. A client whose mean squared core coordinate $\|g_k\|^2/p$ lies at or below the machine epsilon of the server arithmetic receives $q_k = 0$ and enters with the isotropically damped rank-one metric $\lambda I + \mu v_k v_k^\top$ (Appendix A.3).

3.4 COMPLEMENT-PRESERVING RECONSTRUCTION AND RANK-ADAPTIVE READOUT

The consensus determines only the component of the update that lies in the shared subspace. Lifting it alone would discard the complementary part of the mean update, so that the dimensionality reduction itself would alter components to which the adaptive rule was never applied. We therefore require the reconstructed update to match the consensus inside the shared subspace and to retain the product-space mean $\bar{\mathcal{D}} = \sum_k w_k \mathcal{D}_k$, whose core is $\bar{g}$, outside it:

$$\mathcal{D}^\star = (I - P)\bar{\mathcal{D}} + \mathcal{T}\big(g^\star\big) = \bar{\mathcal{D}} + \mathcal{T}\big(g^\star - \bar{g}\big). \tag{9}$$

Equivalently, $\mathcal{D}^\star$ is the update closest to the product-space mean in Frobenius norm among all updates whose shared core equals $g^\star$, and the dimensions (c_L, c_O, c_I) determine where the operator may depart from the mean. The reconstructed per-layer target need not respect the server's rank budget. To return to the nested LoRA protocol, the server takes its best rank-R approximation and stores balanced factors in spectral order,

$$X_\ell^{t+1} = \Pi_R\big(X_\ell^t + \mathcal{D}^\star[\ell, :, :]\big) = U\Sigma V^\top, \qquad B_\ell^{t+1} = U\Sigma^{1/2}/\sqrt{s}, \quad A_\ell^{t+1} = \Sigma^{1/2}V^\top/\sqrt{s}, \tag{10}$$

where Π_R keeps the R leading singular triplets. By the Eckart–Young theorem (Eckart & Young, 1936), this ordering makes each prefix $[X_\ell^{t+1}]_r$ the best rank-r approximation of the global adapter in Frobenius norm for every $r \le R$, so that clients of different ranks receive prefixes of one common server state. Complement preservation holds before this rank-constrained write-back.

3.5 THEORETICAL PROPERTIES AND COMPLEXITY

The construction raises three questions: whether the aggregate depends only on the represented client updates, whether reference correction preserves the fixed point of a round without training, and how far the metric-weighted consensus can move the update away from product-space averaging. Propositions 1–3 hold for any orthonormal bases and any positive-definite metrics computed from the client cores, Proposition 4 requires in addition $\lambda I \preceq M_k \preceq (\lambda + \bar{c})I$ ($\bar{c} = c + \mu$ for (6)), and Proposition 5 concerns (6) with $q_k = 0$ for every client, that is, $M_k = \lambda I + \mu v_k v_k^\top$. Theorem 1 then bounds stationarity under explicit assumptions on the local updates and the rank-constrained write-back. Proofs are in Appendix C, with numerical checks on real uploads and the empirical quantities entering these assumptions in Appendix C.4.

Proposition 1 (Gauge invariance). *Replacing any upload $(\tilde{B}_{k,\ell}, \tilde{A}_{k,\ell})$ by $(\tilde{B}_{k,\ell}Q,\, Q^{-1}\tilde{A}_{k,\ell})$ with Q invertible leaves $\mathcal{D}^\star$ and the new global adapter X^{t+1} unchanged.*

Proposition 2 (Zero-update fixed point). *A client that returns its prefix unchanged has $\mathcal{D}_k = 0$; if all clients of the cohort do so, then $X^{t+1} = X^t$. If uploads are instead read without tail completion, the product-space mean is shifted by the reference tails, $-\sum_k w_k T_k$, independently of the data.*

Proposition 3 (Exact reduction and complement preservation). *If $M_k = M$ for all $k \in S_t$, then $g^\star = \bar{g}$ and $\mathcal{D}^\star = \bar{\mathcal{D}}$, the product-space mean of the reference-corrected client updates. In general $(I - P)\mathcal{D}^\star = (I - P)\bar{\mathcal{D}}$.*

Proposition 4 (Bounded deviation). *$\|\mathcal{D}^\star - \bar{\mathcal{D}}\|_F \le \chi \left(\sum_k w_k \|P(\mathcal{D}_k - \bar{\mathcal{D}})\|_F^2\right)^{1/2}$ with $\chi^2 = (\lambda + \bar{c})/\lambda$. For diagonal metrics ($\mu = 0$) the constant improves to $\chi^2 = \min\{c^2/(4\lambda(c + \lambda)),\ 1/\min_k w_k - 1\}$.*

Proposition 5 (Closed form of the rank-one consensus). *Let $q_k = 0$ for all k and $V = [v_1, \dots, v_C]$. Then $g^\star = Va$ lies in the span of the client directions, and for fixed λ, μ the coefficients $a \in \mathbb{R}^C$ depend only on the client norms $\|g_k\|$, the weights w_k and the cosine matrix $G = V^\top V$ (closed form in Appendix C). In particular, $g^\star = \bar{g}$ if all v_k coincide; $g^\star = \sum_k \frac{\lambda+\mu}{\lambda+\mu w_k} w_k g_k$ if the v_k are orthogonal; and if $w_k = 1/C$, all norms are equal and all pairwise cosines equal $\bar{c}$, then $g^\star = \rho(\bar{c})\,\bar{g}$ with $\rho(\bar{c}) = (\lambda + \mu)/\big(\lambda + \mu(1 + (C-1)\bar{c})/C\big)$.*

Theorem 1 (Stationarity bound). *Under Assumptions 1–3 (Appendix C.1), with effective step $\bar{\eta}$ (local learning rate times τ), bias constants (β, B), variance σ^2 and dispersion D^2 of the local updates and relative write-back residual ϑ, let f be L_f-smooth and bounded below by $f^\star$, let $\omega = \beta + \vartheta(1+\beta) < 1$ and $\bar{\eta} \le (1-\omega)/\big(8L_f(1+\vartheta)^2(1+\beta)^2\big)$. Then the iterates of Fed3DLoRA satisfy*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\big\|\nabla f(X^t)\big\|_F^2 \;\le\; \frac{2\big(f(X^0) - f^\star\big)}{(1-\omega)\,\bar{\eta}\,T} + \frac{2L_f\bar{\eta}\,\Psi}{1-\omega} + \frac{2\,\Theta}{(1-\omega)^2}, \tag{11}$$

with $\Psi = (1+\vartheta)^2(2B^2+\sigma^2+\chi^2D^2)$ *and* $\Theta = \big((1+\vartheta)(B+\chi D)+\vartheta\sigma\big)^2$.

The first two terms are those of product-space local SGD, giving $\mathcal{O}(1/\sqrt{T})$ for $\bar{\eta} \propto 1/\sqrt{T}$; the last does not vanish as T grows. The consensus enters through χD, which bounds $\mathcal{D}^\star - \bar{\mathcal{D}}$ (Proposition 4), and the write-back through ϑ: the SVD guarantees only $\vartheta \leq 1$, so $\vartheta < 1$ is an assumption on the trajectory ($\vartheta \leq 0.30$ on the recorded events; Appendix C.4).

Complexity. Client computation and communication are those of rank-r_k LoRA (§2); bases and cores never leave the server. Per round, the server additionally computes the Gram operators (4), three symmetric eigendecompositions of sizes L, m and n, C projections (5) and one $C \times C$ solve (8); the L truncated SVDs (10) are shared with product-space averaging under the same write-back (Appendix F). The shared bases take $Lc_L + mc_O + nc_I$ numbers per family against $L(mc_O + nc_I c_L/L)$ for layer-wise bases with the same number of core coordinates.

4 Evaluation

4.1 Setup

Protocol. We use 40 clients with 8 participating per round, 10 rounds and multiple fixed seeds. Within a batch all methods share the client data, rank binding, participation sequence, initialisation and evaluation set of a seed, so every comparison is paired. Each participation trains 16 AdamW (Loshchilov & Hutter, 2019) steps of 512 supervised response tokens at maximum length 512 (precisions in Appendix B.4); LoRA is applied to the query and value projections with global rank $R = 64$ and $\alpha = 64$. Client ranks are drawn from $\{8, 16, 32, 64\}$ independently of the task or domain profile.

Datasets. *Commonsense8* pools eight multiple-choice tasks—ARC-Easy and ARC-Challenge (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SocialIQA (Sap et al., 2019) and WinoGrande (Sakaguchi et al., 2020)—with 2–4 candidates per example; its 40 clients hold two tasks each, drawn from four task-pair profiles, and the held-out diagnostic set has 768 examples (96 per task). *MultiNLI* (Williams et al., 2018) is three-way inference over five training genres; its 40 clients are built by genre, eight per genre with 960 examples each (80% from the main genre), every genre holds the same rank multiset and the ranks of every round sum to 256. Main results are reported on the official matched (9,815) and mismatched (9,832) validation sets, and a held-out diagnostic set of 1,500 examples (300 per genre) serves the component analyses; on both datasets, accuracy is the fraction of examples whose gold candidate has the lowest mean token negative log-likelihood, and choice NLL is the negative log-probability of the gold candidate after a softmax over the candidates' scores (lower is better). We report final-round values as mean $\pm$ SD over multiple seeds (client construction in Appendix B).

Models and baselines. We use Qwen3-0.6B-Base and Qwen3-1.7B-Base (Qwen Team, 2025) and Llama-3.2-1B and Llama-3.2-3B (Grattafiori et al., 2024). The external baselines are FedIT (Zhang et al., 2024), Zero-Padding, HetLoRA (Cho et al., 2024), FlexLoRA (Bai et al., 2024), FLoRA (Wang et al., 2024), FSLoRA (Fang et al., 2025), FedHera (Xiao et al., 2026), Fed-PLoRA (Zhang et al., 2026) and GLoRA (Chen et al., 2026). Four internal controls each change one server step of Fed3DLoRA: **product-space averaging** replaces the consensus by the product-space mean of the reference-corrected updates, keeping tail completion and the same write-back (Proposition 3); **shuffled-metric consensus** applies a fixed random coordinate permutation to M_k but not to g_k, keeping the spectrum of the metric and breaking its alignment with the update; **layer-wise consensus** fits separate bases and client metrics for every layer with the same total number of core coordinates; and **without tail completion** removes the reference correction of §3.1 by treating each returned product as the complete adapter. Fixed-input analyses (Appendix E.4) compare the metric terms on the same recorded uploads.

4.2 Main results

Fed3DLoRA is competitive across both backbone families. On Commonsense8 it ranks first on three of the four backbones and second on Llama-3.2-3B, next to Fed-PLoRA (Table 1); on MultiNLI with Llama-3.2-1B it leads on both validation sets, whereas with Qwen3-0.6B Fed-PLoRA leads on both

Table 1: Final-round accuracy (%) on the Commonsense8 diagnostic set, mean ± SD over multiple seeds (base model: pre-trained backbone without fine-tuning); best in bold, second best underlined.

