FEDDOSA: ONE-SHOT FEDERATED LLM TUNING VIA SPARSE ORTHOGONAL SUBSPACE ALLOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

One-shot federated learning makes it affordable to adapt large language models across clients that cannot share data, since each client uploads its update only once. The single round, however, removes any chance to repair conflicts between locally trained updates, so whatever interference the aggregation introduces is permanent. We cast one-shot federated adaptation as a design problem over the subspace each client may train in, and show that its excess risk separates into capture, noise, and interference terms. The interference term acts through the layer output rather than the weights: updates that are orthogonal as matrices can still conflict in the function the layer computes. We prove that allocating clients disjoint *output* indices in a shared orthonormal basis makes their layer outputs orthogonal for every input, which removes the interference term exactly for a squared-loss linear layer, and that the overlap left by coordinate-level allocation is governed by how nearly the basis diagonalizes the activation covariance. These bounds make no distributional assumption on the learned coefficients: their randomness is each client's own index draw. We instantiate the design as `FedDOSA`, which trains sparse coefficients in a two-dimensional discrete cosine basis, and find on language-understanding and mathematical-reasoning benchmarks that its coordinate-level variant achieves a favourable communication–performance trade-off against existing one-shot methods.

1 INTRODUCTION

Large Language Models (LLMs) have revolutionized natural language processing (NLP), demonstrating remarkable capabilities across a diverse range of tasks (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a). However, the current centralized training paradigm raises significant privacy concerns, especially when dealing with sensitive data such as personal conversations, medical records and proprietary business information (Bommasani et al., 2021; Carlini et al., 2021). Federated Learning (FL) offers a promising alternative by enabling collaborative model training without exposing private data (Kairouz et al., 2021; Yang et al., 2020; Ren et al., 2025; Fan et al., 2025). In typical FL settings, clients train models locally and periodically exchange model updates with a server, which aggregates these updates to improve a global model.

Despite its privacy benefits, traditional FL faces a critical challenge when applied to training LLMs: the immense communication overhead. With modern LLMs containing hundreds of billions of parameters (Brown et al., 2020; Chowdhery et al., 2022), each communication round requires transferring enormous amounts of data—potentially hundreds of gigabytes per client. This becomes especially problematic in bandwidth-constrained environments, where network capacity might be limited to just 1-10 MB/s (Yang et al., 2020; Ren et al., 2025; Tang et al., 2023). For instance, transmitting the full parameter set of a 175B-parameter model (approximately 700GB) would require nearly 20 hours at 10 MB/s for each round, making multi-round FL training impractical.

One-shot federated learning (OFL) has emerged as a potential solution by limiting communication to a single round (Guha et al., 2019; Zhang et al., 2022; Li et al., 2021). In this paradigm, clients train models locally and communicate with the server only once, drastically reducing the communication burden. However, the fundamental challenge of one-shot federated learning lies in its inability to leverage iterative refinement to resolve client heterogeneity conflicts, forcing all reconciliation to occur within a single aggregation step (Zhou et al., 2020; Zhang et al., 2022). Unlike standard FL,

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.

which can gradually mitigate the effects of non-IID data distributions across multiple communication rounds, one-shot FL must simultaneously address amplified spatial heterogeneity and parameter space collisions in a single update. This constraint is particularly problematic when client data distributions vary significantly, as the local updates from different clients not only reflect conflicting data characteristics but also compete for overlapping regions in the parameter space during aggregation. The absence of iterative communication rounds eliminates the opportunity to detect and resolve these conflicts progressively, resulting in parameter interference that can severely distort the global model's ability to generalize across diverse client distributions and downstream tasks, thus raising an important question: *How to effectively resolve severe parameter interference in a **single** round in Federated Learning?*

In this paper, we introduce `FedDOSA`, a one-shot federated learning framework for LLMs that mitigates parameter interference by controlling where each client's update lives. Each client learns a sparse set of coefficients in a shared orthonormal basis, the two-dimensional Discrete Cosine Transform (DCT), and the server merges these coefficients in the same coordinate system, so the overlap between client updates is explicit and can be analyzed directly. We show that this overlap must be measured at the layer output $\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{X}$ rather than in weight space: updates that are orthogonal as matrices can still conflict in the function a layer computes, and what makes their outputs orthogonal for every input is disjointness in the *output* index of the basis. Because each client draws its coordinates before training, the expected overlap is determined by the sampler rather than by the learned coefficients, which yields interference bounds without any independence or zero-mean assumption on the coefficients. `FedDOSA` supports both coordinated allocation, which guarantees exact disjointness, and independent random allocation, which requires no coordination and keeps the overlap small with high probability. Through theoretical analysis and experiments, we demonstrate the effectiveness of this allocation strategy for OFL of LLMs across language understanding and reasoning tasks.[1] Our main contributions are as follows:

- We propose `FedDOSA`, which learns and merges sparse client updates in an orthonormal 2D DCT basis. Random subspace allocation keeps client updates nearly disjoint while preserving single-round communication and requiring no coordination between clients.
- We formulate one-shot FL as a subspace design problem whose excess risk separates into capture, noise, and interference terms, show that interference acts at the layer output, and prove that output-index disjointness makes layer outputs orthogonal for every input, removing the interference term exactly for a squared-loss linear layer. Our interference and concentration bounds make no distributional assumption on the learned coefficients.
- We empirically validate our approach on benchmark NLP tasks, showing improved communication–performance trade-offs over existing one-shot FL methods with substantially lower upload cost than multi-round or larger-adapter alternatives.

2 Related Work

Parameter-efficient fine-tuning. LoRA (Hu et al., 2021) reparameterizes the update as a product of low-rank factors, reducing the number of trainable parameters from quadratic to linear in the layer width; adapter methods (Houlsby et al., 2019; Pfeiffer et al., 2020; Chen et al., 2022) insert small trainable modules between frozen blocks, and prefix or prompt tuning (Li & Liang, 2021; Lester et al., 2021) prepends trainable vectors to the input or to intermediate layers. These methods were designed for a single learner: they reduce *how many* parameters are trained but do not specify *which* parameters each participant should occupy, a distinction that becomes decisive when several clients' updates are merged. Recent work combines PEFT with multi-round FL (Zhang et al., 2023a;b); the one-shot regime remains largely unexplored for LLMs.

One-shot federated learning. Guha et al. (2019) introduced one-shot FL with two strategies, ensembling client models and distilling them through a public dataset; data-dependent variants (Guha et al., 2019; Li et al., 2021) require auxiliary data that may not exist in specialized domains. Data-free approaches synthesize surrogate data with generative models (Zhou et al., 2020; Zhang et al., 2022), shifting the burden onto modelling the clients' distributions. Parameter-aggregation methods (Wang

[1] An unrelated earlier method abbreviated FedDCT (Nguyen et al., 2024) performs divide-and-collaborative training of convolutional networks; the two share no mechanism.

et al., 2020; Yurochkin et al., 2019) combine weights directly and degrade under heterogeneity, and Jhunjhunwala et al. (2023) analyse one-shot averaging for over-parameterized networks without treating heterogeneous data. Across these lines the aggregation rule varies while the allocation does not; we treat the allocation as the design variable.

Sparse and spectral parameterizations. A related line of work adapts models in restricted subspaces, through sparse fine-tuning, random subspace optimization, or spectral parameterizations of the weight update. These methods share the view that useful adaptation does not require updating every parameter. In `FedDOSA` the sparse subspace serves a second purpose: it is the interface through which client updates are merged. Clients train coefficients in a common frequency coordinate system, so the overlap between their updates is explicit and can be analysed, which matters when no later round can repair it.

3 `FedDOSA`

We first formalize one-shot federated adaptation as a subspace design problem (Section 3.1), then describe how `FedDOSA` allocates and merges sparse coefficients in a shared orthonormal basis (Section 3.2), and finally analyze the interference that remains between clients (Section 3.3).

3.1 Problem Setting

One-shot federated adaptation. K clients hold private datasets $\mathcal{D}_\kappa$ and share a frozen pretrained model $\boldsymbol{W}$. Client κ learns an update $\Delta\boldsymbol{W}_\kappa$ on its own data and uploads it once; the server merges the uploads linearly into $\Delta\bar{\boldsymbol{W}} = \mathcal{A}(\{\Delta\boldsymbol{W}_\kappa\})$, and the model $\boldsymbol{W} + \Delta\bar{\boldsymbol{W}}$ is final.

The design problem. Once the data, the pretrained weights, and the local optimizer are fixed, a one-shot designer controls where each client may move and how the results are merged. We formalize the first as a subspace $\mathcal{S}_\kappa \subseteq \mathbb{R}^{d_1 \times d_2}$ of dimension at most the per-client budget B, with orthogonal projector P_κ, within which client κ solves

$$\Delta\boldsymbol{W}_\kappa \in \operatorname*{argmin}_{\Delta \in \mathcal{S}_\kappa} \widehat{\mathcal{L}}_\kappa(\boldsymbol{W} + \Delta). \tag{1}$$

With $\mathcal{L}$ the population risk under the client mixture, the design problem is

$$\min_{\{\mathcal{S}_\kappa\}} \mathbb{E}\Big[\mathcal{L}\Big(\boldsymbol{W} + \mathcal{A}\big(\{\Delta\boldsymbol{W}_\kappa\}_{\kappa=1}^K\big)\Big)\Big] \quad \text{subject to} \quad \dim \mathcal{S}_\kappa \le B. \tag{2}$$

We analyze the scalar merge $\mathcal{A} = \sum_\kappa \gamma_\kappa \Delta\boldsymbol{W}_\kappa$, $\gamma_\kappa \propto |\mathcal{D}_\kappa|$; our algorithm's index-wise averaging uses coordinate-dependent weights instead, and all results carry over (Appendix A.1).

Capture, noise, and interference. To see what Eq. (2) rewards, consider one local gradient step $\Delta\boldsymbol{W}_\kappa = -\eta P_\kappa \widehat{g}_\kappa$ with $\widehat{g}_\kappa = g_\kappa + \xi_\kappa$, where g_κ is the population gradient and the sampling noise ξ_κ is centered and independent across clients. Let $H \succeq 0$ be the curvature of $\mathcal{L}$ at $\boldsymbol{W}$ and $\|x\|_H^2 = \langle x, Hx\rangle$.

Proposition 3.1 (Capture–noise–interference decomposition)**.** *Under the above conditions, the expected one-shot excess risk satisfies*

$$\mathbb{E}\big[\mathcal{L}(\boldsymbol{W} + \Delta\bar{\boldsymbol{W}})\big] - \mathcal{L}(\boldsymbol{W}) = \underbrace{\mathcal{T}_{\mathrm{cap}}}_{\textit{capture}} + \underbrace{\frac{\eta^2}{2}\sum_{\kappa=1}^K \gamma_\kappa^2\, \mathbb{E}\big\|P_\kappa \xi_\kappa\big\|_H^2}_{\textit{noise}} + \underbrace{\eta^2 \sum_{\kappa<\kappa'} \gamma_\kappa \gamma_{\kappa'} \big\langle P_\kappa g_\kappa,\, P_{\kappa'} g_{\kappa'}\big\rangle_H}_{\textit{interference}} + o(\eta^2), \tag{3}$$

where $\mathcal{T}_{\mathrm{cap}} = -\eta\big\langle \nabla\mathcal{L}, \sum_\kappa \gamma_\kappa P_\kappa g_\kappa\big\rangle + \frac{\eta^2}{2}\sum_\kappa \gamma_\kappa^2 \|P_\kappa g_\kappa\|_H^2$ *collects the single-client terms.*

The proof is in Appendix A.2. The three terms respond differently to the allocation. *Capture* is the descent actually achieved; it is limited by how much of the useful update direction lies in the union of the allocated subspaces, and it improves with the budget B, faster when the basis concentrates update energy into few coordinates. *Noise* is the price of that freedom: every admitted coordinate also admits local sampling noise, so this term favours a small budget. *Interference* is a sum over client pairs and exists only because the round is single. Its sign is not fixed: aligned client gradients overshoot along

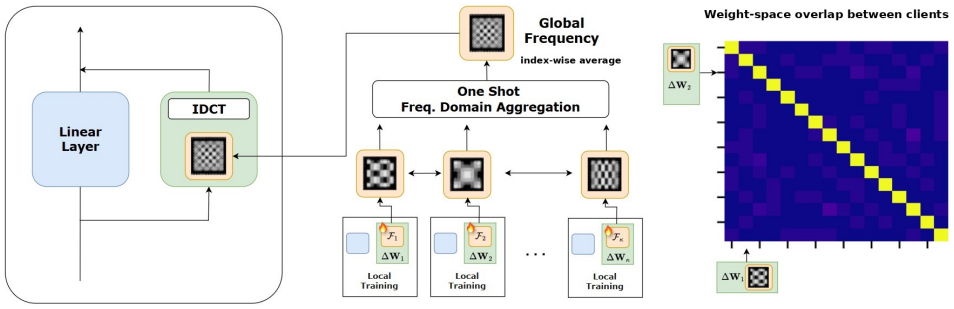


Figure 1: Overview of FedDOSA. Each client trains the coefficients on a sparse support in a shared two-dimensional DCT basis and uploads them once; the server averages coefficients index-wise and reconstructs the global update by the inverse transform. The right panel shows weight-space overlap, which disjoint supports remove; output-space overlap is analyzed in Section 3.3.

the shared direction, anti-aligned ones cancel. Multi-round FL observes and corrects a harmful cross term in a later round; in one shot its sign is unknown in advance and cannot be corrected afterwards. The expansion motivates, but does not by itself prove optimal, three design choices: disjoint directions in a shared orthonormal basis to remove the uncontrolled interference, sparse supports to limit noise, and a basis that concentrates update energy to preserve capture. For a squared-loss linear layer the H-weighted pairing equals the overlap of the two clients' layer outputs; for a transformer trained with cross-entropy we use that overlap as a layer-level diagnostic (Appendix A.1). In either case orthogonality of the weight updates is only the special case of white inputs, which is why Section 3.3 works at the output.

