MECHLEASH-FL: MECHANISTIC LEASHING OF LATENT FEATURES FOR FEDERATED LEARNING UNDER FEATURE SKEW

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Averaging (FedAvg) aggregates client models by averaging their parameters directly, so its performance implicitly depends on clients representing matching features along the same directions in the model's hidden space. We analyze what happens to this match under feature skew and representational superposition. This analysis reveals two ways that parameter averaging can degrade features: reducing feature strength and causing feature interference between clients. To mitigate these issues, we introduce MechLeash-FL, a two-phase federated training procedure that leashes clients' latent feature directions to a shared mechanistic reference during local training. MechLeash-FL uses a sparse autoencoder (SAE), shared across clients, to penalize changes in the clients' latent representations. After each aggregation, the SAE is refit to the merged model's activations and shared with the clients, providing a common reference for latent feature directions without changing the deployed model's architecture. We evaluate MechLeash-FL in two settings: a controlled toy model of superposition, where we directly measure feature shrinkage and interference, and real-data reconstruction on Office-Caltech10, a naturally feature-skewed benchmark. In a synthetic setting, MechLeash-FL reduces endpoint reconstruction error by 14.82%, preserves more feature strength after aggregation, and reduces interference between clients. On Office-Caltech10, adding MechLeash-FL to FedAvg, FedProx, MOON, and Ditto reduces validation mean squared error in uncorrupted reconstruction by 15.29%-33.08%.

1 INTRODUCTION

Federated learning (FL) lets multiple clients train a model together without sharing their data. The standard algorithm, Federated Averaging (FedAvg), alternates between local training and averaging the resulting parameters on a server (McMahan et al., 2017). Averaging assumes that corresponding parameters play the same role across clients. Under heterogeneous data, clients can reach low local loss while learning different internal representations, so averaging their models can perform worse than any individual client.

This paper studies *feature skew*: clients that differ in how often, or how strongly, the features relevant to a shared task occur in their data. FedAvg's averaging rule implicitly assumes that clients represent matching features along the same directions in the model's hidden space. Existing methods for data heterogeneity do not guarantee this. Some limit how far local updates can move from a shared starting point (Li et al., 2020), and others let each client keep personalized parameters (Li et al., 2021b), but none check whether clients actually agree on which direction encodes which feature.

This mismatch can become more pronounced under *superposition*, where a model has to represent more features than it has hidden dimensions, so several features must share the same directions (Elhage et al., 2022). Under superposition, feature skew directly changes the geometry of these shared directions, which is what we analyze here. Our three questions are:

- **RQ1:** How does feature skew, combined with superposition, change the directions clients learn for the same features, and what happens to these directions once local parameters are averaged?

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

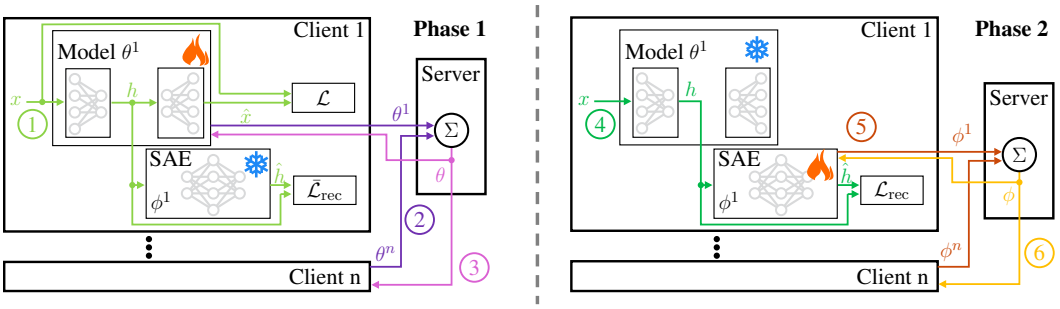


Figure 1: **MechLeash-FL alternates two phases per round.** *Phase 1*: (1) clients train their task model on the task loss $\mathcal{L}$ plus a normalized penalty $\bar{\mathcal{L}}_{\text{rec}}$ from a shared, *frozen* SAE; (2)–(3) the server averages and redistributes the task models. *Phase 2*: with the task model now *frozen*, (4) clients fit a local SAE copy to minimize $\mathcal{L}_{\text{rec}}$; (5)–(6) the server averages and redistributes the SAEs. The SAE only steers training toward consistent latent features and is discarded afterward.

- **RQ2:** Can averaging degrade these directions, and how can this be prevented without knowing which feature is active in a given input, and without changing the deployed model?
- **RQ3:** Does a method built on this idea improve reconstruction quality and representation geometry, both in a controlled synthetic setting and on real, naturally feature-skewed data?

To answer RQ1, we analyze FedAvg in a toy model of superposition. Because a client's feature importance shapes both the strength and the orientation of its learned feature directions, averaging across clients can weaken features that only one client represented strongly, and can create interference between features that were never in conflict on either client alone.

This analysis points to a fix for RQ2. If clients keep their feature directions close to a shared reference during local training, both effects should be limited. Individual feature directions are not directly observable in a general network, so we connect this idea to a more practical one: reconstructing a client's activations using a learned Sparse Autoencoder (SAE, Bricken et al., 2023). Formalizing this connection gives our method, MechLeash-FL, which "leashes" clients' mechanistic features to that shared reference.

MechLeash-FL shares an SAE across clients and alternates between two phases (Figure 1). In the **task-model phase**, the SAE is frozen while each client trains its task model with an added penalty for deviating from the SAE's reconstruction, and the server then aggregates the resulting task models. In the **SAE phase**, the aggregated task model is frozen while each client fits a local copy of the SAE to its own activations, and the server aggregates these local SAEs into the shared SAE for the next round. Because the SAE is only needed during training, it can be discarded afterward, so inference uses the original architecture unchanged.

To answer RQ3, we evaluate MechLeash-FL in two settings. A synthetic setting lets us inspect the mechanism directly by measuring feature strength and pairwise interference through the Gram matrix of the learned directions, as well as task performance. A real-data setting tests whether the improvement in task performance carries over, using Office-Caltech10 as a reconstruction task, where the Amazon, Caltech, DSLR, and Webcam domains form naturally feature-skewed clients. We combine MechLeash-FL with four FL algorithms: FedAvg, FedProx (Li et al., 2020), MOON (Li et al., 2021a), and Ditto (Li et al., 2021b). Our main contributions are:

- A feature-level analysis of federated averaging under feature skew, separating aggregation-induced feature shrinkage from the cross-client overlap that can turn into interference.
- MechLeash-FL, a two-phase FL procedure that uses a shared SAE to keep clients' latent feature directions "leashed", without changing the deployed model's architecture.
- An empirical evaluation connecting reconstruction behavior to representation geometry in a controlled synthetic setting, and demonstrating validation mean squared error reductions of up to 33.08% with FedAvg, FedProx, MOON, and Ditto on Office-Caltech10 reconstruction.

2 Related Work

Superposition and sparse autoencoders. Toy models in Elhage et al. (2022) show how sparse features end up in non-orthogonal directions when there are more features than hidden dimensions, a phenomenon called superposition. Sparse autoencoders (SAEs) express dense activations as sparse combinations of atoms from an overcomplete learned dictionary, revealing interpretable directions in a model's activations (Bricken et al., 2023). The TopK variant used in this work fixes the number of active codes by keeping only the largest encoder responses (Gao et al., 2025). SAEs are normally used in Mechanistic Interpretability after training a model for analysis. We instead train an SAE across clients *during* federated training and use it as a shared reference for features.

FL under data heterogeneity. FedAvg (McMahan et al., 2017) builds a global model by averaging locally updated parameters. Under statistical heterogeneity, local training can move each client's parameters toward a different solution before aggregation, so parameters that are in the same position across clients need not play the same role (Li et al., 2020). One line of work addresses this by matching corresponding neurons across clients before averaging. Probabilistic Federated Neural Matching (Yurochkin et al., 2019) and FedMA (Wang et al., 2020) match neurons from the local weights. Optimal-transport fusion (Singh & Jaggi, 2020) and Git Re-Basin (Ainsworth et al., 2023) reach their best performance by aligning neurons using client activations, which requires a dataset on the server, which is rarely available in FL. We focus on the setting where such data is not available. A second line of work avoids the need for server-side data but requires particular model architectures. FedBN (Li et al., 2021c) keeps batch-normalization parameters local rather than aggregating them, and FedRep (Collins et al., 2021) splits each personalized model into a shared representation and a client-specific head. A third line makes neither assumption. FedProx (Li et al., 2020) limits client drift by penalizing deviation from the current global parameters, and Ditto (Li et al., 2021b) instead treats personalization as a bi-level objective, regularizing each client's own model toward a jointly optimized global model. MOON (Li et al., 2021a) uses a contrastive objective that pulls a client's current representation toward the global model's and away from its own preceding local representation. All of these methods operate on neurons, whether by matching them directly or by keeping a client's hidden activations close to the global model's. None of them addresses alignment at the level of the individual features a network represents in superposition, which is the level we study.

3 Background

3.1 Federated Learning

In FL, a model is trained over R rounds with C clients. In round $r \in [R] = \{1, \dots, R\}$, every client $c \in [C]$ receives the current global parameters θ^r, trains them on its local data $\mathcal{D}^c$ to obtain local parameters $\theta^{r,c}$, and returns $\theta^{r,c}$ to the server, which aggregates the updates into θ^{r+1}. Our analysis focuses on the FedAvg aggregation rule McMahan et al. (2017),

$$\theta^{r+1} = \tfrac{1}{C} \sum\nolimits_{c \in [C]} \theta^{r,c}. \tag{1}$$

3.2 The Superposition Hypothesis

A network often has to represent more features than it has hidden dimensions. Elhage et al. (2022) study how this happens in a toy autoencoder that reconstructs n input features from a hidden space of dimension $m < n$. With weights $W \in \mathbb{R}^{n \times m}$ and bias b (jointly $\theta = (W, b)$), the model is

$$\hat{x} = \mathrm{ReLU}(WW^\top x + b),$$

where $x \in [0,1]^n$ has one coordinate per feature: x_i says how strongly feature i is present. We write W_i for row i of W, the hidden representation the model assigns to feature i. The model is trained on q-sparse inputs with an importance-weighted loss,

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim \mathcal{U}([0,1]^n)^q} \sum\nolimits_{i=1}^{n} \alpha_i (x_i - \hat{x}_i)^2, \tag{2}$$

where $\mathcal{U}([0,1]^n)^q$ is sampled by first choosing q distinct dimensions of a vector. The values of those dimensions are then drawn uniformly from $[0,1]$, and the remaining dimensions are set to 0. The

$\alpha_i > 0$ is the importance of feature i. Elhage et al. (2022) analyze the *1-sparse* case, in which only one feature i is active at a time ($x_i = 1$), which approximates equation 2 by

$$\mathcal{L}(\theta) \approx \sum\nolimits_i \alpha_i \big(1 - \mathrm{ReLU}(\|W_i\|^2 + b_i)\big)^2 + \sum\nolimits_{i \neq j} \alpha_j \,\mathrm{ReLU}(W_j^\top W_i + b_j)^2, \tag{3}$$

where the first term encourages feature reconstruction ($\|W_i\|^2 + b_i \to 1$) and the second penalizes interference between overlapping directions, especially when $\alpha_j \gg \alpha_i$. When n exceeds m, some directions must overlap: the model is forced into *superposition*.

3.3 Sparse Autoencoders

An SAE is trained to reconstruct a hidden activation $h \in \mathbb{R}^m$ as a sparse combination of atoms from an overcomplete dictionary Bricken et al. (2023), and is commonly used after training to find interpretable feature directions inside a network. Given an encoder E, decoder D, and biases e, d, a TopK_k SAE computes a k-sparse code z and a reconstruction $\hat{h}$ as

$$z = \mathrm{TopK}_k\big(\,\mathrm{ReLU}(E(h - d) + e)\big), \qquad \hat{h} = Dz + d, \tag{4}$$

where TopK_k keeps only the k largest non-negative encoder responses and zeroes the rest Gao et al. (2025). We write $\mathrm{SAE}_\phi(h) := \hat{h}$ for an SAE with parameters $\phi = (E, D, e, d)$, and use the mean squared error (MSE) $\mathcal{L}_{\mathrm{rec}}(\phi, h) = \mathrm{MSE}(h - \mathrm{SAE}_\phi(h))$ for its reconstruction error.