Method	Qwen3-0.6B	Qwen3-1.7B	Llama-3.2-1B	Llama-3.2-3B
Base model	50.39	54.56	54.30	56.25
FedIT	58.46±0.47	66.67±0.98	58.77±0.42	67.27±1.12
Zero-Padding	58.68±0.27	67.10±0.59	59.64±1.32	68.92±0.60
HetLoRA	58.59±1.24	68.06±0.78	59.03±1.25	68.66±0.53
FlexLoRA	56.55±0.64	67.14±0.67	59.51±0.57	69.10±0.64
GLoRA	59.12±0.11	68.62±0.50	59.56±0.73	69.14±0.57
FLoRA	56.34±0.33	67.19±0.78	59.85±0.66	69.23±1.01
FSLoRA	58.12±1.06	68.66±0.72	58.16±1.21	68.97±0.40
FedHera	57.77±1.55	67.27±0.27	59.55±1.33	68.32±0.66
Fed-PLoRA	57.03±1.24	66.75±1.26	58.03±0.72	**69.66±0.85**
Fed3DLoRA	**60.03±0.60**	**68.79±0.27**	**60.89±0.96**	69.49±0.21

Table 2: Final-round accuracy (%) on the MultiNLI matched and mismatched validation sets; notation as in Table 1.

Method	Qwen3-0.6B		Llama-3.2-1B	
	matched	mismatched	matched	mismatched
Base model	39.10	39.68	35.45	35.22
FedIT	84.35±0.11	84.51±0.16	84.03±0.41	84.93±0.13
Zero-Padding	83.43±0.15	83.66±0.31	83.14±0.10	84.01±0.10
HetLoRA	83.01±0.20	83.30±0.32	82.81±0.12	83.68±0.20
FlexLoRA	83.00±0.14	83.67±0.08	84.20±0.40	84.87±0.18
GLoRA	81.83±0.04	82.12±0.05	70.45±0.89	71.48±0.91
FLoRA	84.41±0.17	84.58±0.11	84.26±0.30	84.93±0.13
FSLoRA	83.68±0.05	84.04±0.18	82.67±0.17	83.54±0.14
FedHera	83.83±0.23	84.26±0.19	83.89±0.44	84.66±0.19
Fed-PLoRA	**84.56±0.45**	**84.72±0.16**	84.31±0.18	84.96±0.16
Fed3DLoRA	84.30±0.59	84.65±0.32	**84.54±0.31**	**85.21±0.10**

and Fed3DLoRA is second on the mismatched set (Table 2). Reference correction, implemented by tail completion, improves mean accuracy in all eight of these comparisons (Appendix D.1).

4.3 Which components matter

Table 3 changes one server step at a time to test the joint cross-layer design, the metric–update correspondence and reference correction. At matched total core dimensionality, the joint cross-layer consensus attains higher mean accuracy than separate layer-wise subspaces and metrics in all four settings, by 0.23–0.95 points, with approximately 21–23× less basis storage in the two reported Qwen configurations (Appendix F); this supports the joint representation-and-aggregation design, although the comparison does not isolate basis sharing from cross-layer metric coupling. Replacing the consensus by product-space averaging lowers mean accuracy in all four settings, most on Qwen3-1.7B, where the control was re-trained (§4.4). Shuffling the metric coordinates while preserving their spectrum also lowers accuracy, so the correspondence between a metric and its client update matters in the tested configurations. Removing the reference correction, which lets the omitted reference tails enter the cross-layer statistics, lowers accuracy in all four settings, in line with the main-setting comparison of Appendix D.1.

Table 3: Ablations of Fed3DLoRA: paired final-round accuracy differences (points), Fed3DLoRA minus control, mean over multiple seeds, mechanism batch; †re-trained control (Appendix D.3); earlier-batch choice NLL in Table 17.

Fed3DLoRA vs.	Commonsense8		MultiNLI	
	Qwen3-0.6B	Qwen3-1.7B	Qwen3-0.6B	Qwen3-1.7B
Layer-wise consensus	+0.77	+0.95	+0.27	+0.23
Product-space averaging	+1.09	+2.08†	+0.49	+0.29
Shuffled-metric consensus	+0.95	+1.82	+0.69	+0.11
Without tail completion	+0.61	+1.00	+0.60	+0.21

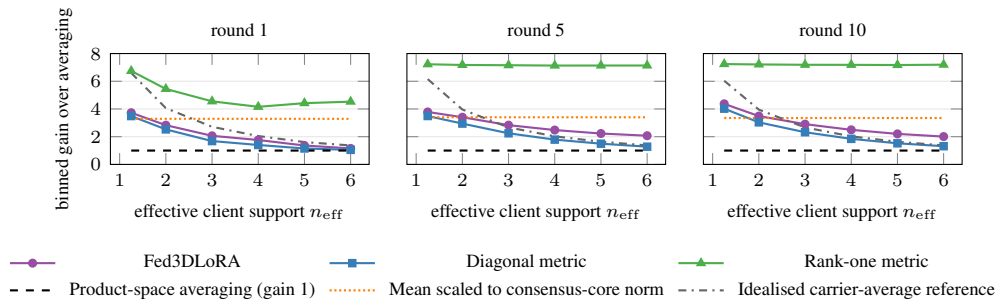

Figure 2: Binned least-squares gain relative to product-space averaging versus effective client support n_{eff} (Commonsense8 / Qwen3-1.7B, query projections, low-conflict coordinates; medians over multiple seeds; binning and controls in Appendix E.4).

Table 4: In-subspace update (medians over events and families): mean pairwise client cosine $\bar{c}$, amplification $\rho = \|g^\star\|/\|\bar{g}\|$ and cosine κ between $g^\star$ and $\bar{g}$ (shaded: Fed3DLoRA; ρ_1, κ_1: rank-one term alone).

Setting	Round 1					Rounds 5 and 10				
	$\bar{c}$	ρ	κ	ρ_1	κ_1	$\bar{c}$	ρ	κ	ρ_1	κ_1
CS8 / Qwen3-0.6B	0.289	2.47	0.807	4.18	0.841	0.041	4.49	0.848	7.01	0.992
CS8 / Qwen3-1.7B	0.156	3.13	0.795	5.74	0.921	0.015	3.36	0.858	7.16	0.998
MNLI / Qwen3-0.6B	0.727	1.25	0.928	1.33	0.982	0.239	3.02	0.978	3.02	0.978

Table 5: Server-step control on Commonsense8 / Qwen3-1.7B, mean over multiple seeds (per run: Appendix D.3); NM: Fed3DLoRA update rescaled to the norm of the averaging update (Appendix E.2).

	Product-space averaging					Fed3DLoRA	Fed3DLoRA-NM
	γ=1	γ=1.5	γ=2	γ=3	FedExP		
Accuracy (%, ↑)	65.06	66.06	64.24	61.72	61.50	67.14	**67.40**
Choice NLL (↓)	0.9495	0.9334	0.9641	1.0253	0.9970	0.9068	**0.8312**

4.4 Mechanism

Task-related structure in the shared cores. We first ask whether the compact shared representation retains task-related differences among client updates. In all 36 recorded event–family combinations on Commonsense8, clients with the same task-pair profile have a higher mean core cosine than clients with different profiles (Figure 6, Appendix E.5). The shared coordinates therefore retain task-related structure in the client updates, and the resulting cosine matrix G of client directions enters the rank-one consensus characterised by Proposition 5.

Coordinate weighting and aggregate geometry. We next examine how the coordinate and direction penalties change the consensus core. Across the recorded settings and rounds, the shared subspace captures medians of 32–65% of the weighted client-update energy, and components outside it are aggregated by their product-space mean (Appendix E.4). Figure 2 reports binned least-squares gains relative to the mean, grouping low-conflict coordinates by their effective client support n_{eff} and separating the two terms of (7). The gain of the diagonal term decreases with client support, a trend comparable to the idealised carrier-average reference $C_{\text{eff}}/\bar{n}$. On the recorded uploads, the rank-one term alone gives a nearly flat common amplification that is larger when client agreement is lower; in general, it also depends on the client norms, the weights and the full cosine matrix (Proposition 5). Table 4 connects this coordinate-level selectivity to changes in the magnitude and direction of the consensus core. In the later Commonsense8 rounds, client agreement $\bar{c}$ is low and the rank-one-only consensus is nearly aligned with the mean at a large amplification ($\rho_1 \approx 7$, $\kappa_1 \approx 0.99$), consistent with the qualitative trend of the symmetric case in Proposition 5, whereas the full metric produces a smaller amplification and a larger directional change ($\kappa \approx 0.85$). On the later MultiNLI rounds the two rules coincide because every client falls under the guard of §3.3 and the diagonal term is inactive (Appendix E.4).

Server-step controls. Because the metrics change both the direction and the magnitude of the update, Table 5 tests whether a global step size can reproduce their end-to-end effect, training product-space

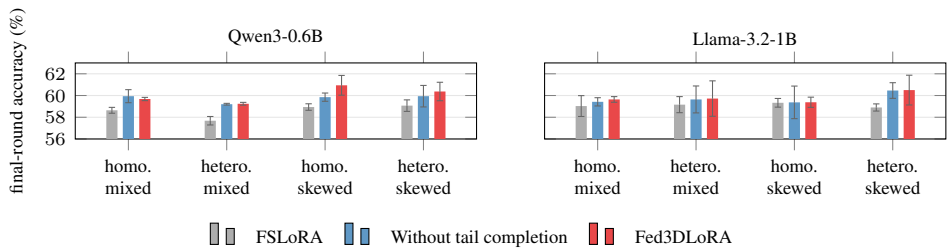

Figure 3: Rank and data heterogeneity at a fixed rank budget of 256 per round (Commonsense8, mean $\pm$ SD over multiple seeds; numbers in Table 19).

averaging with server steps $\gamma \in \{1, 1.5, 2, 3\}$ and with the FedExP step (Jhunjhunwala et al., 2023) next to Fed3DLoRA and Fed3DLoRA-NM (Appendix E.2). A moderate step helps and larger tested steps reduce accuracy; none of the tested global steps reaches the mean accuracy of Fed3DLoRA, which improves on product-space averaging by 2.08 percentage points on average, with a positive difference in each seed. Norm matching lowers choice NLL further in every seed, while its accuracy effect changes sign across seeds. These comparisons do not separate subspace selection, direction and scale.

4.5 Rank and data heterogeneity

The total active rank per round is held fixed at 256, so Figure 3 varies only its allocation across clients (homogeneous rank 32 against heterogeneous $\{8, 16, 32, 64\}$) and the distribution of client data (mixed against skewed) on Commonsense8, against FSLoRA. At this fixed budget the effect of rank allocation is model-dependent: heterogeneous allocation lowers mean accuracy relative to homogeneous allocation for Qwen3-0.6B in both data settings, whereas the mean differences have the opposite sign for Llama-3.2-1B, so heterogeneous ranks show no uniform trend here. Fed3DLoRA remains above FSLoRA in the jointly heterogeneous setting on both models, by 1.30 and 1.60 points. These comparisons evaluate the complete pipeline under different allocations and data distributions and do not isolate the contribution of cross-layer consensus.

5 Related Work

Parameter-efficient federated fine-tuning of language models (Zhang et al., 2023; 2024; Ye et al., 2024), its initialisation (Babakniya et al., 2023) and analyses of factor averaging (Sun et al., 2024; Guo et al., 2025) assume a common rank. HetLoRA (Cho et al., 2024) pads and averages factors with norm-based weights; FlexLoRA (Bai et al., 2024) averages represented updates and redistributes them by SVD; FLoRA (Wang et al., 2024) stacks client adapters; FSLoRA (Fang et al., 2025) trains random sketches of a global adapter; Fed-PLoRA (Zhang et al., 2026) uses parallel rank-one modules; FedHera (Xiao et al., 2026) and GLoRA (Chen et al., 2026) maintain spectral and gauge-aware server states. All of these aggregate each weight matrix on its own with a scalar weight per client or per module (GLoRA's shared reference coordinates are also per weight matrix), whereas Fed3DLoRA aggregates jointly across the layers of a family under matrix-valued client metrics and keeps the mean in the complement. Tensor decompositions parameterise adapters compactly, e.g. Tucker in LoTR (Bershatsky et al., 2024) and tensor trains in LoRETTA (Yang et al., 2024), whereas Fed3DLoRA forms its tensor on the server to define aggregation coordinates and leaves the clients' LoRA parameterisation unchanged; model merging weights parameters by estimated Fisher precisions (Matena & Raffel, 2022) or resolves sign conflicts before averaging (Yadav et al., 2023).