Existing aggregators. LoRA-based one-shot aggregation writes $\Delta \boldsymbol{W}_\kappa = \boldsymbol{A}_\kappa \boldsymbol{B}_\kappa$ with $r \ll \min\{d_1, d_2\}$. *FedAvg* (FedIT; Zhang et al., 2024) averages the factors, $\Delta \bar{\boldsymbol{W}} = \bar{\boldsymbol{A}}\bar{\boldsymbol{B}}$, which introduces cross terms $\boldsymbol{A}_i \boldsymbol{B}_j$ $(i \neq j)$ that distort client-specific updates. *FedStack* (FLoRA; Wang et al., 2024b) concatenates the factors, avoiding cross terms but leaving columns of different clients free to learn overlapping directions. In both, every client adapts the same coordinates, so the interference term in Eq. (3) is left uncontrolled.

3.2 Sparse Orthogonal Subspace Allocation

Parameterization. For a target matrix $\Delta \boldsymbol{W} \in \mathbb{R}^{d_1 \times d_2}$, let $\Omega = \{0, \ldots, d_1 - 1\} \times \{0, \ldots, d_2 - 1\}$ be its coordinate domain in the two-dimensional DCT basis, of size $D = d_1 d_2$. Client κ trains coefficients $F_\kappa(u, v)$ on a support $\Omega_\kappa \subset \Omega$ with $|\Omega_\kappa| = B$, and its update is

$$\Delta \boldsymbol{W}_\kappa(m, n) = \sum_{(u,v) \in \Omega_\kappa} F_\kappa(u, v)\, \alpha_u \alpha_v \cos\tfrac{\pi(2m+1)u}{2d_1} \cos\tfrac{\pi(2n+1)v}{2d_2}, \tag{4}$$

with the standard orthonormal scaling α_u, α_v. Because the basis is orthonormal, clients with disjoint supports are orthogonal as matrices,

$$\langle \Delta \boldsymbol{W}_\kappa, \Delta \boldsymbol{W}_{\kappa'} \rangle_F = 0 \quad \text{whenever } \Omega_\kappa \cap \Omega_{\kappa'} = \emptyset. \tag{5}$$

Aggregation. The server averages each coefficient over the clients that hold it,

$$\bar{F}(u, v) = \frac{1}{|\{\kappa : (u, v) \in \Omega_\kappa\}|} \sum_{\kappa : (u,v) \in \Omega_\kappa} F_\kappa(u, v), \tag{6}$$

and reconstructs $\Delta \bar{\boldsymbol{W}}$ by the inverse transform, treating unselected coefficients as zero. The uploaded object is a list of B coefficients per target matrix, together with the coordinates or the seed that regenerates them.

Allocation. Allocation involves two choices. The first is the index on which disjointness is imposed. *Scattered-coordinate allocation* draws B coordinates uniformly from the whole grid and is the variant evaluated in Section 4; *row-block allocation* draws m output indices and places the budget within

Algorithm 1 `FedDOSA`: one-shot federated adaptation by sparse orthogonal subspace allocation

Require: client datasets $\{\mathcal{D}_\kappa\}_{\kappa=1}^{K}$, frozen weights $\boldsymbol{W}$, budget B, coordinate domain Ω with $|\Omega| = D$.
Ensure: updated global model $\boldsymbol{W} + \Delta\bar{\boldsymbol{W}}$

for all client $\kappa = 1$ to K **in parallel do**
  Draw support $\Omega_\kappa \subset \Omega$, $|\Omega_\kappa| = B$, uniformly without replacement
  Initialize coefficients $\mathcal{F}_\kappa = \{F_\kappa(u, v) \mid (u, v) \in \Omega_\kappa\}$
  for each local batch **do**
    $\Delta\boldsymbol{W}_\kappa \leftarrow \texttt{iDCT}(\mathcal{F}_\kappa)$; update $\mathcal{F}_\kappa$ by a gradient step on $\mathcal{L}(\boldsymbol{W} + \Delta\boldsymbol{W}_\kappa)$
  end for
  Upload $\mathcal{F}_\kappa$ with Ω_κ or its seed ▷ single communication
end for
Merge coefficients by Eq. (6) and set $\Delta\bar{\boldsymbol{W}} \leftarrow \texttt{iDCT}(\bar{\mathcal{F}})$
return $\boldsymbol{W} + \Delta\bar{\boldsymbol{W}}$

them. Section 3.3 shows that only row blocks assigned disjointly by coordination make layer outputs orthogonal for every input. The second choice is coordination. *Coordinated* allocation partitions the indices through the server before training, making the supports disjoint by construction at the cost of an extra round and global allocation state; *independent* allocation lets each client draw its own indices, as in Algorithm 1, and its residual overlap is bounded in Section 3.3.

Choice of basis. Any shared orthonormal basis supports disjoint allocation; the basis matters through capture and through the interference that remains when clients share output indices, which Section 3.3 shows is governed by the off-diagonal mass of the activation covariance in the input basis. The DCT suits both: it compacts the energy of smooth signals, and it is the asymptotic Karhunen–Loève basis of a first-order stationary Markov process (Ahmed et al., 1974; Strang, 1999), so it approximately diagonalizes smooth, slowly decaying covariance; whether trained activations have such covariance is a hypothesis we do not measure. Unlike a principal-component basis, it needs no data exchange. It is also real-valued and applicable in $\mathcal{O}(d \log d)$ time.

3.3 Theoretical Analysis: Interference at the Layer Output

For a projection with input activations $\boldsymbol{X} \in \mathbb{R}^{d_2 \times n}$, client κ perturbs the layer output by $\Delta\boldsymbol{Y}_\kappa = \Delta\boldsymbol{W}_\kappa\boldsymbol{X}$, and two clients interfere through

$$\big\langle \Delta\boldsymbol{W}_\kappa\boldsymbol{X},\, \Delta\boldsymbol{W}_{\kappa'}\boldsymbol{X} \big\rangle_F = \operatorname{tr}\big(\Delta\boldsymbol{W}_\kappa^\top \Delta\boldsymbol{W}_{\kappa'}\boldsymbol{X}\boldsymbol{X}^\top\big). \tag{7}$$

All statements below hold exactly for a fixed batch with $\Sigma := \boldsymbol{X}\boldsymbol{X}^\top$, and in expectation with $\Sigma := \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top]$. The basis is separable, $\mathbf{B}_{uv} = \boldsymbol{c}_u^{(1)}(\boldsymbol{c}_v^{(2)})^\top$, with orthonormal output vectors $\boldsymbol{c}_u^{(1)}$ and input vectors $\boldsymbol{c}_v^{(2)}$ (the columns of $\boldsymbol{C}_2$). We write $\widetilde{\Sigma} = \boldsymbol{C}_2^\top \Sigma \boldsymbol{C}_2 = \mathbf{D} + \mathbf{E}$ for the input second moment in the input basis, split into diagonal and off-diagonal parts; $\boldsymbol{f}_\kappa^{\,u} \in \mathbb{R}^{d_2}$ for client κ's coefficients on output row u; and U_κ for the output rows *allocated* to client κ by its index draw, which are fixed before training.

Theorem 3.2 (Master identity)**.** *For any supports and any coefficients,*

$$\big\langle \Delta\boldsymbol{W}_\kappa\boldsymbol{X},\, \Delta\boldsymbol{W}_{\kappa'}\boldsymbol{X} \big\rangle_F \;=\; \sum_{u \in U_\kappa \cap U_{\kappa'}} \big(\boldsymbol{f}_\kappa^{\,u}\big)^\top \widetilde{\Sigma}\, \boldsymbol{f}_{\kappa'}^{\,u}. \tag{8}$$

Corollary 3.3 (Two sources of interference)**.** *With $\mathcal{C} = \Omega_\kappa \cap \Omega_{\kappa'}$,*

$$\big\langle \Delta\boldsymbol{Y}_\kappa, \Delta\boldsymbol{Y}_{\kappa'} \big\rangle = \underbrace{\sum_{(u,v)\in\mathcal{C}} \widetilde{\Sigma}_{vv}\, F_\kappa(u,v) F_{\kappa'}(u,v)}_{\textit{collision term}} \;+\; \underbrace{\sum_{u \in U_\kappa \cap U_{\kappa'}} \big(\boldsymbol{f}_\kappa^{\,u}\big)^\top \mathbf{E}\, \boldsymbol{f}_{\kappa'}^{\,u}}_{\textit{row-sharing term}}. \tag{9}$$

The sum in Eq. (8) runs only over shared *output* rows, because the output basis is orthonormal; within a shared row, however, different input frequencies are coupled through $\widetilde{\Sigma}$. Disjoint coordinates remove the collision term but not the row-sharing term, so two clients can be exactly orthogonal as matrices and still interfere at the output whenever they share a row and $\mathbf{E} \neq 0$. Weight-space orthogonality

Table 1: A coordinate-disjoint client pair that shares output rows, on a synthetic layer with correlated inputs. Normalized values divide by the product of the two norms.

Inner product	Value	Normalized
Weight space, $\langle \Delta \boldsymbol{W}_\kappa, \Delta \boldsymbol{W}_{\kappa'} \rangle_F$	9.0×10^{-17}	1.0×10^{-17}
Layer output, $\langle \Delta \boldsymbol{W}_\kappa \boldsymbol{X}, \Delta \boldsymbol{W}_{\kappa'} \boldsymbol{X} \rangle_F$	6.12×10^{1}	8.1×10^{-3}

is the special case $\Sigma = I$ (Theorem A.1, Appendix A.1). Table 1 illustrates the difference on a synthetic layer whose inputs have the smooth, correlated structure of trained activations: two clients with disjoint coordinates but shared output rows are orthogonal as matrices to machine precision, yet their output perturbations overlap. Appendix A.9 verifies Eq. (8) over random allocations and the bounds below under coefficient distributions that violate independence and centering.

Theorem 3.4 (Output-disjoint allocation)**.** *If $U_\kappa \cap U_{\kappa'} = \emptyset$ then $\langle \Delta \boldsymbol{W}_\kappa \boldsymbol{X}, \Delta \boldsymbol{W}_{\kappa'} \boldsymbol{X} \rangle_F = 0$ for every input $\boldsymbol{X}$, every input second moment Σ, and every pair of coefficient vectors.*

The guarantee is geometric: client κ's output perturbation lies in the span of its allocated output vectors whatever the input, so clients with disjoint output rows perturb orthogonal subspaces. The row-sharing term can also be reduced by the basis, and it vanishes when the input basis diagonalizes Σ on the occupied coordinates (Theorem A.2).

Bounds without coefficient assumptions. Both terms of Eq. (9) are controlled by the share of a client's coefficient energy that falls on shared indices: θ_κ, the fraction of $\|F_\kappa\|_2^2$ on the collided coordinates $\mathcal{C}$, and ρ_κ, the fraction on the shared output rows $U_\kappa \cap U_{\kappa'}$ (Definition A.3).

Theorem 3.5 (Expected shared energy)**.** *Assume $F_\kappa \neq 0$, and let F_κ be any coefficients produced by client κ's local training.*

(a) Scattered allocation. *If client κ' draws B coordinates uniformly without replacement from the $D = d_1 d_2$ grid, then*

$$\mathbb{E}\big[\theta_\kappa \,\big|\, F_\kappa\big] = \frac{B}{D}, \qquad \mathbb{E}\big[\rho_\kappa \,\big|\, F_\kappa\big] = q := 1 - \binom{D - d_2}{B} \Big/ \binom{D}{B} \approx 1 - (1 - 1/d_1)^B. \quad (10)$$

(b) Row-block allocation. *If client κ' draws m output rows uniformly without replacement from d_1, then $\mathbb{E}[\rho_\kappa \mid F_\kappa] = m/d_1$; if it then draws B coordinates uniformly among the md_2 cells of those rows, $\mathbb{E}[\theta_\kappa \mid F_\kappa] = B/D$.*

These equalities hold because F_κ depends only on client κ's own draw and data, while client κ''s draw is independent of both; no independence, centering, or distributional assumption on the learned coefficients is used. Combining them with Cauchy–Schwarz gives the following, stated informally; the formal statements are Theorem A.4, Corollary A.5, Theorem A.6, and Corollary A.7 in Appendix A.1.

Proposition 3.6 (Interference bounds, informal)**.** *The normalized output interference between two clients is at most $\widetilde{\Sigma}_{\max} \sqrt{\theta_\kappa \theta_{\kappa'}} + \|\mathbf{E}\|_2 \sqrt{\rho_\kappa \rho_{\kappa'}}$ for any coefficients, where $\widetilde{\Sigma}_{\max} = \max_v \widetilde{\Sigma}_{vv}$. Its expectation is therefore at most $\widetilde{\Sigma}_{\max} B/D + \|\mathbf{E}\|_2 \, q$ under scattered allocation and $\widetilde{\Sigma}_{\max} B/D + \|\mathbf{E}\|_2 \, m/d_1$ under row-block allocation. Conditional on the coefficients, both energy fractions obey a Bernstein tail whose only free parameter is the largest single-index energy share, which yields an explicit condition on the number of clients under which every pair stays below a given tolerance with high probability.*

Implication for the evaluated allocation. Scattered allocation makes the collision term small but leaves most output rows shared: at our operating point ($B = 6000$), $\mathbb{E}[\rho_\kappa] = q \approx 0.81$ for a 3584×3584 query projection and ≈ 1 for a 512×3584 value projection, against $\mathbb{E}[\theta_\kappa] \approx 4.7 \times 10^{-4}$ and 3.3×10^{-3}. The output interference is then controlled by $\|\mathbf{E}\|_2$, a property of the basis and the activations, whereas coordinated row blocks remove the row-sharing term at equal cost. A squared-loss simulation (Appendix A.10) confirms both effects: coordinated row blocks set the interference term of the merged risk to zero in every basis, and with smooth input covariance scattered allocation in the DCT gives it a mean below 0.015 of the zero-update risk.