4 Problem Statement

We now study how superposition interacts with federated aggregation, using the toy model of Section 3.2. For client c at round r, local training produces $\theta^{r,c} = (W^{r,c}, b^{r,c})$. We analyze what a single client's local training does to its representations and what happens when two such clients are merged by FedAvg. We write $\angle_{ij}(W_i, W_j) = \angle_{ij} \in [0, \pi]$ for the angle between vectors W_i and W_j.

4.1 Feature geometry under superposition

We analyze one client during one local round, so we drop the round and client indices here. Writing $W_j^\top W_i = \|W_i\|\|W_j\| \cos\angle_{ij}$, equation 3 becomes

$$\mathcal{L}(\theta) \approx \sum\nolimits_i \alpha_i \big(1 - \mathrm{ReLU}(\|W_i\|^2 + b_i)\big)^2 + \sum\nolimits_{i \neq j} \alpha_j \,\mathrm{ReLU}\left(\|W_i\|\|W_j\| \cos\angle_{ij} + b_j\right)^2. \tag{5}$$

This form makes two known effects from Elhage et al. (2022) visible. First, minimizing the loss pushes $\cos\angle_{ij} \to 0$ between important features: they become mutually orthogonal whenever there is room. Second, less important features are pushed toward zero length. Throughout the remainder of this section and the next Section, we set $b = 0$. We note that a nonzero bias is what can allow superposition to coexist with zero loss. Fixing $b = 0$ removes this escape route to isolate shrinkage and interference effects driven by feature skew. In our empirical experiments b is learned.

Lemma 1. ***(Partial Equilibrium of Magnitude).*** *For feature i, fix $b_j = 0$ for all j and every angle $\angle_{ij}$ and magnitude $\|W_j\|$, $j \neq i$, and treat $\|W_i\|^2 \geq 0$ in equation 5 as the only free variable. Its constrained minimizer is*

$$\|W_i\|^2 = \Big[1 - \sum\nolimits_{j \in \mathcal{I}_i^+} \big(\tfrac{1}{2} + \tfrac{\alpha_j}{2\alpha_i}\big)\|W_j\|^2 \cos^2\angle_{ij}\Big]_+, \; [\cdot]_+ = \max(0, \cdot), \; \mathcal{I}_i^+ = \{j \neq i : \cos\angle_{ij} \geq 0\}.$$

The proof for Lemma 1 is given in Appendix A.1. If feature i overlaps positively with a much more important feature j ($\alpha_i \ll \alpha_j$, $\|W_j\|^2 > 0$), the term weighted by α_j/α_i can push the bracket below zero, resulting in feature i being shrunk to zero.

4.2 Federated averaging under feature skew

Consider two clients ($C = 2$) that use the same toy model but assign different importance to the same features, $\{\alpha_i^1\}_{i=1}^n \neq \{\alpha_i^2\}_{i=1}^n$. This difference is our model of *feature skew*. The geometry of a learned representation is captured by its Gram matrix,

$$G = WW^\top, \qquad G_{ij} = W_i \cdot W_j. \tag{6}$$

The diagonal entry $G_{ii} = \|W_i\|^2$ measures the strength of feature i's representation, while the off-diagonal entry G_{ij} measures the overlap between two feature representations. Positive overlap can produce interference in reconstruction. We write $G^c = W^c(W^c)^\top$ for client c's Gram matrix. The loss, if sampling the inputs from $\mathcal{U}([0,1]^n)^n$, is minimized exactly at $G = I_n, b = 0$ (Appendix A.2), where I_n is the identity matrix of dimension n. For $n > m$, $G = WW^\top$ has rank at most m while I_n has rank n. This optimum is unreachable, and superposition is unavoidable.

Substituting the FedAvg update equation 1 into equation 6 gives the merged Gram matrix

$$G^{\text{merged}} = \tfrac{1}{4}\big(W^1(W^1)^\top + W^2(W^2)^\top + W^1(W^2)^\top + W^2(W^1)^\top\big). \tag{7}$$

Together with Lemma 1, this decomposition exposes two problems. The first problem is *Feature Shrinkage*. Suppose client 1 shrinks feature i to zero, $W_i^1 = \mathbf{0}$, while client 2 reconstructs it optimally, $\|W_i^2\|^2 = 1$. Then $G_{ii}^{\text{merged}} = \frac{1}{4}\|W_i^2\|^2 = \frac{1}{4}$, a fourfold drop from the optimal value 1, even though one client represented feature i perfectly. The second problem is *Cross-client interference*, where the interference term in equation 5 only pushes two directions apart when both have non-zero length.

Lemma 2. ***(Interference Force Vanishes with Magnitude).*** *With $b = 0$ and $u_i = W_i/\|W_i\|$, the pull that the interference term $\mathrm{ReLU}(W_j^\top W_i)^2$ exerts on the direction u_i (formally, its Riemannian gradient) satisfies*

$$\big\|\nabla_{u_i}^{\text{Riem}} \mathrm{ReLU}(W_j^\top W_i)^2\big\| = \|W_i\|^2\|W_j\|^2\big[\sin(2\angle_{ij})\big]_+ \;\le\; \|W_i\|^2\|W_j\|^2, \tag{8}$$

with equality iff $\angle_{ij} = \pi/4$. Summing over $j \neq i$ with weights $\alpha_i + \alpha_j$,

$$\Big\|\nabla_{u_i}^{\text{Riem}} \sum\nolimits_{j\neq i}(\alpha_i+\alpha_j)\,\mathrm{ReLU}(W_j^\top W_i)^2\Big\| \le \Gamma_i(W), \Gamma_i(W) := \sum\nolimits_{j\neq i}(\alpha_i+\alpha_j)\|W_i\|^2\|W_j\|^2. \tag{9}$$

See Appendix A.3 for formal definitions and proof. If feature i matters only to client 1 and feature j only to client 2, one of $\|W_i\|, \|W_j\|$ is zero or, more generally, small, so by Lemma 2 the force keeping their directions apart vanishes, or is correspondingly weak, *on both clients*: FedAvg has no mechanism to prevent them from becoming positively correlated once merged, and local training can even pull them together (Appendix B).

5 MechLeash-FL

Section 4 shows that nothing in the training objective keeps a feature direction that matters to one client close to where it started once that feature has become weak for other clients, and that this is when shrinkage (Lemma 1) can happen. This suggests a direct fix. Given a reference direction per feature, penalize rotating away from it. We first study this idea in the toy model (Section 3.2), then turn it into a method that does not require knowing individual feature directions.

5.1 Motivation: an idealized directional leash

Let $\hat{W}_i$ be the direction assigned to feature i at the *start* of a local round, and W_i its direction *during* training. We define $\angle_i := \angle(W_i, \hat{W}_i)$ and add a term penalizing rotation away from $\hat{W}_i$,

$$\mathcal{R}(W, \hat{W}) = \sum\nolimits_i \sin^2 \angle_i, \tag{10}$$

giving the leashed 1-sparse loss with hyperparameter $\lambda_{\mathcal{R}}$

$$\mathcal{L}_{\mathcal{R}}(\theta) = \sum\nolimits_i \alpha_i\big(1 - \mathrm{ReLU}(\|W_i\|^2 + b_i)\big)^2 + \sum\nolimits_{i\neq j}\alpha_j\,\mathrm{ReLU}(W_j^\top W_i + b_j)^2 + \lambda_{\mathcal{R}}\,\mathcal{R}(W, \hat{W}), \tag{11}$$

for which we can show the following lemma.

Lemma 3. ***(Directional Stability).*** *Let $u_i = W_i/\|W_i\|$ be a stationary point of the direction subproblem of equation 11, with $b = 0$, $\|W_i\| > 0$. Then*

$$|\sin(2\angle_i)| \;\le\; \Gamma_i(W)/\lambda_{\mathcal{R}} \tag{12}$$

and

$$\angle_i \;\le\; (\pi\,\Gamma_i(W))/(4\lambda_{\mathcal{R}}) \;=:\; \epsilon_i \tag{13}$$

holds whenever $\angle_i \in [0, \pi/4]$.

The proof is given in Appendix C.1. Raising $\lambda_{\mathcal{R}}$ relative to the interference a feature is exposed to, $\Gamma_i(W)$, tightens how far its direction can drift. This bound on individual drift, together with the triangle inequality, bounds how close two *different* feature directions can become. Once $\hat{W}_i$ and $\hat{W}_j$ are separated by more than $\epsilon_i+\epsilon_j = \frac{\pi}{4\lambda_{\mathcal{R}}}(\Gamma_i(W)+\Gamma_j(W))$, interference between the corresponding features is capped (Lemma 6, Appendix C.2). Repeating the argument behind Lemma 1 with this cap in place of the true interference term shows leashing can also mitigate shrinkage (Lemma 7, Appendix C.3). A large $\lambda_{\mathcal{R}}$ therefore limits both cross-feature drift and the shrinkage it causes.

5.2 From directional to reconstruction leashing

The leash loss equation 10 assumes the model exposes one direction W_i per feature and that $\hat{W}_i$ is known. Neither holds in a general network. A hidden activation can mix several features whose identities are unobserved. We show that in a simplified setting the *same* leash loss can be written as a reconstruction loss of an SAE, something we *can* compute without knowing which feature is active.

Consider the one-sparse case ($h = W_i$), and suppose the leashes $\{\hat{W}_j\}_{j=1}^n$ are separated, every W_i stays closer to its leash $\hat{W}_i$ than to others, and drifts by less than $\|\hat{W}_i\|$, during training.

Proposition 1. ***(Reconstruction leashing Equals Directional Leashing).*** *Under these conditions there is a* TopK_1 *SAE* (E, D, e, d)*, built only from the leashes* $\{\hat{W}_j\}_{j=1}^n$*, such that*

$$\mathcal{R}(W, \hat{W}) = \sum\nolimits_i \left\| W_i - \left(D\,\mathrm{TopK}_1(\mathrm{ReLU}(E(W_i - d) + e)) + d\right)\right\|_2^2 / \|W_i\|^2.$$

Details and the proof are given in Appendix C.4. This equivalence is exact only in the simplified setting above, but it motivates two design choices for a practical method. The SAE should be fit to activations from the *merged* model, so all clients reconstruct against the same reference. Additionally, the SAE should be held fixed while the task model trains, or it would drift along with the local models instead of leashing them.

5.3 The MechLeash-FL algorithm

Following Section 5.2, MechLeash-FL replaces the unknown per-feature leashes with a single SAE (equation 4), shared across clients and fit to the merged model's activations. Clients never share activations, and the SAE is discarded after training. The deployed model's architecture and inference cost are unchanged. Training runs for R rounds over clients $c = 1, \ldots, C$, starting from a task model θ^1 and an SAE ϕ^1. Each round has two phases with the two losses

$$\mathcal{L}^{\mathrm{Phase1}}(\theta, \phi) = \mathcal{L}(\theta) + \lambda_{\mathrm{rec}}\,\bar{\mathcal{L}}_{\mathrm{rec}}(\phi; h) \quad (14) \qquad \mathcal{L}^{\mathrm{Phase2}}(\theta, \phi^{r,c}) = \mathcal{L}_{\mathrm{rec}}(\phi^{r,c}; h) \quad (15)$$

where $\mathcal{L}$ is the task loss, $\bar{\mathcal{L}}_{\mathrm{rec}}(\phi; h) := (\mathcal{L}_{\mathrm{rec}}(\phi; h))/(\max(\mathrm{mean}(h^2), \varepsilon))$, and h denotes a mini-batch of hidden activations. Both means are over batch and hidden dimensions, and $\varepsilon = 10^{-6}$. In the *task-model phase*, the SAE parameters ϕ^r are frozen and each client c optimizes $\theta^{r,c}$ with respect to $\mathcal{L}^{\mathrm{Phase1}}(\theta^{r,c}, \phi^r)$. The server then aggregates the resulting client models into θ^{r+1}. During the first τ warm-up rounds, however, we optimize $\mathcal{L}$ instead of $\mathcal{L}^{\mathrm{Phase1}}$. In the *SAE phase*, the task model is frozen at θ^{r+1} and every client optimizes a local copy $\phi^{r,c}$ of the SAE ϕ^r according to the loss $\mathcal{L}^{\mathrm{Phase2}}(\theta^{r+1}, \phi^{r,c})$. Then the server averages the local SAEs to get ϕ^{r+1}. Algorithm 1 gives the full procedure and Figure 1 provides a figure illustrating the method.

