OUTLIER CAPPING FOR ONE-SHOT FEDERATED LORA: DOMINATED, NOT MISALIGNED

Anonymous authors
Paper under double-blind review

ABSTRACT

One-shot federated merging combines LoRA adapters that clients trained independently into one adapter, in a single round and without data, gradients, or a validation set. In this setting recent merge operators collapse, and on eight financial adapters for Llama-3.1-8B and 26 hub adapters for Mistral-7B several of them score exactly zero. Realigning client subspaces is a natural remedy, but we show that misalignment is not the cause. The cause is *client dominance*. Update norms differ by $708\times$ on average, a few clients carry nearly all the energy of the summed update, and scaling a single client reproduces the collapse. A controlled training study further shows that which LoRA factor looks shared across clients depends on the initialization protocol, so a fix should act on the product BA. We propose **Outlier Capping** (OCap), a data-free server-side step for one-shot federated LoRA that shrinks only the clients whose update norm is an outlier of the pool. It bounds dominance but does not remove it. The alternative bounds (clipping at the median as in robust federated learning, shrinking the whole update, and equalizing every client) fail in the ways the diagnosis predicts. All operators we test except task arithmetic and DC-Merge (which flattens itself) also need each client's spectrum to be flattened first (PolCap). With task arithmetic, OCap has the highest retention among the rules that regress on neither pool, and the best worst-case regression against the unmerged base among the 29 rules we measure on every axis at the largest draw.

1 INTRODUCTION

LoRA (Hu et al., 2022) has made adapters cheap to train and to share. Public hubs now host large collections of adapters for a single base model, trained by unrelated parties on data that cannot be pooled; one study alone fine-tunes 310 of them (Zhao et al., 2024). Combining such adapters is the one-shot limit of federated learning (Tao et al., 2025; Wang et al., 2024). The server receives the client adapters $\{(A_i, B_i)\}_{i=1}^{n}$ once, holds no client data, computes no gradients, and has no validation set. We call this *one-shot federated merging*. It is harder than iterative federated LoRA, where clients restart every round from one broadcast (McMahan et al., 2017; Zhang et al., 2024), because independently trained clients share only the base model and not the recipe, the data volume, or the initialization. On FINLORA (Wang et al., 2025a), eight task adapters for Llama-3.1-8B-Instruct, four recent spectral operators (Iso-C, Iso-CTS (Marczak et al., 2025), TSV-M (Gargiulo et al., 2025), and DC-Merge (Zhang et al., 2026)) score exactly zero on all twelve client tasks, and Iso-CTS returns an empty string on 17,314 of 17,317 prompts (App. K). This is not an implementation problem, since the same DC-Merge code is the best operator on a control pool whose clients have similar magnitudes (App. I). We ask:

Why do merge operators collapse on independently trained adapters?

Spectral merging assumes that clients interfere because their singular directions overlap badly, and it realigns them. We find a different cause. On FINLORA, client update norms span $708\times$, and three of the eight clients carry 99.5% of the energy of the summed update. The operators either build their shared basis from this sum or rebuild the update from each client's own singular values, so the merged adapter mostly reflects these three clients (§4). We call this phenomenon *client dominance*. Dominance causes the collapse and is not merely correlated with it, because scaling a single client

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.

reproduces the collapse on two pools in two modalities. In addition, a 26-client hub on Mistral-7B breaks every published operator we ran (§7.3). To our knowledge, no published operator bounds one client's magnitude relative to the others. Normalizing every client (Zheng et al., 2025) also rescales the clients that are not outliers, and norm clipping ported from federated learning (Tao et al., 2025) acts on raw updates at a preset threshold.

A fix also has to act on the right LoRA factor, and the literature disagrees about which factor is shared across clients. One line of work averages, freezes, or carries over A (Sun et al., 2024; Guo et al., 2025; Qiao & Mahdavi, 2026), and another calibrates B (Tang & Yang, 2026). We resolve this with a controlled study that varies only the initialization (§5). A looks shared exactly when clients start from a shared A_0. For independently trained clients the A_i are unrelated frames, so the merge has to act on the product $\Delta W_i = B_i A_i$.

We propose **Outlier Capping** (OCap), a server-side step for one-shot federated LoRA. Before any operator is applied, OCap shrinks only the clients whose update norm exceeds the pool median by more than three median absolute deviations, and it leaves the other clients, all directions, and all spectral shapes unchanged (§6). OCap needs no data or tuning and is cheap. Alternative bounds, such as clipping at the typical client as in robust federated learning (Nguyen et al., 2022), fail in the way the diagnosis predicts. All operators we test except task arithmetic and DC-Merge (which flattens itself) also need each client flattened first (PolCap), more broadly than we pre-registered.

Our contributions are: (i) we identify client dominance as the failure mode of one-shot federated merging, establish it causally with a dominance sweep, and observe it on two public pools (§4, §7.3); (ii) we explain the A/B-sharing disagreement by showing, with a controlled study and four public pools, that cross-client A-similarity is a property of the initialization protocol (§5); (iii) we propose OCap, test one control per design choice, and report a pre-registered test of the polar step including its failures (§7.2); and (iv) we evaluate 114 rules for no regression and find that no rule wins on every axis, while among the rules that regress on neither pool, OCap with task arithmetic has the highest retention (§7.4, §7.3).

2 Related Work

Merge operators. Training-free operators differ in where they combine the clients. *Entry-wise* operators work coordinate by coordinate. Task arithmetic (Ilharco et al., 2023) averages the updates. TIES (Yadav et al., 2023) and DARE (Yu et al., 2024) first trim, elect, or rescale entries, and KnOTS (Stoica et al., 2025) applies such a rule after aligning LoRA updates in a joint SVD basis. *Spectral* operators work in a singular basis. The *isotropic* ones, Iso-C and Iso-CTS (Marczak et al., 2025), flatten the spectrum of the summed update, Iso-CTS adding task-specific directions; the *cover-space* ones, TSV-M (Gargiulo et al., 2025) and DC-Merge (Zhang et al., 2026), decorrelate the clients' pooled singular vectors, DC-Merge after flattening each client's spectrum. All of them were developed on pools fine-tuned by one group with one recipe, where client magnitudes are comparable. OCap is applied before any of these operators. RegMean (Jin et al., 2023) and LoraHub (Huang et al., 2024) need activations or validation data and fall outside our setting.

Closest related work. Two lines of work control client magnitude, but they differ from OCap in which clients they modify. Tao et al. (2025) port federated remedies to task arithmetic, among them clipping any task vector above a preset norm and renormalizing every one. DO-Merging (Zheng et al., 2025) separates magnitude from direction per column: it normalizes every column of every client, rescales the summed directions by the summed magnitudes, and orthogonalizes by gradient descent. Because it normalizes every client instead of capping outliers, it is close to the "equalize every client" panel of Figure 3. In our re-implementation (no code is released and λ is unset), it collapses on the hub pool and loses answer format and transfer on FinLoRA (§7.3, Table 20). Clipping updates at a fixed, quantile-, or median-derived norm is standard in robust and private federated learning to bound one party's influence (Sun et al., 2019; Andrew et al., 2021; Nguyen et al., 2022). OCap uses the same mechanism but places the bound at the outlier boundary instead of at the typical client, per weight matrix and without data. Clipping at the median, as in FLAME, loses the tasks of the heavy clients (§7.2).

Asymmetry of LoRA A and B. FFA-LoRA (Sun et al., 2024) freezes A. FedSA-LoRA (Guo et al., 2025) aggregates only A, and SLAO (Qiao & Mahdavi, 2026) carries over A and merges B, on evidence that A is more similar across tasks (see also Zhu et al., 2024). Conversely, Pico (Tang & Yang, 2026) finds the B factors reusing a few shared directions, and FedLEASE (Wang et al., 2025b) finds B similar within a task and A unrelated to task across separately trained clients. In §5 we isolate the variable behind the two sets of results, and App. A assigns each work to a regime.

One-shot federated learning. Task arithmetic is one round of federated averaging (Tao et al., 2025), one-round federated LoRA has been studied directly (Wang et al., 2024), and FedEx-LoRA (Singhal et al., 2025) corrects the inexactness of averaging factors within a round. RobustMerge (Zeng et al., 2025) releases MM-MergeBench, which we use as the control pool without dominance.

3 Setting and Evaluation Protocol

Problem. A server receives n client adapters $\{(A_i, B_i)\}_{i=1}^{n}$, trained independently on a shared base W_0, with $A_i \in \mathbb{R}^{r\times d_{\text{in}}}$, $B_i \in \mathbb{R}^{d_{\text{out}}\times r}$ and update $\Delta W_i = B_i A_i$ (the LoRA scale α/r folded into B_i). It must produce one adapter in a single round, with no client data, no gradients, and no validation set. Every operation is per adapted weight matrix. A client's *magnitude* is $\|\Delta W_i\|_F$. Summarizing each client's magnitude over weight matrices by its median (or mean), a pool's *median spread* (*mean spread*) is the ratio of the largest summary to the smallest.

Rules. A merge *rule* is a *client step*, applied to each adapter separately, followed by a pool *operator* (§2) at rank budget nr. We consider four kinds of client steps: (i) none, i.e., the operator alone (constants in App. C); (ii) the *scalar control*, which we add so that published operators can function on a dominated pool and which multiplies B_i by $\min\{1, (m_F/\|\Delta W_i\|_F)^\gamma\}$, with m_F the pool median of $\|\Delta W_j\|_F$ and γ=.5 unless stated, so heavy clients shrink and light ones are never raised (*full equalization* is γ=1 without the minimum, and a *share cap* limits any client's share of the summed norm); (iii) Outlier Capping (OCap, ours), its flattened variant PolCap, and the instrument PolEq, defined in §6; and (iv) two further bounds for the design space: *clipping at the median* of $\|\Delta W_j\|_F$, the bound of robust federated learning (Nguyen et al., 2022), and *shrinking the whole update* to OCap's norm per weight matrix. *Baselines* are the published operators with no client step or with the scalar control. App. C lists every rule with its identifier in the supplementary code.

Pools. The main pool is the eight public FinLoRA task adapters (Wang et al., 2025a) for Llama-3.1-8B-Instruct (Grattafiori et al., 2024) at r=8: sentiment, headline, ner, finer, xbrl_extract, xbrl_term, formula, and financebench. The second dominated pool is the 26 distinct LoRA Land adapters for Mistral-7B (§7.3). The control pool, without dominance, is MM-MergeBench (Zeng et al., 2025): eight LLaVA-1.5-7B adapters (Liu et al., 2024). Apart from the fifteen adapters of the controlled study (§5), we train no adapters.

Three axes. *Retention* is accuracy on the twelve label-scored client tasks (six classification, six XBRL extraction), n=2000 items per task where the test set allows. *Transfer* is accuracy on three held-out QA suites no client saw, FinQA (Chen et al., 2021), ConvFinQA (Chen et al., 2022), and FinExam-10K (Lin et al., 2026), weighted equally; ECTSum (Mukherjee et al., 2022) summarization is scored with ROUGE (Lin, 2004) and reported separately. *Capability* is MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and HellaSwag (Zellers et al., 2019) under the EleutherAI harness (Gao et al., 2024). Every language model runs on the same 8-bit base (Dettmers et al., 2022); the multimodal pool runs in fp16.

Reference points. Two *floors* are models a server could deploy without merging: the unmerged base and the *best single adapter* (the client adapter best on transfer, xbrl_term). Two *ceilings* need information that a merge does not have. The *home expert* scores each task with the client trained on it, and *oracle routing* with the best client for that task.

No-regression score. Since there is no validation data, we rank rules by how much they risk. A rule's *worst* z is its smallest paired-bootstrap z against the unmerged base over four measures: retention, transfer, and GSM8K under the strict filter (the answer in the demonstrated format) and

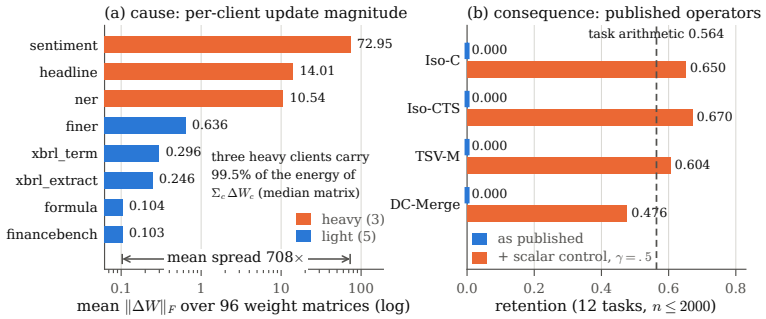


Figure 1: The cause and its consequence. **(a)** Per-client $\|\Delta W_i\|_F$ has a mean spread of $708\times$; three clients carry 99.5% of the energy of the sum. **(b)** Four spectral operators score exactly zero at n=2000; under the scalar control (γ=.5) the same four reach .476–.670.

the flexible one (the last number). A strict score below the flexible one is a loss of *answer format*. MMLU and HellaSwag separate almost no rules and are reported separately.

Statistics. Intervals are task-balanced paired bootstraps (Efron & Tibshirani, 1993) (2000 resamples; App. D). Greedy 8-bit generation is not bit-reproducible, and re-running identical items moves a task score by up to 1.5 points. We therefore do not interpret differences below 4 points at n=100. Headline numbers are at n=2000, and we say so when a claim rests on a smaller n. The smaller item draws are subsets of the larger ones, and every GSM8K comparison uses the same 500 items, so a larger draw refines a smaller one and does not independently confirm it. Models use the benchmark's prompts (Wang et al., 2025a), and floors also use the base's chat template (App. L).

4 Diagnosis I: Client Dominance Is the Failure Mode

Measurement. Figure 1 summarizes the diagnosis. The median spread is $128\times$, so dominance is not limited to a few weight matrices. It is also not limited to this pool, since the LoRA Land hub has a median spread of $52\times$ (§7.3). The three heavy clients are the three trained at α=32 on their full data. The spread comes from the recipe and not from the task, since a sentiment client we trained at α=16 is at least as accurate as the public one at a twentieth of its median norm (App. E).

Intuition. An operator sees the clients only through their sum or through their own spectra, and under dominance both are determined mostly by the heavy clients. The three elementary statements below make this precise, and we check each of them by measurement (proofs in App. B). Write $S = \sum_i \Delta W_i$ and, for a subset H of clients, $S_H = \sum_{i \in H} \Delta W_i$.