Table 2: Comparison with baselines on LLaMA2-7B and Qwen2.5-7B ($K = 9$, $\alpha = 0.001$). Upload is the number of trainable parameters each client transmits per upload. Subscripts give variation over three seeds; bold marks the best one-shot result per column.

Model	Method	Upload Param	MMLU	MMLU-Pro Acc.	MMLU-Pro BLEU	MMLU-Pro METEOR	MMLU-Pro ROUGE-L	GSM-8K
LLaMA2-7B	Centralized FL	0	$38.9_{\pm 0.1}$	$17.6_{\pm 0.1}$	$20.9_{\pm 0.3}$	$28.9_{\pm 0.3}$	$28.9_{\pm 0.4}$	$17.2_{\pm 0.5}$
	FLoRA (30-shots)	2.5M	$36.4_{\pm 0.1}$	$15.7_{\pm 0.1}$	$21.2_{\pm 0.3}$	$29.5_{\pm 0.8}$	$19.9_{\pm 0.4}$	$15.8_{\pm 0.2}$
	One-Shot Methods							
	FLoRA (1-shot)	2.5M	$24.6_{\pm 0.1}$	$10.2_{\pm 0.4}$	$03.2_{\pm 0.4}$	$05.8_{\pm 0.6}$	$01.5_{\pm 0.3}$	$14.6_{\pm 0.9}$
	DENSE	5.1M	$26.3_{\pm 0.2}$	$08.2_{\pm 0.2}$	$0.01_{\pm 0.01}$	$0.02_{\pm 0.01}$	$0.01_{\pm 0.01}$	$11.3_{\pm 0.4}$
	CO-Boosting	5.1M	$32.8_{\pm 0.4}$	$09.5_{\pm 0.5}$	$01.0_{\pm 0.4}$	$0.7_{\pm 0.5}$	$01.8_{\pm 0.3}$	$12.1_{\pm 0.6}$
	FuseFL	2.5M	$34.8_{\pm 0.2}$	$13.6_{\pm 0.2}$	$11.1_{\pm 0.3}$	$\mathbf{11.9}_{\pm 0.2}$	$\mathbf{19.5}_{\pm 0.4}$	$15.0_{\pm 0.3}$
	FedDOSA	**0.36M**	$\mathbf{35.6}_{\pm 0.2}$	$\mathbf{14.0}_{\pm 0.7}$	$\mathbf{13.0}_{\pm 0.1}$	$11.7_{\pm 0.2}$	$18.2_{\pm 0.2}$	$\mathbf{16.2}_{\pm 0.4}$
Qwen2.5-7B	Centralized FL	0	$69.2_{\pm 0.3}$	$44.3_{\pm 0.3}$	$19.4_{\pm 0.9}$	$29.5_{\pm 0.9}$	$20.0_{\pm 0.7}$	$77.8_{\pm 0.1}$
	FLoRA (30-shots)	2.5M	$67.1_{\pm 0.4}$	$39.6_{\pm 0.2}$	$17.7_{\pm 0.1}$	$26.3_{\pm 0.4}$	$13.3_{\pm 0.7}$	$77.1_{\pm 0.7}$
	One-Shot Methods							
	FLoRA (1-shot)	2.1M	$65.7_{\pm 0.3}$	$40.0_{\pm 0.3}$	$12.1_{\pm 0.4}$	$\mathbf{18.6}_{\pm 0.4}$	$13.1_{\pm 0.4}$	$62.6_{\pm 0.2}$
	DENSE	4.2M	$32.3_{\pm 1.0}$	$19.1_{\pm 0.3}$	$01.7_{\pm 0.3}$	$04.7_{\pm 0.6}$	$06.9_{\pm 0.1}$	$52.1_{\pm 0.2}$
	CO-Boosting	4.2M	$45.3_{\pm 0.8}$	$29.5_{\pm 0.4}$	$01.0_{\pm 0.4}$	$01.1_{\pm 0.1}$	$01.8_{\pm 0.2}$	$50.8_{\pm 0.4}$
	FuseFL	2.1M	$67.8_{\pm 0.4}$	$38.7_{\pm 0.6}$	$\mathbf{16.8}_{\pm 0.1}$	$14.4_{\pm 0.3}$	$\mathbf{16.7}_{\pm 0.8}$	$68.4_{\pm 0.9}$
	FedDOSA	**0.36M**	$\mathbf{68.6}_{\pm 0.2}$	$\mathbf{42.3}_{\pm 0.6}$	$16.7_{\pm 0.4}$	$13.5_{\pm 0.9}$	$\mathbf{16.7}_{\pm 0.3}$	$\mathbf{75.4}_{\pm 0.3}$

4 EXPERIMENTS

Setup. Clients fine-tune on Dolly-15K (Conover et al., 2023) for question answering and on GSM-8K (Cobbe et al., 2021) for mathematical reasoning. We evaluate on MMLU (Hendrycks et al., 2021), MMLU-Pro (Wang et al., 2024a) (accuracy, and BLEU, METEOR and ROUGE-L of the generated answers), and GSM-8K accuracy. Data are partitioned across $K = 9$ clients with a Dirichlet distribution (Hsu et al., 2019) with $\alpha = 0.001$, and we additionally vary $\alpha \in \{0.001, 0.01, 0.05\}$. We use LLaMA-2-7B (Touvron et al., 2023b) and Qwen2.5-7B-Instruct (Team, 2024), batch size 8 and 2,000 local steps, and select the learning rate from $\{0.001, 0.005, 0.01, 0.05\}$. FedDOSA trains $B = 6000$ DCT coefficients per query and value projection. We compare with centralized fine-tuning and multi-round FLoRA (Wang et al., 2024b) as references, and with the one-shot methods FLoRA (1-shot), DENSE (Zhang et al., 2022), CO-Boosting (Dai et al., 2024), and FuseFL (Tang et al., 2024). Full settings are in Appendix B.1.

4.1 MAIN RESULTS

Table 3: Aggregate scores from Table 2. Acc-3 averages MMLU, MMLU-Pro accuracy, and GSM-8K; All-6 adds BLEU, METEOR, and ROUGE-L. Best one-shot value in bold.

Method	LLaMA2-7B Acc-3	LLaMA2-7B All-6	Qwen2.5-7B Acc-3	Qwen2.5-7B All-6
FLoRA (30-shot)	22.6	23.1	61.3	40.2
FLoRA (1-shot)	16.5	10.0	56.1	35.4
DENSE	15.3	7.6	34.5	19.5
CO-Boosting	18.1	9.7	41.9	21.6
FuseFL	21.1	17.7	58.3	37.1
FedDOSA	**21.9**	**18.1**	**62.1**	**38.9**

Table 2 is exploratory: its MMLU-Pro subset and learning-rate selection split were not recorded (Appendix B.2). It shows that FedDOSA uploads 0.36M parameters per client against 2.1–5.1M for the one-shot baselines, a 5.8–14.2× reduction, while achieving the highest one-shot accuracy on MMLU, MMLU-Pro, and GSM-8K on both backbones; some margins, such as MMLU-Pro on LLaMA2-7B, lie within the reported variation. FuseFL remains stronger on METEOR and ROUGE-L for LLaMA2-7B, and FLoRA (1-shot) on METEOR for Qwen2.5-7B. Excluding MMLU-Pro, the mean of MMLU and GSM-8K accuracy is 25.9 against 24.9 for the best other one-shot method on LLaMA2-7B and 72.0 against 68.1 on Qwen2.5-7B; multi-round FLoRA is slightly higher on both (26.1, 72.1). All runs use independently drawn scattered coordinates, so they measure that variant's utility, not the output-disjoint guarantee.

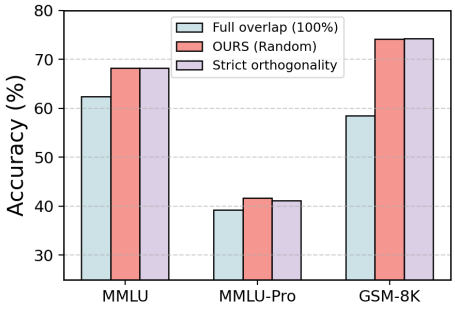


Figure 2: Forced support overlap versus disjoint allocation (Qwen2.5-7B).

Table 6: Coordinated (coordinate-disjoint) vs. independent allocation.

	Metric	K=9	K=40	Avg.
Coordinated	MMLU	68.2	64.2	66.20
	MMLU-Pro	41.7	34.0	37.85
	GSM-8K	74.1	69.3	71.70
Independent	MMLU	68.2	64.3	66.25
	MMLU-Pro	41.1	32.8	36.95
	GSM-8K	74.2	69.3	71.75

Table 7: MMLU accuracy vs. budget B on Dolly-15K (Qwen2.5-7B).

B	500	1000	1500	2000	10k	12k
Acc.	$66.7_{\pm 2.4}$	$68.4_{\pm 0.4}$	$68.7_{\pm 0.2}$	$68.6_{\pm 0.2}$	$68.6_{\pm 0.2}$	$68.8_{\pm 0.3}$

4.2 Analysis

4.2.1 Robustness to Federated Heterogeneity and Participation

1). Client Heterogeneity. A key challenge in one-shot FL is handling non-IID client data. Table 4 reports MMLU accuracy under different levels of heterogeneity controlled by the Dirichlet parameter α. FedDOSA stays within 1.1 points across the range and within 0.6 points of centralized training at the most skewed setting, whereas the synthetic-data methods degrade sharply as heterogeneity grows. At $\alpha = 0.01$ it matches centralized training within the reported variation, so centralized training is best read as a same-task reference rather than a strict upper bound. This stability is consistent with the analysis in Section 3.3: disjoint coordinates keep the collision part of the cross-client term small however the data are split.

Table 4: MMLU accuracy under varying heterogeneity (Qwen2.5-7B).

Method	α=0.001	α=0.01	α=0.05
Centralized FL	$69.2_{\pm 0.3}$	$69.2_{\pm 0.2}$	$69.2_{\pm 0.2}$
DENSE	$32.3_{\pm 1.0}$	$37.6_{\pm 0.9}$	$38.8_{\pm 0.5}$
Co-Boosting	$45.3_{\pm 0.8}$	$45.2_{\pm 0.4}$	$54.2_{\pm 0.8}$
FuseFL	$67.8_{\pm 0.4}$	$67.5_{\pm 0.3}$	$67.7_{\pm 0.4}$
FedDOSA	$\mathbf{68.6}_{\pm 0.2}$	$\mathbf{69.5}_{\pm 0.4}$	$\mathbf{68.4}_{\pm 0.3}$

2). Number of clients. Table 5 varies the number of clients from 2 to 9 on Dolly-15K. Accuracy changes by 0.9 points for FedDOSA and 1.9 points for one-shot FLoRA on the heterogeneous split, while the total upload grows linearly in K for both methods with a per-client constant that differs by roughly $69\times$. Table 8 extends complete federated training to 100 clients, where average accuracy falls from 61.2 at $K = 9$ to 42.4 at $K = 100$. Two causes are plausible and not separated by this experiment: at fixed total data each client's shard shrinks, which worsens the capture and noise terms, and under scattered allocation almost every client pair shares output rows, so the row-sharing term accumulates over more pairs. **3). Client dropout.** Because FedDOSA keeps no allocation state, a client that drops out simply contributes no coefficients and the aggregation rule is unchanged. Table 9 shows that accuracy declines gradually, by 2.5 points, as participation falls from 94% to 80%.

4.2.2 Subspace Allocation and Basis Design

1). Coordinated versus independent allocation. Coordinated allocation makes the supports exactly disjoint, whereas independent allocation leaves an expected collision fraction of B/D. In a single-seed comparison (Table 6) the two agree within 1.2 points on every metric at 9 and 40 clients, consistent with a collision term of order 10^{-4} (Proposition 3.6). Independent allocation is thus a practical option: it avoids the coordination round, the server-side allocation state, and the dependence on all clients being reachable before training (Appendix B.9). Forcing all clients onto the same support, which leaves fewer coefficients, degrades every benchmark by several

Table 5: Accuracy and total upload vs. number of clients on Dolly-15K (Qwen2.5-7B).

Method	K	Hetero	Homo	MB
FedDOSA (B=1000)	2	$69.1_{\pm 0.1}$	$70.8_{\pm 0.2}$	0.278
	3	$69.2_{\pm 0.4}$	$72.4_{\pm 0.5}$	0.417
	4	$68.2_{\pm 0.7}$	$67.1_{\pm 1.2}$	0.556
	5	$68.4_{\pm 0.4}$	$69.1_{\pm 0.4}$	0.695
	9	$68.2_{\pm 0.4}$	$68.6_{\pm 0.4}$	1.250
FLoRA (1-shot)	2	$68.8_{\pm 0.6}$	$68.8_{\pm 0.7}$	19.24
	3	$70.2_{\pm 0.2}$	$70.1_{\pm 0.3}$	28.86
	4	$67.8_{\pm 0.5}$	$68.2_{\pm 0.6}$	38.48
	5	$65.7_{\pm 0.3}$	$68.2_{\pm 0.9}$	48.13
	9	$66.9_{\pm 0.4}$	$67.6_{\pm 0.4}$	86.63

Table 8: Complete federated training with up to 100 clients (single seed).