6 Experimental Setup

We evaluate MechLeash-FL in two settings: a controlled synthetic reconstruction task, where known input features enable analysis of representation geometry during federated learning, and image reconstruction on Office-Caltech10 (Gong et al., 2012), where 4 domains give feature-skewed clients. For the main experiments, we report means and 95% confidence intervals over 10 repetitions.

6.1 Synthetic Setting

Data and training. We use the tied autoencoder from Section 3.2 with $n = 100$ input features and $m = 25$ hidden dimensions. For data generation we sample $(a_i^1, a_i^2) \sim \mathrm{Dirichlet}(0.1, 0.1)$

Algorithm 1: MechLeash-FL

Input: Initial task model θ^1, initial SAE ϕ^1, clients $c = 1, \ldots, C$, rounds R, warmup rounds τ
Notation: $\texttt{optimize}^c(\theta, \mathcal{L})$ denotes optimization of the parameters θ on the data of client c w.r.t. the loss $\mathcal{L}$ (second argument); all other arguments of $\mathcal{L}$ are held fixed.

for $r \leftarrow 1$ **to** R **do**

// Phase 1: update task model, SAE frozen at ϕ^r

$\theta^{r,c} \leftarrow \theta^r$ for all c

$\theta^{r,c} \leftarrow \begin{cases} \texttt{optimize}^c(\theta^{r,c}, \mathcal{L}(\theta^{r,c})) & \text{if } r \le \tau, \\ \texttt{optimize}^c(\theta^{r,c}, \mathcal{L}^{\text{Phase1}}(\theta^{r,c}, \phi^r)) & \text{else} \end{cases}$ for all c in parallel // Warmup

$\theta^{r+1} \leftarrow \frac{1}{C}\sum_{c=1}^{C} \theta^{r,c}$ // aggregate task model

// Phase 2: update SAE, task model frozen at θ^{r+1}

$\phi^{r,c} \leftarrow \phi^r$ for all c

$\phi^{r,c} \leftarrow \texttt{optimize}^c(\phi^{r,c}, \mathcal{L}^{\text{Phase2}}(\theta^{r+1}, \phi^{r,c}))$ for all c in parallel

$\phi^{r+1} \leftarrow \frac{1}{C}\sum_{c=1}^{C} \phi^{r,c}$ // average client SAEs

Output: Task model θ^{R+1}; discard ϕ^{R+1}

for $\tilde{i} \in [n]$ and set $\tilde{p}_{\tilde{i}}^c = 0.4a_{\tilde{i}}^c$ for clients $c \in \{1, 2\}$. We then define a permutation Π satisfying $\tilde{p}^1_{\Pi(1)} - \tilde{p}^2_{\Pi(1)} \le \cdots \le \tilde{p}^1_{\Pi(n)} - \tilde{p}^2_{\Pi(n)}$. Given this ordering, we set $p_i = \tilde{p}_{\Pi(i)}$ for feature $i \in [n]$, which is then used to sample the data. Features are sampled independently: feature i for client c is zero with probability $1 - p_i^c$ and otherwise uniform on $[0, 1]$. The probabilities remain fixed across runs. We use mean squared reconstruction error as the task loss, $\mathcal{L}(\theta) = \mathbb{E}_x\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2\right]$, where inputs are sampled from each client's distribution and $\hat{x}$ is the model's reconstruction. Both clients train their model for $R = 200$ rounds, with 400 local Adam updates per round, learning rate 0.01, and batch size 256. MechLeash-FL uses $\lambda_{\text{rec}} = 0.01$ and an SAE with 100 atoms and $\text{TopK}_{k=5}$. After the task models are averaged, each client trains its SAE for 100 updates at learning rate 0.003 using activations from the merged task model. We use $\tau = 0$ warm-up rounds.

Evaluation and geometry. Each client has 8192 fixed evaluation samples, separate from training and shared across evaluations over rounds, methods and local and merged models. We report pooled loss and *merge barrier*

$$\mathcal{L}_{\text{pool}}(\theta) = \tfrac{1}{2}\sum_{c=1}^{2} \mathcal{L}^c(\theta), \qquad B^r_{\text{merge}} = \mathcal{L}_{\text{pool}}(\theta^{r+1}) - \tfrac{1}{2}\sum_{c=1}^{2} \mathcal{L}_{\text{pool}}(\theta^{r,c}). \tag{16}$$

For the *diagnostic merge*, each client performs 400 additional local updates from the final model, keeping the SAE fixed. To measure feature shrinkage we report *feature strength* as $\frac{1}{n}\sum_i G^{\text{merged}}_{ii}$, the mean diagonal of the Gram matrix. We also measure interference caused by feature skew as *positive overlap*, the mean positive Gram entry between features more important to client 2 and features more important to client 1. Let $\tilde{n} = \max\{i \mid p_i^1 - p_i^2 < 0\}$ so that $[\tilde{n}]$ indexes features more important for client 2 and $[n] \setminus [\tilde{n}]$ indexes features more important for client 1, and define positive overlap as $(\sum_{i\in[\tilde{n}], j\in[n]\setminus[\tilde{n}]} \max(G^{\text{merged}}_{ij}, 0))/(\tilde{n}(n - \tilde{n}))$. More details are given in Appendix D.

6.2 Real Data Setting

Data and models. We use the four domains of Office-Caltech10 (Gong et al., 2012) as clients: Amazon, Caltech, DSLR, and Webcam. We select 62 training and 50 validation images per domain, with no overlap. These subsets remain fixed across runs. Images are resized to 256×256. The encoder has six ReLU convolutional layers with $(16, 32, 64, 64, 64, 64)$ output channels, 4×4 kernels, stride 2, and padding 1. A linear 1×1 convolution produces a $16 \times 4 \times 4$ bottleneck, flattened to 256 dimensions for the SAE. The decoder mirrors this structure using transposed convolutions and a sigmoid RGB output. The SAE has 512 atoms and sparsity $k = 8$.

Training and baselines. All four methods train for 48 rounds with 100 local task epochs per round, Adam, and batch size 16. MechLeash-FL uses 20 local SAE epochs per round and $\tau = 1$ warm-up round. Initial task and SAE learning rates are respectively 10^{-3} and 0.003, except for

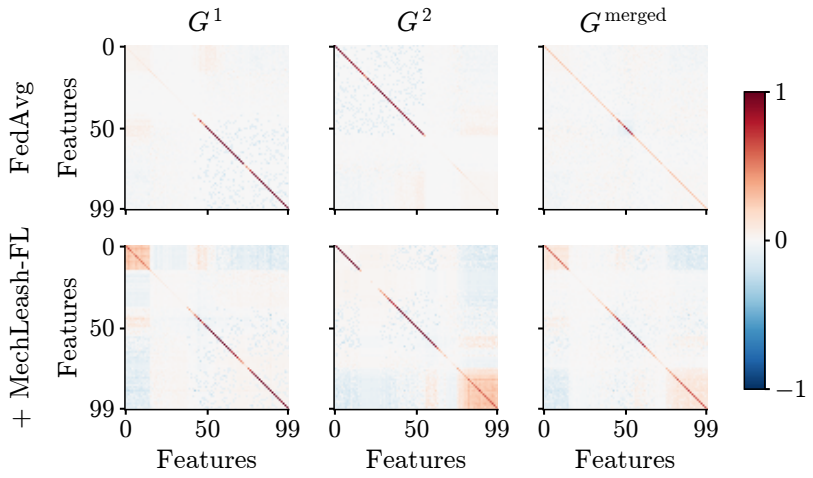


Figure 2: Mean Gram matrices over ten repetitions at the diagnostic merge. Columns: G^1, G^2, and G^{merged}. Top: FedAvg. Bottom: FedAvg+MechLeash-FL.

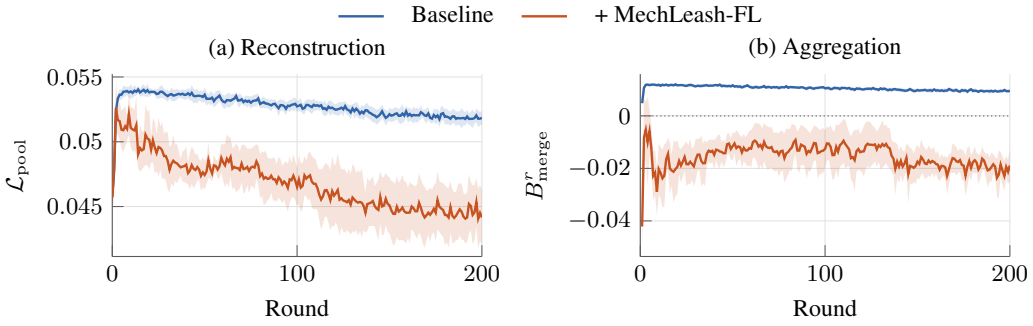


Figure 3: Pooled MSE $\mathcal{L}_{\text{pool}}$ and merge barrier B^r_{merge} over ten repetitions for the synthetic setting. Curves show means with 95% confidence intervals.

MOON, which uses 0.0003 for both. Task learning rates halve each round after round 4, down to 1% of their initial values. SAE learning rates remain constant. For FedAvg, FedProx, MOON, and Ditto, respectively, λ_{rec} is 0.3, 1, 0.03, and 0.3. FedProx uses deviation penalization hyperparameter $\mu = 0.001$; MOON uses contrastive weight 0.1, temperature 1, and no projection head. Ditto uses proximal weight 0.1 and 100 additional personalized epochs per round, with MechLeash-FL applied only to its global update. More details are described in Appendix D.

Reporting. After the last round, we evaluate the global FedAvg, FedProx, and MOON models on all 200 validation images and average squared reconstruction errors over images, pixels, and RGB channels. For Ditto, each client uses its personalized model on its own 50 validation images and the 4 client MSEs are averaged. We use exact one-sided paired Wilcoxon signed-rank tests (Wilcoxon, 1945) for lower final loss with MechLeash-FL, Holm-corrected (Holm, 1979) across baselines.

7 Results Analysis

7.1 Synthetic Setting

At round 200, MechLeash-FL reduces mean pooled MSE from 0.0518 ± 0.0005 to 0.0442 ± 0.0021, an improvement of 14.82% (Figure 3(a)). The paired one-sided Wilcoxon test gives $p = 0.00098$. At round 200, we further see that merging is harmful for FedAvg (merge barrier 0.0095 ± 0.0006) but actually increases performance when trained with MechLeash-FL (-0.0189 ± 0.0061) (Figure 3(b)). Figure 2 shows the mean Gram matrices at the diagnostic merge. For FedAvg, feature strength decreases from 0.532 and 0.529 on the two clients to 0.307 after aggregation, a reduction of $42.1\% \pm$

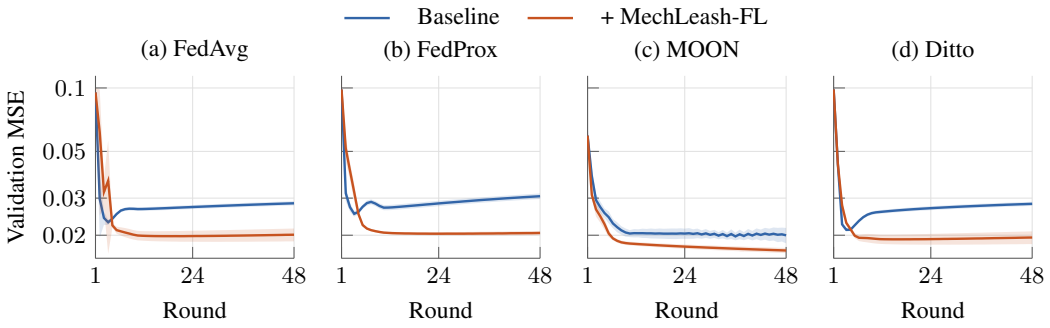


Figure 4: Office-Caltech10 validation MSE over ten repetitions, with 95% confidence intervals and a logarithmic vertical axis.