Proposition 1 (The sum is its heavy subset)**.** *For any* $H \subseteq [n]$, $\|S - S_H\|_F \leq \sum_{i \notin H} \|\Delta W_i\|_F$.

With H the three heavy clients, the pool sum is within 7% of S_H at the median weight matrix. The largest client alone holds .95 of the per-client energy, but the sum is 22% away from it. Therefore, dominance here comes from a heavy subset and not from a single outlier.

Proposition 2 (The common subspace follows the heavy subset)**.** *Let* $U_k(\cdot)$ *be a top-*k *left singular subspace,* $E = S - S_H$*, and* $\delta_k = \sigma_k(S_H) - \sigma_{k+1}(S)$*. If* $\delta_k > 0$ *then* $\|\sin\Theta(U_k(S), U_k(S_H))\|_2 \leq \|E\|_2/\delta_k$.

Iso-C and Iso-CTS take their common subspace from $U_k(S)$. TSV-M and DC-Merge orthogonalize the clients' pooled singular vectors, a step that ignores magnitude, and then rebuild the update from each client's own singular values, so the between-client ratio passes into the merge. Since the gap condition does not hold at the operators' own budget, we measure the angle directly. The mean principal-angle cosine between the top-24 subspaces of S and S_H ($k = \operatorname{rank} S_H$) has median .848, compared with chance levels of .065–.130.

Proposition 3 (Flattening is unsafe in the near-degenerate band)**.** *Let the merged update* M *have rank budget* N*, with* q *directions above* c *and the rest below* $\delta < c$*. Replacing its* N *singular values*

by their mean gives $\cos\angle(\Phi(M), M) = \|M\|_*/(\sqrt{N}\|M\|_F)$*, at most* $\sqrt{q/N}$ *when* $\delta = 0$*, i.e., a share* $1 - q/N$ *of the output energy lies in directions that had no energy.*

In our pools, isotropic operators flatten a sum that is mostly the heavy subset. This raises the light clients' directions to the same level regardless of their energy. §5 tests it without dominance.

Causal experiment. We equalize the pool, scale one client's B by ρ so that its update is ρ times every other client's with no direction changed, and merge with no magnitude control (Figure 5, App. E). The spectral operators fall to essentially zero by ρ=100, and task arithmetic, after first recovering the scaled client's tasks, collapses by ρ=300. The sweep replicates on MM-MergeBench, where task arithmetic, DC-Merge, and Iso-CTS all collapse at ρ=100. PolCap with Iso-CTS (§6) does not collapse (we did not sweep OCap with task arithmetic). After a small drop at ρ=3, where its cap first becomes active, it stays flat up to ρ=700 on FINLORA and ρ=100 on MM-MergeBench.

5 DIAGNOSIS II: WHICH FACTOR IS SHARED IS SET BY INITIALIZATION

To control magnitude, we must know which factor to act on, and the literature disagrees about which one is shared across clients (§2). We isolate the one variable that differs between the two findings.

Controlled study. We train fifteen adapters, three tasks (sentiment, headline, ner) under five regimes. Across the four primary regimes, data, recipe, and seeds are identical; only the initialization A_0 and the learning rate on A differ, and the shared-versus-independent contrast (`S` against `I`) varies A_0 alone. A fifth regime, `I2`, is a second independent draw with more training data, so it is a replication and not a matched seed (App. F). *Overlap* is the mean principal-angle cosine between two clients' subspaces (Björck & Golub, 1973), compared with the same statistic for random subspaces; *drift* is $\|A - A_0\|_F/\|A_0\|_F$, and the *displacement* of A is $A - A_0$.

Result (Table 14). We observe that raw A-similarity only reflects the shared initialization. Overlap is .93 when clients share A_0 and at chance (.05) when they do not. The learned part of A, its displacement, and the column space of B are shared to the same degree (.17–.23) in *both* regimes. This means that the A-shared results (Sun et al., 2024; Guo et al., 2025; Qiao & Mahdavi, 2026) were obtained under protocols in which clients inherit one A_0, from a server broadcast or a preceding task, while the B-shared results (Tang & Yang, 2026; Wang et al., 2025b) were obtained with separately trained adapters. Both findings are correct for their own protocol. The public pools show the same pattern. The LoRA Land hub (Zhao et al., 2024), which shares A_0, has A overlap .80, while the Qwen3-VL-2B (Mou et al., 2026), MM-MergeBench, and FINLORA pools sit near chance (App. F).

Consequence for merging. On an independently trained pool the A_i are unrelated random frames, so aggregating the factors separately does not aggregate the clients. For example, averaging A and B gives an update about five times too small (median over weight matrices 5.2×), and a one-shot port of SLAO's update scores near the unmerged base (App. J). The merge must therefore act on the product $\Delta W_i = B_i A_i$; B-side corrections address the same quantity up to a factor of 3.8 (Prop. 5). This changes only when a shared A_0 moves very little. At drift .05, factor-wise aggregation beats a product-level rule by .12, and at drift .30 it loses by the same amount (three-client pools). Note that this low-drift pool is nearly degenerate, which is the case that Prop. 3 warns about (App. F). Our independent regime drifts .31 and is past the crossover, and `I2` drifts even further (.80).

6 PROPOSED METHOD: OUTLIER CAPPING

Dominance is a between-client property of the products ΔW_i (§4), and on independently trained pools only the products are comparable across clients (§5). We therefore bound each product's magnitude *before* the operator is applied, only for outliers, and change nothing else. The server applies OCap once, to the received adapters, with no further communication. Unlike in robust federated learning, a large update here is not suspicious, since it comes from the training recipe (§4).

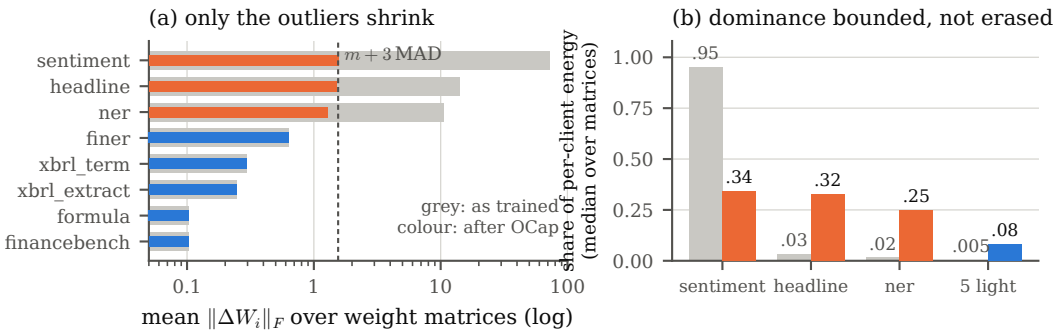


Figure 2: How OCap works on FINLORA. **(a)** Each client's mean $\|\Delta W_i\|_F$ as trained (grey) and after OCap (colour): the bound $m + 3\,\mathrm{MAD}$, computed per weight matrix, is active only for the three heavy clients and leaves the five light ones unchanged (mean spread $708\times \rightarrow 15.1\times$). **(b)** Share of the per-client energy. The largest client's .95 becomes three shares of about a third each, and the light clients' share rises from .005 to .08.

Client step. Per weight matrix, let $n_i = \|\Delta W_i\|_F$, $m = \mathrm{median}_j\, n_j$ and $\mathrm{MAD} = \mathrm{median}_j\, |n_j - m|$. OCap replaces

$$\Delta W_i \;\leftarrow\; \min\Big(1,\; \frac{m + 3\,\mathrm{MAD}}{n_i}\Big)\,\Delta W_i. \tag{1}$$

A client inside the bound is unchanged, and an outlier keeps its directions and spectral shape and is shrunk to the bound. The only design choice is *which* clients to shrink, and OCap shrinks only the outliers. It does not equalize every client, clip at the typical client as robust federated learning does (Nguyen et al., 2022), or shrink the whole update; §7.2 measures each alternative. Any operator can follow, and our default is task arithmetic, the mean of the capped updates (Figure 2).

When the operator is not an average: PolCap. Operators that equalize the singular values of the merged update, Iso-C and Iso-CTS (Marczak et al., 2025), also assume that flattening is safe, and Prop. 3 shows that it is not when the merged update has many near-null directions, which concentrated client spectra can produce. For these operators, and empirically for every operator we test except task arithmetic and DC-Merge, which flattens each client itself, we flatten each client first; why the operators that do not equalize the spectrum also need it remains open. With the compact SVD $\Delta W_i = U_i \Sigma_i V_i^\top$ over the client's r directions and its polar factor $Q_i = U_i V_i^\top$ (Higham, 1986), the polar step replaces ΔW_i by $(\nu_i/\sqrt{r})\, Q_i$ with $\nu_i = \|\Delta W_i\|_*/\sqrt{r}$, and OCap then caps the result; Polar Capping (PolCap) is OCap after the polar step. **Polar Equalization** (PolEq), our instrument for the mechanism experiments, sets every flattened client to the pool median m of the raw norms instead of capping. A narrower criterion we pre-registered, that only spectrum-equalizing operators need the polar step, held for Iso-C and failed for TSV-M, KnOTS-TIES, and TIES (§7.2).

Properties. Proofs and measurements are in Apps. B and C.

P1 *Dominance is bounded but not removed.* With $\bar{n}_j = \min(n_j, m + 3\,\mathrm{MAD})$ the capped norms, no client holds more than $(m + 3\,\mathrm{MAD})^2 / \sum_j \bar{n}_j^2$ of the summed per-client energy $\sum_j \bar{n}_j^2$. On FINLORA the bound is active only for the heavy three; the largest client's share falls from .95 to .34 and the light clients' share rises from .005 to .08, while the heavy three keep most energy.

P2 *Inactive without dominance.* A client inside the bound is unchanged, so between-client ratios change only where the outlier test fires. On MM-MergeBench the bound is active for 0.20 of 8 clients per weight matrix, against 1.89 on FINLORA, and OCap leaves task arithmetic's merged update unchanged in the median weight matrix.

P3 *Data-free, untuned, and cheap.* m and MAD are pool statistics, and the constant 3 was fixed in advance. $n_i^2 = \mathrm{tr}\big((B_i^\top B_i)(A_i A_i^\top)\big)$ needs two $r \times r$ Gram matrices; no SVD is formed. The polar step adds thin QR factorizations of B_i and $A_i^\top$ and one $r \times r$ SVD, and $(\nu_i/\sqrt{r})Q_i$ is the Frobenius-nearest matrix to ΔW_i with r equal nonzero singular values (Prop. 6).

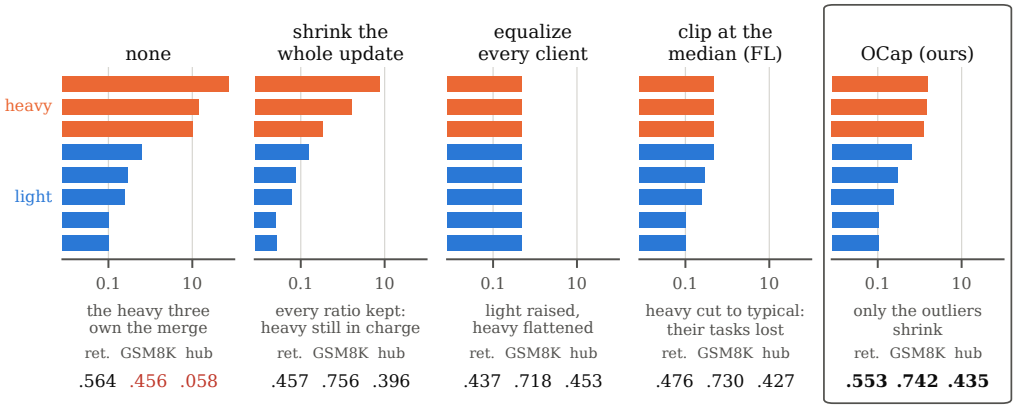


Figure 3: Which clients shrink, with task arithmetic. Each panel shows every client's mean $\|\Delta W_i\|_F$ after the client step (log; orange heavy, blue light) and, below, FINLORA retention and GSM8K strict (n=2000, limit 500) and LoRA Land ("hub") accuracy. Red: significantly below the base (.288, .712, .327). Paired intervals are in the text; transfer in Table 18.

When to use OCap. Based on the diagnosis and §7, we expect OCap to help when:

- **Pool:** independently trained clients whose norms have outliers. Without outliers the bound is inactive (P2), and a relative bound cannot help a pool of only heavy clients (App. I).
- **Operator:** task arithmetic by default; with other operators but DC-Merge, use PolCap (§7.2).
- **Constant:** 3 MADs, fixed in advance. Under PolCap with Iso-CTS only, 2, 3, and 4 MADs move retention and transfer within noise but not capability, which is best at 2 (Table 26); a fixed bound at τ times the median has an interior optimum, which the untuned MAD bound reaches (Tab. 18).

7 EXPERIMENTS

We evaluate 114 merge rules on FINLORA (106 with transfer, and 41 on all three axes besides the cap-removed ablation), and we repeat the key comparisons on a second dominated pool (§7.3). All numbers are at n=2000 items per task unless stated otherwise, and App. G contains every table.

7.1 MAIN RESULTS: COLLAPSE AND RECOVERY

Collapse and recovery. All four spectral operators score exactly zero as published, and they recover to .48–.67 under the scalar control (Figure 1b). This shows that magnitude control is necessary.

OCap vs. task arithmetic. Task arithmetic alone reaches .564 retention, but falls below the base on transfer ($-.022$) and on GSM8K strict ($-.256$). OCap stays within .010 $[.002, .018]$ of its retention and reverses both losses: transfer is $+.013$ $[.003, .023]$ and GSM8K strict is $+.030$ over the base (not significant). The rules that beat OCap on retention lose capability. KnOTS-TIES under the scalar control gives up .71 of GSM8K strict, and Iso-CTS at γ=.6 gives up .19 (Table 22). The exception is PolCap with Iso-CTS at 3 or 2 MADs ($+.371$, $+.365$ retention), which is safe on FINLORA but not on the hub pool (§7.3). The best single adapter beats OCap on transfer by .021.