Metric	K=9	K=40	K=50	K=100
MMLU	68.2	64.3	59.7	53.6
MMLU-Pro	41.1	32.8	29.9	20.8
GSM-8K	74.2	69.3	57.2	52.9
Average	**61.2**	**55.5**	**48.9**	**42.4**

Table 9: Client dropout on LLaMA3.2-1B (MMLU).

Dropout	Active	Accuracy
6%	94%	38.57
10%	90%	38.50
16%	84%	37.71
20%	80%	36.09

Table 10: Orthonormal bases at equal budget (LLaMA3.2-1B). λ: retained-gradient leakage; γ_t: target-gradient capture.

Basis	MMLU	mean λ	mean γ_t
DCT (ours)	**39.3**	**10.51%**	**70.91%**
FFT	39.3	10.71%	70.71%
Hadamard	38.8	10.71%	70.71%
Fourier–Hadamard	—	10.71%	70.72%

Table 11: Local cost and round time on LLaMA3.2-1B at two bandwidths.

	FLoRA (1-shot)	FedDOSA
Local compute	60.4 s	76.5 s
Peak memory	22,813 MiB	**18,729** MiB
Upload	2.10 MB	**0.36** MB
Round @ 0.5 MB/s	**64.6** s	77.2 s
Round @ 0.1 MB/s	81.4 s	**80.1** s

Table 12: One-shot comparison on LLaMA3.2-1B (top) and under matched local compute (bottom), where FedDOSA runs fewer local steps so that its total local computation matches FLoRA's.

		MMLU-Pro				GSM-8K			
Method	**MMLU**	Acc.	BLEU	METEOR	ROUGE-L	Acc.	BLEU	METEOR	ROUGE-L
FLoRA (1-shot)	32.7	12.2	1.6	22.0	8.6	24.9	26.4	25.3	24.3
FedDOSA	**39.3**	**12.7**	**2.9**	**31.8**	**14.2**	**26.4**	**29.9**	**26.5**	**27.0**

Matched compute	Compute for 2,000 steps	Local steps run	Rounds	MMLU
FLoRA (1-shot)	60.4 s	2,000	1	32.7
FedDOSA	76.5 s	1,587	1	**39.8**

points (Figure 2). **2). Coefficient budget.** Table 7 varies B. Accuracy rises from $B = 500$ to $B = 1000$ and is flat within run-to-run variation from $B = 1500$ to $B = 12000$, an eightfold range. This is consistent with a capture term that limits only the smallest budgets, and it indicates that communication can be reduced to the low end of the plateau without loss. **3). Choice of basis.** Table 10 compares orthonormal bases. The structured transforms perform similarly: DCT and FFT reach the same MMLU accuracy, Hadamard is 0.5 points lower, and the gradient-level leakage and capture diagnostics agree within about 0.2 percentage points. This near-parity is expected, since disjoint allocation controls the collision term in any orthonormal basis; we adopt the DCT for its slightly better diagnostics and because it is real-valued, fixed in advance, and fast to apply. Random-basis and original-coordinate controls are synthetic only (Appendix A.10).

Smaller backbone and matched compute. Table 12 repeats the one-shot comparison on LLaMA3.2-1B, where FedDOSA improves on FLoRA across all nine metrics, most clearly on MMLU (39.3 vs. 32.7). Because a FedDOSA step reconstructs the update from its coefficients, it costs more than a LoRA step; granting FedDOSA only the steps that fit FLoRA's compute budget still gives 39.8 against 32.7, so the gain does not come from additional local computation.

Computation and memory. Table 11 measures the local cost of reconstructing the update from coefficients. FedDOSA uses more local computation than FLoRA but 18% less peak memory, since the trainable object is a coefficient list rather than a pair of dense factors, and it uploads $5.8\times$ less.

5 Conclusion

Disjoint client output coordinates in an orthonormal basis eliminate squared-loss layer-output interference without assumptions on local optimization; overlap depends on activation covariance. FedDOSA implements this design with a two-dimensional DCT basis and achieves a favorable one-shot communication-performance trade-off.

AI USE STATEMENT

Generative AI tools were used to revise and edit the manuscript text, to check proof steps and arithmetic, to write the numerical-verification script in Appendix A.9, and to simulate critical reviews of drafts. All AI-assisted content was reviewed by the authors, who take responsibility for the final text, claims, and artifacts.

ETHICS STATEMENT

`FedDOSA` supports privacy-conscious federated fine-tuning of LLMs in sensitive domains such as healthcare and finance, where data sharing is legally prohibited, and reduces communication overhead sufficiently to support resource-constrained participants. As with any method that improves the accessibility of LLM fine-tuning, it could in principle be used to fine-tune models on harmful content in a decentralized and harder-to-monitor manner; practitioners should combine it with secure aggregation, differential privacy on uploaded coefficients, and server-side output monitoring. Non-IID client data may amplify local biases in the aggregated model, and post-aggregation bias auditing is recommended for high-stakes deployments.

REPRODUCIBILITY STATEMENT

All theoretical results are stated with their assumptions in Sec. 3.3, and complete proofs are given in Appendices A.2–A.8. Every number in Appendices A.9 and A.10 is produced by the supplied scripts in `verification/` with a fixed seed, together with their logs. The experimental configuration is given in Sec. 4 and Appendix B.1, and Appendix B.2 lists, for every table, which configuration fields are recorded and which are not.

REFERENCES

Nasir Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL `https://arxiv.org/abs/2005.14165`.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL `https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting`.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *arXiv preprint arXiv:2205.13535*, 2022.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin

Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Rong Dai, Yonggang Zhang, Ang Li, Tongliang Liu, Xun Yang, and Bo Han. Enhancing one-shot federated learning through data and ensemble co-boosting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tm8s3696Ox.

Tao Fan, Hanlin Gu, et al. Ten challenging problems in federated foundation models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *CoRR*, abs/1909.06335, 2019. URL http://arxiv.org/abs/1909.06335.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Towards a theoretical and practical understanding of one-shot federated learning with fisher information. In *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023.

Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *The Annals of Statistics*, 11(1):286–295, 1983.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning, 2021.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1484–1490. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/205. URL https://doi.org/10.24963/ijcai.2021/205. Main Track.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Quan Nguyen, Hieu H. Pham, Kok-Seng Wong, Phi Le Nguyen, Truong Thao Nguyen, and Minh N. Do. FedDCT: Federated learning of large convolutional neural networks on resource-constrained devices using divide and collaborative training. *IEEE Transactions on Network and Service Management*, 2024.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. AdapterHub: A framework for adapting transformers. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 46–54, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.7. URL https://aclanthology.org/2020.emnlp-demos.7/.

Chao Ren, Han Yu, et al. Advances and open challenges in federated foundation models. *IEEE Communications Surveys and Tutorials*, 2025.

Gilbert Strang. The discrete cosine transform. *SIAM Review*, 41(1):135–147, 1999.

Zhenheng Tang, Yuxin Wang, Xin He, Longteng Zhang, Xinglin Pan, Qiang Wang, Rongfei Zeng, Kaiyong Zhao, Shaohuai Shi, Bingsheng He, et al. Fusionai: Decentralized training and deploying llms with massive consumer-level gpus. *The 32nd International Joint Conference on Artificial Intelligence, Symposium on Large Language Models (LLM@IJCAI 2023)*, 2023.

Zhenheng Tang, Yonggang Zhang, Peijie Dong, Yiu ming Cheung, Amelie Chi Zhou, Bo Han, and Xiaowen Chu. Fusefl: One-shot federated learning through the lens of causality with progressive model fusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=E7fZOoiEKl.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra,

Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023b. URL https://arxiv.org/abs/2307.09288.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkluqlSFDS.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a. URL https://openreview.net/forum?id=y10DM6R2r3.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. FLoRA: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=TcCorXxNJQ.

Qiang Yang, Lixin Fan, and Han Yu. *Federated Learning: Privacy and Incentive*. Springer, Cham, 2020.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. *arXiv preprint arXiv:1905.12022*, 2019.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Yufan Zhou, Guoyin Wang, and Yiran Chen. Towards building the federated gpt: Federated instruction tuning, 2024. URL https://arxiv.org/abs/2305.05644.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems*, 35, 2022.

Zhen Zhang, Jialu Wang, and Xin Eric Wang. Parameter-efficient cross-lingual transfer of vision and language models via translation-based alignment. *arXiv preprint arXiv:2305.03510*, 2023a.

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 Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 9963–9977, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.632. URL https://aclanthology.org/2023.findings-acl.632/.

Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020.

A Appendix

A.1 Deferred Statements and Discussion

This section collects the formal statements summarized in Sections 3.1 and 3.3, together with the supporting discussion. Proofs follow in Appendices A.2–A.8.

The implemented merge. The algorithm we evaluate (Algorithm 1, Eq. (6)) does not use scalar weights γ_κ: it averages coefficients *index-wise over the clients occupying each index*, so a coordinate held by one client keeps its full coefficient. This is still linear in the client updates, but with a weight that varies by coordinate: $\mathcal{A}(\{\Delta \boldsymbol{W}_\kappa\}) = \sum_\kappa \Gamma_\kappa(\Delta \boldsymbol{W}_\kappa)$, where Γ_κ is diagonal in the shared basis and scales coordinate j by $1/|\{\kappa' : j \in \Omega_{\kappa'}\}|$. Conditional on the supports, Γ_κ is fixed and commutes with P_κ, so the expansion of Proposition 3.1 goes through unchanged with $\gamma_\kappa P_\kappa$ replaced by $\Gamma_\kappa P_\kappa$ (Appendix A.2). Under disjoint supports Γ_κ acts as the identity on client κ's support, so the implemented merge is the plain sum $\sum_\kappa \Delta \boldsymbol{W}_\kappa$. We state results for scalar γ_κ for readability; nothing below depends on that choice.

Scope of the interference term. One feature of Eq. (3) deserves emphasis immediately, because it determines what the rest of the paper must analyze. The interference term carries the H-weighted pairing, not the plain Frobenius one. In the tractable case of a single linear projection trained with squared loss, the curvature acts on the input index as the input second moment $\Sigma = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top]$, so that $\langle \Delta \boldsymbol{W}_\kappa, \Delta \boldsymbol{W}_{\kappa'} \rangle_H = \mathbb{E}\langle \Delta \boldsymbol{W}_\kappa \boldsymbol{X}, \Delta \boldsymbol{W}_{\kappa'} \boldsymbol{X} \rangle_F$ exactly: the interference term *is* the overlap of the two clients' output perturbations. For a transformer trained with cross-entropy this identity does not hold — the Hessian also involves the output-side curvature of the loss and the downstream layers — and we do not claim it does. There we use the layer-output inner product as a *diagnostic*: it is the first factor through which any client's update reaches the loss, and it is exactly what the squared-loss case makes rigorous. In either reading the Frobenius inner product is the special case $\Sigma = I$, white isotropic activations, so making updates orthogonal as matrices is *not* the same as making them non-interfering at the layer output. Sec. 3.3 is organized around that distinction.

The three terms. The three terms pull on the same variable in different directions. **Capture** is the descent actually achieved, limited by how much of $\nabla \mathcal{L}$ lies in $\bigcup_\kappa \mathcal{S}_\kappa$ at all: a direction no client can represent is one the federation cannot move in. It improves with B, and improves faster when the basis concentrates update energy so that a small B already spans what matters. **Noise** is the price of that freedom: when the projected noise is roughly isotropic across the admitted coordinates, $\mathbb{E}\|P_\kappa \xi_\kappa\|_H^2$ grows with B, so this term favours small B; for anisotropic noise the growth depends on how the admitted coordinates align with the noise covariance. **Interference** exists only because the round is single. It is a sum over client pairs, so the number of terms grows quadratically in K, although with normalized weights γ_κ its magnitude need not. Its sign is not fixed: for positive semidefinite H, two clients whose projected population gradients align contribute a *positive* cross term — the combined step overshoots along the shared direction — while anti-aligned gradients contribute a negative one. The difficulty in one shot is that the sign is unknown in advance and cannot be corrected afterwards, so the design goal is to remove this uncontrolled term rather than to gamble on its sign. Multi-round FL can repair a harmful cross term in a later round; one-shot FL cannot. The term vanishes whenever the two clients' output perturbations are orthogonal, which is a condition on $\Delta \boldsymbol{W}_\kappa \boldsymbol{X}$ rather than on $\Delta \boldsymbol{W}_\kappa$, and Sec. 3.3 shows the two differ in a way that changes which allocation one should choose.

Weight-space localization.

Theorem A.1 (Exact interference localization in weight space)**.** *In a shared orthonormal basis, $\langle \Delta \boldsymbol{W}_\kappa, \Delta \boldsymbol{W}_{\kappa'} \rangle_F = \sum_{j \in \mathcal{C}} F_{\kappa,j} F_{\kappa',j}$, which vanishes when $\mathcal{C} = \emptyset$. No assumption on the coefficients is required.*

Basis diagonalization.

Theorem A.2 (Basis diagonalization)**.** *Let $\mu(\mathcal{B}, \Sigma) = \|\mathbf{E}\|_2$ be the off-diagonal operator norm of the input second moment in the input basis. Then the row-sharing term of Eq. (9) is bounded by $\mu(\mathcal{B}, \Sigma) \sqrt{\rho_\kappa \rho_{\kappa'}}\, \|\Delta \boldsymbol{W}_\kappa\|_F \|\Delta \boldsymbol{W}_{\kappa'}\|_F$, where ρ_κ is defined in Eq. (A.3). The row-sharing term*

vanishes for all coefficients on given supports if and only if every entry $\mathbf{E}_{vv'}$ coupling an input index of client κ to a distinct input index of client κ' on a shared row is zero; in particular it vanishes whenever $\mathcal{B}$ diagonalizes Σ on the occupied input coordinates. When these bounds are combined with Theorem 3.5 in expectation, Σ is held fixed with respect to the support draws; activations of a model trained on the drawn supports need not satisfy this.