0.8%. With MechLeash-FL, the reduction is $(23.1 \pm 3.5)\%$, leaving a mean diagonal of 0.483 after aggregation. At the same diagnostic merge, positive overlap increases by 0.0050 ± 0.0006 under FedAvg but decreases by 0.0042 ± 0.0012 under MechLeash-FL. Together, the reduced shrinkage and overlap show that MechLeash-FL preserves stronger feature representations while limiting the positive overlap that can cause interference, consistent with our theoretical motivation.

7.2 Real Data Setting

MechLeash-FL lowers mean validation MSE for all four baselines (Figure 4). At round 48, FedAvg improves from 0.0284 ± 0.0007 to 0.0201 ± 0.0014, a reduction of 29.08%, while FedProx improves from 0.0306 ± 0.0009 to 0.0205 ± 0.0005, a reduction of 33.08%. MOON achieves the lowest final MSE, improving from 0.0200 ± 0.0017 to 0.0169 ± 0.0005 (15.29%). Ditto's personalized models also benefit, improving from 0.0282 ± 0.0006 to 0.0195 ± 0.0014 (30.78%), even though MechLeash-FL is applied only to its global-model update. All four comparisons remain significant after Holm correction ($p = 0.00391$). We note that MechLeash-FL requires about $1.54\times$ more communication than the baselines, yet even within the same data-upload budget, it still achieves a 12.7-33.6% lower MSE than the four baselines (Appendix, Figure 6). These consistent gains demonstrate that the practical benefits of our theoretically motivated reconstruction leash extend beyond the controlled synthetic setting to naturally feature-skewed image data. The improvement across parameter regularization, representation regularization, and personalization baselines further shows that a shared SAE reference can complement these existing approaches to heterogeneity. A hyperparameter sweep showed similar mean losses across $k = 2$-16. The task performance is more sensitive to the reconstruction coefficient λ_{rec}, as small values do not leash the features enough, while large values are too restrictive (Appendix D.4).

8 Conclusion

We analyzed how feature skew and superposition cause parameter averaging to weaken features and introduce cross-client interference. Once a feature weakens on one client, the interference force keeping unrelated directions apart vanishes with its magnitude, letting shrunk features drift or collapse together after merging. A directional leash counteracts this by binding drift, limiting both interference and shrinkage. Since individual feature directions are unobservable in general networks, we showed this leash is realized by an SAE reconstruction loss, giving MechLeash-FL. Experiments confirm this mechanism. In a synthetic setting, MechLeash-FL reduces feature shrinkage from 42.1% to 23.1%, reverses the harmful merge barrier, and cuts reconstruction error by 14.82%. On Office-Caltech10, the same leash lowers validation MSE by 15.29%-33.08% across FedAvg, FedProx, MOON, and Ditto. These results support leashing clients' latent features to a shared mechanistic reference as an effective, architecture-agnostic fix for feature skew under superposition.

Limitations and Future Work. Our analysis uses a synthetic toy model, and the real-data evaluation covers one dataset and reconstruction architecture. Future work should evaluate settings with more clients, additional architectures, and tasks beyond reconstruction.

AI USE STATEMENT

We used generative AI tools to assist with experimental implementation, analysis, and manuscript preparation. This assistance included developing and debugging training code and preparing scripts for submitting and monitoring experiments on our Slurm cluster. AI tools also assisted with organizing and documenting the anonymized submission code.

In our theoretical work, AI was used within an iterative back-and-forth process: proposing and refining hypotheses, suggesting critical ingredients for proofs, and assisting with proof writing. No proof was AI-written in isolation. Every proof was developed through this collaborative process and manually checked by the authors. Human refinement and ideation played a strong role in most proofs, most notably for our main proposition establishing the equivalence between the directional leash and the SAE reconstruction loss.

For analysis, AI assistance supported extracting recorded measurements, calculating summary statistics and significance tests, preparing tables and figures, and discussing experimental design choices and possible interpretations of the results. AI tools were not used to generate any data or clean/reformat the Office-Caltech10 dataset directly, only to help design and code the procedures that did. For writing, these tools assisted with section organization, wording, LaTeX formatting, and consistency checks between the manuscript, implementation, and reported results. We take responsibility for the final manuscript and accompanying artifacts, including all AI-assisted text, code, analyses, and claims.

REPRODUCIBILITY STATEMENT

Section 6 and Appendix D describe the experimental protocols, data preparation, hyperparameters, and statistical procedures. Proofs are provided in Appendices A and C. The accompanying anonymized code includes configurable training scripts and scripts for regenerating the reported tables and figures from recorded results. We provide the Office input tensors and fixed data splits, random seeds, experiment configurations, per-run measurements, and recorded software environments. The code package also includes dependency specifications, validation tests, and instructions for rerunning each experiment. Numerical results from new training runs may vary across hardware and software environments.

REFERENCES

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008. ISBN 9780691132983. URL http://www.jstor.org/stable/j.ctt7smmk.

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CQsmMYmlP5T.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Earl A. Coddington and Norman Levinson. *Theory of Ordinary Differential Equations*. McGraw-Hill, 1955.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2089–2099. PMLR, 2021. URL https://proceedings.mlr.press/v139/collins21a.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/42ef3308c230942d223c411adf182c88-Abstract-Conference.html.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073. IEEE, 2012. doi: 10.1109/CVPR.2012.6247911.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 1979. doi: 10.2307/4615733.

Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, 3 edition, 2002.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10713–10722, 2021a. URL https://openaccess.thecvf.com/content/CVPR2021/html/Li_Model-Contrastive_Federated_Learning_CVPR_2021_paper.html.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020. URL https://proceedings.mlsys.org/paper_files/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6357–6368. PMLR, 2021b. URL https://proceedings.mlr.press/v139/li21h.html.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-iid features via local batch normalization. In *International Conference on Learning Representations*, 2021c. URL https://openreview.net/forum?id=6YEQUn0QICG.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Dragoslav S. Mitrinović. *Analytic Inequalities*. Grundlehren der mathematischen Wissenschaften. Springer Berlin, Heidelberg, 1970. ISBN 978-3-642-99970-3. doi: 10.1007/978-3-642-99970-3.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22045–22055. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/fb2697869f56484404c8ceee2985b01d-Abstract.html.

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.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. ISSN 00994987. URL http://www.jstor.org/stable/3001968.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7252–7261. PMLR, 2019. URL https://proceedings.mlr.press/v97/yurochkin19a.html.

A Proofs for Section 4

A.1 Partial Equilibrium of Magnitude

Lemma 1. ***(Partial Equilibrium of Magnitude).*** *For feature i, fix $b_j = 0$ for all j and every angle $\angle_{ij}$ and magnitude $\|W_j\|$, $j \neq i$, and treat $\|W_i\|^2 \geq 0$ in equation 5 as the only free variable. Its constrained minimizer is*

$$\|W_i\|^2 = \Big[1 - \sum\nolimits_{j \in \mathcal{I}_i^+} \big(\tfrac{1}{2} + \tfrac{\alpha_j}{2\alpha_i}\big) \|W_j\|^2 \cos^2 \angle_{ij}\Big]_+, \; [\cdot]_+ = \max(0, \cdot), \; \mathcal{I}_i^+ = \{j \neq i : \cos \angle_{ij} \geq 0\}.$$

Proof. Fix $b_j = 0$ for all j, and treat $\|W_i\|^2 \geq 0$ as the only free variable in equation 5, holding every angle $\angle_{ij}$ and $\|W_j\|$, $j \neq i$, fixed. The first-order condition $\partial \mathcal{L} / \partial \|W_i\|^2 = 0$ is

$$0 = 2\alpha_i \big(\|W_i\|^2 - 1\big) + \sum_{j \neq i} (\alpha_i + \alpha_j) \, \mathbf{1}\{\cos \angle_{ij} \geq 0\} \, \|W_j\|^2 \cos^2 \angle_{ij}, \tag{17}$$

which rearranges to

$$\|W_i\|^2 = 1 - \sum_{j \in \mathcal{I}_i^+} \Big(\tfrac{1}{2} + \tfrac{\alpha_j}{2\alpha_i}\Big) \|W_j\|^2 \cos^2 \angle_{ij}, \qquad \mathcal{I}_i^+ = \{j \neq i : \cos \angle_{ij} \geq 0\}. \tag{18}$$

Since $\|W_i\|^2 \geq 0$ is required, the true stationary point clips this at 0, giving the stated bound. If $\alpha_i \ll \alpha_j$ for some $j \in \mathcal{I}_i^+$ with $\|W_j\|^2 \approx 1$, the term weighted by $\alpha_j / (2\alpha_i)$ can drive the unclipped expression negative, so the optimum is exactly $\|W_i\|^2 = 0$. A less important feature overlapping with a more important one can be shrunk away entirely. □

A.2 Optimality of the Toy Model

Lemma 4 (Optimality of the Toy Model)**.** *Let $x \sim \mathrm{Unif}([0,1]^n)^n$ with independent coordinates, $\hat{x} = \mathrm{ReLU}(Gx + b)$ for $G \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, and $\alpha_i > 0$. Then $\mathcal{L} = \mathbb{E}_x \sum_i \alpha_i (x_i - \hat{x}_i)^2 = 0$ if and only if $G = I_n$ and $b = 0$.*

Proof. Since each summand $\alpha_i (x_i - \hat{x}_i)^2 \geq 0$, a nonnegative integral is zero iff its integrand is zero a.e., so $\mathcal{L} = 0$ iff, for every i, $x_i = \hat{x}_i$ for almost every x.

Fix i. The set $\{x_i = \hat{x}_i\}$ has measure 1 by the argument above, and $\{x_i > 0\}$ also has measure 1 (its complement $\{x_i = 0\}$ is a hyperplane, hence measure 0). The intersection of two measure-1 sets in $[0,1]^n$ again has measure 1, so on a measure-1 set we have both $x_i > 0$ and $x_i = \hat{x}_i = \mathrm{ReLU}((Gx + b)_i)$. Since $\mathrm{ReLU}(y) = x_i > 0$ forces $y = x_i$ (the only way $\mathrm{ReLU}(y)$ can equal a strictly positive number is if y already equals it), we get $(Gx + b)_i = x_i$ on this measure-1 set.

Define the affine function $f_i(x) := (Gx + b)_i - x_i = (G_{i,:} - e_i^\top)x + b_i$. We have just shown f_i vanishes on a set of measure 1. Write $a := G_{i,:} - e_i^\top \in \mathbb{R}^n$.

- If $a \neq 0$, then $\{x : f_i(x) = 0\} = \{x : a^\top x = -b_i\}$ is a hyperplane, which has Lebesgue measure zero. A measure-zero set cannot contain a measure-1 subset of $[0,1]^n$ (which has positive volume), a contradiction. Hence $a = 0$, i.e., $G_{i,:} = e_i^\top$.
- With $a = 0$, $f_i(x) = b_i$ is constant. A nonzero constant function vanishes on a set of measure 0 (the empty set), again contradicting that it vanishes on a set of measure 1 unless $b_i = 0$.