7.2 THE DESIGN SPACE OF THE CLIENT STEP

Which clients shrink (Figure 3). Clipping at the median, which is the bound used in robust federated learning, loses .078 $[.070, .086]$ retention compared with OCap. All of this loss comes from the heavy clients' classification tasks ($-.191$), which depend on their magnitude, and clipping ties OCap on every other measure of both pools. Equalizing every client has the opposite trade-off. It gives up .305 on the heavy clients' classification tasks to gain .073 on the light clients' extraction tasks and .023 transfer, and it is .018 more accurate on the hub pool. As a result, OCap leads by .116 $[.108, .124]$ retention, but not on every measure. Shrinking the whole update to OCap's

Table 1: Both pools, difference to the unmerged base: FINLORA retention (n=2000), LoRA Land mean accuracy on eight client tasks (n=200 each), and GSM8K strict (limit 500). Bold: 95% paired interval below zero. Every constant is as on FINLORA, including Iso-CTS's k_c=32.

		FINLORA (Llama-3.1-8B)		LoRA Land (Mistral-7B)	
client step	operator	retention	GSM8K strict	accuracy	GSM8K strict
none (published)	Iso-CTS	**−.288**	—	**−.327**	—
none (published)	DC-Merge	**−.288**	—	**−.281**	—
DO-Merging		+.085	**−.614**	**−.327**	—
full equalization	task arithmetic	+.150	+.006	+.126	+.004
full equalization, TA norm	Iso-CTS	+.178	+.020	−.001	+.018
scalar control	DC-Merge	+.188	+.034	+.214	**−.392**
clip at the median	task arithmetic	+.188	+.018	+.101	+.020
shrink the whole update	task arithmetic	+.169	+.044	+.069	+.040
OCap	**task arithmetic**	+.266	+.030	+.108	+.014
OCap	DC-Merge	+.179	+.018	+.185	**−.390**
OCap	Iso-CTS	+.403	**−.400**	−.016	**−.390**
PolCap	task arithmetic	+.184	+.028	+.036	+.036
PolCap	Iso-CTS	+.371	+.028	**−.071**	**−.372**
PolCap	DC-Merge	+.247	+.020	+.159	**−.382**

norm keeps every ratio between clients, so the heavy subset still dominates the merge. It is .097 [.090, .104] below OCap on retention, on both halves (−.130 classification, −.063 extraction), and .039 [.024, .054] below it on the hub pool's client tasks, while it is level on the other measures. This shows that the accuracy of OCap comes from the bound itself and not from the smaller update. Finally, removing the bound from PolCap brings Iso-CTS back to zero (τ=∞, Table 2).

When to flatten. With Iso-CTS, which equalizes the merged spectrum, unflattened updates lose answer format under every magnitude control (GSM8K strict .312 under OCap and .584 under full equalization). Flattening each client first restores the format. PolCap reaches .740, which is +.428 [.380, .474] over OCap on the same items, at a cost of .032 retention (.659 vs. .690). With task arithmetic, flattening only has costs (.082 retention, .011 transfer, and .072 hub accuracy), and GSM8K does not change. We pre-registered, from the operator definitions alone, which operators need the step, and we then tested four more operators (App. H). The prediction held only for Iso-C. TSV-M, KnOTS-TIES, and TIES also lose answer format without the step (.002–.006 vs. .176–.744), so the rule we state after the fact is broader. We flatten before every operator except task arithmetic and DC-Merge, which flattens each client itself. Note that the step is necessary but not sufficient, since Iso-C, TSV-M, and KnOTS-TIES still lose answer format with it.

7.3 A SECOND DOMINATED POOL

The 26 distinct LoRA Land adapters for Mistral-7B (Zhao et al., 2024; Jiang et al., 2023) share their A_0 (§5), but they are still dominated, with a mean spread of 251×. OCap is active for the same four clients in every weight matrix and reduces the spread to 6.9×. We score eight of the clients' own classification tasks under each adapter's published prompt (Table 1, App. I.1).

What transfers. The diagnosis transfers to this pool. Every published operator, and also DO-Merging, falls far below the base. The fix also transfers. Six rules in Table 1 do not regress on any measure of either pool, and all of them keep the merged update at the scale of task arithmetic. These are OCap, clipping at the median, shrinking the whole update, full equalization, and PolCap before task arithmetic, and full equalization with Iso-CTS rescaled to task arithmetic's norm. Among the six, OCap has the highest FINLORA retention, .078 above the next rule, while full equalization is .018 more accurate on the hub pool's client tasks.

What does not transfer. PolCap with Iso-CTS, the safe rule with the highest retention on FINLORA (+.371), falls below the base on this pool. The likely reason is the common subspace of Iso-CTS. Its 32 directions are half of FINLORA's budget but only 15% of this pool's budget, and at half the budget PolCap is above the base on client accuracy (App. I.1). Note that k_c=32 was fixed

Table 2: No-regression ranking against the unmerged base (difference, paired-bootstrap 95% interval) for the eleven rules of twenty-nine never below it, the next rule, task arithmetic (eighteenth) and the $\tau{=}\infty$ ablation (cap removed); bold marks an interval below zero; asym-CTS takes its common subspace from the B factors (App. C). Full table: Table 20.

rule	retention	transfer	GSM8K strict	GSM8K flexible	worst z
OCap+task arithmetic (ours)	$+.266$ [$+.257, +.274$]	$+.013$ [$+.003, +.023$]	$+.030$ [$-.010, +.070$]	$+.008$ [$-.024, +.038$]	$+0.51$
clip at median+task arithmetic	$+.188$ [$+.180, +.197$]	$+.016$ [$+.008, +.024$]	$+.018$ [$-.016, +.054$]	$+.000$ [$-.032, +.030$]	$+0.00$
PolCap+Iso-CTS, 2 MADs	$+.365$ [$+.356, +.373$]	$+.018$ [$+.009, +.029$]	$+.054$ [$+.016, +.092$]	$-.002$ [$-.034, +.032$]	-0.12
Iso-CTS, full equalization, TA norm	$+.178$ [$+.171, +.185$]	$+.007$ [$-.000, +.015$]	$+.020$ [$-.012, +.054$]	$-.002$ [$-.030, +.024$]	-0.15
PolCap+task arithmetic	$+.184$ [$+.175, +.192$]	$+.002$ [$-.005, +.010$]	$+.028$ [$-.004, +.062$]	$-.002$ [$-.030, +.024$]	-0.15
task arithmetic, shrunk to OCap's norm	$+.169$ [$+.161, +.177$]	$+.014$ [$+.006, +.023$]	$+.044$ [$+.012, +.076$]	$-.002$ [$-.030, +.024$]	-0.15
PolCap+DC-Merge	$+.247$ [$+.238, +.255$]	$-.002$ [$-.011, +.006$]	$+.020$ [$-.012, +.054$]	$+.012$ [$-.016, +.040$]	-0.52
task arithmetic, full equalization	$+.150$ [$+.142, +.158$]	$+.036$ [$+.026, +.045$]	$+.006$ [$-.030, +.042$]	$-.010$ [$-.040, +.018$]	-0.68
PolCap+Iso-CTS	$+.371$ [$+.363, +.379$]	$+.015$ [$+.005, +.026$]	$+.028$ [$-.012, +.068$]	$-.016$ [$-.048, +.016$]	-0.98
DC-Merge, $\gamma{=}.5$	$+.188$ [$+.181, +.196$]	$-.006$ [$-.015, +.002$]	$+.034$ [$+.000, +.070$]	$+.000$ [$-.028, +.030$]	-1.54
OCap+DC-Merge	$+.179$ [$+.171, +.187$]	$-.007$ [$-.016, +.001$]	$+.018$ [$-.016, +.054$]	$+.000$ [$-.028, +.028$]	-1.66
PolEq+asym-CTS, $k_c{=}16$	$+.352$ [$+.343, +.360$]	$+.039$ [$+.028, +.050$]	$\mathbf{-.068}$ [$-.110, -.026$]	$-.004$ [$-.036, +.026$]	-3.17
task arithmetic	$+.276$ [$+.267, +.285$]	$\mathbf{-.022}$ [$-.033, -.011$]	$\mathbf{-.256}$ [$-.306, -.210$]	$\mathbf{-.098}$ [$-.136, -.058$]	-10.45
PolCap+Iso-CTS, $\tau{=}\infty$ (cap removed)	$\mathbf{-.288}$ [$-.296, -.280$]	$\mathbf{-.467}$ [$-.481, -.454$]	$\mathbf{-.712}$ [$-.750, -.672$]	$\mathbf{-.784}$ [$-.818, -.748$]	-69.01

before this pool was scored. On GSM8K, every rule with a large merged update collapses to near zero regardless of its client step. Only the rules at task arithmetic's scale stay at or above the base, and PolCap with task arithmetic ($+.036$ each) and task arithmetic shrunk to OCap's norm ($+.069$, $+.040$) are significantly above it on both measures of this pool.

7.4 No Rule Wins; Eleven Never Regress

The top six rules on retention are within .007 of each other, and they reorder under a second item draw. The first of them is OCap with Iso-CTS, which loses answer format (§7.2). A server without validation data should therefore ask how much a rule risks compared with not merging. We rank the twenty-nine rules measured on all three axes at the largest draw by worst z (Table 2).

Eleven rules are never significantly below the base, and every other rule is at $z = -3.17$ or below. The six rules at task arithmetic's scale are safe because they stay close to the base. Among them, OCap has the highest retention ($+.266$) and the best worst z ($+0.51$; third without the flexible filter), although worst z hardly separates the safe rules. Rules with a large merged update are safe on FinLoRA only when each client is flattened, by PolCap or inside DC-Merge, and they lose GSM8K on the hub pool. Caveat: OCap with task arithmetic and the other bounds of Figure 3 were measured after the no-regression criterion was fixed, and PolCap at 2 MADs after its multiplier sweep.

7.5 Floors, Capability, and a Pool without Dominance

Floors. Under its chat template the base gains .17 retention, and OCap with task arithmetic stays .10 [.09, .11] above it on the same 500 items (App. D). **Capability.** OCap with task arithmetic matches the base on MMLU (.687 vs. .680) and HellaSwag (.714 vs. .706), and is level with it on GSM8K strict (.742 vs. .712, not significant). **A pool without dominance.** On MM-MergeBench the bound is active for 0.20 of 8 clients per weight matrix, and OCap with task arithmetic scores exactly as task arithmetic alone does (54.3 seen, 37.0 unseen). With PolCap, DC-Merge is level with its published version (56.8 vs. 56.7 seen), and Iso-CTS loses less format (48.3 vs. 41.9 strict; App. I).

8 Discussion and Conclusion

We show that one-shot federated merging of public LoRA adapters fails because client magnitudes are incommensurable, and not because client subspaces are misaligned, and we propose OCap, which bounds only the outlier clients. Operators other than an average also need each client to be flattened first. Bounding intruder dimensions (Shuttleworth et al., 2025) is left to future work.

Limitations. Both dominated pools are text pools. OCap with task arithmetic was measured after the criterion that ranks it was fixed (§7.4), and most robustness checks (Apps. E–I) were run on PolCap with Iso-CTS before it. A relative bound cannot help an all-heavy pool (App. I).

ETHICS STATEMENT

All authors have read and adhere to the ICLR Code of Ethics. This work involves no human subjects and collects no personal data. Every adapter, base model, and evaluation set we use is publicly released, and the fifteen adapters we train use training data from the same public benchmark. Our financial tasks are built from public filings, news, and exam questions, and the merged models are research artifacts, not tools for financial advice. One risk concerns deployment. A server that merges adapters from unrelated parties inherits whatever each adapter has learned, including undesirable behavior. OCap limits how much a single client with an unusually large update can influence the merge, but we did not evaluate adversarial clients, and OCap should not be read as a defense against them.

REPRODUCIBILITY STATEMENT

The supplementary material contains the code, every saved generation, and the per-item GSM8K outputs, so every number in the paper can be recomputed offline, without a GPU pass, by the scripts that generate its tables. The main pool, the sibling pool, the LoRA Land hub, and the multimodal pool are public; the fifteen adapters of the controlled study are the only models we train, and the second independent draw differs from the first in data volume as well as in seed, which we state because it makes that draw a replication rather than a matched second seed. Appendix L reproduces representative prompts verbatim, generated directly from the evaluation code rather than transcribed (the supplementary code prints all of them on demand), together with the decoding configuration, the answer-cue rule, the harness configuration, and a three-way scorer-sensitivity analysis. Appendix B contains the proofs and Appendix G the full result grid. An automated integrity pass over the result tree checks for stale generations, partial models, unequal n within a comparison, and rank misspecification; its output is included in the supplementary material.

AI USE STATEMENT

We used a generative AI coding assistant in this work and disclose its role. It was used for software development (the evaluation harness, the merge-rule implementations, and the analysis and table-generating scripts), for running and monitoring the experimental queue, for literature search, for drafting mathematical arguments that we then checked, for structuring, drafting, and editing the text of this paper, and for automated audits that recomputed the paper's numbers from the saved outputs and checked its internal consistency.

The research is the authors'. The research question, the experimental design, the choice of pools, baselines, and evaluation axes, the reading of the results, and the decisions about which claims the evidence supports—including the decision to withdraw several earlier claims (Appendix J) and to rank rules by worst-case regression rather than by a single axis—are ours. We reviewed the code, re-derived the headline numbers from the saved generations, checked the proofs, verified that every cited work exists and says what we attribute to it, and take full responsibility for all content. No synthetic data was generated: every evaluation item comes from a public benchmark, and every model output we score was produced by the evaluated models themselves.

REFERENCES

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

Justin Zhao, Timothy Wang, Wael Abid, Geoffrey Angus, Arnav Garg, Jeffery Kinnison, Alex Sherstinsky, Piero Molino, Travis Addair, and Devvret Rishi. LoRA Land: 310 fine-tuned LLMs that rival GPT-4, a technical report. *arXiv preprint arXiv:2405.00732*, 2024.

Zhixu Silvia Tao, Ian Mason, Sanjeev Kulkarni, and Xavier Boix. Task Arithmetic through the lens of one-shot Federated Learning. *Transactions on Machine Learning Research (TMLR)*, 2025.

Ziyao Wang, Bowei Tian, Yexiao He, Zheyu Shen, Guoheng Sun, Yuhan Liu, Luyang Liu, Meng Liu, and Ang Li. Revisiting federated fine-tuning: A single communication round is enough for foundation models. *arXiv preprint arXiv:2412.04650*, 2024.

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 (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017.