Energy fractions and the two-term bound.

Definition A.3 (Collision and row-sharing energy fractions). $\theta_\kappa = \dfrac{\|F_\kappa|_{\mathcal{C}}\|_2^2}{\|F_\kappa\|_2^2}$ *and* $\rho_\kappa = \dfrac{\sum_{u\in U_\kappa\cap U_{\kappa'}}\|\boldsymbol{f}_\kappa^{\,u}\|_2^2}{\|F_\kappa\|_2^2}$,

both in $[0,1]$: the share of client κ's energy on collided *coordinates*, and on shared *output indices*, respectively.

Theorem A.4 (Two-term interference bound). *For any coefficients with $\Delta\boldsymbol{W}_\kappa, \Delta\boldsymbol{W}_{\kappa'} \neq 0$, with $\widetilde{\Sigma}_{\max} = \max_v \widetilde{\Sigma}_{vv}$,*

$$\frac{\left|\langle \Delta\boldsymbol{Y}_\kappa, \Delta\boldsymbol{Y}_{\kappa'}\rangle\right|}{\|\Delta\boldsymbol{W}_\kappa\|_F\,\|\Delta\boldsymbol{W}_{\kappa'}\|_F} \;\le\; \widetilde{\Sigma}_{\max}\sqrt{\theta_\kappa\theta_{\kappa'}} \;+\; \|\mathbf{E}\|_2\sqrt{\rho_\kappa\rho_{\kappa'}}. \tag{11}$$

Bounds for the merged update. The implemented merge scales each client's coefficient at coordinate j by $\Gamma_{\kappa,j} = 1/|\{\kappa'' : j \in \Omega_{\kappa''}\}| \in (0,1]$, which depends on the other clients' supports. The bounds above are stated for the raw client updates, but they dominate the merged contributions: since $0 < \Gamma_{\kappa,j} \le 1$ entrywise, $\|(\Gamma_\kappa F_\kappa)|_{\mathcal{C}}\|_2 \le \|F_\kappa|_{\mathcal{C}}\|_2$ and $\sum_u \|\Gamma_\kappa \boldsymbol{f}_\kappa^{\,u}\|_2^2 \le \sum_u \|\boldsymbol{f}_\kappa^{\,u}\|_2^2$ over shared rows, so every step of the proof of Theorem A.4 applies to the merged contributions with the right-hand side of Eq. (11) unchanged (normalized by the raw norms). The expectations and tail bounds below are then applied to the raw fractions $\theta_\kappa, \rho_\kappa$, which are functions of F_κ and the other client's draw only.

Expected interference, concentration, and capacity.

Corollary A.5 (Expected function-space interference). *Combining Theorems A.4 and 3.5 with $\sqrt{xy} \le (x+y)/2$ and taking expectations, the expected normalized output interference between two clients is at most*

- $\widetilde{\Sigma}_{\max}\, B/D + \|\mathbf{E}\|_2\, q$ *under* scattered-coordinate *allocation (the evaluated variant), with q as in Theorem 3.5;*
- $\widetilde{\Sigma}_{\max}\, B/D + \|\mathbf{E}\|_2\, m/d_1$ *under* row-block *allocation in which each client draws m output rows uniformly and then B coordinates uniformly among the md_2 cells of those rows, since a given cell of client κ is then held by client κ' with probability $(m/d_1)\cdot B/(md_2) = B/D$.*

Coordinated row-block allocation makes the second term zero exactly (Theorem 3.4).

Theorem A.6 (Concentration). *Write ψ for either θ_κ or ρ_κ, p for its conditional mean from Eq. (10), and $\|w_\kappa\|_\infty$ for the largest single-index energy share of the corresponding partition of F_κ (an F_κ-measurable quantity). Then for any $t > 0$,*

$$\Pr\left[\psi \ge p + t \,\middle|\, F_\kappa\right] \;\le\; \exp\!\left(\frac{-t^2}{2\|w_\kappa\|_\infty\,(p + t/3)}\right). \tag{12}$$

Corollary A.7 (Client capacity). *Fix a peakiness cap $\bar{w} \in (0,1]$ and let $L = \log(K(K-1)/\delta)$. Then*

$$\Pr\left[\exists\,\kappa \neq \kappa' :\; \psi_{\kappa\to\kappa'} > p + \sqrt{2\bar{w}pL} + \tfrac{2}{3}\bar{w}L \;\text{ and }\; \|w_\kappa\|_\infty \le \bar{w}\right] \;\le\; \delta,$$

where $\psi_{\kappa\to\kappa'}$ is client κ's shared-energy fraction with respect to client κ'. This is a bound on the joint *event; it is not a conditional probability given the cap. If in addition every client satisfies the cap with probability at least $1-\delta'$, then all pairs satisfy $\psi \le \varepsilon$ with probability at least $1-\delta-\delta'$ provided*

$$p + \sqrt{2\bar{w}\,p\,L} + \tfrac{2}{3}\bar{w}\,L \;\le\; \varepsilon, \qquad \textit{i.e.} \qquad K(K-1) \;\le\; \delta\, e^{L^\star(\varepsilon,p,\bar{w})}, \tag{13}$$

where $L^\star$ is the largest L satisfying the left inequality. This requires $p < \varepsilon$; when $p \ll \varepsilon$, $L^\star \approx 3(\varepsilon - p)/(2\bar{w})$.

Scattered allocation: comparing the two bound terms. In Eq. (11) the row-sharing *bound term* therefore exceeds the collision *bound term* unless $\|\mathbf{E}\|_2/\widetilde{\Sigma}_{\max}$ is below roughly 10^{-3} to 10^{-2}. We are careful about what this licenses. It compares two upper bounds, not the two signed terms of Eq. (9), so it does not establish which term actually dominates in trained layers; that depends on $\|\mathbf{E}\|_2$ and on coefficient signs, neither of which we have measured. What it does establish is structural: *scattered-coordinate allocation leaves the row-sharing term uncontrolled by the allocation*, so any function-space guarantee it enjoys must come from $\|\mathbf{E}\|_2$ being small — a property of the basis and the activation statistics. Theorem 3.4 removes that term exactly, for any input distribution, by allocating over the output index: each client receives its own output rows and may use any input frequencies within them. This costs nothing in communication and, by Theorems 3.5–A.6, inherits the same assumption-free expectation and conditional concentration guarantees with D replaced by d_1 and B by m.

On the basis argument. Two things this argument is not. It is not a claim that the DCT is uniquely optimal: it predicts that other structured transforms with comparable diagonalization behaviour perform comparably, and that an unstructured random orthogonal basis — which has no reason to diagonalize Σ — performs worse. Only the synthetic simulation of Appendix A.10 tests the random-basis part of that prediction; Sec. 4.2 compares structured transforms only. Nor is it a claim we have verified directly: the argument stands or falls on the measured value of $\|\mathbf{E}\|_2/\widetilde{\Sigma}_{\max}$ for these layers, which is a concrete quantity we flag as outstanding in Sec. 3.3 rather than assume.

A.2 Proof of Proposition 3.1

Proof. Write $\Delta\bar{\boldsymbol{W}} = -\eta \sum_\kappa \gamma_\kappa P_\kappa \widehat{g}_\kappa$ with $\widehat{g}_\kappa = g_\kappa + \xi_\kappa$, where $\mathbb{E}[\xi_\kappa] = 0$ and the ξ_κ are independent across clients. Assume $\mathcal{L}$ is twice differentiable at $\boldsymbol{W}$ with curvature $H \succeq 0$, the projectors are fixed, ξ_κ has finite second moment, and the Hessian of $\mathcal{L}$ is bounded and continuous on a neighbourhood containing $\boldsymbol{W} + \Delta\bar{\boldsymbol{W}}$ for all small η (for instance, ξ_κ bounded), so that by dominated convergence the remainder is $o(\eta^2)$ in expectation. A second-order expansion of $\mathcal{L}$ at $\boldsymbol{W}$ gives

$$\mathcal{L}(\boldsymbol{W} + \Delta\bar{\boldsymbol{W}}) - \mathcal{L}(\boldsymbol{W}) = \big\langle \nabla\mathcal{L}, \Delta\bar{\boldsymbol{W}} \big\rangle + \tfrac{1}{2}\big\|\Delta\bar{\boldsymbol{W}}\big\|_H^2 + o(\|\Delta\bar{\boldsymbol{W}}\|^2). \tag{14}$$

Taking expectations, the first-order term contributes $-\eta\langle \nabla\mathcal{L}, \sum_\kappa \gamma_\kappa P_\kappa g_\kappa\rangle$ because $\mathbb{E}[\xi_\kappa] = 0$ and P_κ is deterministic given the support. Expanding the quadratic term, the pure-noise cross terms $\mathbb{E}\langle P_\kappa\xi_\kappa, P_{\kappa'}\xi_{\kappa'}\rangle_H$ and the signal–noise cross terms both vanish by independence and zero mean, leaving

$$\mathbb{E}\big\|\Delta\bar{\boldsymbol{W}}\big\|_H^2 = \eta^2 \sum_\kappa \gamma_\kappa^2 \big\|P_\kappa g_\kappa\big\|_H^2 + 2\eta^2 \sum_{\kappa<\kappa'} \gamma_\kappa\gamma_{\kappa'} \big\langle P_\kappa g_\kappa, P_{\kappa'} g_{\kappa'}\big\rangle_H + \eta^2 \sum_\kappa \gamma_\kappa^2 \mathbb{E}\big\|P_\kappa\xi_\kappa\big\|_H^2. \tag{15}$$

Collecting the single-client population-gradient terms into $\mathcal{T}_{\mathrm{cap}}$ and halving the quadratic contributions gives Eq. (3).

For the implemented index-wise merge, replace each $\gamma_\kappa P_\kappa$ by $\Gamma_\kappa P_\kappa$ with Γ_κ the fixed diagonal weighting defined in Sec. 3.1. Every step above uses only linearity of the merge and that the operator applied to $\widehat{g}_\kappa$ is deterministic given the supports, both of which still hold, so the same three-term decomposition results with $\gamma_\kappa\gamma_{\kappa'}\langle P_\kappa g_\kappa, P_{\kappa'} g_{\kappa'}\rangle_H$ replaced by $\langle \Gamma_\kappa P_\kappa g_\kappa, \Gamma_{\kappa'} P_{\kappa'} g_{\kappa'}\rangle_H$. □

Which inner product the interference term is. The cross term carries the H-weighted pairing. For a single linear projection with squared loss, $\mathcal{L}(\boldsymbol{W}) = \frac{1}{2}\mathbb{E}\|\boldsymbol{W}\boldsymbol{X} - \boldsymbol{Y}\|_F^2$, the loss is exactly quadratic in $\boldsymbol{W}$ and its Hessian acts as $H[\Delta] = \Delta\,\Sigma$ with $\Sigma = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top]$, so

$$\langle \Delta\boldsymbol{W}_\kappa, \Delta\boldsymbol{W}_{\kappa'}\rangle_H = \mathrm{tr}\big(\Delta\boldsymbol{W}_\kappa^\top \Delta\boldsymbol{W}_{\kappa'}\Sigma\big) = \mathbb{E}\,\big\langle \Delta\boldsymbol{W}_\kappa\boldsymbol{X}, \Delta\boldsymbol{W}_{\kappa'}\boldsymbol{X}\big\rangle_F, \tag{16}$$

and the remainder term in Proposition 3.1 vanishes. In this case the interference term is exactly the expected overlap of the clients' output perturbations, Eq. (7). For a nonlinear network trained with cross-entropy the Hessian with respect to one layer's weights has the Gauss–Newton form $\mathbb{E}[\mathbf{J}^\top\mathbf{H}_{\mathrm{out}}\mathbf{J}]$ plus a residual term, where the input side contributes through the layer's activations and the output side through $\mathbf{H}_{\mathrm{out}}$ and downstream Jacobians; the identity above is then not exact, and we use the layer-output inner product only as a diagnostic of how two updates overlap at the point where they first reach the network. In both readings, replacing it by the Frobenius inner product amounts to setting $\Sigma = I$, which is why Sec. 3.3 analyzes Eq. (7) rather than $\langle \Delta\boldsymbol{W}_\kappa, \Delta\boldsymbol{W}_{\kappa'}\rangle_F$.

Which independence is used. Only that the finite-sample noise ξ_κ is centered and independent *across clients*, which we assume; it is natural for independently drawn disjoint datasets but is an assumption, not a consequence of the protocol. Nothing is assumed about the learned coefficients.