6 Conclusion

Fed3DLoRA performs update-dependent aggregation in a shared cross-layer tensor representation and preserves the product-space mean outside the consensus subspace before rank truncation. It is competitive across heterogeneous fine-tuning settings and attains higher mean accuracy than the tested layer-wise and product-space averaging controls, with unchanged client training and communication and additional server-side computation.

AI USE STATEMENT

We used large language models to aid and polish the writing of this paper and to assist in proving the mathematical claims of Section 3.5, whose proofs are given in Appendix C. The authors reviewed and verified all AI-assisted text and every proof, and take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

The server operator is specified by Equations (3)–(10) and Algorithm 1, and its hyperparameters are listed in Appendix B.7. The datasets and backbones are publicly available; the client construction, rank assignment, training configuration including learning rates, and evaluation protocol are described in Section 4.1 and Appendix B. Assumptions and complete proofs are given in Appendix C. The implementation of Fed3DLoRA and its configuration are provided as anonymous supplementary material.

REFERENCES

Ahmad Ajalloeian and Sebastian U. Stich. On the convergence of SGD with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.

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. *arXiv preprint arXiv:2308.06522*, 2023.

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

Daniel Bershatsky, Daria Cherniuk, Talgat Daulbaev, Aleksandr Mikhalev, and Ivan Oseledets. LoTR: Low tensor rank weight adaptation. *arXiv preprint arXiv:2402.01376*, 2024.

Rajendra Bhatia and Chandler Davis. A better bound on the variance. *The American Mathematical Monthly*, 107(4):353–357, 2000.

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

Jinqian Chen, Chang Liu, and Jihua Zhu. Beyond factor aggregation: Gauge-aware low-rank server representations for federated LoRA. *arXiv preprint arXiv:2605.06733*, 2026.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. Heterogeneous LoRA for federated fine-tuning of on-device foundation models. In *Conference on Empirical Methods in Natural Language Processing*, 2024.

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

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

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

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. *arXiv preprint arXiv:2501.19389*, 2025.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.

Aaron Grattafiori, Abhimanyu Dubey, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *International Conference on Learning Representations*, 2025.

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. In *International Conference on Learning Representations*, 2022.

Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. FedExP: Speeding up federated averaging via extrapolation. In *International Conference on Learning Representations*, 2023.

Peter Kairouz, H. Brendan McMahan, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3): 455–500, 2009.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Michael Matena and Colin Raffel. Merging models with Fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, 2022.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 2017.

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

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *AAAI Conference on Artificial Intelligence*, 2020.

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

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

Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3): 279–311, 1966.

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 *Advances in Neural Information Processing Systems*, 2024.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

Ke Xiao, Qiyuan Wang, Christos Anagnostopoulos, Zhuoran Tan, and Wenhao Li. FedHera: Towards drift-resilient federated fine-tuning with heterogeneous resources. In *International Conference on Machine Learning*, 2026.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

Yifan Yang, Jiajun Zhou, Ngai Wong, and Zheng Zhang. LoRETTA: Low-rank economic tensor-train adaptation for ultra-low-parameter fine-tuning of large language models. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2024.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. OpenFedLLM: Training large language models on decentralized private data via federated learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

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

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the FederatedGPT: Federated instruction tuning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2024.

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*, 2023.

Zikai Zhang, Rui Hu, and Jiahao Xu. Heterogeneous federated fine-tuning with parallel one-rank adaptation. In *International Conference on Learning Representations*, 2026.

APPENDIX OVERVIEW

Appendix A specifies the implementation of Fed3DLoRA: notation, the client interface, the server operator and its numerical conventions. Appendix B documents the data, the federated protocol, the rank assignments, the training configuration, the baselines and controls, and the evaluation and reporting rules. Appendix C contains the assumptions, the proofs, the numerical checks of the operator identities on recorded uploads and the measured quantities that enter the assumptions of Theorem 1. Appendix D gives complete tables behind the main results, Appendix E the controlled analyses of the server update, and Appendix F the computation, communication and storage accounting.

A Additional method details

A.1 Notation and tensor dimensions

Table 6: Symbols used in the paper. Sizes refer to one projection family; the query and value families are processed independently.

Symbol	Size	Meaning
L, m, n	–	number of adapted layers of the family; output and input dimension of the adapted matrix
R, r_k	–	global (server) rank; rank of client k, $r_k \le R$
$X_\ell^t = sB_\ell^t A_\ell^t$	$m \times n$	represented global adapter of layer ℓ at round t, factors in spectral order, $s = \alpha/R$
$[X_\ell]_r$	$m \times n$	leading rank-r prefix of X_ℓ (the part a rank-r client downloads)
$(\tilde{B}_{k,\ell}, \tilde{A}_{k,\ell})$	$m \times r_k, r_k \times n$	factors returned by client k after local training
$\Delta_{k,\ell}, \mathcal{D}_k$	$m \times n, L \times m \times n$	reference-corrected update of client k in layer ℓ; its cross-layer update tensor
S_t, C, w_k	–	cohort of round t, its size, aggregation weight of client k ($\sum_{k \in S_t} w_k = 1$)
H_L, H_O, H_I	L^2, m^2, n^2	energy-weighted Gram operators of the three modes
U_L, U_O, U_I	$L \times c_L, m \times c_O, n \times c_I$	shared mode bases (leading eigenvectors of the Gram operators)
g_k, p	p	client core of client k; $p = c_L c_O c_I$
$\mathcal{T}, P$	–	lift from the shared subspace to the represented space; orthogonal projector $\mathcal{T}\mathcal{T}^*$
M_k, λ, μ, c	$p \times p$	metric of client k; damping, rank-one weight, truncation of the energy profile
v_k, q_k	p	unit direction of g_k; truncated energy profile of g_k
$\bar{\mathcal{D}}, \bar{g}, g^\star, \mathcal{D}^\star$	–	product-space mean, its core, consensus core, aggregated update
Π_R, ϑ	–	truncated SVD to rank R; relative write-back residual (Assumption 3)

A.2 Operator details beyond the main text

Communication. Clients exchange only their rank-specific factors, as specified in §2; communication accounting is given in Appendix F.1.

Woodbury solve. Let $d = \lambda \mathbf{1} + \sum_k w_k q_k \in \mathbb{R}^p$ and $V = [\sqrt{w_k}\, v_k]_{k \in S_t} \in \mathbb{R}^{p \times C}$, so that $\sum_k w_k M_k = \mathrm{Diag}(d) + \mu V V^\top$. The right-hand side is formed exactly as

$$b = \sum_k w_k M_k g_k = \sum_k w_k \big((\lambda + q_k) \odot g_k + \mu\, v_k\, (v_k^\top g_k) \big), \tag{12}$$

and with $y = b/d$ (element-wise)

$$g^\star = y - \mathrm{Diag}(d)^{-1} V \big(\mu^{-1} I_C + V^\top \mathrm{Diag}(d)^{-1} V \big)^{-1} V^\top y, \tag{13}$$

which costs $\mathcal{O}(pC^2 + C^3)$ and never forms a $p \times p$ matrix.

Order of the operations. The core correction $\mathcal{T}(g^\star - \bar{g})$ is added to the product-space mean before any truncation, so the complement of the mean is preserved exactly in $\mathcal{D}^\star$ (Proposition 3); only then is $X_\ell^t + \mathcal{D}^\star[\ell, :, :]$ truncated to rank R. The balanced factors $B = U\Sigma^{1/2}/\sqrt{s}$, $A = \Sigma^{1/2} V^\top / \sqrt{s}$ are determined by X_ℓ^{t+1} up to the sign of each singular pair and rotations within repeated singular values; the represented adapter, and hence every prefix $[X_\ell^{t+1}]_r$ for r at a gap of the spectrum, does not depend on this choice.

A.3 End-to-end pseudocode and numerical conventions

Algorithm 1 Fed3DLoRA (one projection family; families are processed independently)

Require: global factors $(B_\ell^0, A_\ell^0)_{\ell=1}^L$ of rank R in spectral order, client ranks r_k, mode ranks (c_L, c_O, c_I), constants λ, c, μ

1: **for** round $t = 0, 1, \ldots, T-1$ **do**
2: sample a cohort S_t of C clients with weights w_k
3: **for** client $k \in S_t$ **in parallel do**
4: download the prefix $\big(B^t_{\ell,[:,\,:r_k]}, A^t_{\ell,[:r_k,\,:]}\big)_\ell$; train it for τ local steps; upload $(\tilde{B}_{k,\ell}, \tilde{A}_{k,\ell})_\ell$
5: **end for**
6: **tail completion:** $\Delta_{k,\ell} \leftarrow s\big(\tilde{B}_{k,\ell}\tilde{A}_{k,\ell} - B^t_{\ell,[:,\,:r_k]}A^t_{\ell,[:r_k,\,:]}\big)$; stack $\mathcal{D}_k \in \mathbb{R}^{L\times m\times n}$ (3)
7: **shared bases:** $U_L, U_O, U_I \leftarrow$ leading eigenvectors of H_L, H_O, H_I (4)
8: **client cores:** $g_k \leftarrow \mathrm{vec}\big(\mathcal{D}_k \times_1 U_L^\top \times_2 U_O^\top \times_3 U_I^\top\big)$ (5)
9: **metric:** $M_k \leftarrow \lambda I + \mathrm{Diag}(q_k) + \mu v_k v_k^\top$ (6)
10: **consensus core:** $g^\star \leftarrow \big(\sum_k w_k M_k\big)^{-1} \sum_k w_k M_k g_k$ via the Woodbury identity (8)
11: **reconstruction:** $\mathcal{D}^\star \leftarrow \sum_k w_k \mathcal{D}_k + \mathcal{T}\big(g^\star - \sum_k w_k g_k\big)$ (9)
12: **write-back:** $(B_\ell^{t+1}, A_\ell^{t+1}) \leftarrow$ balanced rank-R SVD factors of $X_\ell^t + \mathcal{D}^\star[\ell, :, :]$ (10)
13: **end for**

Numerical conventions. The server runs in fp32 arithmetic on the CPU, with dense represented updates. The implementation uses $v_k = g_k/(\|g_k\| + \epsilon)$ with the machine epsilon ϵ of the server arithmetic ($\epsilon = 1.19 \times 10^{-7}$ for fp32), which differs from the unit vector of (6) by the factor $\|g_k\|/(\|g_k\| + \epsilon)$ and maps a zero update to $v_k = 0$; the metric and the right-hand side above are formed with this v_k, so the solve is exact for the implemented metric. The energy profile is defined piecewise,

$$q_k = \begin{cases} \min\{g_k^{\odot 2}/(\|g_k\|^2/p),\, c\} & \text{if } \|g_k\|^2/p > \epsilon, \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

so a client below the guard enters with the rank-one metric $\lambda I + \mu v_k v_k^\top$ alone. The guard is fixed by the arithmetic rather than tuned; because its threshold is absolute, whether it is active depends on the scale of the client updates, which differs across models and rounds, and when active it changes which terms of the metric are in effect. Table 21 reports its activation on the recorded events of Fed3DLoRA. Bit-for-bit reproducibility across thread counts is not claimed: with dense fp32 arithmetic, changing the number of CPU threads changes the output of one of the 27 recorded events by a relative 5.4×10^{-3}, because near-degenerate eigenvalues of the layer-mode Gram operator rotate the basis inside the retained subspace and the diagonal term of the metric is not invariant to such rotations; the same event agrees with an independent float64 implementation to 2.3×10^{-12}. Proposition 1 concerns re-parameterisations of the uploaded factors and does not extend to rotations of the shared bases.