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

Dannong Wang, Jaisal Patel, Daochen Zha, Steve Y. Yang, and Xiao-Yang Liu. FinLoRA: Benchmarking LoRA methods for fine-tuning LLMs on financial datasets. *arXiv preprint arXiv:2505.19819*, 2025a.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43177–43199. PMLR, 2025.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705, 2025.

Han-Chen Zhang, Zi-Hao Zhou, Mao-Lin Luo, Shimin Di, Min-Ling Zhang, and Tong Wei. DC-Merge: Improving model merging with directional consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22248–22258, 2026.

Shenghe Zheng, Hongzhi Wang, Chenyu Huang, Xiaohui Wang, Tao Chen, Jiayuan Fan, Shuyue Hu, and Peng Ye. Decouple and orthogonalize: A data-free framework for LoRA merging. *arXiv preprint arXiv:2505.15875*, 2025.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving LoRA in privacy-preserving federated learning. In *International Conference on Learning Representations (ICLR)*, pp. 17978–17994, 2024.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *International Conference on Learning Representations (ICLR)*, pp. 99003–99027, 2025.

Fuli Qiao and Mehrdad Mahdavi. Merge before forget: A single LoRA continual learning via continual merging. In *International Conference on Learning Representations (ICLR)*, pp. 8133–8159, 2026.

Yixuan Tang and Yi Yang. Crowded in B-space: Calibrating shared directions for LoRA merging. *arXiv preprint arXiv:2604.16826*, 2026.

Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. FLAME: Taming backdoors in federated learning. In *31st USENIX Security Symposium*, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 7093–7115, 2023.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are Super Mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR, 2024.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the Knots. In *International Conference on Learning Representations (ICLR)*, pp. 4501–4519, 2025.

Xisen Jin, Xiang Ren, Daniel Preoţiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. LoraHub: Efficient cross-task generalization via dynamic LoRA composition. In *Conference on Language Modeling (COLM)*, 2024.

Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019. NeurIPS 2019 Workshop on Federated Learning for Data Privacy and Confidentiality.

Galen Andrew, Om Thakkar, H. Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17455–17466, 2021.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez de Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62369–62385. PMLR, 2024.

Lei Wang, Jieming Bian, Letian Zhang, and Jie Xu. Adaptive LoRA experts allocation and selection for federated fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025b.

Raghav Singhal, Kaustubh Ponkshe, and Praneeth Vepakomma. FedEx-LoRA: Exact aggregation for federated and efficient fine-tuning of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1316–1336, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.67.

Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. RobustMerge: Parameter-efficient model merging for MLLMs with direction robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, 2024.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3697–3711. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.300.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6279–6292, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.421.

Yan Lin, Jingyu Sun, Zhongliang Guo, Qing Li, Zhuohan Xie, and Yuxia Wang. FinExam-10K: When retrieval helps financial reasoning? *arXiv preprint arXiv:2608.28155*, 2026.

Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. ECTSum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 10893–10906, 2022.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

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.

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.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4791–4800, 2019.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The Language Model Evaluation Harness. Zenodo, version v0.4.3, 7 2024. URL `https://zenodo.org/records/12608602`.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 30318–30332, 2022.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall, New York, 1993.

Åke Björck and Gene H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

Lichao Mou, Shilan Zhang, Chunlei Li, Bingcong Yan, Jingliang Hu, Yilei Shi, Shengwu Xiong, Xiao Xiang Zhu, Lei Li, and Yaxiong Chen. Model merging for medical LVLMs: A benchmark and a winner-take-all approach. *arXiv preprint arXiv:2607.15661*, 2026.

Nicholas J. Higham. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. LoRA vs full fine-tuning: An illusion of equivalence. In *Advances in Neural Information Processing Systems*, 2025.

Per-Åke Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA. *arXiv preprint arXiv:2312.03732*, 2023.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796, 2014.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. WWW'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of The Web Conference 2018 (WWW)*, pp. 1941–1942. ACM, 2018. doi: 10.1145/3184558.3192301.

Ricky Costa. Twitter financial news sentiment dataset. Hugging Face dataset (zeroshot/twitter-financial-news-sentiment), 2022. URL `https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment`.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. FinGPT: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023. FinLLM Symposium at IJCAI 2023.

Ankur Sinha and Tanmay Khandait. Impact of news on the commodity market: Dataset and results. In *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 2*, volume 1364 of *Advances in Intelligent Systems and Computing*, pp. 589–601. Springer, 2021.

Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pp. 84–90, 2015.

Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Georgios Paliouras. FiNER: Financial numeric entity recognition for XBRL tagging. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4419–4431, Dublin, Ireland, 2022. Association for Computational Linguistics.

Shijie Han, Haoqiang Kang, Bo Jin, Xiao-Yang Liu, and Steve Y. Yang. XBRL agent: Leveraging large language models for financial report analysis. In *Proceedings of the 5th ACM International Conference on AI in Finance (ICAIF)*, pp. 856–864. ACM, 2024. doi: 10.1145/3677052.3698614.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. FinanceBench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2023.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations (ICLR)*, 2020.

A Extended Related Work

Why no published operator controls between-client magnitude. Entry-wise operators are linear in each client's magnitude. Task arithmetic (Ilharco et al., 2023) degrades toward the heavy subset but not to zero (.5638 at n=2000). TIES (Yadav et al., 2023) first trims each client separately, which is scale-free. It then elects a sign from the summed signed magnitude, which is not scale-free (.1592 at n=100; .6176 at n=2000 under the scalar control). DARE (Yu et al., 2024) rescales its surviving entries to preserve each client's expected contribution, so it inherits the same linearity. KnOTS (Stoica et al., 2025) first aligns the clients in a joint SVD basis. The failure comes from the aggregation inside this basis and not from the basis. KnOTS reaches .5705 with plain averaging (by construction the same merge as task arithmetic, .5638, so the gap is generation noise) and .1337 with TIES (Table 18). Spectral operators never compare clients' magnitudes. A single global scale cannot change a between-client ratio, and rescaling each client to its own norm, as DC-Merge does, preserves it. This is harmless when magnitudes are comparable but fatal on our pool. DO-Merging (Zheng et al., 2025) equalizes every client's column directions and rescales their sum by the summed magnitudes, which the heavy clients determine. In our implementation (Appendix C), it collapses on LoRA Land (Table 1).

The A/B disagreement comes from different protocols. Table 3 matches each cited work to the regime of our controlled study (Table 14). When A is inherited from a server broadcast or a preceding task, cross-client A-similarity reflects the starting point and not what was learned. The single-adapter asymmetry result (Zhu et al., 2024) shows that a frozen random A with a trained B nearly matches training both. This result concerns one adapter and not a pool, so it constrains neither side.

Table 3: Protocols behind the published A/B claims, as the cited works describe them, and the matching regime of our controlled study. Overlaps are mean principal-angle cosines over $q/k/v$; chance is .037–.073.

work	client protocol	matching regime and its measurement
FFA-LoRA (Sun et al., 2024)	iterative; one broadcast A_0, then frozen	`S_fz`: A overlap 1.000 by construction
FedSA-LoRA (Guo et al., 2025)	iterative; A re-broadcast every round	`S`: A .921–.937, B .182–.229
SLAO (Qiao & Mahdavi, 2026)	sequential; each adapter starts from the previous task's (A re-orthogonalized by QR)	`S` is the closest regime; we ran no chain
FedLEASE (Wang et al., 2025b)	independent local warm-up per client; clusters on B	`I`: A .043–.048, B .179–.220
Pico (Tang & Yang, 2026)	separately trained; initialization not stated	`I` is the closest regime

Cross-client B overlap is .182–.229 under `S` and .179–.220 under `I`. In both regimes this is far above chance and far below one, so "B is client-specific" and "B is crowded" may be the same number read from two sides. Training one task under both regimes with the same data and recipe but a different A_0 gives A overlap .061–.096 against B overlap .442–.492. Our practical disagreement with the A-shared line is narrower than the empirical one. It concerns where factor-wise aggregation pays off, which the drift crossover of §5 locates. In our setting, freezing A saves communication but does not improve accuracy.

Multimodal merging. RobustMerge (Zeng et al., 2025) derives a scaling from statistics of A and applies it to B. We use its released pool but not its operator. MergeMedBench (Mou et al., 2026) is used only in the initialization diagnostic.

What we do not claim. *A new merge operator.* The pool step is a published operator (task arithmetic by default, or any other), and under the PolEq client step three spectral operators are within .016 of one another. *Per-client spectrum flattening.* This is the first step of DC-Merge (Zhang et al., 2026), and PolCap differs from DC-Merge in the magnitude it assigns to each client. *Superiority without dominance.* On MM-MergeBench, DC-Merge as published leads on the seen tasks. *A*

statement about LoRA itself. Neither side of the A/B disagreement is wrong, and §5 attributes the disagreement to the training protocol.

B PROOFS

Throughout, $\Delta W_i = B_i A_i \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is client i's update at *one* weight matrix, with $A_i \in \mathbb{R}^{r \times d_{\text{in}}}$, $B_i \in \mathbb{R}^{d_{\text{out}} \times r}$ and the LoRA scale folded into B_i, so that $\operatorname{rank} \Delta W_i \le r$. We write $S = \sum_{i=1}^{n} \Delta W_i$, and for $H \subseteq [n]$, $S_H = \sum_{i \in H} \Delta W_i$ and $E = S - S_H = \sum_{i \notin H} \Delta W_i$. Norms are Frobenius $\|\cdot\|_F$, spectral $\|\cdot\|_2$ and nuclear $\|\cdot\|_*$. Index 1 is the heaviest client, $\|\Delta W_1\|_F = \max_i \|\Delta W_i\|_F$. For a matrix M, $U_k(M)$ is the span of k leading left singular vectors and $V_k(M)$ of k leading right ones; when $\sigma_k(M) = \sigma_{k+1}(M)$ these spans are not unique, and every statement below holds for *any* choice of them. $\Theta(\mathcal{U}, \mathcal{U}')$ denotes principal angles between subspaces of equal dimension, and $\|\sin\Theta\|_2 = \sqrt{1 - \cos^2\theta_{\max}}$. Every non-score number below is recomputed by the supplementary code from the eight public adapters, per weight matrix, and reported as the median over the 96 adapted matrices. Scores have the provenance stated in their own appendix. At the settings of §3 the pool has $n = 8$, $r = 8$, rank budget $nr = 64$ and common-subspace budget $k_c = nr/2 = 32$.

B.1 A PERTURBATION BOUND FOR SINGULAR SUBSPACES

Proposition 2 relies on one inequality, which we state and prove here. It is the standard $\sin\Theta$ bound, i.e., Wedin's theorem for singular subspaces (Wedin, 1972) and Davis–Kahan on a symmetric pair (Davis & Kahan, 1970).

Proposition 4 ($\sin\Theta$ bound)**.** *Let $M, F \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and $N = M + F$. Fix $k \ge 1$ and fix singular value decompositions of M and N. If $\delta = \sigma_k(N) - \sigma_{k+1}(M) > 0$, then*

$$\big\|\sin\Theta\big(U_k(M), U_k(N)\big)\big\|_2 \;\le\; \frac{\|F\|_2}{\delta}, \qquad \big\|\sin\Theta\big(V_k(M), V_k(N)\big)\big\|_2 \;\le\; \frac{\|F\|_2}{\delta}.$$

Proof. Take *full* decompositions $M = [P_1 \;\; P_2]\,\operatorname{diag}(\Gamma_1, \Gamma_2)[Q_1 \;\; Q_2]^\top$ and, likewise, $N = [\hat{P}_1 \;\; \hat{P}_2]\,\operatorname{diag}(\hat{\Gamma}_1, \hat{\Gamma}_2)[\hat{Q}_1 \;\; \hat{Q}_2]^\top$. The bracketed factors are orthogonal matrices, of order d_{out} on the left and d_{in} on the right, and Γ_1, $\hat{\Gamma}_1$ contain the leading k singular values. So P_1 spans $U_k(M)$ and P_2 its orthogonal complement, and likewise for N. Since $N\hat{Q}_1 = \hat{P}_1\hat{\Gamma}_1$ and $P_2^\top M = \Gamma_2 Q_2^\top$,

$$P_2^\top \hat{P}_1 \hat{\Gamma}_1 = P_2^\top N \hat{Q}_1 = P_2^\top M \hat{Q}_1 + P_2^\top F \hat{Q}_1 = \Gamma_2 Q_2^\top \hat{Q}_1 + P_2^\top F \hat{Q}_1.$$

Because $\sigma_k(N) = \sigma_{\min}(\hat{\Gamma}_1) > 0$, the matrix $\hat{\Gamma}_1$ is invertible and $\|X\|_2 \le \|X\hat{\Gamma}_1\|_2 / \sigma_k(N)$ for every X. Taking spectral norms and using $\|\Gamma_2\|_2 = \sigma_{k+1}(M)$, $\|P_2\|_2 = \|\hat{Q}_1\|_2 = 1$,

$$\sigma_k(N)\,\big\|P_2^\top \hat{P}_1\big\|_2 \;\le\; \sigma_{k+1}(M)\,\big\|Q_2^\top \hat{Q}_1\big\|_2 + \|F\|_2.$$

Applying the same argument to $M^\top$ and $N^\top$, which exchanges the roles of the left and right factors and leaves $\|F\|_2$ and every singular value unchanged, gives $\sigma_k(N)\|Q_2^\top \hat{Q}_1\|_2 \le \sigma_{k+1}(M)\|P_2^\top \hat{P}_1\|_2 + \|F\|_2$, since $\|F^\top\|_2 = \|F\|_2$. Put $x = \max\big(\|P_2^\top \hat{P}_1\|_2, \|Q_2^\top \hat{Q}_1\|_2\big)$. Bounding the right-hand side of each inequality by $\sigma_{k+1}(M)\,x + \|F\|_2$ and taking whichever of the two left-hand sides equals $\sigma_k(N)\,x$ gives $x\big(\sigma_k(N) - \sigma_{k+1}(M)\big) \le \|F\|_2$, i.e. $x \le \|F\|_2/\delta$. Finally $\|P_2^\top \hat{P}_1\|_2 = \|\sin\Theta(U_k(M), U_k(N))\|_2$ because P_2 spans $U_k(M)^\perp$, and likewise on the right. □

We note two points that are used below. First, the gap pairs the k-th singular value of *one* matrix with the $(k+1)$-th of the *other*, and the labelling determines which is which. Applying the proposition to (N, M) with perturbation $-F$ gives the mirrored gap $\sigma_k(M) - \sigma_{k+1}(N)$ for the same angles. Second, $\delta > 0$ forces $\sigma_k(N) > 0$, hence $k \le \operatorname{rank} N$, so the bound says nothing about directions that the matrix supplying the gap does not have. Note also that the bound is not asymptotic.