Hence $f_i \equiv 0$, i.e., $G_{i,:} = e_i^\top$ and $b_i = 0$.

Conversely, if $G_{i,:} = e_i^\top$ and $b_i = 0$ for every i, then $(Gx + b)_i = x_i$ for every x, so $\hat{x}_i = \mathrm{ReLU}(x_i) = x_i$ off the null set $\{x_i = 0\}$, giving $x_i = \hat{x}_i$ a.e. Applying this to every i gives $G = I_n$, $b = 0$. □

Applied to $G = WW^\top$, this is the fact used in Section 4.2.

A.3 Interference Force Vanishes with Magnitude

Lemma 2. ***(Interference Force Vanishes with Magnitude).*** *With $b = 0$ and $u_i = W_i/\|W_i\|$, the pull that the interference term $\mathrm{ReLU}(W_j^\top W_i)^2$ exerts on the direction u_i (formally, its Riemannian gradient) satisfies*

$$\left\|\nabla^{\mathrm{Riem}}_{u_i} \mathrm{ReLU}(W_j^\top W_i)^2\right\| = \|W_i\|^2\|W_j\|^2\left[\sin(2\angle_{ij})\right]_+ \;\le\; \|W_i\|^2\|W_j\|^2, \tag{8}$$

with equality iff $\angle_{ij} = \pi/4$. Summing over $j \neq i$ with weights $\alpha_i + \alpha_j$,

$$\left\|\nabla^{\mathrm{Riem}}_{u_i} \sum\nolimits_{j\neq i}(\alpha_i+\alpha_j)\,\mathrm{ReLU}(W_j^\top W_i)^2\right\| \le \Gamma_i(W), \Gamma_i(W) := \sum\nolimits_{j\neq i}(\alpha_i+\alpha_j)\|W_i\|^2\|W_j\|^2. \tag{9}$$

Proof. For a differentiable function f on the unit sphere, the *Riemannian gradient* at u is the component of the ordinary gradient tangent to the sphere (Absil et al., 2008),

$$\nabla^{\mathrm{Riem}}_u f = (I - uu^\top)\nabla_u f, \tag{19}$$

i.e., the part that actually moves u along the sphere; f is stationary at u exactly when this vanishes.

Write $W_i = \|W_i\|u_i$, $W_j = \|W_j\|u_j$, so $\mathrm{ReLU}(W_j^\top W_i)^2 = \|W_i\|^2\|W_j\|^2\,\mathrm{ReLU}(\cos\angle_{ij})^2$. Differentiating w.r.t. u_i using $\nabla_{u_i}(u_j^\top u_i) = u_j$ gives ordinary gradient $2\|W_i\|^2\|W_j\|^2\cos_+(\angle_{ij})u_j$, with $\cos_+ = \max(\cos(\cdot), 0)$. Projecting onto $u_i^\perp$ and using $\|(I - u_iu_i^\top)u_j\| = \sin\angle_{ij}$,

$$\left\|\nabla^{\mathrm{Riem}}_{u_i} \mathrm{ReLU}(W_j^\top W_i)^2\right\| = 2\|W_i\|^2\|W_j\|^2\cos_+(\angle_{ij})\sin(\angle_{ij}).$$

For $\angle_{ij} \in [0, \pi/2]$ this equals $\|W_i\|^2\|W_j\|^2\sin(2\angle_{ij}) \ge 0$; for $\angle_{ij} \in (\pi/2, \pi]$, $\cos_+ = 0$ while $\sin(2\angle_{ij}) \le 0$. Both cases equal $\|W_i\|^2\|W_j\|^2[\sin(2\angle_{ij})]_+$, proving equation 8; the bound follows since $|\sin(\cdot)| \le 1$. Writing $I_i(W) = \sum_{j\neq i}(\alpha_i + \alpha_j)\,\mathrm{ReLU}(W_j^\top W_i)^2$, the weight $\alpha_i + \alpha_j$ arises because W_i appears in equation 3 both as the feature reconstructing itself (weight α_i) and as a source of interference in feature j's term (weight α_j). The triangle inequality over j gives equation 9. □

B Shared Background Features Can Pull Client-Specific Ones to a Common Direction

Section 4.2 shows only that nothing keeps a client-specific feature direction away from another client's. Here we show something stronger. Local training can *pull* two such directions together.

B.1 Setup

Fix $b = \mathbf{0}$. Suppose, as in Section 4.2 (e.g. after the shrinkage of Lemma 1), that $W_j^1 = \mathbf{0}$ and $W_i^2 = \mathbf{0}$ exactly, while every other row agrees between clients, $W_k^1 = W_k^2 =: W_k^{(0)}$ for $k \neq i, j$. From here, we assume that only the directions $u_i = W_i/\|W_i\|$ (client 1) and $u_j = W_j/\|W_j\|$ (client 2) continue training, each at its own fixed magnitude $\rho_i := \|W_i\|, \rho_j := \|W_j\| > 0$; all other rows, including the zero rows, stay fixed. Assume shared background importance $\alpha_k^1 = \alpha_k^2 =: \alpha_k$ for $k \neq i, j$, and equal importance for the two heterogeneous features, $\alpha_i^1 = \alpha_j^2 =: \alpha^\star$.

B.2 Both clients follow the same potential

Client 1's loss restricted to u_i at fixed ρ_i is, using $W_j^1 = \mathbf{0}$ so the interference term with j vanishes identically,

$$f_1(u_i) = \alpha^\star(1-\rho_i^2)^2 + \rho_i^2\,\Phi^\star(u_i), \qquad \Phi^\star(u) := \sum_{k\neq i,j} (\alpha^\star + \alpha_k)\,\mathrm{ReLU}(W_k^{(0)\top} u)^2. \tag{20}$$

The identical computation for client 2 (using $W_i^2 = \mathbf{0}$) gives $f_2(u_j) = \alpha^\star(1-\rho_j^2)^2 + \rho_j^2\,\Phi^\star(u_j)$: *the same* function $\Phi^\star$ governs both clients, because the term that could have differed, interference with the other client's now-zero feature, vanishes identically.

B.3 From the losses to the angular dynamics

Since ρ_i is fixed, the constant term in equation 20 depends only on ρ_i, not on u_i, so it contributes nothing to the gradient in u_i. Gradient descent on the direction u_i, directly on the sphere $\mathbb{S}^{m-1}$, is therefore

$$\dot{u}_i = -\nabla_{u_i}^{\mathrm{Riem}} f_1(u_i) = -\rho_i^2\,\nabla_{u_i}^{\mathrm{Riem}} \Phi^\star(u_i), \tag{21}$$

using that ρ_i^2 is a fixed positive constant and so factors out of the gradient in u_i. The identical computation for client 2 gives

$$\dot{u}_j = -\rho_j^2\,\nabla_{u_j}^{\mathrm{Riem}} \Phi^\star(u_j). \tag{22}$$

Both are constant-rate rescalings of the *same* gradient flow $\dot{u} = -\nabla_u^{\mathrm{Riem}}\Phi^\star(u)$ on $\mathbb{S}^{m-1}$, by the positive constants ρ_i^2 and ρ_j^2 respectively.

Lemma 5 (Global Existence and Uniqueness of the Flow)**.** *For any $u(0) \in \mathbb{S}^{m-1}$, the initial value problem $\dot{u} = -\nabla_u^{\mathrm{Riem}}\Phi^\star(u)$, $u(0)$ given, has a unique solution $u(\cdot)$ defined for all $t \geq 0$, called* the trajectory started at $u(0)$. *The same holds for the rescaled flows in equation 21 and equation 22, and more generally for $\dot{u} = -\rho^2\nabla_u^{\mathrm{Riem}}\Phi^\star(u)$ for any constant $\rho > 0$.*

Proof. Write $f(u) := -\nabla_u^{\mathrm{Riem}}\Phi^\star(u)$; the equation is autonomous, i.e., f does not depend on t. We verify the hypotheses of the Picard-Lindelöf theorem (Coddington & Levinson, 1955) globally on $\mathbb{S}^{m-1}$, rather than only locally, using that $\mathbb{S}^{m-1}$ is compact.

Continuity. $\Phi^\star$ is a finite sum of terms $\mathrm{ReLU}(w^\top u)^2$ for fixed vectors w, each continuously differentiable with derivative $2\,\mathrm{ReLU}(w^\top u)\,w$, itself a continuous (piecewise-linear) function of u. Hence $\Phi^\star$ is C^1, i.e., differentiable with continuous gradient $\nabla_u\Phi^\star$. The tangent-space projection $u \mapsto (I - uu^\top)$ is smooth, so $f(u) = -(I-uu^\top)\nabla_u\Phi^\star(u)$ is continuous on all of $\mathbb{S}^{m-1}$.

Boundedness. A continuous function on a compact set is bounded and attains a maximum. Since $\mathbb{S}^{m-1}$ is compact and f is continuous, there is a single constant $M < \infty$ with $\|f(u)\| \leq M$ for every $u \in \mathbb{S}^{m-1}$, not just near one starting point, but everywhere.

Lipschitz continuity. Each $\mathrm{ReLU}(w^\top u)^2$ has gradient $2\,\mathrm{ReLU}(w^\top u)w$, which is piecewise-linear in u and therefore Lipschitz with some constant depending on w. A finite sum of Lipschitz functions is Lipschitz, so $\nabla_u\Phi^\star$ is Lipschitz on $\mathbb{S}^{m-1}$ with some constant L_0. The projection $u \mapsto (I-uu^\top)$ is smooth with bounded derivative on the compact set $\mathbb{S}^{m-1}$, so composing it with $\nabla_u\Phi^\star$ preserves Lipschitz continuity: f is Lipschitz on $\mathbb{S}^{m-1}$ with some constant L. Crucially, L is a single global constant, valid over the entire sphere, not only in a neighborhood of one point.

From local to global existence. The Picard-Lindelöf theorem, applied at any point $u_0 \in \mathbb{S}^{m-1}$ using the global bound M and global Lipschitz constant L above, yields a unique solution on some time interval $[0, b]$ with $b > 0$ independent of u_0 (since M and L do not depend on u_0). Starting the theorem again from the endpoint $u(b)$ extends the solution to $[0, 2b]$, and repeating this indefinitely extends it to all $t \geq 0$; uniqueness on each sub-interval guarantees these extensions agree, so the full trajectory on $[0, \infty)$ is itself unique. (This patching argument works because the solution can never leave $\mathbb{S}^{m-1}$, so the same global constants M, L apply at every step.)

Finally, for $\rho > 0$, $-\rho^2\nabla_u^{\mathrm{Riem}}\Phi^\star(u)$ is f scaled by the constant ρ^2, hence itself continuous, bounded by $\rho^2 M$, and Lipschitz with constant $\rho^2 L$, so the same argument applies. □

Proposition 2 (Identical Angular Dynamics up to Speed)**.** *Let $u(\cdot)$ denote the trajectory of the unit-speed flow $\dot{u} = -\nabla_u^{\mathrm{Riem}} \Phi^\star(u)$ started at $u(0)$. If $u_i^1(0) = u(0)$, then $u_i^1(t) = u(\rho_i^2 t)$ for all $t \geq 0$; likewise if $u_j^2(0) = u(0)$ then $u_j^2(t) = u(\rho_j^2 t)$.*

Proof. Define $v(t) := u(\rho_i^2 t)$. By the chain rule, $\dot{v}(t) = \rho_i^2\, u'(\rho_i^2 t) = \rho_i^2 \big(-\nabla_u^{\mathrm{Riem}} \Phi^\star(u)\big)\big|_{u=u(\rho_i^2 t)} = -\rho_i^2 \nabla_{v(t)}^{\mathrm{Riem}} \Phi^\star(v(t))$, so v solves equation 21 with $v(0) = u(0)$. Since $u_i^1(\cdot)$ solves the same ordinary differential equation (ODE) with the same initial condition, and Lemma 5 guarantees a *unique* trajectory through any starting point, $u_i^1 \equiv v$, i.e., $u_i^1(t) = u(\rho_i^2 t)$. The argument for u_j^2 is identical with ρ_j^2 in place of ρ_i^2. □

In particular, u_i^1 and u_j^2 trace out exactly the same path on $\mathbb{S}^{m-1}$ as the reference trajectory $u(\cdot)$, only traversed at different (constant) speeds; since $\rho_i^2, \rho_j^2 > 0$, both $\rho_i^2 t \to \infty$ and $\rho_j^2 t \to \infty$ as $t \to \infty$, so $u_i^1(t)$ and $u_j^2(t)$ converge to the same limit points as $u(s)$ does as $s \to \infty$, whenever that limit exists.