Implementation defaults are summarised in Table 13.

B Experimental protocols and reproducibility

B.1 Datasets, splits and input formatting

Scoring. On both datasets an example is scored by the mean token negative log-likelihood of each candidate, end-of-sequence token included; accuracy is the fraction of examples whose gold candidate has the lowest mean token negative log-likelihood, and the choice NLL is the negative log-probability of the gold candidate after a softmax over the candidates' scores (lower is better). Training supervises response tokens only: each local step consumes 512 supervised response tokens, with a maximum sequence length of 512.

Table 7: Datasets, splits and evaluation roles. The diagnostic sets are held out from all training data and are never used for model or hyperparameter selection.

Dataset	Tasks / genres and format	Training pool	Evaluation sets
Commonsense8	ARC-Easy, ARC-Challenge, BoolQ, HellaSwag, OpenBookQA, PIQA, SocialIQA, WinoGrande; multiple choice with 2–4 candidates; tasks normalised to one format and globally deduplicated by input hash	frozen task-skewed partition of 48 clients (two tasks, 128 records each), of which 40 are used (partition seed 20260902)	held-out diagnostic set of 768 examples (96 per task)
MultiNLI	three-way inference over the genres fiction, government, slate, telephone and travel	40 clients built by genre, 960 training examples each (320 per label)	held-out diagnostic set of 1,500 examples (300 per genre); official matched (9,815) and mismatched (9,832) validation sets

B.2 Federated partitions and client participation

Commonsense8 clients come from the frozen task-skewed partition of Table 7: ten clients are taken from each of four task-pair profiles (the two ARC tasks; BoolQ and HellaSwag; OpenBookQA and PIQA; SocialIQA and WinoGrande), 40 in total. MultiNLI clients are built by genre, eight per genre; each client draws 80% of its 960 examples from its main genre and 5% from each of the other four. In every round the server samples 8 of the 40 clients; a run has 10 rounds. Aggregation weights are proportional to the local dataset sizes. We use multiple seeds (Table 9); each seed fixes the client data, the rank binding, the participation sequence, the initialisation and the evaluation examples, and within a seed every compared method uses the same values of all five, so all comparisons are paired. The component analysis of Table 3 uses a second batch with a different participation sequence (the mechanism batch, whose uploads were recorded for the analyses of Appendices C.4 and E); comparisons are paired within that batch.

B.3 Resource heterogeneity and rank assignment

Table 8: Client populations and rank assignments. Ranks are bound to clients once per seed and remain fixed across rounds; the server always keeps a rank-$R = 64$ adapter with mode ranks $(c_L, c_O, c_I) = (16, 128, 128)$.

Setting	Clients	Active ranks	Data assignment
Commonsense8, main (Table 1)	40, 8 per round	drawn from $\{8, 16, 32, 64\}$ independently of the task profile	two tasks per client, four task-pair profiles
MultiNLI, main (Table 2)	40, 8 per round	every genre holds the multiset $\{8, 8, 16, 32, 32, 32, 64, 64\}$; the ranks of every round sum to 256	one main genre per client (80%)
Factorial, homogeneous (Table 19)	40, 8 per round	32 for every client; the ranks of every round sum to 256	mixed or skewed
Factorial, heterogeneous (Table 19)	40, 8 per round	drawn from $\{8, 16, 32, 64\}$; the ranks of every round sum to 256	mixed or skewed

The active rank of a client determines what it downloads, trains and uploads ($r_k(m + n)$ numbers per adapted matrix per direction). It is a resource budget on client memory, compute and communication; it is not coupled to the client's data. The server rank R and the mode ranks are server-side quantities and identical in every setting; the factorial of Table 19 fixes the per-round rank budget at 256 for both the homogeneous and the heterogeneous population, so that the two differ in how the budget is distributed across clients.

B.4 BACKBONES, ADAPTERS AND TRAINING CONFIGURATION

Table 9: Backbones and local training configuration, identical for every method of a batch.

Item	Value
Backbones	Qwen3-0.6B-Base, Qwen3-1.7B-Base (Qwen Team, 2025); Llama-3.2-1B, Llama-3.2-3B (Grattafiori et al., 2024)
Adapted modules	query and value projections of every Transformer layer
Global rank, scaling	$R = 64$, $\alpha = 64$ ($s = 1$); client ranks as in Table 8
Precision	bf16 base model, fp32 adapters, fp32 server arithmetic
Local optimiser	AdamW (Loshchilov & Hutter, 2019), re-initialised at every participation (no optimiser state is carried across rounds)
Local budget	16 steps per participation, 512 supervised response tokens per step (8,192 per participation), maximum length 512
Learning rates	selected once per rank on a separate development split and frozen; shared by all methods of a batch (Table 10)
Rounds, seeds	10 rounds; seeds 11, 29, 47

Table 10: Local learning rates by client rank.

Dataset	Backbone	$r = 8$	$r = 16$	$r = 32$	$r = 64$
Commonsense8	Qwen3-0.6B	5×10^{-4}	2.5×10^{-4}	1.25×10^{-4}	6.25×10^{-5}
Commonsense8	Qwen3-1.7B	2×10^{-3}	1×10^{-3}	5×10^{-4}	2.5×10^{-4}
Commonsense8	Llama-3.2-1B	2×10^{-3}	1×10^{-3}	5×10^{-4}	2.5×10^{-4}
Commonsense8	Llama-3.2-3B	2×10^{-3}	1×10^{-3}	5×10^{-4}	2.5×10^{-4}
MultiNLI	Qwen3-0.6B	1.25×10^{-4}	1.25×10^{-4}	1.25×10^{-4}	1.25×10^{-4}
MultiNLI	Qwen3-1.7B	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
MultiNLI	Llama-3.2-1B	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}

B.5 BASELINE IMPLEMENTATIONS AND COMPARISON CONTROLS

All baselines run under the protocol of Section 4.1 with the same client data, rank bindings and participation sequences. Table 11 lists, for every method, what a client trains and uploads, how the server aggregates, and what a client of rank r receives.

FedIT requires a common rank, which we match to the parameter budget of the heterogeneous population; for GLoRA we use server rank 64 and core fraction 0.5. The internal controls change one server step of Fed3DLoRA each and, like the server-step arms of Table 18, are trained end to end: they share data partitions, rank assignments, initialisation and participation schedules, but because the server rule changes the next global adapter, the uploads of later rounds differ between arms. The fixed-input analyses of Appendix E.4 instead compare the consensus, its individual metric terms and the mean on the same recorded uploads of a round. The control without tail completion differs from the other controls in that it also changes the reference: it reads the returned factors as the whole adapter.

Table 11: Interfaces of the compared methods and internal controls. "Factor mean" averages B and A separately; "product mean" averages represented updates BA; "prefix" denotes the leading-r spectral components of the global adapter.

Method	Client trains / uploads	Server aggregation	Readout to a rank-r client
FedIT	common-rank factors	factor mean	common-rank factors
Zero-Padding	rank-r_k factors	factor mean after zero-padding to rank R	prefix of the padded mean
HetLoRA	rank-r_k factors with self-pruning	factor mean after padding, weights $\propto$ Frobenius norm of the client update	prefix
FlexLoRA	rank-r_k factors	product mean, SVD	leading components within rank r
FLoRA	rank-r_k factors	stacking (product equals the weighted sum of client updates), folded into the base weights	re-initialised adapter
FSLoRA	random sketch of r_k components of the rank-R global adapter	mean of the sparse factor deltas	sketch of r components
FedHera	rank-r_k factors; receive and train ranks per layer	aggregation in a global SVD basis	leading components within the receive rank
Fed-PLoRA	a selected subset of rank-one modules; the rest folded into the base weights	each module averaged over the clients that selected it	selected modules
GLoRA	gauge-fixed factors	aggregation in a consensus reference basis (server rank 64, core fraction 0.5)	readout from the spectral state
Product-space averaging (control)	prefix, rank-r_k factors	tail completion, product-space mean (Proposition 3)	prefix
Shuffled-metric consensus (control)	as Fed3DLoRA	fixed random coordinate permutation applied to M_k but not to g_k	prefix
Layer-wise consensus (control)	as Fed3DLoRA	layer-wise subspaces and metrics with the same total number of core coordinates	prefix
Without tail completion (control)	as Fed3DLoRA	no tail completion (returned factors read as the whole adapter)	prefix
Product-space averaging with step γ; FedExP	as product-space averaging	mean scaled by $\gamma \in \{1.5, 2, 3\}$ or by the FedExP adaptive step	prefix
Fed3DLoRA-NM (control)	as Fed3DLoRA	Fed3DLoRA update rescaled so that its written-back norm equals that of product-space averaging	prefix

B.6 Evaluation, model selection and statistical reporting

No result is selected on the diagnostic sets: learning rates were chosen on a separate development split before the main experiments, every table reports the final round, and the readout and fixed-input analyses characterise trained adapters and recorded uploads after the fact. Differences between methods are computed within seed; we report per-seed signs where they are available and do not claim statistical significance.

Table 12: Evaluation protocols of the experiment groups. All accuracies are final-round values of the global adapter unless stated.

Group	Measured object	Data	Reporting
Main results (Tables 1, 2, 16, 19)	rank-64 global adapter after round 10; base-model rows: pre-trained backbone without fine-tuning	Commonsense8 diagnostic set; MultiNLI matched and mismatched validation sets	mean $\pm$ sample SD over multiple seeds; base-model rows evaluated once
Component analysis (Tables 3 and 17)	as above; accuracy from the mechanism batch, choice NLL from an earlier batch	diagnostic sets	paired difference to each control, within seed; means over seeds
Server-step control (Table 18, Appendix D.3)	rank-64 global adapter after round 10	Commonsense8 diagnostic set (768)	per-seed values and means; pre-registered readings of Appendix E.2
Spectral-order readout control (Appendix E.3)	rank-r prefixes under spectral, previous-round and rotated orderings	diagnostic sets	post hoc; paired differences to spectral readout, medians over events
Numerical checks (Appendix C.4)	server operator replayed on recorded uploads	27 events $\times$ 2 families	relative errors, ranges and medians

B.7 METHOD-SPECIFIC HYPERPARAMETERS

Table 13: Hyperparameters of Fed3DLoRA and of the controls. All values are fixed across datasets, models and seeds; no task-specific override is used.

Quantity	Value	Role
mode ranks (c_L, c_O, c_I)	$(16, 128, 128)$	size of the shared subspace, $p = 262{,}144$ coordinates per family
Gram weighting	energy-weighted	w_k times the squared Frobenius energy of each client's update
λ	0.01	damping of every metric ($M_k \succeq \lambda I$)
μ	1	weight of the rank-one term $v_k v_k^\top$
c	10	truncation of the energy profile q_k
small-energy guard	fp32 machine epsilon	$q_k = 0$ when $\|g_k\|^2/p \le \epsilon$ (fixed by the arithmetic)
global rank R, α	64, 64	server adapter; $s = \alpha/R = 1$
aggregation weights w_k	$\propto \|D_k\|$	cohort weights in every mean and Gram operator
layer-wise consensus	layer-wise subspaces	same total number of core coordinates as (c_L, c_O, c_I)
scalar arms	$\gamma \in \{1.5, 2, 3\}$; FedExP	pre-registered diagnostic steps of Table 18
Fed3DLoRA-NM	one scalar per round	written-back update norm matched to that of product-space averaging, all adapted matrices
GLoRA	server rank 64, core fraction 0.5	baseline setting

B.8 HARDWARE, SOFTWARE AND MEASUREMENT PROTOCOLS

Table 14: Execution environment and measurement boundaries.