B.2 PROPOSITION 1: THE SUM IS ITS HEAVY SUBSET

Proof. For any $H \subseteq [n]$, $S - S_H = \sum_{i \notin H} \Delta W_i$, so by the triangle inequality $\|S - S_H\|_F = \big\|\sum_{i \notin H} \Delta W_i\big\|_F \le \sum_{i \notin H} \|\Delta W_i\|_F \le (n - |H|) \max_{i \notin H} \|\Delta W_i\|_F$. □

The bound concerns the merged object itself and does not depend on any operator applied afterwards. Its usefulness depends entirely on how small the right-hand side is on a given pool. With $H = \{\text{sentiment, headline, ner}\}$ the right-hand side is $.124\,\|S_H\|_F$ and the realized $\|S - S_H\|_F$ is $.068\,\|S_H\|_F$ at the median weight matrix. This shows that the pool sum is those three clients to within 7%. In contrast, the single-client specialization does not hold on this pool. Taking $H = \{1\}$ and $\varepsilon = \max_{i\neq 1} \|\Delta W_i\|_F / \|\Delta W_1\|_F = .176$ gives $\left\|\frac{1}{n}S - \frac{1}{n}\Delta W_1\right\|_F \leq \frac{n-1}{n}\varepsilon\|\Delta W_1\|_F = .154\,\|\Delta W_1\|_F$, and the sharper $\frac{1}{n}\sum_{i\neq 1}\|\Delta W_i\|_F$ reads $.052\,\|\Delta W_1\|_F$, against a realized normalized distance of .028. Without normalization, the realized $\|S - \Delta W_{\max}\|_F / \|\Delta W_{\max}\|_F$ has median .224, which is far from the .068 of the heavy subset. Note that the spread of §3 is the ratio of the heaviest to the *lightest* client and is *not* $1/\varepsilon$. Substituting it would understate the bound by two orders of magnitude. On this pool, dominance is a property of a heavy subset and not of a single outlier. This is why Proposition 1 is stated for arbitrary H.

B.3 Proposition 2: The Common Subspace Follows the Heavy Subset

Iso-C and Iso-CTS take their common subspace as $U_{k_c}(S)$, so the proposition applies to them directly. TSV-M and DC-Merge orthogonalize all nr of the clients' concatenated singular vectors, which does not depend on magnitude. They then rebuild the update from each client's own singular values (flattened, for DC-Merge). As a result, the between-client ratio enters them linearly through the reconstruction, and no bound on a truncated subspace applies to them.

Hypotheses in full. Fix $k \geq 1$ and fix singular value decompositions of S and S_H. If $\delta_k = \sigma_k(S_H) - \sigma_{k+1}(S) > 0$ then $\|\sin\Theta(U_k(S), U_k(S_H))\|_2 \leq \|E\|_2/\delta_k$, and the same bound holds on the right.

Proof. Apply Proposition 4 with $M = S$, $F = -E$ and $N = M + F = S_H$. Its gap is $\sigma_k(N) - \sigma_{k+1}(M) = \sigma_k(S_H) - \sigma_{k+1}(S) = \delta_k > 0$, its perturbation has $\|F\|_2 = \|-E\|_2 = \|E\|_2$, and its conclusion is a bound on $\|\sin\Theta(U_k(M), U_k(N))\|_2 = \|\sin\Theta(U_k(S), U_k(S_H))\|_2$, and likewise on the right. Note that the matrix whose k-th singular value enters the gap is S_H and not S. This is the direction in which δ_k is written in §4. □

The hypothesis fails at the budget the operators use, and this failure is the diagnosis. Since $\delta_k > 0$ requires $\sigma_k(S_H) > 0$, the bound is available only for $k \leq \operatorname{rank} S_H \leq |H|r$. We measure $\operatorname{rank} S_H = 24$ and $\operatorname{rank} S = 64$ at every one of the 96 weight matrices, against $k_c = 32$. Therefore, at the operator's own budget $\sigma_{32}(S_H) = 0$ and $\delta_{32} \leq 0$, and the proposition says nothing on all 96 matrices. It says almost nothing at $k = 24$ either, since δ_{24} has median $-.048$, is positive on 2 of 96 matrices, and gives $\|E\|_2/\delta_{24} < 1$ on one. This does not make the diagnosis wrong. It only means that this inequality does not cover all of it. Because $\operatorname{range} S \subseteq \operatorname{range} S_H + \operatorname{range} E$ with $\operatorname{rank} E \leq (n - |H|)r$, at least $k_c - \operatorname{rank} S_H = 8$ of the 32 common directions must lie outside $\operatorname{range} S_H$, regardless of the angles. The light clients contribute these directions, and the pool step then flattens them to $\bar{\sigma}$ (§B.4). The measured dimension $\|(I - \Pi_{S_H})U_{32}(S)\|_F^2$ of $U_{32}(S)$ outside $\operatorname{range} S_H$ has median 8.14 against that floor of 8, so the excess comes from the rank budget and essentially nothing else. Since a vacuous bound certifies nothing either way, we measure the angle it bounds directly. At $k = \operatorname{rank} S_H = 24$ the mean principal-angle cosine between $U_{24}(S)$ and $U_{24}(S_H)$ has median .848 over the 96 matrices (range .768–1.000), against a random-subspace chance level (Björck & Golub, 1973) of .130 for the 1024-row matrices and .065 for the 4096-row ones. The common subspace therefore follows the heavy subset as a measured result and not as a corollary.

B.4 Proposition 3: Flattening in the Near-Degenerate Band

The merged object and two means that are easy to confuse. Let M be the merged update the operator forms *before* flattening, let N be its rank budget, and let $\sigma_1 \geq \cdots \geq \sigma_N \geq 0$ be its singular values, padded with zeros if $\operatorname{rank} M < N$. Iso-C and Iso-CTS form M from S and a per-client residual block, so $N = nr$. The proposition concerns these isotropic operators. Write $\bar{\sigma} = N^{-1}\sum_{j\leq N}\sigma_j = \|M\|_*/N$ and $\bar{\sigma}_{\text{rms}} = \|M\|_F/\sqrt{N}$, and let $\Phi(M) = \bar{\sigma}\sum_{j\leq N} u_j v_j^\top$ be the isotropic operator. By Cauchy–Schwarz $\bar{\sigma} \leq \bar{\sigma}_{\text{rms}}$, with equality if and only if the spectrum is

flat. For an isotropic rule the merged adapter has a flat spectrum, so the two means coincide for it and $\bar{\sigma} = \|\Phi(M)\|_2$. For any X, $\|X\|_F/\sqrt{\operatorname{rank} X} \le \|X\|_2$ with equality only for a flat spectrum. Therefore, for a rule that does not flatten, the two means differ by the dispersion of the spectrum, and the indicator does not transfer.

Hypotheses in full. Fix $q < N$, $c > 0$ and $0 \le \delta < c$, and assume $\sigma_j \ge c$ for $j \le q$, $\sigma_j \le \delta$ for $j > q$, and $\bar{\sigma} \ge \delta$. Then (a)–(d) below hold.

Proof. (a) *Exact cost, and the angle.* Since $\sum_j \sigma_j = N\bar{\sigma}$ and $\|\Phi(M)\|_F^2 = N\bar{\sigma}^2$,

$$\|\Phi(M) - M\|_F^2 = \sum_{j \le N} (\bar{\sigma} - \sigma_j)^2 = \|M\|_F^2 - N\bar{\sigma}^2,$$

$$\cos\angle\big(\Phi(M), M\big) = \frac{\langle \Phi(M), M\rangle}{\|\Phi(M)\|_F \|M\|_F} = \frac{\|M\|_*}{\sqrt{N}\,\|M\|_F}.$$

(b) *The near-null block, two-sided and with no asymptotics.* For $j > q$ we have $0 \le \sigma_j \le \delta \le \bar{\sigma}$, so $\bar{\sigma} - \delta \le \bar{\sigma} - \sigma_j \le \bar{\sigma}$; squaring and summing over the $N - q$ such indices, $(\bar{\sigma} - \delta)\sqrt{N-q} \le \big(\sum_{j>q}(\bar{\sigma} - \sigma_j)^2\big)^{1/2} \le \bar{\sigma}\sqrt{N-q}$. We need the two-sided form because the additive error has the same factor $\sqrt{N-q}$ as the leading term. Writing it as $\bar{\sigma}\sqrt{N-q} - O(\delta)$ understates the correction by that factor.

(c) *Size of $\bar{\sigma}$, and the n-dependence.* From $N\bar{\sigma} = \sum_{j \le q} \sigma_j + \sum_{j>q} \sigma_j$ with the first sum in $[qc, q\sigma_1]$ and the second in $[0, (N-q)\delta]$, we get $qc/N \le \bar{\sigma} \le q\sigma_1/N + \delta$. With $N = nr$ and $q = r$ this is $c/n \le \bar{\sigma} \le \sigma_1/n + \delta$, so at $\delta = 0$ the near-null block of (b) is at most $\sigma_1\sqrt{r/n}$ and *decreases* in n like $n^{-1/2}$.

(d) *What does grow with the pool.* Because $\Phi(M)$ places $\bar{\sigma}^2$ on each of its N directions and $N - q$ of them had at most δ in M, a share $1 - q/N$ of $\|\Phi(M)\|_F^2$ lies on directions that had almost no weight. This holds for every δ. For the signal block, $\sigma_j - \bar{\sigma} \ge c - \bar{\sigma} \ge c - q\sigma_1/N - \delta$ termwise by (c), so $\big(\sum_{j \le q}(\sigma_j - \bar{\sigma})^2\big)^{1/2} \ge \sqrt{q}\,(c - q\sigma_1/N - \delta)$ whenever the right-hand side is non-negative, and trivially otherwise. Finally, at $\delta = 0$ we have $\sigma_j = 0$ for $j > q$, so Cauchy–Schwarz gives $\|M\|_* = \sum_{j \le q} \sigma_j \le \sqrt{q}\,\|M\|_F$ and (a) yields

$$\cos\angle\big(\Phi(M), M\big) \le \sqrt{q/N}, \qquad \frac{\|\Phi(M) - M\|_F}{\|M\|_F} = \sqrt{1 - \cos^2\angle} \;\ge\; \sqrt{1 - q/N}.$$

At $N = nr$ and $q = r$ these read $\cos\angle \le n^{-1/2}$ and $\sqrt{1 - 1/n}$. □

The damage grows as a fraction of the output and not in norm. In the near-degenerate band, the absolute perturbation injected into the near-null block does not grow with the pool, and by (c) it falls like $n^{-1/2}$ at $\delta = 0$. Instead, the spurious share of the output grows. By (d), a fraction $1 - 1/n$ of the merged update's energy lies in directions that had none, and the q genuine directions are attenuated from $\ge c$ to $\bar{\sigma} \le \sigma_1/n$. At small client drift the pool is near-degenerate, so factor-wise aggregation, which never forms the spurious directions, performs better. At large drift all N directions are real, and flattening is safe. This is the crossover of §5.

Scope. Note that the FINLORA pool itself is not in this band. After the PolEq client step, mutually orthogonal client directions would put the merged flat level at $m_F/\sqrt{r} = .354\, m_F$, and we measure $\bar{\sigma}/m_F = .318 \pm .005$ across the nine distinct PolEq sub-pool merges of Appendix I. So the pool after the client step is close to fully non-degenerate.

B.5 WHY THE CORRECTION IS PLACED ON THE B SIDE

Proposition 5. *Let $A \in \mathbb{R}^{r \times d_{\text{in}}}$ with $r \le d_{\text{in}}$ and write $\sigma_{\min}(A) = \sigma_r(A)$. For every $B \in \mathbb{R}^{d_{\text{out}} \times r}$,*

$$\sigma_{\min}(A)\,\|B\|_F \;\le\; \|BA\|_F \;\le\; \sigma_{\max}(A)\,\|B\|_F.$$

Proof. Take the thin SVD $A = U\Sigma V^\top$. Here U is $r \times r$ and orthogonal, Σ is the $r \times r$ diagonal matrix of singular values of A, and V has r orthonormal columns in $\mathbb{R}^{d_{\text{in}}}$, so that $V^\top V = I_r$. Then

$\|XV^\top\|_F^2 = \mathrm{tr}(VX^\top XV^\top) = \mathrm{tr}(X^\top X\, V^\top V) = \|X\|_F^2$ for every X, so $\|BA\|_F^2 = \|BU\Sigma\|_F^2 = \sum_{j=1}^r \sigma_j(A)^2\, \|(BU)_{:,j}\|_2^2$, which lies between $\sigma_{\min}(A)^2$ and $\sigma_{\max}(A)^2$ times $\sum_j \|(BU)_{:,j}\|_2^2 = \|BU\|_F^2 = \|B\|_F^2$. □

The inequalities do not need full row rank, but full row rank makes the lower bound informative. Note that the proposition is *not* needed for rescaling, because rescaling B_i by a scalar c rescales $\Delta W_i = B_iA_i$ by exactly c, with no condition number involved. It is needed for the converse, i.e., for reading $\|\Delta W_i\|_F$ off $\|B_i\|_F$ and equalizing one by equalizing the other, and this inference is cross-client. The binding constant is therefore a pool quantity. Setting $\|B_i\|_F$ equal for every i leaves the products spread by at most $\kappa_{\text{pool}} = \max_i \sigma_{\max}(A_i)/\min_j \sigma_{\min}(A_j)$, which is not $\max_i \mathrm{cond}(A_i)$ unless the A_i share a scale, and on this pool they do not.