B.4 Convergence to a common direction

A critical point $u^\star$ of a flow is *asymptotically stable* if it is Lyapunov stable and locally attracting, and its *basin of attraction* is the set of starting points whose trajectory converges to $u^\star$ (Khalil, 2002). By Proposition 2, $u_i^1(0)$ lies in the basin of attraction of $u^\star$ for the flow equation 21 if and only if $u(0)$ does for the reference flow, and likewise for $u_j^2(0)$. So u_i^1 and u_j^2 share the basins of attraction of $\Phi^\star$'s asymptotically stable critical points.

Corollary 1 (Convergence to a Common Direction)**.** *If $u_i^1(0)$ and $u_j^2(0)$ lie in the basin of attraction of the same asymptotically stable critical point $u^\star$ of $\Phi^\star$, then $\angle(W_i^1(t), W_j^2(t)) \to 0$ as $t \to \infty$, even though $\angle(W_i^1(0), W_j^2(0)) > 0$ in general.*

Proof. By Proposition 2, $u_i^1(t) \to u^\star$ and $u_j^2(t) \to u^\star$ as $t \to \infty$, since the underlying reference trajectory converges to $u^\star$ and the time rescalings $\rho_i^2 t, \rho_j^2 t \to \infty$. By the triangle inequality, $\angle(u_i^1(t), u_j^2(t)) \leq \angle(u_i^1(t), u^\star) + \angle(u^\star, u_j^2(t)) \to 0$. □

Once two idiosyncratic features have shrunk away on the two clients, the two clients' remaining losses are driven by the very same background landscape, and descending the same landscape from different starting points can bring those points together.

C Proofs for Section 5

C.1 Directional Stability

Lemma 3. ***(Directional Stability).*** *Let $u_i = W_i/\|W_i\|$ be a stationary point of the direction subproblem of equation 11, with $b = 0$, $\|W_i\| > 0$. Then*

$$|\sin(2\angle_i)| \;\leq\; \Gamma_i(W)/\lambda_{\mathcal{R}} \qquad (12) \qquad \textit{and} \qquad \angle_i \;\leq\; (\pi\,\Gamma_i(W))/(4\lambda_{\mathcal{R}}) \;=:\; \epsilon_i \qquad (13)$$

holds whenever $\angle_i \in [0, \pi/4]$.

Proof. Fix i and view $\mathcal{L}_{\mathcal{R}}$ (equation 11) as a function of $u_i = W_i/\|W_i\|$ alone, with $\|W_i\|$, $b = 0$, and every other W_j fixed. The feature-benefit term does not depend on u_i, so only the interference term $I_i(W) = \sum_{j \neq i} (\alpha_i + \alpha_j)\,\mathrm{ReLU}(W_j^\top W_i)^2$ and the leash term $\lambda_{\mathcal{R}} \sin^2 \angle_i$ exert any force on u_i.

By Lemma 2, $\|\nabla_{u_i}^{\mathrm{Riem}} I_i(W)\| \leq \Gamma_i(W)$. For the leash term, write $\hat{u}_i = \hat{W}_i/\|\hat{W}_i\|$ and $\kappa = u_i^\top \hat{u}_i = \cos\angle_i$, so $\sin^2\angle_i = 1 - \kappa^2$. Since $\nabla_{u_i}\kappa = \nabla_{u_i}(u_i^\top \hat{u}_i) = \hat{u}_i$, the ordinary gradient of the leash term is

$$\nabla_{u_i}\big[\lambda_{\mathcal{R}}(1-\kappa^2)\big] = -2\lambda_{\mathcal{R}}\kappa\,\nabla_{u_i}\kappa = -2\lambda_{\mathcal{R}}\kappa\,\hat{u}_i.$$

Projecting onto the tangent space $u_i^\perp$ via equation 19,

$$\nabla_{u_i}^{\mathrm{Riem}}\big[\lambda_{\mathcal{R}}(1-\kappa^2)\big] = (I - u_i u_i^\top)\big(-2\lambda_{\mathcal{R}}\kappa\,\hat{u}_i\big) = -2\lambda_{\mathcal{R}}\kappa\big(\hat{u}_i - (u_i^\top \hat{u}_i)u_i\big) = -2\lambda_{\mathcal{R}}\kappa\,(\hat{u}_i - \kappa\,u_i).$$

Since $\|u_i\| = \|\hat{u}_i\| = 1$ and $u_i^\top \hat{u}_i = \kappa$,

$$\|\hat{u}_i - \kappa\,u_i\|^2 = \|\hat{u}_i\|^2 - 2\kappa(u_i^\top \hat{u}_i) + \kappa^2\|u_i\|^2 = 1 - 2\kappa^2 + \kappa^2 = 1 - \kappa^2,$$

so $\|\hat{u}_i - \kappa\,u_i\| = \sqrt{1-\kappa^2} = \sin\angle_i$ (using $\sin\angle_i \geq 0$ for $\angle_i \in [0,\pi]$). Hence

$$\big\|\nabla_{u_i}^{\mathrm{Riem}}[\lambda_{\mathcal{R}}\sin^2\angle_i]\big\| = 2\lambda_{\mathcal{R}}|\kappa|\sin\angle_i. \tag{23}$$

Finally, on the range $\angle_i \in [0,\pi/4]$ assumed by the lemma, $\cos\angle_i \geq 0$ so $|\kappa| = \cos\angle_i$, and $2\cos\angle_i\sin\angle_i = \sin(2\angle_i) \geq 0$ (as $2\angle_i \in [0,\pi/2]$), so equation 23 becomes

$$\big\|\nabla_{u_i}^{\mathrm{Riem}}[\lambda_{\mathcal{R}}\sin^2\angle_i]\big\| = \lambda_{\mathcal{R}}\sin(2\angle_i) = \lambda_{\mathcal{R}}|\sin(2\angle_i)|. \tag{24}$$

Unlike the interference force, this does not depend on $\|W_i\|$: the leash stays equally strong even as a feature shrinks, exactly where the interference force disappears.

At a stationary point the two forces cancel exactly, $\nabla_{u_i}^{\mathrm{Riem}} I_i(W) = 2\lambda_{\mathcal{R}}\kappa\,(I - u_i u_i^\top)\hat{u}_i$, so by equation 24, $\|\nabla_{u_i}^{\mathrm{Riem}} I_i(W)\| = \lambda_{\mathcal{R}}|\sin(2\angle_i)|$ exactly. Combining with $\|\nabla_{u_i}^{\mathrm{Riem}} I_i(W)\| \leq \Gamma_i(W)$ gives equation 12. For equation 13, Jordan's inequality (Mitrinović, 1970) gives $\sin(x) \geq \frac{2}{\pi}x$ for $x \in [0,\pi/2]$; at $x = 2\angle_i$ this gives $\angle_i \leq \frac{\pi}{4}\sin(2\angle_i)$, which combined with equation 12 gives the result.

We note one complementary mechanism not covered by this bound. When a pre-activation $W_j^\top W_i + b_j$ crosses zero, the ReLU discretely removes that pair's interference term regardless of magnitude. The argument above covers the continuous, magnitude-driven route to interference, which is the one behind the shrinkage feedback loop of Section 4.1. □

C.2 Leashed Interference Bound

Lemma 6. ***(Leashed Interference Bound).*** *Let $\delta_{ij} := \angle(\hat{W}_i, \hat{W}_j) - \angle_i - \angle_j$ and assume $\delta_{ij} \geq 0$ (note $\delta_{ij} \leq \pi$ holds automatically). Then*

$$\alpha_j\,\mathrm{ReLU}(W_j^\top W_i + b_j)^2 \;\leq\; \alpha_j\,\mathrm{ReLU}\big(\|W_i\|\|W_j\|\cos\delta_{ij} + b_j\big)^2, \tag{25}$$

where the right-hand side is non-increasing in δ_{ij}. Since δ_{ij} grows as $\angle_i, \angle_j$ shrink, the bound is non-increasing *as $\angle_i, \angle_j$ shrink: tighter leashing gives a smaller interference bound. By Lemma 3, the hypothesis $\delta_{ij} \geq 0$ is guaranteed if $\angle_i, \angle_j \in [0, \pi/4]$ and*

$$\angle(\hat{W}_i, \hat{W}_j) \;\geq\; \epsilon_i + \epsilon_j = \frac{\pi}{4\lambda_{\mathcal{R}}}\big(\Gamma_i(W) + \Gamma_j(W)\big). \tag{26}$$

Proof. By the triangle inequality for angles on the sphere, $\angle_{ij} \geq \angle(\hat{W}_i, \hat{W}_j) - \angle_i - \angle_j = \delta_{ij}$. Since $\delta_{ij} \in [0,\pi]$ by hypothesis and cos is non-increasing on $[0,\pi]$,

$$W_j^\top W_i = \|W_i\|\|W_j\|\cos\angle_{ij} \leq \|W_i\|\|W_j\|\cos\delta_{ij}.$$

Adding b_j and applying the non-decreasing map $\mathrm{ReLU}(\cdot)^2$ preserves the inequality, giving equation 25. As $\angle_i, \angle_j$ shrink, δ_{ij} *increases*; since cos is non-increasing on $[0,\pi]$, $\cos\delta_{ij}$, and hence the bound, decreases. By Lemma 3, $\angle_i \leq \epsilon_i$ and $\angle_j \leq \epsilon_j$, so $\angle_i + \angle_j \leq \epsilon_i + \epsilon_j$. Hence $\epsilon_i + \epsilon_j \leq \angle(\hat{W}_i, \hat{W}_j)$ implies $\delta_{ij} \geq 0$, which is equation 26. □

C.3 Effect on the Equilibrium Magnitude

Lemma 7. ***(Leashed Partial Equilibrium of Magnitude).*** *Under the same restriction as Lemma 1, and with $\angle_i \leq \epsilon_i(\lambda_{\mathcal{R}}) = \pi\Gamma_i/(4\lambda_{\mathcal{R}})$ from Lemma 3 (treating Γ_i as a fixed constant, like the other magnitudes), define the guaranteed separation floor*

$$\delta_{ij}^{\mathrm{floor}}(\lambda_{\mathcal{R}}) := \Big[\angle(\hat{W}_i, \hat{W}_j) - \epsilon_i(\lambda_{\mathcal{R}}) - \epsilon_j(\lambda_{\mathcal{R}})\Big]_+. \tag{27}$$

Bounding interference as in equation 25 with δ_{ij} replaced by $\delta_{ij}^{\mathrm{floor}}$, the resulting magnitude bound

$$\|W_i\|^2_{\mathrm{bound}}(\lambda_{\mathcal{R}}) = \Big[1 - \tfrac{1}{2\alpha_i}\sum_{j\neq i}(\alpha_i+\alpha_j)\|W_j\|^2\cos^2_+\big(\delta_{ij}^{\mathrm{floor}}(\lambda_{\mathcal{R}})\big)\Big]_+ \tag{28}$$

bounds the magnitude equilibrium from Lemma 1 ($\|W_i\|^2$) from below ($\|W_i\|^2 \geq \|W_i\|^2_{\mathrm{bound}}$) and is non-decreasing in $\lambda_{\mathcal{R}}$. This is a statement about the worst-case guarantee, not about the realized δ_{ij} along a trajectory.