Item	Value
Server aggregation	dense fp32 on the CPU; the recorded audits use an Intel Xeon Gold 5318Y with 4 intra-op and 1 inter-op threads
GPU checks	NVIDIA A100 80 GB PCIe, TF32 disabled
Software	PyTorch 2.10.0 (CUDA 12.8 build)
Logged server time (Appendix F.2)	`aggregation_seconds` per round from the run logs of the mechanism batch, together with the eight clients' `elapsed_seconds`; thread counts were not logged
Timed replays (Appendix F.2)	CPU, 4 threads, fp32; one warm-up and five repetitions, medians reported; excludes logging I/O, model inference and input copies; tail completion performed before timing
Communication accounting	numbers transmitted per adapted matrix per direction, $r_k(m+n)$; bases and cores never leave the server

C ASSUMPTIONS, PROOFS AND NUMERICAL CHECKS

C.1 SCOPE AND ASSUMPTIONS

Throughout, $\langle \cdot, \cdot \rangle$ is the Frobenius inner product on the represented space (the concatenation of all adapted matrices), $\mathcal{T}$ is the lift of Section 3.2, which is an isometry because the three bases have orthonormal columns, $P = \mathcal{T}\mathcal{T}^*$, and $g_k = \mathcal{T}^*\mathcal{D}_k$. Propositions 1–3 hold for any orthonormal bases and any positive-definite metrics computed from the client cores, Proposition 4 requires in addition $\lambda I \preceq M_k \preceq (\lambda + \bar{c})I$, and Proposition 5 concerns the metric of Section 3.3 with $q_k = 0$ for every client; all five concern the server step itself, and rank truncation enters the convergence analysis only through Assumption 3. For the convergence result we write the server recursion as $X^{t+1} = X^t + \mathcal{D}_t^\star + e_t$, where e_t is the write-back residual, and $\mathbb{E}_t$ for the expectation conditional on X^t.

Assumption 1 (Smoothness)**.** *f is L_f-smooth in the represented space and bounded below by $f^\star$.*

Assumption 2 (Local updates)**.** *There are an effective step $\bar{\eta} > 0$ and constants $\beta \in [0, 1)$, $B, \sigma, D \geq 0$ such that the reference-corrected updates satisfy, for every round, (a) $\mathbb{E}_t[\bar{\mathcal{D}}_t] = -\bar{\eta}\,(\nabla f(X^t) + b_t)$ with $\|b_t\|_F \leq \beta\|\nabla f(X^t)\|_F + B$; (b) $\mathbb{E}_t\|\bar{\mathcal{D}}_t - \mathbb{E}_t\bar{\mathcal{D}}_t\|_F^2 \leq \bar{\eta}^2\sigma^2$; (c) $\mathbb{E}_t \sum_{k\in S_t} w_k\|\mathcal{D}_{k,t} - \bar{\mathcal{D}}_t\|_F^2 \leq \bar{\eta}^2 D^2$.*

Assumption 3 (Write-back residual)**.** *$\|e_t\|_F \leq \vartheta\|\mathcal{D}_t^\star\|_F$ for some $\vartheta \in [0, 1)$.*

Assumption 2 is the usual description of local training in biased-gradient analyses (Ajalloeian & Stich, 2020): the mean update is a step along the gradient up to a relative and an absolute bias, which absorb client drift over τ local steps, data heterogeneity and the rank budget, and (b)–(c) bound its variance and the dispersion within a cohort. It is not derived from the local optimiser (rank-wise AdamW on the factors); Theorem 1 is conditional on it. The truncated SVD guarantees $\vartheta \leq 1$ in Assumption 3: X^t has rank at most R, so the distance of $X^t + \mathcal{D}_t^\star$ to the set of rank-R matrices is at most $\|\mathcal{D}_t^\star\|_F$, and Π_R attains this distance (Eckart & Young, 1936). The strict inequality is an additional assumption on the trajectory: with $R = 1$, $X^t = \mathrm{Diag}(1, 0)$ and $\mathcal{D}_t^\star = \mathrm{Diag}(0, 0.1)$, truncation returns X^t and $\vartheta = 1$. On the recorded events ϑ_t lies between 0.09 and 0.30 (Table 15).

C.2 PROOFS OF PROPOSITIONS 1–5

Proof of Proposition 1. The uploads enter the server computation only through the products $\tilde{B}_{k,\ell}\tilde{A}_{k,\ell}$ in (3), and $(\tilde{B}_{k,\ell}Q)(Q^{-1}\tilde{A}_{k,\ell}) = \tilde{B}_{k,\ell}\tilde{A}_{k,\ell}$. Hence $\mathcal{D}_k$, the Gram operators, the bases, g_k, M_k, $g^\star$ and $\mathcal{D}^\star$ are unchanged, and so is $X_\ell^{t+1} = \Pi_R(X_\ell^t + \mathcal{D}^\star[\ell, :, :])$. The published factors are determined by X_ℓ^{t+1} up to the sign convention of the SVD (Appendix A.2). □

Proof of Proposition 2. If $\tilde{B}_{k,\ell}\tilde{A}_{k,\ell} = B^t_{\ell,[:,\,:r_k]}A^t_{\ell,[:r_k,\,:]}$, then $\Delta_{k,\ell} = 0$ by (3). If this holds for the whole cohort, then $\bar{\mathcal{D}} = 0$, $g_k = 0$ for all k, hence $g^\star = \bar{g} = 0$ and $\mathcal{D}^\star = 0$; since X_ℓ^t has rank at most

R, $\Pi_R(X_\ell^t) = X_\ell^t$. Without tail completion the server would use $X_{k,\ell}^{\mathrm{NT}} = s\tilde{B}_{k,\ell}\tilde{A}_{k,\ell} = X_{k,\ell} - T_{k,\ell}$, so every update is shifted by $-T_{k,\ell}$ and the mean by $-\sum_k w_k T_{k,\ell}$, whatever the data. □

Proof of Proposition 3. If $M_k = M$, then $g^\star = (\sum_k w_k M)^{-1} M \sum_k w_k g_k = \bar{g}$ because $\sum_k w_k = 1$, and (9) gives $\mathcal{D}^\star = \bar{\mathcal{D}}$. In general, $P\mathcal{T} = \mathcal{T}$ implies $(I-P)\mathcal{T}(g^\star - \bar{g}) = 0$, hence $(I-P)\mathcal{D}^\star = (I-P)\bar{\mathcal{D}}$. □

Proof of Proposition 4. Let $\bar{M} = \sum_k w_k M_k$ and $y_k = g_k - \bar{g}$. Since $\bar{M}^{-1}\sum_k w_k M_k \bar{g} = \bar{g}$, we have $a := g^\star - \bar{g} = \bar{M}^{-1}\sum_k w_k M_k y_k$. Then, by the Cauchy–Schwarz inequality,

$$a^\top \bar{M} a = \sum_k w_k \langle M_k^{1/2} a, M_k^{1/2} y_k \rangle \le \Big(\sum_k w_k a^\top M_k a\Big)^{1/2} \Big(\sum_k w_k y_k^\top M_k y_k\Big)^{1/2}$$
$$= \big(a^\top \bar{M} a\big)^{1/2} \Big(\sum_k w_k y_k^\top M_k y_k\Big)^{1/2},$$

so $a^\top \bar{M} a \le \sum_k w_k y_k^\top M_k y_k$. Using $\bar{M} \succeq \lambda I$ and $M_k \preceq (\lambda + \bar{c})I$ gives $\lambda \|a\|^2 \le (\lambda + \bar{c}) \sum_k w_k \|y_k\|^2$. Because $\mathcal{T}$ is an isometry, $\|\mathcal{D}^\star - \bar{\mathcal{D}}\|_F = \|a\|$ and $\|y_k\| = \|\mathcal{T}^*(\mathcal{D}_k - \bar{\mathcal{D}})\| = \|P(\mathcal{D}_k - \bar{\mathcal{D}})\|_F$, which proves the first claim.

For diagonal metrics, fix a coordinate i and write $m_k = \lambda + q_k[i]$, $\bar{m} = \sum_k w_k m_k$, $a_k = w_k m_k / \bar{m}$. Since $\sum_k (a_k - w_k)\bar{g}[i] = 0$,

$$\big(g^\star[i] - \bar{g}[i]\big)^2 = \Big(\sum_k (a_k - w_k)\, y_k[i]\Big)^2 \le \Big(\sum_k \frac{(a_k - w_k)^2}{w_k}\Big)\Big(\sum_k w_k\, y_k[i]^2\Big).$$

The first factor equals $\sum_k w_k (m_k/\bar{m} - 1)^2 = \mathrm{Var}_w(q[i])/(\lambda + \bar{q})^2$ with $\bar{q} = \sum_k w_k q_k[i]$. As $q_k[i] \in [0, c]$, the Bhatia–Davis inequality (Bhatia & Davis, 2000) gives $\mathrm{Var}_w(q[i]) \le \bar{q}(c - \bar{q})$, and maximising $u(c-u)/(\lambda+u)^2$ over $u \in [0, c]$ (the maximiser is $u = c\lambda/(c+2\lambda)$) yields the bound $c^2/(4\lambda(c+\lambda))$. The same factor also equals $\sum_k a_k(a_k/w_k) - 1 \le \max_k a_k/w_k - 1 \le 1/\min_k w_k - 1$ because $a_k \le 1$. Summing over i gives the second claim. □

Proof of Proposition 5. Write $V = [v_1, \ldots, v_C]$, $n_k = \|g_k\|$, $n = (n_1, \ldots, n_C)$, $W = \mathrm{Diag}(w)$ and $G = V^\top V$. With $q_k = 0$, $M_k = \lambda I + \mu v_k v_k^\top$, so $\sum_k w_k M_k = \lambda I + \mu V W V^\top$ and $\sum_k w_k M_k g_k = \sum_k w_k (\lambda + \mu) n_k v_k = (\lambda + \mu) V (w \odot n)$, using $v_k^\top g_k = n_k$. The push-through identity $(\lambda I + \mu V W V^\top)^{-1} V = V(\lambda I + \mu W V^\top V)^{-1}$ gives the closed form $g^\star = Va$ with $a = (\lambda + \mu)(\lambda I + \mu W G)^{-1}(w \odot n)$. If all $v_k = v$, then $M_k = \lambda I + \mu v v^\top$ is the same for every client, and Proposition 3 gives $g^\star = \bar{g}$ whatever the norms n_k. If the v_k are orthogonal, $G = I$ and $(\lambda I + \mu W)^{-1}$ is diagonal with entries $1/(\lambda + \mu w_k)$. If all $n_k = n$ and $G = (1 - \bar{c})I + \bar{c}\mathbf{1}\mathbf{1}^\top$ with $w_k = 1/C$, then $w \odot n = (n/C)\mathbf{1}$ and $\mathbf{1}$ is an eigenvector of $\lambda I + \mu W G$ with eigenvalue $\lambda + \mu(1 + (C-1)\bar{c})/C$, so $g^\star = \rho(\bar{c})\,(n/C) V \mathbf{1} = \rho(\bar{c})\,\bar{g}$. □

Remark 1 (Masked averaging as a limit). Let $\mu = 0$ and suppose that only the clients in $S_i \subset S_t$ have a non-zero i-th core coordinate. The uniform mean returns $\sum_{k \in S_i} w_k g_k[i]$, which is diluted by the mass of $S_t \setminus S_i$. As $\lambda \to 0$, the diagonal-term weights of §3.3 return $\sum_{k \in S_i} w_k q_k[i] g_k[i] / \sum_{k \in S_i} w_k q_k[i]$, an average over S_i only. Fed3DLoRA thus behaves as a soft, gauge-invariant form of support-aware averaging.