What the pool measures, and what is at chance. The per-client median $\mathrm{cond}(A_i)$ ranges from 1.08 to 2.53, with median 1.32 over the eight clients. A fresh A has 1.07, which is not a client measurement. We measure $\kappa_{\text{pool}} = 3.8$ at the median weight matrix, and the realized cross-client spread of $\|\Delta W_i\|_F/\|B_i\|_F$ is 2.8. This is larger than any single client's $\mathrm{cond}(A_i)$, which the pool constant permits but a per-client one does not predict. Compared with the 708× spread of $\|\Delta W_i\|_F$, a factor of 3.8 is small. So a B-side equalization and a product-level one cannot differ by enough to explain the dominance, which is why B-side corrections in the literature address the same quantity as our product-level correction. Note that the A_i do not stay at their draws. The A drift is .312 in the independently trained regime (Table 14) and $\|A_{\text{sentiment}}\|_F = 3.81$ against a fresh draw's 1.63 (Table 11). What is at chance is the *cross-client overlap* of the A_i (.045 on FinLoRA, Table 17; .043–.048 in regime I, Table 14), and together with bounded conditioning this is all the argument needs. On a shared-initialization pool, where A contains structure common to the clients, the argument does not apply. This is consistent with the methods that aggregate A in that regime.

B.6 The Polar Step as a Projection

The polar step of PolCap and PolEq (§6) replaces ΔW_i by $(s_i/\sqrt{r})\, Q_i$, with $s_i = \nu_i$ for PolCap before OCap caps it and $s_i = m_F$ for PolEq, and $Q_i = U_iV_i^\top$ from the compact SVD $\Delta W_i = U_i\Sigma_iV_i^\top$, where $U_i \in \mathbb{R}^{d_{\text{out}}\times r}$ and $V_i \in \mathbb{R}^{d_{\text{in}}\times r}$. So $Q_i^\top Q_i = V_iV_i^\top$ is a rank-r orthogonal projector and *not* the identity. In other words, Q_i is a partial isometry and not a matrix with orthonormal columns. This distinction matters in practice. For the k and v projections $d_{\text{out}} = 1024$ and $d_{\text{in}} = 4096$, so 64 of the 96 adapted matrices admit *no* matrix with orthonormal columns at all, and the nearest-orthonormal-columns problem is posed over the empty set. The characterization below also fixes the scale, so it justifies both the scale and the direction of the polar step.

Proposition 6. *Let ΔW have compact SVD $U\Sigma V^\top$ with $\mathrm{rank}\,\Delta W = r$, and let $\mathcal{Q}_r = \{Q : Q^\top Q$ is an orthogonal projector of rank $r\}$. For every fixed $c > 0$, $Q = UV^\top$ is the unique minimizer of $\|\Delta W - cQ\|_F$ over $Q \in \mathcal{Q}_r$. Consequently, for every $t > 0$,*

$$\frac{t}{\sqrt{r}}\, UV^\top = \arg\min\Big\{\|\Delta W - X\|_F \;:\; \mathrm{rank}\, X = r,\ \sigma_1(X) = \cdots = \sigma_r(X) = \tfrac{t}{\sqrt{r}}\Big\},$$

and that constraint set is exactly $\{X : \mathrm{rank}\, X = r,\ X$ has a flat spectrum, $\|X\|_F = t\}$. Minimizing over $c > 0$ as well gives $c^* = \|\Delta W\|_*/r$, so $(\nu/\sqrt{r})\, UV^\top$ with $\nu = \|\Delta W\|_*/\sqrt{r}$ is the nearest matrix to ΔW with r equal nonzero singular values. The polar step takes $t = \nu_i$; OCap (Eq. 1) then lowers it to $\min(\nu_i, m_\nu + 3\,\mathrm{MAD}_\nu)$, and PolEq sets $t = m_F$.

Proof. For $Q \in \mathcal{Q}_r$ the eigenvalues of $Q^\top Q$ are r ones and the rest zeros, so $\sigma_1(Q) = \cdots = \sigma_r(Q) = 1$, $\|Q\|_2 = 1$ and $\|Q\|_F^2 = r$. Hence $\|\Delta W - cQ\|_F^2 = \|\Delta W\|_F^2 - 2c\,\langle \Delta W, Q\rangle + c^2 r$, and minimizing over $\mathcal{Q}_r$ is equivalent to maximizing $\langle \Delta W, Q\rangle$. Writing $\langle \Delta W, Q\rangle = \mathrm{tr}(V\Sigma U^\top Q) = \sum_{j\le r} \sigma_j\, u_j^\top Q v_j$ and using $|u_j^\top Q v_j| \le \|Q\|_2 = 1$ gives $\langle \Delta W, Q\rangle \le \sum_{j\le r}\sigma_j$, which $Q = UV^\top$ attains. For uniqueness, equality forces $u_j^\top Q v_j = 1$ for every $j \le r$; since $\|Qv_j\|_2 \le 1$ and $\|u_j\|_2 = 1$, Cauchy–Schwarz gives $Qv_j = u_j$ for each such j. Then $\mathrm{range}(Q^\top Q) \supseteq \mathrm{span}\{v_1, \ldots, v_r\}$, and as $Q^\top Q$ has rank r the two coincide, so $Q = QVV^\top = UV^\top$. The second display is the case $c = t/\sqrt{r}$, and $\|X\|_F = \sqrt{r}\cdot(t/\sqrt{r}) = t$ for every X in the set. At $Q = UV^\top$ the objective is $\|\Delta W\|_F^2 - 2c\|\Delta W\|_* + c^2 r$, minimized at $c^* = \|\Delta W\|_*/r = \nu/\sqrt{r}$. □

The factorization $\Delta W_i = Q_i P_i$ with $P_i = (\Delta W_i^\top \Delta W_i)^{1/2} = V_i \Sigma_i V_i^\top \succeq 0$ is a polar decomposition (Higham, 1986). Because ΔW_i is rank deficient the partial-isometry factor is unique only on $\mathrm{row}(\Delta W_i)$, and Q_i is the canonical choice supported on the row and column spaces. For this reason we call it "a polar factor" and not "the orthogonal factor". The polar step replaces P_i by its mean eigenvalue times the row-space projector and keeps only $\mathrm{tr}\, P_i = \|\Delta W_i\|_*$. As a result, within-client concentration is removed, while between-client magnitude remains in ν_i, which OCap then caps. We report the cost of forming Q_i in Appendix C.

C Rules, Identifiers, and the Precondition in Detail

Rules. Every rule is a client step followed by an operator (§3). Table 4 lists the rules named in the paper, with the identifiers used by the supplementary code and every appendix table (we drop the prefix `merged_`). Unlisted identifiers change a single constant of a listed rule and are named where they appear. In the appendix, "PolCap" alone denotes PolCap with Iso-CTS, the rule used for the robustness checks. With any other operator, we name the operator.

Where the constants come from. TIES and KnOTS-TIES keep 20% of entries and TSV-M uses output multiplier 1, the authors' defaults. DC-Merge runs with the settings of its public description, and we did not tune them. Iso-C and Iso-CTS run at k_c=32 with $(nr - k_c)/n$ task-specific directions and output multiplier 1. The authors default to $k_c = .8$ of the budget and tune the multiplier on validation data, which we do not have. Task arithmetic is the mean (λ=1 on the mean), which is the peak of a λ sweep (Appendix E). Every baseline that uses the scalar control shares γ=.5, and we swept γ only for Iso-CTS. The 3 MADs of OCap were fixed before the sweep of Appendix H, which was run under PolCap. We chose the constants of DO-Merging ourselves (below).

DO-Merging as we run it. Zheng et al. (2025) give the rule but not its optimizer or λ, and release no code. Per weight matrix, we perturb each client's factor V_i (A_i, orthogonalized across clients by rows, or B_i, by columns) to minimize $\sum_{i \neq j} \|(V_i + \delta_i)(V_j + \delta_j)^\top\|_F^2 + \sum_i \|\delta_i\|_F^2$ (their Eq. 4). We use Adam for 100 steps from δ=0 with learning rate .01 times the root-mean-square of the factors. On a sampled FinLoRA matrix, this lowers the mean cross-client overlap from .020 to .006 for A and from .057 to .024 for B, while moving each factor by 5–11%. We then take the per-column magnitudes α_i of $\Delta W_i = B_i A_i$ and its column-normalized $\bar{W}_i$, and we compute the output $\lambda(\sum_i \alpha_i) \odot \sum_i \bar{W}_i$ (their Eq. 5) exactly, in rank-nr factored form. Without validation data, we set $\lambda = 1/n^2$, which returns the mean when all clients coincide, and also report $\lambda = 1/n$. Both settings collapse on LoRA Land. On FinLoRA, $\lambda = 1/n$ scores at most .009 on every one of the twelve tasks (retention .001, n=500), and $\lambda = 1/n^2$ is in Table 1.

The two magnitude controls we add. The *scalar control* multiplies B_i by $c_i = \min\{1, (m_F / \|\Delta W_i\|_F)^\gamma\}$ per weight matrix, where m_F is the pool median of $\|\Delta W_j\|_F$. It leaves directions and spectra unchanged. It is a baseline device and not our method, although with DC-Merge's per-client flattening it becomes a soft analogue of PolCap. It gives every published operator a chance to function on a dominated pool, and its exponent must be chosen. The *share cap* scales down the heaviest clients until no client holds more than a fixed share of $\sum_i \|\Delta W_i\|_F$.

OCap and its variants. OCap is Eq. 1. For each weight matrix, it scales ΔW_i by $\min(1, (m + 3\,\mathrm{MAD})/n_i)$, where $n_i = \|\Delta W_i\|_F$ and m and MAD are the pool median and unscaled median absolute deviation of n_i. Both polar steps replace ΔW_i by $(t_i/\sqrt{r})Q_i$ and differ only in t_i. PolCap applies OCap to the flattened updates, i.e., $t_i = \min(\nu_i, m_\nu + 3\,\mathrm{MAD}_\nu)$ with $\nu_i = \|\Delta W_i\|_*/\sqrt{r}$, the Frobenius norm of DC-Merge's per-client flattening, and m_ν and MAD_ν are computed over the ν_j. PolEq sets $t_i = m_F$ for every client. When a client's update has numerical rank $k < r$ at some matrix, we replace $\sqrt{r}$ by $\sqrt{k}$ so that $\|\Delta W_i\|_F = t_i$ still holds.

Dominance, measured. The three heavy clients account for 99.5% of the energy of the summed update, i.e., the median over weight matrices of $\|S_H\|_F^2 / \|S\|_F^2$ is .995. The median share of summed per-client energies, $\sum_{i \in H} \|\Delta W_i\|_F^2 / \sum_i \|\Delta W_i\|_F^2$, is also .995.

Table 4: Rule taxonomy. Operators: task arithmetic (TA), TIES, TIES-trim (TIES without sign election), KnOTS-TIES and KnOTS-TA (entry-wise, after KnOTS alignment), Iso-C and Iso-CTS (isotropic), TSV-M and DC-Merge (cover-space; DC-Merge also trims entries in its cover space), asym-CTS (isotropic: a common output subspace of size k_c is taken from the clients' stacked B factors, each B_i keeps its component there and a fraction β of the rest, and the sum's spectrum is flattened). Every rule is run at rank budget nr=64 on FINLORA, except `slao`, `avg_factors` and `svd_prod8` (rank 8), `proj_a16` and `proj_b16` (16), `proj_b32` (32) and `ties128_g50` (128).

client step	operator	identifiers
none (published settings)	TA; FedAvg-weighted TA	`concat`; `concat_weighted`
	TIES, KnOTS-TIES, KnOTS-TA	`ties64`, `knots_ties`, `knots_ta`
	Iso-C, Iso-CTS, TSV-M, DC-Merge	`isoc`, `isocts`, `tsvm`, `dcmerge`
	DO-Merging (λ=$1/n^2$; $1/n$)	`domerging`; `domerging_l1n`
scalar control, γ=.5	Iso-C, Iso-CTS, TSV-M, DC-Merge, KnOTS-TIES, TIES; TIES-trim	`isoc_g50`, `isocts_g50`, `tsvm_g50`, `dcmerge_g50`, `knots_ties_g50`, `ties64_g50`; `trim_g50`
scalar control, other γ	Iso-CTS	`isocts_g55`, `_g60`, `_g70`, `_g100` (γ=1 with the minimum)
full equalization (γ=1)	Iso-CTS; TA; DC-Merge	`isocts_neq`; `normeq_med`; `dcmerge_neq`
share cap $\frac{1}{4}$	Iso-CTS	`isocts_cap25`
OCap (ours)	**TA**; Iso-CTS; DC-Merge	**`capmad_concat`**; `capmad_isocts`; `capmad_dcmerge`
	Iso-C, TSV-M, KnOTS-TIES, TIES (pre-registered test)	`capmad_isoc`, `_tsvm`, `_knots`, `_ties`
clipping at the median	TA	`capmed_concat`
shrinking the whole update to OCap's norm	TA	`concat_nm_capmad`
PolCap (OCap after the polar step)	Iso-CTS	`adaptive_mad`
	Iso-C, TSV-M, DC-Merge, KnOTS-TIES, TIES, TA	`adaptive_mad_isoc`, `_tsvm`, `_dcmerge`, `_knots`, `_ties`, `_concat`
fixed-cap variants of PolCap (τm)	Iso-CTS	`adaptive_t1`, `adaptive` (τ=3), `adaptive_t10`; `adaptive_tinf` (τ=∞, cap removed)
PolEq	Iso-CTS; Iso-C; TA	`isocts_white`; `isoc_white`; `concat_white`
	TSV-M, DC-Merge, KnOTS-TIES, TIES	`tsvm_white`, `dcmerge_white`, `knots_white`, `ties_white`
	asym-CTS (k_c, β=.5)	`asym_cts` (32), `asym_cts_k16`, `_k48`, `_k56`; `asym_shared_only` (β=0)
a client step, then rescaled to the norm of task arithmetic after the same client step	Iso-CTS (absolute anchor)	suffix `_nm`: `isocts_neq_nm`; `isocts_white_nm` (built by `scripts/heavy_pool_fix.py`)
floors: unmerged base; best single adapter		`base`; `client_xbrl_term`

What the bound does on each pool. On FINLORA, the bound of OCap has median $3.31\,m$ over weight matrices. It binds on sentiment in 96 of 96 matrices, on headline in 68, on ner in 17, and never on the other five clients. The bound of PolCap has median $3.23\,m_\nu$ and binds on the same three clients in 96, 77 and 27. Table 5 follows the spread through the client step. Flattening alone (DC-Merge's first step) barely changes the spread, while the cap reduces it from $604\times$ to $13.0\times$. OCap leaves nearly the same spread and energy shares as PolCap, so the two caps differ only in the within-client shape they keep. On MM-MergeBench, the bound of OCap binds on 0.20 of the 8 clients per weight matrix and that of PolCap (median $2.18\,m_\nu$) on 0.42, compared with 1.89 and 2.08 on FINLORA.