Proof. Substituting the bound equation 25 for the true interference term in equation 11, with $b = 0$ and all directions and $\|W_j\|$, $j \neq i$, fixed, gives an upper bound on the loss as a function of $\|W_i\|^2$ alone,

$$L_i^{\mathrm{bound}}(\|W_i\|) = \alpha_i\big(1 - \|W_i\|^2\big)^2 + \sum_{j\neq i}(\alpha_i+\alpha_j)\|W_i\|^2\|W_j\|^2\cos^2_+(\delta_{ij}).$$

This is identical in form to the loss analyzed in Appendix A.1, with $\cos^2\angle_{ij}$ replaced by $\cos^2_+(\delta_{ij})$; the same argument gives equation 28. The substitution is valid because $\angle_{ij} \geq \delta_{ij}^{\mathrm{floor}}$: by the triangle inequality $\angle_{ij} \geq \angle(\hat{W}_i, \hat{W}_j) - \angle_i - \angle_j \geq \angle(\hat{W}_i, \hat{W}_j) - \epsilon_i - \epsilon_j$, and $\angle_{ij} \geq 0$ trivially. Since $\cos_+$ is non-increasing on $[0, \pi]$, $\cos^2_+(\angle_{ij}) \leq \cos^2_+(\delta_{ij}^{\mathrm{floor}})$. For monotonicity, $\epsilon_i(\lambda_{\mathcal{R}})$ is decreasing in $\lambda_{\mathcal{R}}$, so $\delta_{ij}^{\mathrm{floor}}$ is non-decreasing in $\lambda_{\mathcal{R}}$. Then $\cos^2_+(\delta_{ij}^{\mathrm{floor}})$ is non-increasing, the subtracted sum is non-increasing, and $\|W_i\|^2_{\mathrm{bound}}$ is non-decreasing in $\lambda_{\mathcal{R}}$.

Every inequality behind this bound is tight under an explicit condition: equation 8 at $\angle_{ij} = \pi/4$, the triangle inequality when interfering gradients align, and equation 25 when $\angle_{ij} = \delta_{ij}$ exactly. Like the unleashed case, this bound treats one feature at a time, while in the full objective all directions move jointly.

Why treating magnitude and direction separately loses no stationary points. Let f be any C^1 function of $W_i \in \mathbb{R}^m$ with all other W_j held fixed (the losses above are C^1, since $\mathrm{ReLU}(\cdot)^2$ is). Let $\|W_i\| > 0$, $u_i = W_i/\|W_i\|$ and $P_i := I - u_i u_i^\top$ is the projector onto $u_i^\perp$. Since $u_i^\top u_i = 1$, every vector splits orthogonally into its component along u_i and its component in $u_i^\perp$; applied to $\nabla_{W_i} f$,

$$\nabla_{W_i} f = \underbrace{u_i u_i^\top \nabla_{W_i} f}_{\text{radial}} + \underbrace{P_i \nabla_{W_i} f}_{\text{tangential}}. \tag{29}$$

We identify the two parts with the derivatives used in the magnitude and direction subproblems.

Radial part. Fix u_i and vary the magnitude. The chain rule gives

$$\frac{\partial f}{\partial \|W_i\|} = \frac{\mathrm{d}}{\mathrm{d}s} f(\|W_i\|\, u_i) = u_i^\top \nabla_{W_i} f, \tag{30}$$

so the radial term in equation 29 equals $\frac{\partial f}{\partial \|W_i\|}\, u_i$. This is the derivative set to zero in Lemma 1; because $\frac{\partial f}{\partial \|W_i\|} = 2\|W_i\| \frac{\partial f}{\partial \|W_i\|^2}$ and $\|W_i\| > 0$, the two vanish together.

Tangential part. Fix the magnitude and vary the direction: define $g(u) := f(\|W_i\|\, u)$ for $u \in \mathbb{S}^{m-1}$. The chain rule gives $\nabla_u g(u) = \|W_i\|\, \nabla_{W_i} f(\|W_i\| u)$, so by equation 19, evaluated at $u = u_i$,

$$\nabla^{\mathrm{Riem}}_{u_i} f = P_i\, \nabla_u g = \|W_i\|\, P_i\, \nabla_{W_i} f \quad \Longleftrightarrow \quad P_i \nabla_{W_i} f = \frac{1}{\|W_i\|} \nabla^{\mathrm{Riem}}_{u_i} f. \tag{31}$$

The factor $\|W_i\|$ appears because rotating u_i by a small angle ϑ moves W_i by a distance $\|W_i\|\vartheta$. This is the quantity set to zero in Lemma 3.

Combining. Inserting equation 30 and equation 31 into equation 29 yields

$$\nabla_{W_i} f = \frac{\partial f}{\partial \|W_i\|}\, u_i + \frac{1}{\|W_i\|}\, \nabla^{\mathrm{Riem}}_{u_i} f.$$

The first term lies in $\mathrm{span}(u_i)$ and the second lies in $u_i^\perp$, so

$$\|\nabla_{W_i} f\|^2 = \Big(\frac{\partial f}{\partial \|W_i\|}\Big)^2 + \frac{1}{\|W_i\|^2}\, \big\|\nabla^{\mathrm{Riem}}_{u_i} f\big\|^2.$$

Both summands are non-negative, hence

$$\nabla_{W_i} f = \mathbf{0} \iff \frac{\partial f}{\partial \|W_i\|} = 0 \text{ and } \nabla^{\mathrm{Riem}}_{u_i} f = \mathbf{0}.$$

Any stationary point of the full objective in W_i (with $\|W_i\| > 0$) is therefore stationary for the magnitude subproblem (Lemma 1) and for the direction subproblem (Lemma 3), so analyzing the two separately does not discard any stationary points. The argument concerns first-order stationarity only and does not cover $\|W_i\| = 0$, where the direction is undefined. □

C.4 Reconstruction Leashing

Assumptions. With $x = e_i$ one-sparse and $h = W_i$, write $\hat{u}_j := \hat{W}_j/\|\hat{W}_j\|$ for the leash directions, and assume: (A1) the leash directions $\{\hat{u}_j\}$ are separated enough that each has a larger inner product with itself than with any other leash; (A2) during local training W_i stays close enough to $\hat{W}_i$ that $\hat{u}_i$ remains its best cosine match among all leash directions; (A3) the drift $\delta := W_i - \hat{W}_i$ satisfies $\|\delta\| < \|\hat{W}_i\|$, i.e. a feature has not drifted farther than its own original length.

The active feature recovers its own index. Define $q^*(i) := \arg\max_j \hat{u}_j^\top W_i$, i.e. the leash direction with largest cosine similarity to W_i, so that q^* is invariant to the scale of W_i.

Lemma 8 (Self-Recovery)**.** *Under (A1)-(A2), $q^*(i) = i$ for every active feature i.*

Proof. Let $\delta := W_i - \hat{W}_i$ and $m_{ij} := \|\hat{W}_i\| - \hat{u}_j^\top \hat{W}_i$ for $j \neq i$. By Cauchy-Schwarz, $m_{ij} \geq 0$, and (A1) makes it strictly positive. Since $\hat{u}_i^\top \hat{W}_i = \|\hat{W}_i\|$,

$$\hat{u}_i^\top W_i - \hat{u}_j^\top W_i = m_{ij} + (\hat{u}_i - \hat{u}_j)^\top \delta \;\geq\; m_{ij} - 2\|\delta\|,$$

where the inequality uses Cauchy-Schwarz and $\|\hat{u}_i\| = \|\hat{u}_j\| = 1$. By (A2), $\|\delta\| < \frac{1}{2}\min_{j\neq i} m_{ij}$, so the right-hand side is positive for every $j \neq i$. Hence $\hat{u}_i^\top W_i > \hat{u}_j^\top W_i$ for all $j \neq i$, i.e. $q^*(i) = i$. □

Rewriting the leash loss without the feature label. Lemma 8 lets us replace the (unknown) true index i with the computable index q^*,

$$\mathcal{R}(W, \hat{W}) = \sum_i \sin^2 \angle(W_i, \hat{W}_i) = \sum_i \sin^2 \angle\big(W_i, \hat{W}_{q^*(i)}\big), \tag{32}$$

losing no information, since $q^*(i) = i$. By the standard identity relating a vector's angle to its projection residual,

$$\sin^2 \angle(W_i, \hat{W}_{q^*}) = \frac{\big\|W_i - \hat{W}_{q^*}\hat{W}_{q^*}^\top W_i/\|\hat{W}_{q^*}\|^2\big\|_2^2}{\|W_i\|^2}, \tag{33}$$

so equation 32 is a normalized reconstruction error: W_i compared to its projection onto the single, activation-dependent leash $\hat{W}_{q^*}$. This is exactly the form of an SAE reconstruction loss with dictionary size n and sparsity 1, provided the encoder recovers q^* and the decoder reproduces this projection.

An SAE that realizes this exactly. Take a TopK_1 SAE with $e = d = \mathbf{0}$, encoder rows $E_{[i,:]} = \hat{u}_i^\top$, and decoder columns $D_{[:,i]} = \hat{u}_i$, both unit-norm, matching the architecture of equation 4 including its ReLU.

Proposition 3 (Exact Reconstruction Form)**.** *Under (A1)-(A3), with $h = W_i$ and this SAE,*

$$\big\|h - \big(D\,\mathrm{TopK}_1(\mathrm{ReLU}(E(h-d)+e)) + d\big)\big\|_2^2 = \big\|W_i - \hat{u}_{q^*}\hat{u}_{q^*}^\top W_i\big\|_2^2. \tag{34}$$

Proof. With $e = d = \mathbf{0}$, the j-th encoder pre-activation is $(Eh)_j = \hat{u}_j^\top W_i$. By Lemma 8, $q^* = i$ is the unique maximizer of these pre-activations, and $\hat{u}_i^\top W_i = \|\hat{W}_i\| + \hat{u}_i^\top \delta \geq \|\hat{W}_i\| - \|\delta\| > 0$ by (A3). Since ReLU is non-decreasing and leaves a positive input unchanged, it fixes the winning

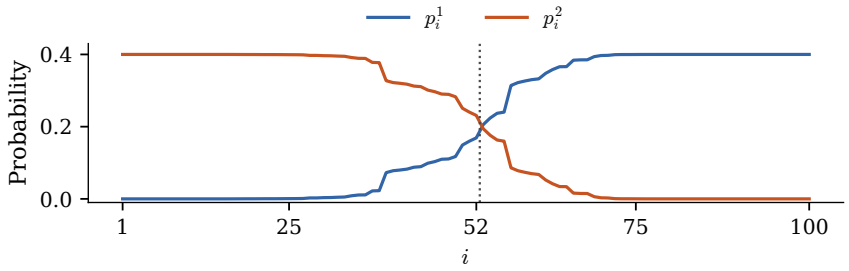


Figure 5: Synthetic feature activation probabilities, ordered by $p_i^1 - p_i^2$. The dotted line separates the 52 features preferred by client 2 from the 48 preferred by client 1. The two client probabilities average to 0.2 for every feature.

Table 1: Office-Caltech10 hyperparameters (LR = Learning Rate).

Baseline	Task LR	Minimum LR	SAE LR	λ_{rec}
FedAvg	10^{-3}	10^{-5}	0.003	0.3
FedProx	10^{-3}	10^{-5}	0.003	1
MOON	0.0003	3×10^{-6}	0.0003	0.03
Ditto	10^{-3}	10^{-5}	0.003	0.3

pre-activation at $\hat{u}_i^\top W_i$ while leaving every other coordinate no larger than before, so the arg max is unchanged. Thus TopK_1 still selects coordinate $q^* = i$ after the ReLU, with the same value $z_{q^*} = \hat{u}_{q^*}^\top W_i$. Decoding, $\hat{h} = D_{[:,q^*]} z_{q^*} = \hat{u}_{q^*} \hat{u}_{q^*}^\top W_i$, exactly the projection subtracted in equation 34. Since $\hat{u}_{q^*} \hat{u}_{q^*}^\top = \hat{W}_{q^*} \hat{W}_{q^*}^\top / \|\hat{W}_{q^*}\|^2$, this is the same orthogonal projection used in equation 33. □

Dividing equation 34 by $\|W_i\|^2$, summing over i, and comparing with equation 32-equation 33 gives the following Proposition.