C.3 Proof of Theorem 1

Proof of Theorem 1. Let $G = \|\nabla f(X^t)\|_F$, $r_t = \mathcal{D}_t^\star - \bar{\mathcal{D}}_t$ and $\delta_t^2 = \sum_k w_k \|P(\mathcal{D}_{k,t} - \bar{\mathcal{D}}_t)\|_F^2$. By Proposition 4, $\|r_t\|_F \le \chi \delta_t$, and by Assumption 2(c) and the non-expansiveness of P, $\mathbb{E}_t \delta_t \le (\mathbb{E}_t \delta_t^2)^{1/2} \le \bar{\eta} D$. Smoothness gives

$$f(X^{t+1}) \le f(X^t) + \big\langle \nabla f(X^t), \bar{\mathcal{D}}_t + r_t + e_t \big\rangle + \tfrac{L_f}{2} \|\mathcal{D}_t^\star + e_t\|_F^2.$$

Linear terms. By Assumption 2(a), $\mathbb{E}_t \langle \nabla f, \bar{\mathcal{D}}_t \rangle \le -\bar{\eta}(1-\beta)G^2 + \bar{\eta} B G$, and $\mathbb{E}_t \langle \nabla f, r_t \rangle \le G\chi\, \mathbb{E}_t \delta_t \le \bar{\eta}\chi D G$. Moreover $\mathbb{E}_t \|\bar{\mathcal{D}}_t\|_F \le \|\mathbb{E}_t \bar{\mathcal{D}}_t\|_F + (\mathbb{E}_t \|\bar{\mathcal{D}}_t - \mathbb{E}_t \bar{\mathcal{D}}_t\|_F^2)^{1/2} \le \bar{\eta}((1+\beta)G + B + \sigma)$, hence by Assumption 3

$$\mathbb{E}_t \langle \nabla f, e_t \rangle \le G\vartheta\, \mathbb{E}_t \|\mathcal{D}_t^\star\|_F \le G\vartheta \big(\mathbb{E}_t \|\bar{\mathcal{D}}_t\|_F + \chi \mathbb{E}_t \delta_t\big) \le \bar{\eta}\vartheta(1+\beta)G^2 + \bar{\eta}\vartheta(B + \sigma + \chi D)G.$$

Adding the three bounds, with $\omega = \beta + \vartheta(1+\beta)$ and $\theta = (1+\vartheta)(B+\chi D)+\vartheta\sigma$, the linear terms are at most $-\bar{\eta}(1-\omega)G^2+\bar{\eta}\theta G \le -\frac{3}{4}\bar{\eta}(1-\omega)G^2+\bar{\eta}\theta^2/(1-\omega)$, where we used $\theta G \le \frac{1-\omega}{4}G^2+\frac{\theta^2}{1-\omega}$.

Quadratic term. $\|\mathcal{D}_t^\star + e_t\|_F \le (1+\vartheta)\|\mathcal{D}_t^\star\|_F$ and $\mathbb{E}_t\|\mathcal{D}_t^\star\|_F^2 \le 2\mathbb{E}_t\|\bar{\mathcal{D}}_t\|_F^2 + 2\chi^2\mathbb{E}_t\delta_t^2$, with $\mathbb{E}_t\|\bar{\mathcal{D}}_t\|_F^2 = \|\mathbb{E}_t\bar{\mathcal{D}}_t\|_F^2 + \mathbb{E}_t\|\bar{\mathcal{D}}_t - \mathbb{E}_t\bar{\mathcal{D}}_t\|_F^2 \le \bar{\eta}^2\big(2(1+\beta)^2G^2 + 2B^2 + \sigma^2\big)$. Therefore

$$\tfrac{L_f}{2}\mathbb{E}_t\|\mathcal{D}_t^\star + e_t\|_F^2 \le L_f\bar{\eta}^2(1+\vartheta)^2\big(2(1+\beta)^2G^2 + 2B^2 + \sigma^2 + \chi^2D^2\big) \le \tfrac{1}{4}\bar{\eta}(1-\omega)G^2 + L_f\bar{\eta}^2\Psi,$$

where the last step uses the step-size condition and $\Psi = (1+\vartheta)^2(2B^2+\sigma^2+\chi^2D^2)$.

Conclusion. Combining, $\mathbb{E}_t f(X^{t+1}) \le f(X^t) - \frac{1}{2}\bar{\eta}(1-\omega)G^2 + \bar{\eta}\theta^2/(1-\omega) + L_f\bar{\eta}^2\Psi$. Taking total expectations, summing over $t = 0, \dots, T-1$, using $f \ge f^\star$ and dividing by $\frac{1}{2}\bar{\eta}(1-\omega)T$ gives (11) with $\Theta = \theta^2$. □

C.4 Numerical equivalence and known-answer checks on real uploads

All checks in this section replay the server operator on the recorded uploads of the mechanism batch (9 runs of Fed3DLoRA, rounds 1, 5 and 10, i.e. 27 events, query and value families separately) with the production configuration; the small-energy guard of Appendix A is pinned to the fp32 machine epsilon so that the active-proxy pattern matches the published runs exactly (432 of 432 client–family observations), and inner products and norms are accumulated in float64. Table 15 gives one measured quantity per statement of §3.5; Figure 4 shows how the amplification varies with the agreement of the cohort, measured by the mean $\bar{c} = \frac{2}{C(C-1)}\sum_{k<j}\langle g_k, g_j\rangle/(\|g_k\|\|g_j\|)$ over the off-diagonal client pairs, next to the symmetric case of Proposition 5. Both rules of the figure are evaluated on the same 54 event–family inputs, so each pair of points shares its abscissa. On these inputs, every rank-one point lies at or above the symmetric reference (54 of 54); where the diagonal term is active for at least one client, Fed3DLoRA has the smaller amplification (42 of 42); and where every client has $q_k = 0$, the two outputs coincide exactly (12 of 12). The known-answer checks of the coordinate-level analysis (Appendix E.4), which require the product-space mean to have gain 1 in every bin and every scalar control to have its scalar, pass in all 2,880 instances, and all rules of that analysis are evaluated on identical bins (the coordinate counts agree in all 864 bin groups).

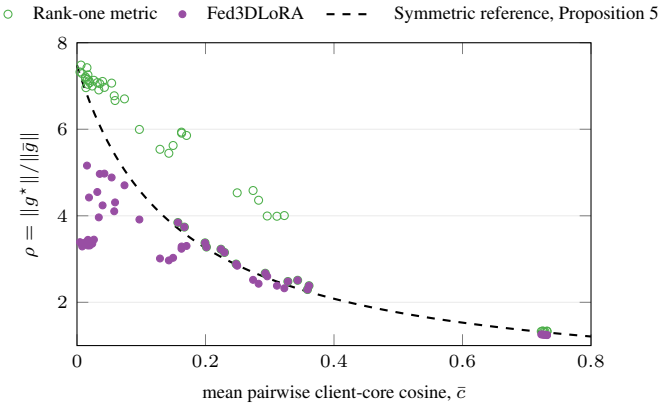

Figure 4: Core-norm amplification versus mean pairwise client-core cosine. Both aggregation rules are evaluated on the same 54 event–family inputs (27 run–round events, query and value). The dashed curve is the symmetric reference of Proposition 5 ($C = 8$, $\lambda = 0.01$, $\mu = 1$), not an exact prediction for arbitrary cohorts.

Table 15: Numerical checks of the statements of §3.5. Relative errors are Frobenius norms of the represented update over all layers of a family unless stated.

Statement	Check	Result (54 event–families)
Prop. 1, gauge invariance	every client prefix re-parameterised as $(\tilde{B}Q, Q^{-1}\tilde{A})$ with random invertible Q (condition number ≤ 10); relative change of the output	Fed3DLoRA: at most 4.0×10^{-12}. Zero-padded factor averaging: 0.88–1.05; FedIT-style equal-rank factor averaging: 0.89–1.34
Prop. 2, zero-update fixed point	every client returns its prefix unchanged; drift $\|X^{t+1} - X^t\|/\|X^t\|$; retention coefficient of component j under the uncompleted mean vs. $\sum_{k:r_k>j} w_k$	drift 0 before write-back, 1.2×10^{-6} after the fp32 SVD; coefficients match to 3.7×10^{-12} (18 events, rounds 5 and 10)
Prop. 3, exact reduction	all clients given the same metric; relative distance of the pre-write-back output to the plain mean	median 2.0×10^{-8}, maximum 6.3×10^{-8} (27 events). With equal scalar weights but client-specific metrics the distance is 0.31 (median)
Prop. 4, bounded deviation	$\|\mathcal{D}^\star - \bar{\mathcal{D}}\|_F/(\sum_k w_k \|P(\mathcal{D}_k - \bar{\mathcal{D}})\|_F^2)^{1/2}$ vs. $\chi = \sqrt{(\lambda + c + \mu)/\lambda} = 33.2$	0.87–1.69 (median 1.18; round 1 ≈ 1.05, rounds 5 and 10 ≈ 1.34); 54/54 within the bound
Prop. 5, closed form	rank-one consensus of the implementation vs. the closed form; Spearman correlation between the mean pairwise client cosine $\bar{c}$ and $\rho = \|g^\star\|/\|\bar{g}\|$ across the 27 events	agreement 1.9×10^{-13} in float64 (4.2×10^{-4} in fp32); Spearman -0.95 (query) and -0.96 (value) for the rank-one consensus, -0.78 / -0.80 for Fed3DLoRA (Figure 4)
Assumption 3, write-back residual	$\vartheta_t = \|X_t^\star - \Pi_R(X_t^\star)\|_F/\|\mathcal{D}_t^\star\|_F$ per event	0.09–0.30 (round 1 $\approx$ 0.09–0.11; rounds 5 and 10 0.10–0.30); largest single layer 0.33; the rank-64 write-back retains 98.9–99.98% of the energy of $X^\star$

D Complete benchmark results

D.1 Reference correction on the main settings

Table 16 compares Fed3DLoRA with its variant without tail completion, which reads each returned product as the complete adapter, on the evaluation sets of Tables 1 and 2. Reference correction gives the higher mean accuracy in all eight columns.

Table 16: Final-round accuracy (%) of Fed3DLoRA with and without tail completion on the main settings, mean ± SD over multiple seeds (Commonsense8 diagnostic set; MultiNLI matched and mismatched validation sets).

Method	Commonsense8				MultiNLI / Qwen3-0.6B		MultiNLI / Llama-3.2-1B	
	Qwen3-0.6B	Qwen3-1.7B	Llama-3.2-1B	Llama-3.2-3B	matched	mismatched	matched	mismatched
Fed3DLoRA	60.03±0.60	68.79±0.27	60.89±0.96	69.49±0.21	84.30±0.59	84.65±0.32	84.54±0.31	85.21±0.10
without tail completion	59.41±0.57	68.36±0.23	60.16±0.45	69.05±0.15	83.26±0.64	83.66±0.54	83.94±0.09	84.91±0.19

D.2 Component analysis: choice NLL from the earlier batch

Table 17 reports the choice-NLL differences of the four internal controls from an earlier training batch of the same controls, and it cannot be paired row by row with the accuracy differences of Table 3, which come from the mechanism batch (Appendix B.2). A difference below zero means that Fed3DLoRA has the lower NLL. Fed3DLoRA has lower choice NLL in 12 of the 16 comparisons; the remaining four favour a control.