Table 5: Client magnitudes on FINLORA through the client step. Mean and median spread as defined in §3; per-matrix ratio is the median over the 96 weight matrices of the largest over the smallest $\|\Delta W_i\|_F$; shares are of $\sum_j \|\Delta W_j\|_F^2$, medians over matrices, for the largest client and for the five light clients.

pool state	mean spread	median spread	per-matrix ratio	largest share	light share
public adapters	708×	128×	108×	.953	.005
per-client flattening only (DC-Merge step 1)	604×	118×	101×	.958	.004
OCap (no flattening)	15.1×	15.4×	14.9×	.343	.082
PolCap	13.0×	13.2×	13.0×	.334	.087

Merged magnitude. At the first layer's query projection the merged update has Frobenius norm 412 under task arithmetic, 1149 under Iso-C and Iso-CTS, 1583 under DC-Merge, 1946 under KnOTS-TIES, and 3263 under TSV-M as published. It is 1.96 under PolCap, 1.37 under PolEq, and 0.86 for DC-Merge under full equalization.

Cost. With thin QR factorizations $B_i = Q_B R_B$ and $A_i^\top = Q_A R_A$ and the SVD $R_B R_A^\top = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$ of an $r \times r$ matrix, $Q_i = (Q_B\tilde{U})(Q_A\tilde{V})^\top$ is kept in factored rank-r form. The step costs $O((d_{\text{out}} + d_{\text{in}})r^2)$ per client and weight matrix. This is well below the $O(d_{\text{out}}d_{\text{in}}r)$ cost of forming ΔW_i, and the step does not need to form it, because n_i follows from two $r \times r$ Gram matrices and ν_i from the $r \times r$ SVD. Our implementation still forms ΔW_i once per client and weight matrix in float32 to compute the norms (relative error of a norm below 10^{-4}).

D PROTOCOL, REFERENCE POINTS, AND CHECKS

Reference points. All are scored with the same prompts, decoding, and scorer as the merges (Table 6). We choose the *best single adapter* on transfer, where `client_xbrl_term` leads the eight clients (.5250 at n=200, all nine models scored under one answer-cue version; .5015 at n=2000). We keep it fixed on every axis, so the retention floor is not a maximum over clients. `client_xbrl_extract` (.4633) and `client_finer` (.4520) score higher on retention because their own tasks are scored. The other rows come from the 8×12 expert matrix, in which every client is scored alone on every task. *Random client* is the mean of the eight clients' means. The *home expert* and *oracle routing* ceilings need the task identity at inference time. They differ on one task, `formula`, where `client_xbrl_term` (.425) beats the client trained on it (.360).

Table 6: Floors on the public FINLORA pool at n=2000 requested items per task, raw prompt format; a task with fewer test items uses all of them, and each `xbrl_finer` prompt carries four tag queries, so its items are counted in tags. The first two rows come from the main pool run (`results/e1_pool2000`); the rest from the 8×12 expert matrix (`results/e8_2000`), in which every client adapter is scored alone on every task. `class6` and `xbrl6` are the six classification and six extraction tasks. The matrix is not printed row by row; the lone-client means are in the text. Two clients own only a free-text task, which carries no accuracy and enters no mean here. `client_xbrl_term` appears in both runs, on identical items in identical order; its two 12-task means differ by .0006.

floor	source	retention	class6	xbrl6
unmerged base	main pool run	.2876	.130	.446
best single adapter (`client_xbrl_term`)	main pool run	.3935	.370	.417
random client (mean of 8)	expert matrix	.3283	.297	.359
home expert (owner of each task)	expert matrix	.8225	.833	.812
oracle routing (per-task best client)	expert matrix	.8279	.833	.823

The prompt-format probe. Table 7 and Figure 4 report every model measured in both formats on one item draw. We fixed the rule (each floor at its stronger format) before the numbers existed, and it works against us, since the base's retention rises from .2776 to .4476. Five of the eleven probed merges change retention rank between formats, and all of them use an isotropic operator. The entry-wise and cover-space rules keep their ranks. The largest per-task changes of the base are

`headline` (+.598), `fiqa` (+.549) and `fpb` (+.514). We have no chat-format run at n=2000. OCap with task arithmetic was not run under the chat template. We therefore score its n=2000 generations on the same n=500 items as the chat-template base, and it is +.098 [.085, .111] above that floor over all twelve tasks (483 of the 500 `xbrl_finer` items match). The same computation gives PolCap +.206, compared with +.202 from its own n=500 run.

Table 7: Prompt-format probe, all thirteen models measured in both formats, n=500 per task. "Raw" is the benchmark's own prompt (`results/e1_pool500`, `results/e1_heldout500`); "chat" wraps it in the base checkpoint's own chat template, one user turn plus a generation prompt (`results/chat_pool500`, `results/chat_heldout500`). Every entry carries its directory's format stamp. The item draw is identical in the two formats (checked on the item identifiers of all 195 model–task pairs). Merges are sorted by raw retention; the two floors follow.

	retention (12 tasks)			transfer (3 suites)		
model	raw	chat	Δ	raw	chat	Δ
`asym_cts_k56`	.6853	.6954	+.0101	.5167	.4367	−.0800
`isocts_cap25`	.6852	.7044	+.0191	.5000	.3887	−.1113
`isoc_white`	.6829	.6924	+.0095	.5120	.4373	−.0747
`asym_cts_k48`	.6805	.7016	+.0211	.5080	.4200	−.0880
`isocts_g60`	.6775	.6933	+.0158	.5073	.4067	−.1007
`isocts_white`	.6671	.7005	+.0335	.4993	.3913	−.1080
`knots_ties_g50`	.6651	.6907	+.0256	.4400	.3427	−.0973
`adaptive_mad`	.6498	.6768	+.0270	.4780	.3153	−.1627
`trim_g50`	.6310	.6462	+.0152	.4780	.3613	−.1167
`tsvm_g50`	.6016	.6361	+.0345	.3480	.3340	−.0140
`concat`	.5573	.5386	−.0187	.4453	.3133	−.1320
`client_xbrl_term`	.3842	.5004	+.1162	.4940	.3400	−.1540
`base`	.2776	.4476	+.1700	.4553	.2600	−.1953

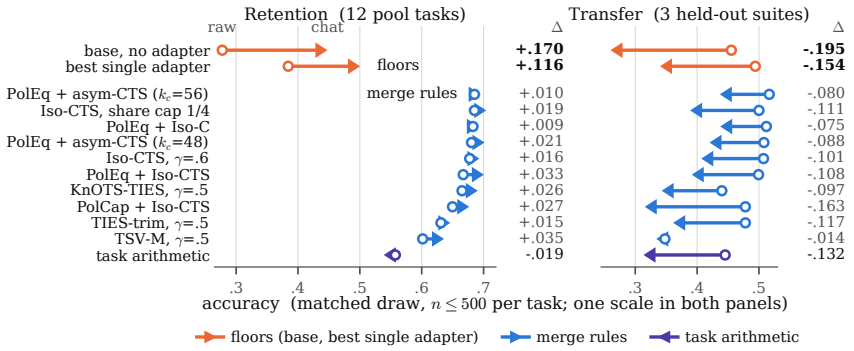


Figure 4: The prompt-format probe. Each marker pair connects the raw-format and chat-format scores of one model. On retention, the floors move sharply while every merge barely moves. On transfer, the chat template lowers every model.

Task-balanced intervals. Retention is an unweighted mean over twelve tasks, so its interval must weight tasks equally. `xbrl_finer` scores four tags per paragraph and contributes 8,000 of the 23,317 scored rows at n=2000 (34.3%, against its 8.3% share of the mean). `formula` and `fiqa` together contribute under 2%. For a rule that trades classification for extraction, pooling can flip the verdict. For example, PolEq vs. KnOTS-TIES under the scalar control is +.000 [−.007, +.007] task-balanced and −.027 [−.033, −.021] pooled over examples. We therefore resample items within each task and average the per-task mean differences with equal weight, using 2000 resamples. Transfer uses the same construction over three suites. The pooled binomial SE $\frac{1}{T}\sqrt{\sum_t p_t(1-p_t)/n_t}$ depends on the model. At n=2000, it spans .0030–.0049 on retention and .0017–.0071 on transfer.

The noise floor, measured three ways. (i) We evaluated `client_xbrl_term` twice at n=2000 on identical items in identical order. Of the 17,317 generations, 88.1% are byte-identical, the largest per-task shift is .0050, and the 12-task means are .3935 and .3929. (ii) We ran PolEq with Iso-CTS twice on `xbrl_finer` at n=200 (Table 8). The two runs are 76.0% byte-identical and 1.5 points

apart. This pair was queued as a batch-size contrast, but it is not one because the runner fixes the batch size at 4 for every XBRL task. (iii) A forced re-run of one of the six leave-one-out arms moved its mean by .0019. All three checks hold the items and their order fixed. The likely cause is 8-bit matrix multiplication, but we ran no full-precision twin. Hence the rule of §3.

Table 8: Run-to-run reproducibility: the same merged adapter and prompts, greedy decoding, run twice (`results/repro_bs1`, `results/repro_bs8`). The runs were queued as batch sizes 1 and 8, but the runner fixes the batch size at 4 for every XBRL task, so both ran in one configuration. "Identical" counts generations that are byte-for-byte equal, aligned on the item index.

model	task	prompts	labels	acc., run 1	acc., run 2	Δ	identical
`isocts_white`	`xbrl_finer`	200	800	.6025	.6175	+.0150	152/200 (76.0%)

A second item draw. Redrawing the items with seed 7 moves retention by $-.0021$ to $+.0081$ over seven models at n=500 and by $-.0017$ to $+.0029$ over three at n=2000, and the two leading models swap places in both blocks (Table 9). This understates the instability, because tasks smaller than the requested sample are scored in full under both seeds.

Table 9: Second evaluation draw: retention under the seed-42 item draw used everywhere else and under a seed-7 redraw (`results/e1_pool500_seed7`, `results/e1_pool2000_seed7` against `results/e1_pool500`, `results/e1_pool2000`). Ranks are within the block. The pooled binomial SE of one retention mean is .0057 and .0045 in the two blocks.

model	seed 42	rank	seed 7	rank	Δ
n=500, seven models, 10 of 12 tasks redrawn					
`asym_cts_k56`	.6853	1	.6857	2	+.0005
`isocts_cap25`	.6852	2	.6877	1	+.0025
`isoc_white`	.6829	3	.6808	5	−.0021
`asym_cts_k48`	.6805	4	.6818	4	+.0013
`isocts_g60`	.6775	5	.6832	3	+.0057
`isocts_white`	.6671	6	.6752	6	+.0081
`concat`	.5573	7	.5618	7	+.0045
n=2000, three models, 7 of 12 tasks redrawn					
`asym_cts_k56`	.6872	1	.6882	2	+.0011
`isocts_cap25`	.6868	2	.6897	1	+.0029
`adaptive_mad`	.6586	3	.6569	3	−.0017

Decontamination. We drop a test item if its normalized context appears in any client's training data or repeats an earlier item. Nearly all removals come from two tasks. The `headline` task loses 11.2% of its test split to training overlap, and `ner` loses 16.0% to repeats. The three transfer suites have no training overlap. The filter runs before subsampling, so on a large task it draws replacement items. PolCap moves by $-.0016$ on retention and $-.0027$ on transfer. Two pairs swap rank, each within .005: `isocts_cap25` and `isoc_white` at n=500, and PolCap and `adaptive` (PolCap with a fixed cap τ=3) at n=2000.

The label prior of `nwgi`. `nwgi` is a 7-way ordinal sentiment task with a skewed gold distribution. Always answering the majority label scores .3830, which beats most of the grid, including PolCap. Accuracy and weighted F1 disagree on the ordering below the top rule (Table 10). We keep the task in every mean, because dropping it after seeing which rules it favours would be a choice made on the test set.

Scoring. Every language model, including the capability runs, runs on the same 8-bit base (the multimodal pool runs in fp16). Therefore, comparisons are unaffected by quantization, but absolute numbers are not comparable to full-precision ones. FinQA and ConvFinQA compare the first generated number to the gold answer within 1% relative tolerance and accept x, $x/100$ and $100x$, since the gold answers mix percentage scales. FinExam-10K takes the first standalone letter A–D. Two pool tasks, `xbrl_term` and `financebench`, declare BERTScore as their metric. We generate them but exclude both from every mean, which is why retention has twelve tasks and two clients have no scored task. ECTSum uses ROUGE-Lsum over 495 documents.

Table 10: `nwgi` at n=2000, every model in the main pool run (`results/e1_pool2000`), sorted by accuracy. The task is 7-way ordinal sentiment with a skewed gold distribution: always answering "moderately positive" scores .3830 (766/2000 gold labels, computed from the saved targets, which are identical for every model). The rule inside the table marks that constant; 40 of 46 models fall below it. F1 is support-weighted. In three columns: read down each column in turn.

model	accuracy	weighted F1
tsvm_g50	.5835	.5667
concat_weighted	.5145	.4973
knots_ties_g50	.4895	.4341
ties64_g50	.4435	.4020
knots_ta	.3940	.3964
dcmerge_g50	.3835	.5318
constant: .3830 / .2121		
adaptive_mad_dcmerge	.3730	.5062
concat	.3720	.3785
capmad_dcmerge	.3650	.5060
knots_ties	.3345	.3927
adaptive_mad_concat	.3210	.4515
adaptive_t1	.3135	.4582
client_xbrl_term	.2890	.3701
adaptive	.2875	.3952
adaptive_mad_k2	.2860	.3960
isocts_neq_nm	.2790	.3893
capmad_concat	.2695	.3716
isocts_white	.2695	.3760
trim_g50	.2595	.2491
adaptive_mad	.2515	.3094
domerging	.2425	.2108
isocts_g70	.2385	.3204
isocts_g50	.2325	.1714
isocts_g55	.2305	.1820
isocts_g60	.2305	.1993
asym_cts	.2080	.2809
isocts_neq	.2075	.2878
asym_cts_k16	.2060	.2939
capmed_concat	.2060	.3263
isocts_cap25	.2060	.1883
capmad_isocts	.1965	.1548
isoc_white	.1955	.1469
asym_cts_k48	.1950	.1905
asym_cts_k56	.1900	.1482
isoc_g50	.1670	.1113
isocts_g100	.1635	.2626
asym_shared_only	.1485	.1779
adaptive_t10	.1460	.0914
concat_nm_capmad	.1430	.2335
normeq_med	.1420	.2376
base	.0800	.1370
adaptive_tinf	.0000	.0000
dcmerge	.0000	.0000
isoc	.0000	.0000
isocts	.0000	.0000
tsvm	.0000	.0000

E CLIENT DOMINANCE IN FULL

We recompute norms from the public adapters with α/r folded into B. Several diagnostics exist only at the n=100 tier (transfer at n=200), so we state the tier with every score. At that tier, the pooled SE is at most .013 on retention and .020 on transfer.