A.3 Proof of Theorem 3.2 and Corollary 3.3

Proof. Group each client's expansion by output index. Since $\mathbf{B}_{uv} = \boldsymbol{c}_u^{(1)}(\boldsymbol{c}_v^{(2)})^\top$,

$$\Delta \boldsymbol{W}_\kappa = \sum_{(u,v)\in\Omega_\kappa} F_\kappa(u,v)\, \boldsymbol{c}_u^{(1)}(\boldsymbol{c}_v^{(2)})^\top = \sum_{u\in U_\kappa} \boldsymbol{c}_u^{(1)}\big(\boldsymbol{C}_2\, \boldsymbol{f}_\kappa^{\,u}\big)^\top, \tag{17}$$

where $\boldsymbol{f}_\kappa^{\,u} \in \mathbb{R}^{d_2}$ collects the row-u coefficients and $\boldsymbol{C}_2$ has the input basis vectors as columns. Substituting into Eq. (7) with $\Sigma := \boldsymbol{X}\boldsymbol{X}^\top$ for a fixed batch (the expected version follows by taking expectations of both sides and using linearity of the trace),

$$\begin{aligned}\big\langle \Delta \boldsymbol{W}_\kappa \boldsymbol{X}, \Delta \boldsymbol{W}_{\kappa'} \boldsymbol{X}\big\rangle_F &= \mathrm{tr}\Big(\big[\textstyle\sum_u (\boldsymbol{C}_2 \boldsymbol{f}_\kappa^{\,u})(\boldsymbol{c}_u^{(1)})^\top\big]\big[\textstyle\sum_{u'} \boldsymbol{c}_{u'}^{(1)}(\boldsymbol{C}_2 \boldsymbol{f}_{\kappa'}^{\,u'})^\top\big]\Sigma\Big) \\ &= \sum_u \sum_{u'} \big\langle \boldsymbol{c}_u^{(1)}, \boldsymbol{c}_{u'}^{(1)}\big\rangle \big(\boldsymbol{C}_2 \boldsymbol{f}_{\kappa'}^{\,u'}\big)^\top \Sigma \big(\boldsymbol{C}_2 \boldsymbol{f}_\kappa^{\,u}\big). \end{aligned} \tag{18}$$

Orthonormality of the output basis gives $\langle \boldsymbol{c}_u^{(1)}, \boldsymbol{c}_{u'}^{(1)}\rangle = \mathbf{1}[u = u']$, collapsing the double sum to a single sum over output indices occupied by both clients. With $\widetilde{\Sigma} = \boldsymbol{C}_2^\top \Sigma \boldsymbol{C}_2$ symmetric, each surviving term is $(\boldsymbol{f}_\kappa^{\,u})^\top \widetilde{\Sigma}\, \boldsymbol{f}_{\kappa'}^{\,u}$, which is Eq. (8).

For Corollary 3.3, substitute $\widetilde{\Sigma} = \mathbf{D} + \mathbf{E}$ with $\mathbf{D} = \mathrm{diag}(\widetilde{\Sigma})$. The diagonal part contributes $\sum_u \sum_v \widetilde{\Sigma}_{vv} (\boldsymbol{f}_\kappa^{\,u})_v (\boldsymbol{f}_{\kappa'}^{\,u})_v$, and a term is nonzero only when both clients hold coefficients at the same (u, v), i.e. on $\mathcal{C} = \Omega_\kappa \cap \Omega_{\kappa'}$. The off-diagonal part gives the remaining sum. Note the asymmetry between the two: the collision term requires agreement in *both* indices, while the row-sharing term requires agreement only in the output index. □

A.4 Proof of Theorem A.1

Proof. Expanding both updates in the shared orthonormal basis $\{b_j\}$ and using $\langle b_j, b_{j'}\rangle = \mathbf{1}[j = j']$ leaves only terms with $j = j' \in \mathcal{C}$, giving $\langle \Delta \boldsymbol{W}_\kappa, \Delta \boldsymbol{W}_{\kappa'}\rangle_F = \sum_{j\in\mathcal{C}} F_{\kappa,j} F_{\kappa',j}$, which is empty and hence zero when $\mathcal{C} = \emptyset$. No property of the coefficients is used. Equivalently this is Corollary 3.3 at $\Sigma = I$, for which $\widetilde{\Sigma} = I$, so $\mathbf{E} = 0$ and the row-sharing term is absent. □

A.5 Proof of Theorem 3.4

Proof. If $U_\kappa \cap U_{\kappa'} = \emptyset$ the index set of the sum in Eq. (8) is empty, so the inner product is zero for every $\widetilde{\Sigma}$ and hence every Σ and every $\boldsymbol{X}$.

The geometric statement behind this is worth recording, since it explains why the guarantee needs no assumptions. For any input, $\Delta \boldsymbol{Y}_\kappa = \Delta \boldsymbol{W}_\kappa \boldsymbol{X} = \sum_{u\in U_\kappa} \boldsymbol{c}_u^{(1)} \big[(\boldsymbol{C}_2 \boldsymbol{f}_\kappa^{\,u})^\top \boldsymbol{X}\big]$, so every column of $\Delta \boldsymbol{Y}_\kappa$ lies in $\mathrm{span}\{\boldsymbol{c}_u^{(1)} : u \in U_\kappa\}$ whatever $\boldsymbol{X}$ is. Clients with disjoint output-index sets therefore perturb the layer output within mutually orthogonal subspaces of $\mathbb{R}^{d_1}$, and orthogonality of subspaces is a property of the basis alone. No distributional, independence, or zero-mean assumption enters, and the conclusion is exact rather than in expectation. □

A.6 Proof of Theorem A.4 and Theorem A.2

Proof. Bound the two terms of Eq. (9) separately. For the collision term, Cauchy–Schwarz on $\mathbb{R}^{|\mathcal{C}|}$ and $\widetilde{\Sigma}_{vv} \le \widetilde{\Sigma}_{\max}$ give

$$\Big|\sum_{(u,v)\in\mathcal{C}} \widetilde{\Sigma}_{vv} F_\kappa(u,v) F_{\kappa'}(u,v)\Big| \le \widetilde{\Sigma}_{\max} \big\|F_\kappa|_{\mathcal{C}}\big\|_2 \big\|F_{\kappa'}|_{\mathcal{C}}\big\|_2 = \widetilde{\Sigma}_{\max} \sqrt{\theta_\kappa \theta_{\kappa'}}\, \|F_\kappa\|_2 \|F_{\kappa'}\|_2, \tag{19}$$

using Definition A.3. For the row-sharing term, apply the operator-norm bound row-by-row and then Cauchy–Schwarz across rows:

$$\Big|\sum_u \big(\boldsymbol{f}_\kappa^{\,u}\big)^\top \mathbf{E}\,\boldsymbol{f}_{\kappa'}^{\,u}\Big| \le \|\mathbf{E}\|_2 \sum_u \big\|\boldsymbol{f}_\kappa^{\,u}\big\|_2 \big\|\boldsymbol{f}_{\kappa'}^{\,u}\big\|_2 \le \|\mathbf{E}\|_2 \Big(\sum_u \big\|\boldsymbol{f}_\kappa^{\,u}\big\|_2^2\Big)^{1/2} \Big(\sum_u \big\|\boldsymbol{f}_{\kappa'}^{\,u}\big\|_2^2\Big)^{1/2}, \quad (20)$$

where all sums run over $u \in U_\kappa \cap U_{\kappa'}$. By Eq. (A.3) the two factors are $\sqrt{\rho_\kappa}\|F_\kappa\|_2$ and $\sqrt{\rho_{\kappa'}}\|F_{\kappa'}\|_2$. This proves Theorem A.2. Adding the two bounds and dividing by $\|F_\kappa\|_2\|F_{\kappa'}\|_2 = \|\Delta \boldsymbol{W}_\kappa\|_F\|\Delta \boldsymbol{W}_{\kappa'}\|_F$, the last equality by Parseval, gives Eq. (11).

For the vanishing condition in Theorem A.2: the row-sharing term is $\sum_u \sum_{v\neq v'} (\boldsymbol{f}_\kappa^{\,u})_v\, \mathbf{E}_{vv'}\, (\boldsymbol{f}_{\kappa'}^{\,u})_{v'}$ over shared rows u, with v ranging over client κ's input indices on row u and v' over client κ''s. If every such $\mathbf{E}_{vv'}$ is zero the term vanishes for all coefficients. Conversely, if some $\mathbf{E}_{vv'} \neq 0$ with $(u, v) \in \Omega_\kappa$, $(u, v') \in \Omega_{\kappa'}$, $v \neq v'$, then setting those two coefficients to one and all others to zero makes the term equal to $\mathbf{E}_{vv'} \neq 0$. Diagonalization of Σ on the occupied input coordinates is one sufficient case. □

A.7 Proof of Theorem 3.5

Proof. Both statements follow from one argument applied at different granularities, and the argument turns on the order of operations in the protocol. Client κ first draws its indices, then trains on private data $\mathcal{D}_\kappa$ to produce F_κ. Hence F_κ is measurable with respect to the σ-algebra generated by $(\Omega_\kappa, \mathcal{D}_\kappa)$, and may depend on both in an arbitrary, nonlinear, data-dependent way. Client κ' draws independently with no knowledge of client κ, so $\Omega_{\kappa'} \perp (\Omega_\kappa, \mathcal{D}_\kappa)$ and therefore $\Omega_{\kappa'} \perp F_\kappa$. Conditioning on F_κ consequently does not change the law of client κ''s draw.

Coordinate version. For every coordinate j, $\Pr[j \in \Omega_{\kappa'} \mid F_\kappa] = B/D$ exactly, by symmetry of a uniform B-subset of a domain of size D; no sparse approximation is needed. Since F_κ is supported on Ω_κ, writing $w_{\kappa,j} = F_{\kappa,j}^2/\|F_\kappa\|^2$ with $\sum_{j\in\Omega_\kappa} w_{\kappa,j} = 1$ gives $\theta_\kappa = \sum_{j\in\Omega_\kappa} w_{\kappa,j} Z_j$ with $Z_j = \mathbf{1}[j \in \Omega_{\kappa'}]$, so

$$\mathbb{E}\big[\theta_\kappa \,\big|\, F_\kappa\big] = \sum_{j\in\Omega_\kappa} w_{\kappa,j}\, \Pr\big[j \in \Omega_{\kappa'} \,\big|\, F_\kappa\big] = \frac{B}{D} \sum_{j\in\Omega_\kappa} w_{\kappa,j} = \frac{B}{D}. \quad (21)$$

Output-index version. Identical, with the partition taken over rows instead of coordinates. Put $w_{\kappa,u} = \|\boldsymbol{f}_\kappa^{\,u}\|_2^2/\|F_\kappa\|^2$, which also sums to one over $u \in U_\kappa$, and $Z_u = \mathbf{1}[u \in U_{\kappa'}]$ with $\Pr[Z_u = 1 \mid F_\kappa] = m/d_1$ under row-block allocation. Then $\rho_\kappa = \sum_{u\in U_\kappa} w_{\kappa,u} Z_u$ and $\mathbb{E}[\rho_\kappa \mid F_\kappa] = m/d_1$. Under scattered allocation, $\Pr[Z_u = 1 \mid F_\kappa]$ is the probability that a uniform B-subset of the D grid cells meets the d_2 cells of row u, namely $1 - \binom{D-d_2}{B}/\binom{D}{B}$ for every u, and the same linearity argument gives the value q stated in Theorem 3.5. For θ_κ under row-block allocation with uniform within-row sampling, a given cell of client κ is held by client κ' exactly when its row is among κ''s m rows (probability m/d_1) and it is among the B cells then drawn from those md_2 cells (conditional probability $B/(md_2)$); the two draws are independent of F_κ, so the same linearity argument gives B/D.

In neither case were the weights constrained: they may be arbitrarily skewed, the underlying coefficients may carry any sign pattern, and they may be correlated with client κ''s coefficients. The conclusion is unaffected because the only randomness used is that of client κ''s index draw. □

A.8 Proof of Theorem A.6 and Corollary A.7

Proof. All probabilities in this proof are conditional on F_κ, so the weights below are fixed. From the previous proof, $\psi = \sum_i w_i Z_i$ with $\sum_i w_i = 1$, $w_i \in [0, \|w_\kappa\|_\infty]$, $\mathbb{E}[\psi \mid F_\kappa] = p$, and the Z_i indicators determined by client κ''s draw, which is independent of F_κ. For the coordinate version the Z_i are membership indicators of a uniform fixed-size subset, which are negatively associated (Joag-Dev & Proschan, 1983). For the row version under row-block allocation the same holds; under scattered-coordinate allocation each Z_u is a nondecreasing function of the coordinate indicators on row u, and nondecreasing functions of disjoint subsets of negatively associated variables are again negatively associated (Joag-Dev & Proschan, 1983). The exponential moment bound for sums of

negatively associated variables is dominated by that of independent copies, so Bernstein's inequality applies. The summands satisfy $|w_i(Z_i - p_i)| \le \|w_\kappa\|_\infty$, and with $p_i \le p$ the conditional marginals,

$$\sum_i \mathrm{Var}\big[w_i Z_i \,\big|\, F_\kappa\big] \;\le\; p \sum_i w_i^2 \;\le\; p\,\|w_\kappa\|_\infty, \tag{22}$$

using $\sum_i w_i^2 \le \|w_\kappa\|_\infty \sum_i w_i = \|w_\kappa\|_\infty$. Bernstein then gives

$$\Pr[\psi \ge p + t \mid F_\kappa] \le \exp\!\left(\frac{-t^2/2}{p\|w_\kappa\|_\infty + \|w_\kappa\|_\infty t/3}\right) = \exp\!\left(\frac{-t^2}{2\|w_\kappa\|_\infty(p + t/3)}\right), \tag{23}$$

which is Eq. (12).