Table 17: Paired final-round choice-NLL differences, Fed3DLoRA minus control, mean over multiple seeds, earlier batch of the same controls; below zero means Fed3DLoRA has the lower NLL (green), above zero the control (red).

Fed3DLoRA vs.	Commonsense8		MultiNLI	
	Qwen3-0.6B	Qwen3-1.7B	Qwen3-0.6B	Qwen3-1.7B
Layer-wise consensus	−0.0089	−0.0001	+0.0352	−0.0045
Product-space averaging	−0.0022	−0.0257	−0.0305	+0.0001
Shuffled-metric consensus	−0.0015	+0.0223	−0.0285	−0.0054
Without tail completion	−0.0004	+0.0065	−0.0204	−0.0161

D.3 Server-step control: per-seed values

Table 18: Per-seed final-round accuracy (%, higher is better) and choice NLL (lower is better) for the arms of the server-step control (Commonsense8 / Qwen3-1.7B, diagnostic set, 768 examples; one run per seed).

	Accuracy (%, ↑)				Choice NLL (↓)			
Arm	run 1	run 2	run 3	mean	run 1	run 2	run 3	mean
Product-space averaging, $\gamma = 1$	63.93	66.02	65.23	65.06	0.9404	0.9596	0.9485	0.9495
Product-space averaging, $\gamma = 1.5$	66.41	65.49	66.28	66.06	0.9229	0.9495	0.9279	0.9334
Product-space averaging, $\gamma = 2$	64.45	65.49	62.76	64.24	0.9563	0.9615	0.9745	0.9641
Product-space averaging, $\gamma = 3$	61.33	63.54	60.29	61.72	1.1267	0.9117	1.0376	1.0253
Product-space averaging, FedExP	61.07	63.02	60.42	61.50	1.0477	0.9471	0.9960	0.9970
Fed3DLoRA	66.41	69.01	66.02	67.14	0.9136	0.8915	0.9152	0.9068
Fed3DLoRA-NM	68.62	65.62	67.97	67.40	0.8158	0.8549	0.8231	0.8312

D.4 Rank and data heterogeneity: full table

Table 19: Capacity-matched rank × data factorial on Commonsense8 (rank budget per round 256), plotted in Figure 3: homogeneous rank 32 vs. heterogeneous $\{8, 16, 32, 64\}$, mixed vs. skewed client data; final-round accuracy (%).

Model	Method	homo / mixed	hetero / mixed	homo / skewed	hetero / skewed
Qwen3-0.6B	FSLoRA	58.64±0.27	57.68±0.39	58.94±0.30	59.07±0.53
	Fed3DLoRA without tail completion	**59.94±0.60**	59.20±0.08	59.85±0.38	59.94±0.99
	Fed3DLoRA	59.68±0.15	**59.24±0.13**	**60.94±0.91**	**60.37±0.85**
Llama-3.2-1B	FSLoRA	59.03±0.96	59.16±0.74	59.33±0.40	58.90±0.33
	Fed3DLoRA without tail completion	59.42±0.38	59.64±1.24	59.37±1.50	60.46±0.72
	Fed3DLoRA	**59.64±0.26**	**59.72±1.63**	**59.38±0.47**	**60.50±1.37**

E Controlled analyses of the server update

E.1 Summary of exceptions

The analyses below distinguish end-to-end training from the behaviour of a single aggregation on fixed inputs. Their outcomes depend on the model, on the stage of training and on whether the diagonal term of the metric is active, so not every comparison favours Fed3DLoRA. Each exception is reported with the corresponding experiment, in Appendix D.2 and Appendices E.2–E.4.

E.2 Scalar step-size controls and matched-norm training

The server-step control of Table 18 trains seven arms end to end on Commonsense8 / Qwen3-1.7B under the protocol of Appendix B: the product-space mean with server step $\gamma \in \{1, 1.5, 2, 3\}$, the mean with the FedExP adaptive step (Jhunjhunwala et al., 2023), Fed3DLoRA, and Fed3DLoRA-NM. Fed3DLoRA-NM multiplies the Fed3DLoRA update in every round by one scalar, found by bisection along the rank-R write-back path, so that the written-back update has the Frobenius norm, over all adapted matrices of both projection families, of the written-back product-space-averaging update.

Three readings were fixed before the runs, each counted over seeds. (H) The $\gamma = 3$ arm or the FedExP arm is below the mean ($\gamma = 1$) in at least two seeds; it holds in every seed. (S) The best scalar arm ($\gamma \in \{1.5, 2, 3\}$ or FedExP) is at least Fed3DLoRA in at least two seeds; it holds in one seed outright and in a second only through an exact tie at $\gamma = 1.5$. (D) The norm-matched arm is at least the mean and Fed3DLoRA is at least every scalar arm, each in at least two seeds; the first clause holds in two seeds, and the second holds in one seed outright and in a second only through the same tie, so D also depends on it.

On this setting a moderate step on the mean helps, larger tested steps and FedExP reduce accuracy, and none of the tested scalar-step rules reaches the mean final-round accuracy of Fed3DLoRA, which is above the mean in every seed. Fed3DLoRA-NM has the lowest choice NLL of all arms in every seed, while its accuracy relative to Fed3DLoRA changes sign across seeds (Table 18).

E.3 Spectral ordering and prefix stability

Order of the components matters. Table 20 evaluates the rank-r prefix of the published rank-64 adapter of each event on the diagnostic set under three orderings of its components: the spectral order of (10), the order of the previous round matched by subspace overlap, and a random orthogonal rotation of the factors that preserves the full rank-64 product but destroys the nesting. Reading a prefix from rotated factors loses accuracy at $r = 8$ in every setting, whereas the previous-round order stays close to the spectral order. The exception is Qwen3-1.7B at rounds 5 and 10, where the rotated rank-32 prefix has higher accuracy and lower choice NLL than the spectral one; a rotated prefix is, in expectation, a scaled-down copy of the full adapter, and on this model a smaller update also has the lower choice NLL (Appendix E.2).

Table 20: Readout control: paired difference to the spectral order (accuracy in points, choice NLL; positive accuracy and a NLL difference below zero, i.e. lower NLL, favour the alternative order), median over events.

		rotation, $r = 8$		rotation, $r = 32$		previous order, $r = 8$	
Setting	rounds	ΔAcc	ΔNLL	ΔAcc	ΔNLL	ΔAcc	ΔNLL
CS8 / Qwen3-0.6B	1	-7.81	$+0.072$	-4.04	$+0.049$	–	–
CS8 / Qwen3-0.6B	5, 10	-8.20	$+0.054$	-1.50	-0.003	$+0.07$	-0.00004
CS8 / Qwen3-1.7B	1	-4.95	$+0.073$	0.00	-0.011	–	–
CS8 / Qwen3-1.7B	5, 10	-3.19	$+0.009$	$+2.67$	-0.089	$+0.07$	-0.00011
MNLI / Qwen3-0.6B	1	-34.20	$+0.306$	-12.93	$+0.090$	–	–
MNLI / Qwen3-0.6B	5, 10	-35.23	$+0.355$	-5.10	$+0.091$	-0.07	-0.00045

The nested prefix is stable across rounds. For the published adapters of consecutive rounds, the mean squared cosine between the leading-r left (right) singular subspaces is 0.95 (0.96) at $r = 8$ and 0.89 (0.91) at $r = 32$ on Commonsense8 / Qwen3-0.6B, 0.95 (0.93) and 0.89 (0.88) on Qwen3-1.7B, and 0.98 (0.99) and 0.98 (0.98) on MultiNLI, medians over layers and seeds; on Commonsense8 the overlap rises from about 0.69 between rounds 1 and 2 to 0.97 between rounds 9 and 10. The leading 8 components hold 88–91% of the energy of the published adapter and the leading 32 hold 97.5–98.8%, so a rank-8 client downloads a prefix that changes little from round to round and already carries most of the adapter.

E.4 Coordinate-level selectivity and proxy activity

Measurement. For every event (run $\times$ round $\in \{1, 5, 10\}$) and projection family, the client cores g_k of the cohort are computed by the production pipeline in float64 arithmetic, with the small-energy guard of Appendix A pinned to the fp32 machine epsilon so that the active-proxy pattern matches the published runs (checked in Appendix C.4). For coordinate i, the relative energy of client k is $e_k[i] = g_k[i]^2 / \mathrm{mean}_j g_k[j]^2$, i.e. $q_k[i]$ before clipping and gating, and the effective client support is $n_{\mathrm{eff}}[i] = (\sum_k w_k e_k[i])^2 / \sum_k (w_k e_k[i])^2 \in [1, C]$, and $C_{\mathrm{eff}} = 1/\sum_k w_k^2$ is the effective number of clients. In our experiments, the client weights are equal, so $C_{\mathrm{eff}} = C = 8$. Coordinates with $\sum_k w_k e_k[i] = 0$ or $\sum_k w_k |g_k[i]| = 0$ are dropped, and a coordinate is low-conflict if $1 - |\sum_k w_k g_k[i]| / \sum_k w_k |g_k[i]| \le 0.5$. The bins in n_{eff} have edges $1, 1.5, 2.5, \ldots, 6.5, 7.5, 8$, are closed on the right with $n_{\mathrm{eff}} = 1$ in the first bin, and are placed at their centres; a bin is reportable if it holds at least 100 coordinates and at least 0.1% of the total energy. For a fixed event and projection

Table 21: Activity of the diagonal term: fraction of client–family observations with $q_k = 0$ (24 client observations and 2 families per cell) and median of $(\|g_k\|^2/p)/\epsilon$ with its range.

Setting	Round	$q_k = 0$ (client–families)	$(\|g_k\|^2/p)/\epsilon$, median [min, max]
CS8 / Qwen3-0.6B	1	12.5%	6.0 [0.47, 48.7]
CS8 / Qwen3-0.6B	5	45.8%	1.4 [0.072, 31.3]
CS8 / Qwen3-0.6B	10	37.5%	1.96 [0.147, 39.8]
CS8 / Qwen3-1.7B	1	0.0%	192 [3.37, 1840]
CS8 / Qwen3-1.7B	5	0.0%	76.4 [7.6, 1670]
CS8 / Qwen3-1.7B	10	0.0%	109 [11, 1830]
MNLI / Qwen3-0.6B	1	0.0%	3.26 [1.88, 5.76]
MNLI / Qwen3-0.6B	5	100.0%	0.061 [0.023, 0.164]
MNLI / Qwen3-0.6B	10	100.0%	0.104 [0.039, 0.303]

family, the gain of a rule with core output h in bin b is the least-squares coefficient without an intercept, fitted between h and the mean core $\bar{g}$ on the same retained coordinates $\mathcal{I}_b$,

$$G_b(h) = \frac{\sum_{i\in\mathcal{I}_b} h[i]\,\bar{g}[i]}{\sum_{i\in\mathcal{I}_b} \bar{g}[i]^2},$$

reported as missing if the denominator is below $10^{-12}\sum_i \bar{g}[i]^2$. All rules are evaluated on the same cores, bins and masks. Figure 2 shows, for Commonsense8 / Qwen3-1.7B, the query family and low-conflict coordinates, the median over multiple seeds of the per-seed gains $G_{b,s}$; a bin is drawn only if it is reportable in every seed, which holds for the six bins up to $n_{\text{eff}} = 6.5$ in every round (927 to 44,294 coordinates per bin and seed), whereas the two highest bins hold at most 244 coordinates or less than 0.1% of the energy. The figure has no error bars.