Proposition 1. ***(Reconstruction leashing Equals Directional Leashing).*** *Under these conditions there is a* TopK_1 *SAE* (E, D, e, d)*, built only from the leashes* $\{\hat{W}_j\}_{j=1}^n$*, such that*

$$\mathcal{R}(W, \hat{W}) = \sum\nolimits_i \left\| W_i - \left(D \,\text{TopK}_1(\text{ReLU}(E(W_i - d) + e)) + d \right) \right\|_2^2 / \|W_i\|^2.$$

D Additional Experimental Details

D.1 Synthetic Setting

Since $p_i^1 + p_i^2 = 0.4$, every feature has an average activation probability

$$\frac{p_i^1 + p_i^2}{2} = 0.2$$

when the clients are weighted equally. Figure 5 shows the sampled feature activation probabilities, which remain fixed across runs. Of the 100 features, 52 are more frequent on client 2 and 48 on client 1. Many probabilities are close to either zero or 0.4, producing distinct client preferences. Features are sampled independently within each client, so several features can be active in one input.

D.2 Office Dataset: Experimental Setup

Data and optimization. For each Office-Caltech10 domain, we use the first 62 entries of the released training list for training and the last 50 for validation, with no overlap. These subsets remain fixed across runs. Table 1 summarizes the method-specific learning rates and reconstruction coefficients.

Table 2: Office-Caltech10 validation MSE over ten repetitions. Values show means with 95% confidence intervals. Reductions compare mean losses.

Baseline	Plain	+ MechLeash-FL	Reduction
FedAvg	0.0284 ± 0.0007	0.0201 ± 0.0014	29.08%
FedProx	0.0306 ± 0.0009	0.0205 ± 0.0005	33.08%
MOON	0.0200 ± 0.0017	0.0169 ± 0.0005	15.29%
Ditto	0.0282 ± 0.0006	0.0195 ± 0.0014	30.78%

Table 3: Office-Caltech10 merge barriers at the last round. Values show means with 95% confidence intervals over ten repetitions. Ditto reports its global-model barrier.

Baseline	Plain	+ MechLeash-FL
FedAvg	-0.00186 ± 0.00011	-0.00249 ± 0.00031
FedProx	-0.00391 ± 0.00064	-0.00265 ± 0.00026
MOON	0.00195 ± 0.00179	-0.00094 ± 0.00007
Ditto (global)	-0.00186 ± 0.00011	-0.00249 ± 0.00031

Task learning rates remain constant for rounds 1-4, then halve each round until reaching the listed minimum. SAE learning rates remain constant. Task optimization uses Adam without weight decay. Each round contains one task-model aggregation and, with MechLeash-FL, one SAE aggregation after 20 local SAE epochs. The first task round has no reconstruction penalty; its SAE phase provides the shared reference for round two.

Statistical analysis. For each baseline, we compare it against its MechLeash-FL extension over ten paired repetitions. We apply an exact one-sided paired Wilcoxon signed-rank test (Wilcoxon, 1945) to the validation losses, with the alternative hypothesis that MechLeash-FL achieves lower loss. The test assumes that the paired differences are independent across repetitions and that their distribution is symmetric. Because MechLeash-FL improves on its baseline in all ten repetitions, the exact one-sided p-value is $1/1024$ for each of the four baseline comparisons.

To control the family-wise error rate at $\alpha = 0.05$ across the four comparisons, we apply the Holm step-down correction (Holm, 1979): the p-values are sorted in ascending order, and the j-th smallest, $p_{(j)}$, is compared against $0.05/(m - j + 1)$ with $m = 4$, stopping at the first failure to reject. Since all four raw p-values coincide at $1/1024$, the Holm-adjusted value is the same for every comparison, $4 \times (1/1024) \approx 0.0039$, and all four null hypotheses are rejected at $\alpha = 0.05$.

D.3 Main Office Results

Table 2 reports final reconstruction losses. FedAvg, FedProx, and MOON use their global models on the pooled validation set. Ditto uses each client's personalized model on its own validation images, averaging the four client losses.

Merge barriers. We extend Equation equation 16 to four equally weighted clients:

$$B^r_{\text{merge}} = \mathcal{L}_{\text{pool}}(\theta^{r+1}) - \frac{1}{4}\sum_{c=1}^{4} \mathcal{L}_{\text{pool}}(\theta^{r,c}), \qquad \mathcal{L}_{\text{pool}} = \frac{1}{4}\sum_{c=1}^{4} \mathcal{L}^c.$$

Every local model and the merged model are evaluated on the same 200 pooled validation images. A negative barrier means that aggregation lowers loss relative to the mean local-model loss. Ditto's global branch matches FedAvg in our setup and therefore has identical barriers; its personalized models are not aggregated.

MechLeash-FL lowers the merge barrier for FedAvg and MOON (Table 3). For MOON, the mean barrier changes from positive to negative, with a lower barrier in all ten paired repetitions. FedProx's

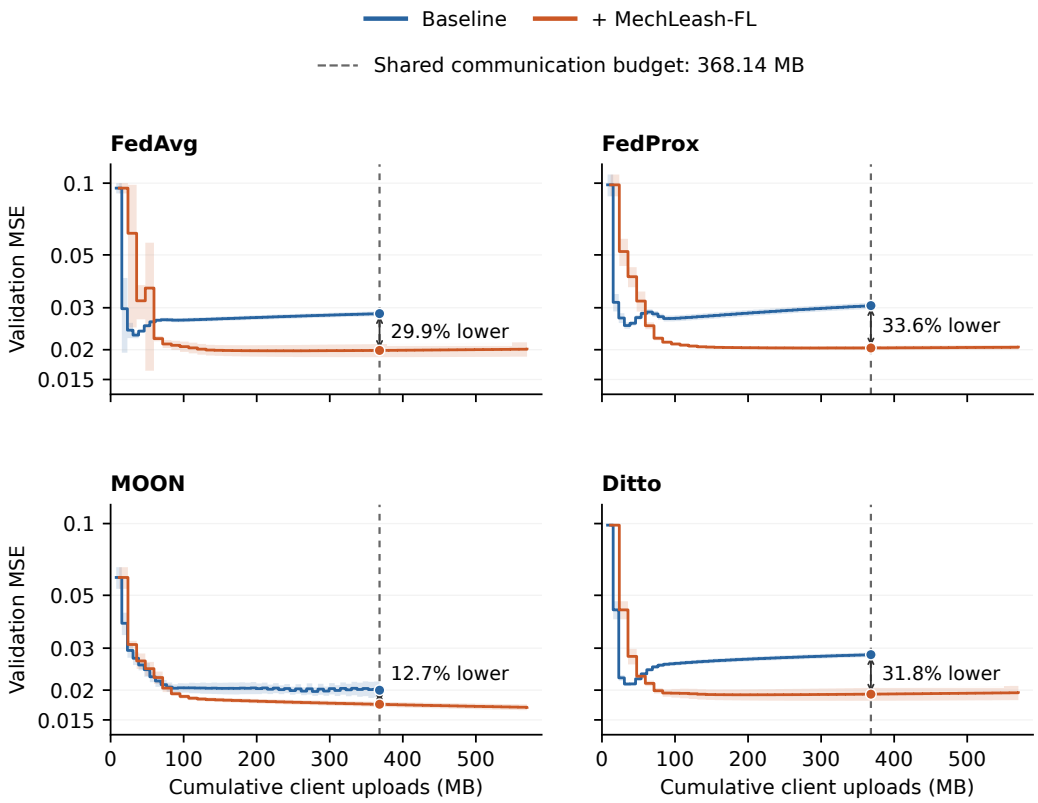


Figure 6: Office-Caltech10 validation MSE versus cumulative client uploads, including task-model and SAE parameters. Curves show means and 95% confidence intervals over ten repetitions.

barrier becomes less negative despite its lower final reconstruction loss, showing that improved final loss does not uniformly correspond to a lower merge barrier.

Communication. MechLeash-FL uploads both task-model and SAE parameters, whereas the baselines upload only task-model parameters. To account for this additional communication, we compare 48 baseline rounds with 30 MechLeash-FL rounds within the same upload budget of 368.14 MB (Figure 6). We count 32-bit client parameter uploads summed across all four clients.

D.4 Hyperparameter Sweep

For the sparsity sweep (Table 4), we retain each method's reconstruction coefficient from Table 1. For the coefficient sweep (Table 5), we fix $k = 8$. All other training hyperparameters remain unchanged. We report validation MSE over three paired repetitions, using personalized models for Ditto and global models for the other methods.

Mean reconstruction loss changes little across $k = 2$-16, while $k = 32$ gives worse results for MOON and Ditto. The choice of λ_{rec} has a larger effect. Values close to the configurations used in Section 6 give the lowest mean losses. Increasing or decreasing λ_{rec} too much degrades reconstruction performance. We hypothesize that this is because too small λ_{rec} values behave like FedAvg, while too big λ_{rec} values are too restrictive.

D.5 Masked Experiments

Each image is divided into 256 16×16 non-overlapping patches. We independently mask each 16×16 input patch with probability 0.5 during training and evaluation, keeping the reconstruction targets clean. Training masks are resampled, while validation masks remain fixed across rounds and

Table 4: Effect of SAE sparsity k on final Office-Caltech10 validation MSE. Values show means with 95% confidence intervals over three repetitions, multiplied by 1000. Each method uses its default reconstruction coefficient.

k	FedAvg	FedProx	MOON	Ditto
2	17.19 ± 1.65	20.29 ± 0.67	18.10 ± 1.01	17.82 ± 4.26
4	17.22 ± 1.23	20.32 ± 2.27	18.08 ± 0.12	18.35 ± 2.94
8	17.94 ± 2.41	20.25 ± 1.53	17.32 ± 1.62	17.79 ± 1.56
16	17.90 ± 4.37	21.33 ± 3.01	17.87 ± 6.94	17.93 ± 2.92
32	17.90 ± 4.51	20.98 ± 2.87	22.82 ± 6.51	22.67 ± 21.92

Table 5: Effect of λ_{rec} on final Office-Caltech10 validation MSE, with $k = 8$. Values show means with 95% confidence intervals over three repetitions, multiplied by 1000.

λ_{rec}	FedAvg	FedProx	MOON	Ditto
0.01	23.87 ± 3.02	28.53 ± 7.73	21.20 ± 4.67	23.94 ± 4.79
0.1	17.47 ± 3.08	22.63 ± 2.80	19.18 ± 0.39	18.00 ± 2.21
1	19.43 ± 1.87	20.25 ± 1.53	22.26 ± 0.81	19.05 ± 2.10
10	25.13 ± 9.78	22.76 ± 2.34	31.78 ± 13.62	27.46 ± 5.49

paired methods. We retain the uncorrupted experiment's hyperparameters and 48-round training schedule without retuning.

Table 6 reports results over three paired repetitions per method. MechLeash-FL lowers mean MSE by 9.07%-20.87%, with lower loss in all twelve paired comparisons. These results suggest that MechLeash-FL's benefits can extend to tasks that demand more than reconstructing an observed input.

Table 6: Office-Caltech10 reconstruction from masked inputs. Values show final validation MSE with 95% confidence intervals over three repetitions. Ditto uses personalized models; the other methods use global models. Reductions compare mean losses.

Baseline	Plain	+ MechLeash-FL	Reduction
FedAvg	0.0239 ± 0.0027	0.0192 ± 0.0021	19.83%
FedProx	0.0241 ± 0.0058	0.0219 ± 0.0011	9.07%
MOON	0.0266 ± 0.0029	0.0210 ± 0.0007	20.87%
Ditto	0.0236 ± 0.0038	0.0189 ± 0.0008	19.78%