For Corollary A.7, fix an ordered pair and let $A_{\kappa\kappa'}$ be the event $\psi_{\kappa\to\kappa'} > p+t$. Since $\{\|w_\kappa\|_\infty \le \bar{w}\}$ is F_κ-measurable and the bound above is increasing in $\|w_\kappa\|_\infty$,

$$\Pr\left[A_{\kappa\kappa'} \cap \{\|w_\kappa\|_\infty \le \bar{w}\}\right] = \mathbb{E}\Big[\mathbf{1}\{\|w_\kappa\|_\infty \le \bar{w}\}\,\Pr[A_{\kappa\kappa'} \mid F_\kappa]\Big] \le \exp\!\left(\frac{-t^2}{2\bar{w}(p + t/3)}\right).$$

Setting the right side to $\delta/(K(K-1))$ gives $t^2 = 2\bar{w}L(p + t/3)$, whose positive root satisfies $t = \frac{1}{3}\bar{w}L + \sqrt{\frac{1}{9}\bar{w}^2L^2 + 2\bar{w}pL} \le \sqrt{2\bar{w}pL} + \frac{2}{3}\bar{w}L$. A union bound over the $K(K-1)$ ordered pairs gives the displayed statement. The left inequality in Eq. (13) is a quadratic in $\sqrt{L}$, $\frac{2}{3}\bar{w}\,x^2 + \sqrt{2\bar{w}p}\,x + (p - \varepsilon) \le 0$ with $x = \sqrt{L}$, which has a positive solution only if $p < \varepsilon$; its largest root gives $L^\star$, and for $p \ll \varepsilon$ the linear term is negligible, giving $L^\star \approx 3(\varepsilon - p)/(2\bar{w})$. The admissible K is therefore finite for every fixed tolerance, but grows like $\exp(L^\star/2)$ rather than being capped at $\lfloor d_1/m \rfloor$ as under coordinated output-index allocation. The statement bounds the probability that a pair violates the bound *and* the cap holds; converting it into a high-probability statement requires a separate bound δ' on the probability that some client exceeds the cap $\bar{w}$, which is a property of the trained coefficients and must be measured or bounded, not assumed. □

A.9 Numerical verification

Two families of claims above are directly falsifiable by simulation, and we check both. All checks use a separable orthonormal DCT basis on both indices and an input second moment with the smooth, slowly-decaying correlation structure characteristic of trained activations ($\Sigma = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top]$ with $\boldsymbol{X}$ drawn from an AR(1) process).

Weight-space orthogonality does not imply function-space non-interference. We construct two clients with strictly disjoint supports that nevertheless share output indices — the situation created by scattered-coordinate allocation — and measure both inner products.

The collision term contributes exactly 0 here and the row-sharing term contributes the entire interference. By contrast, imposing $U_\kappa \cap U_{\kappa'} = \emptyset$ and re-drawing the input 300 times, including across different input distributions, gives a maximum normalized function-space interference of 4.3×10^{-17} — zero to machine precision, as Theorem 3.4 requires and independently of $\boldsymbol{X}$. The master identity Eq. (8) was verified to a maximum relative error of 4.3×10^{-13} over 300 random allocations, and the bound Eq. (11) was never violated over 400 random allocations. All numbers in this subsection are produced by `verification/reproduce_appendix.py` with a fixed seed; the grid is small (24×20, $n = 400$) so that every check runs in seconds, and the input is synthetic, so these checks verify the identities, not the magnitude of interference in trained layers.

The expectation in Theorem 3.5 holds for arbitrary coefficients. Because Eq. (10) claims to hold whatever the learned coefficients are, we test it under distributions that violate the independence and zero-mean assumptions that a coefficient-level analysis would require. We use the operating point of the main experiments, $d_1 = d_2 = 3584$ so $D = 12{,}845{,}056$, and $B = 6000$, giving $p = 4.671 \times 10^{-4}$. Trial counts are scaled with peakiness, since for a concentrated coefficient vector θ_κ is dominated by whether one rare coordinate collides. Client κ''s support is sampled exactly as in the protocol, as a uniform fixed-size subset: the number of its coordinates falling on client κ's support is hypergeometric, and given that number the hit set is a uniform subset of client κ's support.

Table 13: Simulated collision energy against the prediction of Theorem 3.5 ($p = 4.671 \times 10^{-4}$). The last column reports whether the coefficient-level zero-mean conclusion $\mathbb{E}[\langle \Delta \boldsymbol{W}_\kappa, \Delta \boldsymbol{W}_{\kappa'} \rangle] = 0$ survives the same distribution.

Coefficient distribution	**trials**	$\|w\|_\infty$	**measured** $\mathbb{E}[\theta]$	**dev.**	**zero-mean claim**
Gaussian, zero-mean	40,000	0.0023	4.679×10^{-4}	0.3σ	holds (0.8σ)
All-positive (nonzero mean)	40,000	0.0014	4.688×10^{-4}	0.9σ	**violated** (196σ)
Heavy-tailed (Cauchy)	4,000,000	0.939	4.663×10^{-4}	0.1σ	inconclusive
Peaked (one dominant coeff.)	1,784,822	0.288	4.618×10^{-4}	1.2σ	holds (1.5σ)
Power-law decay (DCT-like)	4,000,000	0.413	4.697×10^{-4}	0.5σ	holds (1.4σ)

The measured shared energy agrees with p in every case, all deviations lying within Monte-Carlo error, including for heavy-tailed and highly peaked profiles. For the final column, each client first draws a full B-length coefficient profile from the stated distribution, and the collided coordinates then sit at uniformly random positions in each client's own profile, matching the protocol order. Under all-positive coefficients the coefficient-level zero-mean conclusion fails by 196 standard errors: any systematic sign shared across clients — plausible when all clients start from the same pretrained model — produces a nonzero mean interference that no amount of sparsity removes in expectation, only shrinks. Under the other profiles it is not contradicted at this sample size, and for the heavy-tailed case it cannot be assessed, since the interference has infinite variance and its sample mean does not concentrate. Theorem 3.5 holds in every case because it asks nothing of the coefficients. All numbers come from `verification/reproduce_appendix.py` (fixed seed; log in `verification/reproduce_appendix.log`).

A.10 Controlled Simulation of Allocation Schemes

The LLM experiments evaluate only independently drawn scattered coordinates. To test the predictions of Section 3.3 where they are exact, we simulate a single linear layer trained with squared loss, for which the interference term of Eq. (3) equals the layer-output inner product. The layer is 64×64 ($D = 4096$), with $K = 8$ clients and budget $B = 128$ per client. In a given basis, client κ's target coefficients are $F^*_\kappa = F_\mathrm{c} + 0.5\, G_\kappa$, where F_c and G_κ have independent Gaussian entries whose standard deviation decays as $1/(1 + u + v)$. Targets are drawn with this same profile in each basis's own coordinates; this is a synthetic target family, not one fixed task viewed in three bases, so cross-basis comparisons hold only for this family. Each client solves its least-squares problem exactly on its support, and the server merges by Eq. (6). We compare five allocations at equal per-client budget: one support shared by all clients; scattered coordinates drawn independently or partitioned by coordination; and row blocks (8 rows $\times$ 16 columns) drawn independently or with coordinated, disjoint rows. The shared support exposes only B distinct coordinates in total, whereas the other schemes expose up to KB, so its risk also reflects lower coverage.

Table 14 reports three quantities over 50 random draws of targets, supports, and (for the random bases and random-spectrum covariance) the basis and covariance. *Overlap* is the mean pairwise normalized absolute inner product of the raw client output perturbations, a geometric diagnostic. *Cross* is the signed interference term of the merged risk: writing the merged update as $\sum_\kappa \Gamma_\kappa F_\kappa$ with the index-wise weights of Appendix A.1, it is $2 \sum_{\kappa<\kappa'} \langle \Gamma_\kappa F_\kappa, \Gamma_{\kappa'} F_{\kappa'} \rangle_{\widetilde{\Sigma}}$, divided by the excess risk R_0 of the zero update. R/R_0 is the merged excess risk relative to R_0. Because R_0 depends on the basis and covariance, R/R_0 and Cross should be compared only within a column group.

The results match the theory in three respects. First, coordinated row blocks give exactly zero overlap and zero cross term in every basis and covariance (Theorem 3.4). Second, under AR(1) inputs, which the DCT nearly diagonalizes, scattered allocation in the DCT leaves a cross term of at most $0.015\, R_0$ (Theorem A.2). Within the random-orthogonal and original-coordinate groups, by contrast, scattered allocation leaves a cross term of 2.0–$2.9\, R_0$ that alone exceeds the whole gap between the merged risk and the zero update, so scattered merging is worse than no update ($R/R_0 > 1$), while coordinated row blocks reach the lowest risk in the group. Because targets are drawn separately in each basis, we do not compare these magnitudes across groups. Third, under a covariance with no structure the DCT can exploit, all three bases behave alike (the full results are in the log). Within the DCT under AR(1), the non-shared allocations reach similar risk; coordinated row blocks are lowest,

Table 14: Controlled simulation (squared-loss linear layer; mean over 50 draws, standard errors in the log). Ov.: raw pairwise output overlap (diagnostic). Cr.: signed cross term of the merged risk, relative to R_0. R/R_0: merged excess risk relative to the zero update (lower is better). Compare within a column group.

	AR(1) input covariance									**Random-spectrum**		
	DCT			Random orth.			Original coord.			DCT		
Allocation	Ov.	Cr.	R/R_0	Ov.	Cr.	R/R_0	Ov.	Cr.	R/R_0	Ov.	Cr.	R/R_0
Shared support	0.769	0.015	0.98	0.793	0.17	0.80	0.794	0.16	0.82	0.768	0.030	0.96
Scattered, indep.	0.026	0.015	0.87	0.223	2.56	2.25	0.214	2.01	1.93	0.034	0.037	0.80
Scattered, coord.	0.006	0.007	0.81	0.200	2.92	2.60	0.207	2.75	2.58	0.016	0.031	0.76
Row block, indep.	0.026	0.013	0.83	0.083	0.37	0.79	0.069	0.25	0.75	0.029	0.032	0.79
Row block, coord.	0	0	0.78	0	0	0.37	0	0	0.48	0	0	0.78

but within about one standard error of coordinated scattered coordinates. The simulation checks the mechanism in the squared-loss setting and is not evidence about trained transformers; it is produced by `verification/simulate_allocation.py` with a fixed seed.

B Additional Experimental Detail and Results

B.1 Details

Dataset We trained our model with two NVIDIA A100 GPUs (each with 39 GB of memory). Training took approximately 1.5 hours, while testing required about 2 hours. We reproduced all baselines because they are adopted other datasets and models.

Metrics: **BLEU (Bilingual Evaluation Understudy)**: This metric compares the machine-generated text to one or several reference texts by checking how many n-grams (word sequences) they share. **METEOR (Metric for Evaluation of Translation with Explicit ORdering)**: Matches words from the generated output to the reference text and calculates a score that combines precision and recall, taking synonyms and word stems into account. **ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation)**: Measures the longest common subsequence between the generated and reference summaries, which is useful for evaluating paraphrased or rephrased text.

Dolly-15K dataset Conover et al. (2023) An open-source collection of 15,000 text samples generated by Databricks employees. It covers brainstorming, classification, closed-QA, generation, information extraction, open-QA, and summarization. Data are partitioned by category, and we randomly sample 1,000 for a concise yet comprehensive evaluation.
For example:

GSM8K (Grade School Math 8K) Cobbe et al. (2021) A set of 8,500 high-quality, linguistically diverse grade-school math word problems designed for multi-step reasoning. Partitioned by problem topic.
For example:

Rosetta Code (`https://rosettacode.org`) A programming chrestomathy site presenting solutions to the same task in many languages, partitioned by programming language. It was used only for the code-generation evaluation, which is not reported because its protocol was not recorded.

MMLU and MMLU-Pro test set Hendrycks et al. (2021); Wang et al. (2024a) MMLU contains 14,024 multiple-choice questions across 57 subjects to evaluate LLM reasoning and MMLU-Pro. We randomly sample 1,000 for a concise yet comprehensive evaluation. We adopt a one-shot setting(Just given one example) on MMLU and a five-shots on MMLU-Pro.

Experimental setting For Table 2, all methods use AdamW optimizer, bf16 mixed precision, batch size 8, 2,000 local training steps on A100 GPUs, and Dirichlet $\alpha = 0.001$ with 9 clients (other tables vary α or K as listed in Table 15); MMLU and MMLU-Pro are evaluated after training on Dolly-15K and GSM-8K accuracy after training on GSM-8K. The MMLU-Pro evaluation subset and the split

used to select the learning rate are not recorded. Table 2 averages three seeds, and the seeds of every other table are listed in Table 15. FLoRA applies LoRA with rank $r = 8$, $\alpha = 32$, dropout 0.1, and learning rate 10^{-2} to all attention projections, using the original stacking aggregation of Wang et al. (2024b) without SVD truncation. DENSE uses a GAN generator with noise dimension 512 and sequence length 512 tokens (top-k filtering, $k = 50$), distillation temperature $\tau = 4$, synthesis batch size 64, adversarial and cross-entropy loss weights both 1.0, 1,000 synthesis iterations, and 500 distillation steps per round. Co-Boosting uses temperature $\tau = 3$, perturbation strength 0.01, 3 ensemble update steps, hard sample weight 2.0 (capped to prevent instability), 5 boosting rounds, and 300 distillation steps per round. Centralized FL and individual training use the same LoRA configuration as FLoRA and serve as the upper and lower performance references, respectively.

B.2 Experimental Provenance

Table 15 lists, for every quantitative table, the configuration as stated in this paper. Where a field was not recorded we say so rather than infer it; results whose configuration could not be established at all have been removed from the paper.

Table 15: Provenance of each quantitative result. Every LLM experiment uses scattered-coordinate allocation. "n.r." marks a field not recorded.

Table / figure	Backbone	Data	Clients	B	Seeds
Table 2 (main)	LLaMA2-7B, Qwen2.5-7B	Dolly / GSM-8K, α=0.001	9	6000	3
Table 4	Qwen2.5-7B	MMLU eval, $\alpha \in \{0.001, 0.01, 0.05\}$	n.r.	n.r.	3
Table 5	Qwen2.5-7B	Dolly-15K	2–9	1000	n.r.
Table 6	Qwen2.5-7B†	MMLU / MMLU-Pro / GSM-8K	9, 40	n.r.	1
Table 7	Qwen2.5-7B	Dolly-15K	n.r.	500–12000	3
Table 10	LLaMA3.2-1B†	MMLU	n.r.	n.r.	n.r.
Table 8	Qwen2.5-7B†	MMLU / MMLU-Pro / GSM-8K	9–100	n.r.	1
Tables 9, 12, 12, 11	LLaMA3.2-1B	MMLU etc.	n.r.	n.r.	n.r.