Controls and reference in Figure 2. Product-space averaging is $h = \bar{g}$, and the rescaled mean is $h = a\bar{g}$ with $a = \|g^\star\|_2/\|\bar{g}\|_2$, computed for each event and projection family from the Fed3DLoRA consensus core ($\|\bar{g}\|_2 > 0$ in every event–family). By construction $G_b(\bar{g}) = 1$ and $G_b(a\bar{g}) = a$ in every bin; these known answers are checked in Appendix C.4. The rank-one, diagonal and full-metric curves are computed from the actual consensus outputs of (8) and are not constrained to be flat. The rescaled mean acts in the core space and differs from Fed3DLoRA-NM of Table 5, which rescales the Fed3DLoRA update over all adapted matrices during training (Appendix E.2). The reference of bin b is $C_{\text{eff}}/\bar{n}_b$ with $\bar{n}_b = \sum_{i\in\mathcal{I}_b} E[i]\,n_{\text{eff}}[i]/\sum_{i\in\mathcal{I}_b} E[i]$ and $E[i] = \sum_k w_k g_k[i]^2$, the gain of averaging only over $\bar{n}_b$ carriers of equal weight and energy. It is an idealised reference rather than a prediction of the diagonal-term output, because the metric uses the clipped and gated q_k instead of e_k.

Verdicts. An event–family passes if the Spearman correlation between bin centre and gain is at most -0.8 and the lowest bin gains at least $1.5\times$ the highest bin. Rounds 5 and 10: Qwen3-1.7B / Commonsense8 11 of 12 (query 6/6, value 5/6); Qwen3-0.6B / Commonsense8 3 of 12; on MultiNLI / Qwen3-0.6B the diagonal term is inactive for every client of every cohort at rounds 5 and 10, so no criterion applies and the curve is flat (median ratio between the largest and smallest bin gain 1.10). Round 1 is reported separately: there the criterion holds in every event–family of every setting, with the highest-bin gain between 1.0 and 1.3. On MultiNLI at rounds 5 and 10 the output of Fed3DLoRA coincides with that of the rank-one term alone and lies in the span of the client cores (out-of-span fraction $\leq 10^{-14}$), yet it is not a scalar multiple of the mean: in the medians over multiple seeds of each round and family, the coefficient of variation of the client coefficients is 0.32–0.54, and the norm of the component orthogonal to the mean core, $h - (\langle h, \bar{g}\rangle/\|\bar{g}\|^2)\bar{g}$, is 0.52–0.80 times the norm of the mean core, i.e. $\rho\sqrt{1-\kappa^2}$. Where the diagonal term is active, the median fraction of the norm of the output that lies outside the span of the client cores is 0.49–0.61 on Commonsense8 and 0.35–0.36 on MultiNLI at round 1.

When the diagonal term is active. Table 21 records, for the 27 events, the fraction of client–family observations in which the small-energy guard sets $q_k = 0$, together with the ratio of the mean squared core coordinate to the fp32 machine epsilon. At the recorded rounds the guard is not active on Qwen3-1.7B, is active for part of the Commonsense8 / Qwen3-0.6B cohorts (12.5% of client–families at round 1, 38–46% at rounds 5 and 10), and is active for every client on MultiNLI / Qwen3-0.6B at rounds 5 and 10, where the local updates of a genre-homogeneous client are small. In this regime, each client uses the isotropically damped rank-one metric $\lambda I + \mu v_k v_k^\top$; for fixed λ and μ, the combination coefficients depend on the client norms, aggregation weights and cosine matrix, as characterised by Proposition 5.

Energy captured by the shared subspace. With mode ranks $(16, 128, 128)$ the shared subspace retains, of the weighted energy of the cohort's own updates, a median of 32–57% on Commonsense8 / Qwen3-0.6B (query 32.5, 50.3, 51.6% and value 42.2, 51.9, 56.6% at rounds 1, 5, 10), 37–65% on Commonsense8 / Qwen3-1.7B (query 47.9, 40.2, 37.3%; value 64.5, 44.2, 43.8%) and 40–42% on MultiNLI / Qwen3-0.6B (query 40.4, 40.1, 39.7%; value 42.1, 41.3, 41.8%). Components outside the subspace are aggregated by their product-space mean (Proposition 3): the shared subspace limits where aggregation may depart from averaging, and the orthogonal-complement component of the product-space mean is preserved before rank truncation.

E.5 Client agreement and task profiles

Table 4 and Figure 4 summarise the aggregate geometry; here we examine its relation to client task profiles. Figure 5 shows the matrices behind the $\bar{c}$ column of Table 4 for one seed of each Commonsense8 setting: the cosine matrix G of the client cores, through which the rank-one term depends on the client directions (Proposition 5). Clients are ordered by task-pair profile (Appendix B.2), and in the cohorts shown the two clients of each profile hold different ranks. At round 1 the clients of a profile form clear blocks; at rounds 5 and 10 agreement is lower throughout, as in Table 4. In all 36 event–families of the two Commonsense8 settings, the mean cosine between clients of the same profile exceeds the mean cosine between clients of different profiles, by at least 0.019 (Figure 6). On Commonsense8, the client agreement that enters the rank-one term therefore reflects the task profiles of the cohort, and clients of different rank are compared in the same coordinates.

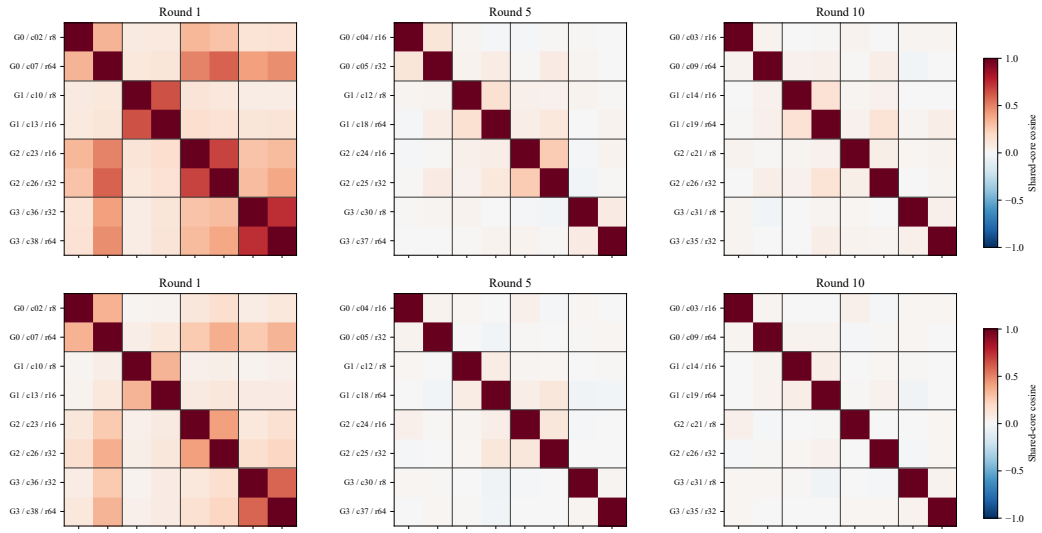

Figure 5: Cosine matrix G of the client cores on Commonsense8 at rounds 1, 5 and 10 (query family, one seed; top: Qwen3-0.6B, bottom: Qwen3-1.7B). Rows are the eight clients of the round, labelled by task-pair profile, client index and rank, and columns follow the same order; lines separate the profiles, and the diagonal is 1 by definition. Profiles: G0 ARC-Challenge / ARC-Easy, G1 BoolQ / HellaSwag, G2 OpenBookQA / PIQA, G3 SocialIQA / WinoGrande. The mean of the off-diagonal entries is the per-event value summarised by $\bar{c}$ in Table 4.

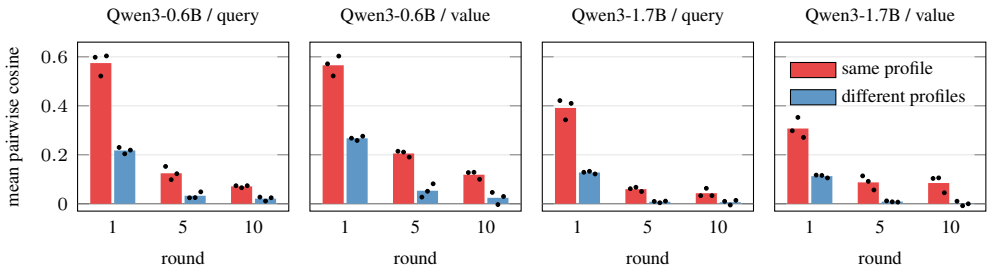

Figure 6: Mean cosine between the cores of clients with the same task-pair profile and with different profiles on Commonsense8, per round and family. Bars are means over multiple seeds, dots the individual seeds; every event has four same-profile and 24 cross-profile client pairs.

F Computation, communication and storage

F.1 Accounting definitions

Communication is counted in numbers transmitted per adapted matrix per direction; for a rank-r_k client this is $r_k(m+n)$ in both directions, and neither the mode bases nor the cores leave the server. Client computation is that of rank-r_k LoRA training and does not depend on the server rule. Server computation per round consists of tail completion, the three Gram operators and their eigendecompositions, the C projections, the $C \times C$ solve of Appendix A.2, the reconstruction and the L truncated SVDs of the write-back. Storage of the shared bases is $Lc_L + mc_O + nc_I$ numbers per family, against $L(mc_O + nc_I c_L/L)$ for layer-wise bases with the same number of core coordinates.

F.2 Server time

Table 22: Logged server time of the mechanism batch: median `aggregation_seconds` per round (30 rounds per setting) and its ratio to the summed and to the largest local training time of the eight clients of the round.

Setting	seconds per round	of summed client time	of slowest client
CS8 / Qwen3-0.6B	19.27	24.9%	161%
CS8 / Qwen3-1.7B	62.48	42.0%	298%
MNLI / Qwen3-0.6B	19.38	10.9%	71%

Table 23: Timed CPU replay of one aggregation of both families per model (4 threads, fp32, median of five repetitions): total time of Fed3DLoRA, the share of the SVD write-back, the time of the shared-subspace stages (bases, cores, metrics, consensus), and the same replay of the product-space mean with the same dense-SVD write-back.

Model	Fed3DLoRA (s)	SVD write-back share	subspace stages (s)	product-space mean (s)
Qwen3-0.6B-Base	32.4	29.6%	17.6	9.8
Qwen3-1.7B	81.8	36.1%	40.9	30.0
Llama-3.2-1B	37.9	35.8%	19.1	13.6
Llama-3.2-3B	201.6	37.7%	100.6	77.4

Both tables time the dense, correctness-first CPU implementation of Appendix A.3, whose server time does not depend on the local data or on the number of local steps. Table 22 reports the aggregation time logged during training, relative to the local training time of the clients of the same round; Table 23 replays single aggregations under controlled conditions and compares Fed3DLoRA with the product-space mean under the same dense-SVD write-back. The additional work of Fed3DLoRA lies in the shared-subspace stages (Gram eigendecompositions, projections, metrics and consensus), which make one aggregation 2.6–3.3$\times$ slower than the product-space mean. The logged server time is a sizeable fraction of the summed client time because the local training budget of the protocol is small (8,192 supervised tokens per participation).

F.3 STORAGE OF THE SHARED BASES

With $(c_L, c_O, c_I) = (16, 128, 128)$ the shared bases of both families occupy 2.5 MiB for Qwen3-0.6B and 3.5 MiB for Qwen3-1.7B, against 58.0 MiB and 74.0 MiB for layer-wise bases with the same total number of core coordinates; the layer-wise control of Table 3 is the one that pays this larger storage.