†Inferred from matching values elsewhere in the paper, not stated by the source run. The following are also not recorded in this manuscript, and we flag them rather than assert them: the fixed evaluation subsets, and whether learning rates were selected on a held-out split. Code-generation evaluations were run but are omitted because their benchmark variant and pass@k protocol are unrecorded. Communication is compared throughout by uploaded *parameter count*, which follows directly from the budget and the number of target matrices. The megabyte figures in Tables 5 and 11 are as logged by those runs; their byte accounting (value precision, index versus seed transmission) is not recorded, so we compare megabytes only within a table, never across tables.

B.3 Choice of Orthonormal Basis

Theorem A.2 makes the basis choice an empirical question with a prediction attached. Disjoint allocation controls the collision term in *any* orthonormal basis, so the basis cannot matter for that purpose; what it governs is capture and the off-diagonal mass $\|\mathbf{E}\|_2$ of the activation covariance, which bounds the row-sharing term. The hypothesis is therefore that structured transforms that approximately diagonalize smooth activation covariance should behave similarly to one another and better than unstructured ones, and that the DCT should hold no large advantage over other structured transforms. We compare bases at fixed coefficient count, target matrices, optimizer, and local steps (upload bytes are not claimed equal, since FFT's complex-coefficient encoding is not recorded), so that only the basis varies. The diagnostics below are gradient-level and do not measure $\|\mathbf{E}\|_2$ directly. Its MMLU values (≈ 39) match the LLaMA3.2-1B runs of Appendix B.5; client count and seeds are not recorded (Table 15).

Alongside accuracy we report two diagnostics measurable from saved coefficients: the retained-gradient leakage λ (distinct from the row-sharing energy fraction ρ of Definition A.3) (the fraction of a retained client's gradient norm lying in the collided subspace) and the target-gradient capture γ_t, together with the Spearman correlation between λ and the induced retained-loss change ΔL. These are gradient-level proxies for functional interference, not measurements of the quantities in Theorem A.4.

The structured transforms are close to indistinguishable. DCT and FFT reach the same MMLU accuracy and Hadamard is 0.5 points lower, a gap we do not treat as meaningful since no seed variation is reported; the leakage and capture diagnostics agree to within about 0.2 percentage points across all four. This is the predicted outcome, not a negative result: the interference mechanism is a property of disjoint allocation in an orthonormal basis, and these bases are all orthonormal, so near-parity is what Theorem A.1 implies. We select the DCT for its marginally better average diagnostics together with its practical properties — real-valued, fixed in advance, and $\mathcal{O}(d \log d)$ — and explicitly do not claim it is uniquely optimal. FFT or Hadamard remain reasonable choices where spectral interpretability or addition-only arithmetic matters.

Two caveats. First, the λ–ΔL correlations are mostly not significant at the 0.05 level (only FFT approaches it, at $p = 0.059$), so we present these as mechanistic diagnostics and not as evidence that parameter overlap causes functional degradation; the sample of clients is small and the test is underpowered. Second, and more importantly, the two comparisons that would actually discriminate were not run on LLMs (a synthetic version is in Appendix A.10). An unstructured random orthogonal basis is the control for the coherence argument of Sec. 3.2, and sparse allocation in the original parameter coordinates is the control that separates the benefit of the *basis* from the benefit of *sparsity plus disjointness* alone. Without them, the claim that a covariance-aligned basis helps is argued but not demonstrated, and we state it only as a hypothesis.

B.4 Scale and Participation

We run complete federated training at 9, 40, 50, and 100 clients and report downstream accuracy.

Accuracy falls substantially as the client count grows, from an average of 61.2 to 42.4, and we report this as a limitation of the method rather than as evidence for it. Two contributors are plausible and this table cannot separate them. One is data fragmentation: at fixed total data, more clients means less data per client and noisier local updates, which worsens the capture and noise terms of Eq. (3). The other is interference: by Theorem 3.5 each pair's expected collision energy stays at B/D regardless of K, but the number of pairs grows quadratically, and under scattered allocation essentially every pair shares output rows (Sec. 3.3), so the uncontrolled row-sharing term accumulates. Separating the two would require varying K at fixed per-client data, or comparing against output-disjoint allocation; we make no causal claim here.

Client dropout. Real deployments lose participants mid-round. Because `FedDOSA` requires no pre-training coordination, a client that disappears simply contributes no coefficients, and the aggregation rule is unchanged — there is no allocation state to repair. Table 9 confirms the behaviour is graceful.

Accuracy falls by 2.5 points as participation drops from 94% to 80%, without the discontinuity that a coordination-dependent scheme would exhibit if the dropped clients held assigned coordinates. We note that severe data-size imbalance, stragglers, and asynchronous arrival are not evaluated.

B.5 Smaller Backbone and Matched Compute

Two questions are naturally asked together: does the method work outside the two 7B models, and are its gains an artifact of a larger local compute budget? We address both on LLaMA3.2-1B.

Matched local compute. `FedDOSA` reconstructs a structured update during local training, so a step costs more than a LoRA step. Comparing at equal step counts therefore compares unequal compute, and the honest comparison equalizes the latter. Table 12 gives FLoRA its full $2{,}000$ steps and grants `FedDOSA` only the number of steps that fits the same wall-clock budget.

Under matched compute `FedDOSA` reaches 39.8% against FLoRA's 32.7%, so the advantage does not come from spending more local computation. It is in fact slightly *higher* than the 39.3% obtained with the full step count, which we attribute to run-to-run variation rather than to any benefit of training less.

The compute column is the total local computation of a $2{,}000$-step round, the same quantity as in Table 11 (where $60.370 + 2.10/0.5 = 64.570$ s). Exact matching would give $2000 \times 60.370/76.464 \approx 1{,}579$ steps; the run used $1{,}587$, about 0.5% more compute than FLoRA, which does not affect the comparison at this margin.

B.6 Computation, Memory, and End-to-End Round Time

Communication savings are only real if they are not purchased with prohibitive local cost, so we measure both sides of that exchange.

The trade is explicit. FedDOSA spends about 27% more local computation reconstructing structured updates, and we make no claim to faster local training. It uses less peak memory ($18{,}729$ against $22{,}813$ MiB), because the trainable object is a sparse coefficient list rather than a pair of dense adapter factors, so the transform does not impose a memory penalty. And it uploads $5.8\times$ less. Whether the total favours it depends entirely on bandwidth: at 0.5 MB/s the extra computation dominates and FedDOSA is slower end-to-end, while at 0.1 MB/s the upload dominates and it is faster. Since one-shot FL is motivated by bandwidth-constrained deployment, we regard the latter regime as the relevant one, but we report the crossover rather than selecting the favourable column.

B.7 Backbone Sensitivity of FLoRA

FLoRA performs markedly differently across our two 7B backbones (Table 2), weakly on LLaMA2-7B and competitively on Qwen2.5-7B. Since this affects how the main comparison should be read, we state what we observe. The implementation, training protocol, and evaluation are identical across backbones, so the difference is not procedural; we do not diagnose its cause further. Because hyperparameters were tuned once per method rather than per backbone, readers should not treat FLoRA's LLaMA2-7B numbers as its best achievable performance.

B.8 Aggregate Communication–Utility Comparison

Because FedDOSA does not lead on every individual metric, we also report aggregates, under two stated formulas; the choice changes one conclusion.

Among one-shot methods FedDOSA has the highest aggregate on both backbones under both formulas while uploading 5.8–$14.2\times$ fewer parameters. On LLaMA2-7B its margin over FuseFL (21.9 vs. 21.1; 18.1 vs. 17.7) is small relative to the per-metric spreads, so we do not claim it as an improvement there. Against the *multi-round* 30-shot FLoRA reference, FedDOSA is lower on LLaMA2-7B under both formulas and, on Qwen2.5-7B, higher under Acc-3 (62.1 vs. 61.3) but lower under All-6 (38.9 vs. 40.2); we therefore make no claim that it matches multi-round training. Averages are summaries, not statistical tests; the claim these exploratory results support is a favourable communication–utility trade-off among one-shot methods.

We report no paired significance tests. Accordingly, throughout the paper we do not claim as an improvement any difference smaller than the larger of the two reported standard deviations.

B.9 Communication Overhead

Both allocation strategies upload the same number of coefficients per client, B per target matrix. They differ in protocol, not in upload size. Coordinated allocation requires a server-to-client step before training, in which each client receives its assigned coordinates; independent allocation omits that step because a client's coordinates are regenerated from a seed or uploaded alongside its coefficients. The additional step adds a synchronization barrier, requires the server to hold allocation state for every participant, and makes the round depend on all clients being reachable before training starts. We do not quantify its byte cost, because the byte accounting of our runs is not recorded (Appendix B.2); the argument above concerns protocol structure only.

B.10 Robustness Analysis

B.10.1 Capacity under Coordinated and Independent Allocation

Coordinated allocation faces a hard combinatorial ceiling: with a per-matrix domain of size $D = d_1 d_2$ and a budget of B coefficients, no disjoint allocation exists beyond $\lfloor D/B \rfloor$ clients. For Qwen2.5-7B's value projection (512×3584, so $D = 1{,}835{,}008$) at the main-experiment budget $B = 6000$ this ceiling is $\lfloor D/B \rfloor = 305$ clients. Independent allocation replaces this counting ceiling with the tolerance-dependent condition Eq. (13) of Corollary A.7: overlap grows smoothly with the client

count, and for a fixed tolerance the admissible count is finite but not tied to D/B. Appendix B.4 reports complete federated training up to 100 clients.

B.10.2 Effect of Forced Overlap on Aggregation Quality

As shown in Figure 2, forcing all clients onto the same support degrades performance substantially across all benchmarks. This drop is associated with overlap, but full overlap also reduces the number of distinct coefficients, so the figure does not isolate interference. The informative comparison, however, is between the other two conditions: independent and coordinated allocation differ by at most 0.6 points at $K = 9$ and 1.2 points at $K = 40$ (single seed). The figure orders the two near-equal MMLU-Pro bars oppositely to Table 6 (41.7 coordinated, 41.1 independent); we rely on the table for that comparison and use the figure only for the gap to full overlap, which is several points on every metric and unaffected by the ordering.

Corollary 3.3 explains all three conditions together, and it is worth separating which term moves in each. Independent and coordinated coordinate allocation differ only in θ, the collision energy fraction, and by Theorem 3.5 that differs between them by at most $B/D \approx 10^{-4}$ in expected collision energy. That bounds output overlap, not accuracy, so it is consistent with — but does not establish — the observation that the two are indistinguishable. Neither of them, however, differs in ρ: both leave output-index energy fraction ρ near its scattered-allocation value of 0.81 or above (Sec. 3.3), so both carry the same row-sharing term. Whether that term limits their accuracy is not something this figure can show. Forced full overlap is different in kind: it drives θ to 1, activating the collision term at full strength and adding $\widetilde{\Sigma}_{\max}$ to the bound.

The figure is therefore consistent with the collision term mattering when it is large (full overlap) and with coordinating the coordinate draw adding nothing measurable once it is small; being a single-seed comparison whose configuration is not recorded, it illustrates rather than tests these readings. What it does *not* test is the row-sharing term, which no condition in this figure varies. Doing so requires the output-disjoint allocation of Theorem 3.4, which is the experiment flagged as outstanding in Sec. 3.3.

C Privacy Analysis

C.1 Reduced Attack Surface for Membership Inference

Our one-shot design removes access to multi-round model evolution, which may reduce the attack surface for trend-based membership inference attacks (MIAs). Traditional MIAs can exploit loss trajectories or prediction confidence curves across rounds; a single communication round provides less temporal information. This observation should not be interpreted as a formal privacy guarantee: `FedDOSA` does not by itself provide differential privacy, and sparse frequency coefficients may still leak information about local data. Instead, we view frequency sparsification as a mechanism that can reduce the amount of transmitted information relative to full-parameter updates and should be combined with standard privacy tools when formal guarantees are required.

C.2 Reconstruction Attack Mitigation

Effective gradient inversion attacks often benefit from multiple gradient updates because parameter evolution patterns provide additional constraints to the attacker. `FedDOSA` exposes only a sparse one-shot coefficient snapshot, which may make reconstruction more difficult than from repeated dense updates. However, collision probability is not a privacy definition, and we therefore do not claim a formal leakage bound from the DCT analysis alone. A complete privacy guarantee would require a specified threat model together with mechanisms such as differential privacy, secure aggregation, or cryptographic protection of the uploaded coefficients.

We deliberately do not report a reconstruction-success comparison here. A privacy number reported without a threat model, the attack implementation, and the evaluation data is not interpretable. Establishing such a claim requires a stated adversary, a named attack with released hyperparameters, and a matched-budget comparison, which we regard as separate work. The argument of this section is therefore confined to attack surface and transmitted volume, and should not be read as a privacy guarantee of any kind.

C.3 Subspace Isolation and Cross-Client Leakage

Orthogonal subspace allocation reduces cross-client overlap in parameter space. Under coordinated allocation different clients occupy disjoint coordinates; under independent allocation occasional collisions occur, but the expected energy on them is bounded by Theorem 3.5. This may reduce direct cross-client mixing of coefficient values, but it should not be equated with a formal bound on cross-client information leakage: a bound on interference is not a bound on inference. The privacy benefit is therefore best understood as reduced exposure and reduced parameter overlap, not as a replacement for formal privacy mechanisms.