Per-client norms. Table 11 gives the norms behind the spreads of §4. The mean exceeds the median because the three heavy clients peak at the first layer's query projection. They are exactly the α=32 clients. α accounts for a factor of two and the trained B for the rest. A freshly drawn A has $\|A\|_F = 1.63$ and condition number 1.07, and the two lightest clients are within .02 of it on both. Dominance is also not a first-layer artifact. Clipping each client's weight matrices at $10\times$ ($3\times$) its own median norm caps sentiment's peak at 103 (31), but it moves task arithmetic only from .5540 to .5788 (.5706) at n=100, vs. .6458 for PolCap. The median spread of $128\times$ remains after the clip.

Table 11: Per-client norms on FINLORA. $\|A\|_F$, cond(A), $\|B\|_F$ and the median of $\|\Delta W\|_F$ are medians over the 96 weight matrices; *mean* is the mean of $\|\Delta W\|_F$ over them and *layer-0 q* its value at the first query projection. B includes α/r.

					$\|\Delta W\|_F$	
client	α	$\|A\|_F$	cond(A)	$\|B\|_F$	median	mean	layer-0 q
formula	16	1.64	1.08	0.14	0.08	0.10	0.14
financebench	16	1.65	1.08	0.15	0.09	0.10	0.15
xbrl_extract	16	1.68	1.16	0.32	0.20	0.25	0.28
xbrl_term	16	1.70	1.21	0.36	0.26	0.30	0.33
finer	16	1.82	1.46	0.65	0.51	0.64	0.73
ner	32	1.79	1.43	1.35	1.07	10.5	879
headline	32	1.89	1.65	2.08	1.93	14.0	634
sentiment	32	3.81	2.53	5.18	10.3	73.0	3077

The spread is a recipe artifact. Our sentiment client from the shared-initialization regime (α=16, 10,000 rows) has mean and median norms 0.62 and 0.50, vs. 73.0 and 10.3 for the public client. On FPB it scores .87 vs. .80 for the public client (n=100; the latter scores .838 at n=1212), so we claim only that the light client is no worse. More data does not close the gap, since our recipe on all 72,824 rows gives a mean norm of 2.27, still $32\times$ lower. We do not isolate which other recipe choice is responsible. The heavy client also reduces capability on its own, with MMLU .325, GSM8K strict .000 and HellaSwag .485 at harness limit 200, vs. .682/.680/.705 for the base.

Scale alone does not explain the collapse. At the first query projection, the published operators produce 2.8–7.9× the merged norm of task arithmetic (Appendix C), and task arithmetic at λ=8 matches TSV-M's norm to within 1%. However, Iso-C and Iso-CTS at 2.8× score .0000, while task arithmetic at λ=2 and 4 still scores .4665 and .2608. The difference is that these two flatten a spectrum set by the heavy subset (Prop. 3), which task arithmetic does not do. DC-Merge flattens each client at its own mean singular value, which barely changes a ratio between clients (median spread 128× → 118×, Table 5).

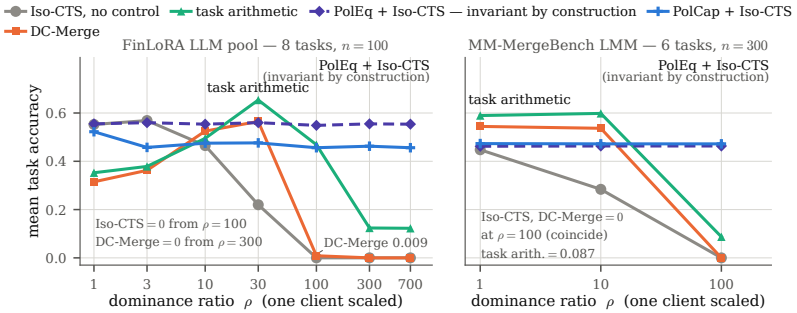


Figure 5: The dominance sweep. The uncontrolled spectral operators fall to at most .009 by ρ=100 on both pools and to .000 by ρ=300; task arithmetic first recovers the scaled client's tasks and then fails. PolCap, whose cap clips the scaled client, does not collapse in the measured sweep. The PolEq line is dashed because it is invariant by construction.

How the sweep is built. The steps are the same on both pools. (i) At every weight matrix, we rescale each B_i so that $\|\Delta W_i\|_F$ equals the pool median (full equalization). (ii) We multiply the heaviest client's B by ρ (sentiment on FINLORA, OCRVQA on MM-MergeBench), so its update is exactly ρ times that of every other client and no direction changes. (iii) Each operator runs with its registered arguments at rank 64 (language) or 128 (multimodal). (iv) On FINLORA, $\rho \in \{1, 3, 10, 30, 100, 300, 700\}$, ending at the measured spread, with eight tasks scored at n=100. On MM-MergeBench, $\rho \in \{1, 10, 100\}$, with six tasks scored at n=300 (Tables 12 and 13). The ρ=1 point is the equalized pool and not the natural one, and there uncontrolled Iso-CTS is identical to `isocts_neq`. The flat line of PolEq moves by at most nine correct answers in 800, which is generation noise. PolCap is not invariant by construction. On FINLORA it falls from .522 at ρ=1 to .458 at ρ=3, where the cap begins to bind on the scaled client and keeps it no higher than m+3 MAD of ν. It then stays within .02 up to ρ=700. On MM-MergeBench it reads .473/.472/.472. There, uncontrolled Iso-CTS and the two polar rules all keep k_c=32 at rank 128. However, uncontrolled Iso-CTS keeps its registered 4 task-specific directions per client, while PolEq and PolCap derive $(128 - 32)/8 = 12$, so the rules also differ in this constant.

Reading the sweep. Every uncontrolled operator ends at or below .123, but where it turns down depends on the operator, and task arithmetic is non-monotone. Four of the eight scored tasks are the scaled client's. At ρ=1 they average .240, because every client enters the uniform mean at one eighth of a median-norm update. Raising ρ restores these tasks (.795 at ρ=100), while the other four rise to .633 at ρ=30 and then fall to .143 and, at ρ=300, to .000. DC-Merge rises from .315 to .565 before collapsing for the same reason. On MM-MergeBench, the ordering and the collapse replicate. At $\rho \leq 10$, task arithmetic leads PolEq by .13–.14 strict and PolCap by .12–.13. Therefore, robustness to dominance does not make a rule superior.

The scalar control on DC-Merge in full. On FINLORA, DC-Merge as published scores .0000 at n=100 and n=2000. It scores .4029 under full equalization (n=100), .4546 and .4760 under the scalar control (n=100, 2000), and .4829 under the PolEq client step (n=500). The control also repairs transfer (.0283 as published vs. .4833 with the scalar control). On MM-MergeBench, DC-Merge is the best operator on seen tasks (56.7 as published, 56.9 under the scalar control, 56.8 under PolCap). Full equalization gives 54.0 normalized and 48.8 strict, a cost of 2.7 or 7.9 points depending on the scorer. Even after the repair, DC-Merge stays below task arithmetic (.4760 vs.

Table 12: Dominance sweep on the FINLORA language pool (Llama-3.1-8B-Instruct), all values. Every client is first equalized to the per-matrix median norm; then one client (`sentiment`) is scaled by ρ and the pool is merged. Score is the mean accuracy over the same 8 tasks (`fpb`, `fiqa`, `tfns`, `nwgi`, `headline`, `ner`, `xbrl_tags_extract`, `formula`), n=100 items per task. The last column is the range of the printed cells. The `isocts_white` row is ρ-invariant by construction, so its spread is a noise floor for the evaluation and not a response to ρ: the merged adapters at ρ=3 and ρ=300 are byte-identical and score .560 and .555. Source: `rho_pool`.

rule	identifier	ρ=1	3	10	30	100	300	700	range
PolCap + Iso-CTS	`adaptive_mad`	.522	.458	.475	.476	.456	.463	.456	.066
PolEq + Iso-CTS	`isocts_white`	.555	.560	.554	.560	.549	.555	.554	.011
TIES-trim, γ=.5	`trim_g50`	.544	.550	.551	.573	.610	.641	.551	.097
task arithmetic	`concat`	.352	.379	.495	.654	.469	.124	.122	.532
DC-Merge (default)	`dcmerge`	.315	.362	.525	.565	.009	.000	.000	.565
Iso-CTS (default)	`isocts`	.552	.569	.465	.220	.000	.000	.000	.569

Table 13: Dominance sweep on the MM-MergeBench multimodal pool (LLaVA-1.5-7B), all values. The construction and the columns are those of Table 12, with `OCRVQA` as the scaled client. Score is the mean accuracy over the same 6 tasks (`ScienceQA`, `VQAv2`, `OCRVQA`, `IconQA`, `AOKVQA`, `ImageNet-R`), n=300 items per task. Accuracy is under the strict scorer; the normalized means are in Appendix I. The `isocts_white` row is ρ-invariant by construction, so its spread is a noise floor for the evaluation and not a response to ρ: the merges at ρ=10 and ρ=100 give the same accuracy on every task. A dash marks a rule that was not run on this pool. Source: `lmm_rho`.

rule	identifier	ρ=1	10	100	range
PolCap + Iso-CTS	`adaptive_mad`	.473	.472	.472	.001
PolEq + Iso-CTS	`isocts_white`	.462	.463	.463	.001
TIES-trim, γ=.5	`trim_g50`	–	–	–	–
task arithmetic	`concat`	.589	.598	.087	.511
DC-Merge (default)	`dcmerge`	.544	.537	.000	.544
Iso-CTS (default)	`isocts`	.448	.284	.000	.448

.5638 at n=2000). So the control determines whether the operator functions, but it does not make it competitive.

Is task arithmetic fairly scaled? Task arithmetic is usually reported with a tuned global λ, but we use λ=1 because there is no validation set. At the n=100 tier, $\lambda = .5, 1, 2, 4, 6, 8$ gives retention .5292, .5540, .4665, .2608, .0242, .0000 and transfer .5117, .4667, .2067, .0467, .0017, .0000. Retention peaks at λ=1. λ=.5 gives up .025 of retention for .045 of transfer. It then matches PolCap on transfer (.5117 vs. .5100) but trails it by .117 on retention. Every $\lambda > 1$ loses on both axes.

F THE CONTROLLED INITIALIZATION STUDY

Table 14: Controlled initialization study. Only A_0 and the learning rate on A differ across regimes. Overlaps are mean principal-angle cosines pooled over the $q/k/v$ projections; chance is .037–.073, and every displacement overlap is at least $+18.9\sigma$ above it. `I2` is a second independent draw.

regime	A_0 (A learning rate)	A drift	A overlap	B overlap	displacement overlap
`S_fz`	shared (frozen)	.000	1.000	—	—
`S_lr01`	shared ($\times$.1)	.046	.998	—	—
`S`	shared (full)	.303	**.921–.937** ($+268\sigma$)	.182–.229 ($+21\sigma$)	.171–.176
`I`	independent (full)	.312	**.043–.048** ($+1.8\sigma$)	.179–.220 ($+18\sigma$)	.170–.179
`I2`	independent (full)	.80 (.23–.97)	**.042–.044** ($+1.5\sigma$)	.118–.187 ($+7\sigma$)	.099–.108

This appendix expands §5. Each client stores its own A_0, and the supplementary material records the seed that regenerates it. So every number is measured against its own starting point.

Recipe. We follow FINLORA's training template (Wang et al., 2025a): Llama-3.1-8B-Instruct in 8-bit with bfloat16 compute; LoRA with r=8, α=16, dropout .05, on the query, key, and value projections of all 32 layers (96 weight matrices); AdamW at learning rate 10^{-4}, cosine schedule, 10 warm-up steps, no weight decay, batch 8 with gradient accumulation 2, four epochs. We disclose two departures. The sequence budget is 1024 instead of 4096, which never binds because the longest training pair is 508 tokens, and AdamW is full-precision instead of 8-bit. We use α=16, the light-group value of the public pool, although the public sentiment, headline, and ner clients use α=32. Training sets come from the benchmark: sentiment (72,824 rows), headline (82,161) and ner (13,549). One data seed (42) fixes the subsample, the shuffling, and the dropout. A separate seed fixes the A_0 draw and nothing else. Since $B_0 = 0$, every B overlap is learned. Training the fifteen adapters took 49 GPU-hours.

The five regimes. `S_fz` freezes A at A_0, as FFA-LoRA does (Sun et al., 2024). `S_lr01` scales the learning rate of A by .1. `S` runs the full recipe from a shared A_0, as a server broadcast would give (Guo et al., 2025). `I` runs it from independent draws, as in a public pool (Wang et al., 2025a). `I2` is a second independent draw (Table 15).

Table 15: The fifteen trained adapters, listed sentiment / headline / ner within each regime. Drift is $\|A - A_0\|_F / \|A_0\|_F$; energy ratio is $\|B(A - A_0)\|_F^2 / \|BA\|_F^2$ (we call it a ratio and not a share because $B(A - A_0)$ and BA_0 are not orthogonal). Both are medians over the 96 weight matrices.

regime	A_0 seed	rows	train loss	drift	energy ratio
`S_fz`	1000	10,000 / 10,000 / 13,549	.2659 / .1004 / .0336	.000 / .000 / .000	.000 / .000 / .000
`S_lr01`	1000	10,000 / 10,000 / 13,549	.2400 / .0884 / .0301	.058 / .046 / .044	.009 / .006 / .006
`S`	1000	10,000 / 10,000 / 13,549	.2006 / .0729 / .0231	.423 / .303 / .243	.378 / .260 / .193
`I`	2002 / 2003 / 2001	10,000 / 10,000 / 13,549	.2019 / .0730 / .0229	.428 / .312 / .222	.371 / .263 / .147
`I2`	101 / 102 / 103	72,824 / 82,161 / 13,549	.1114 / .0309 / .0231	.969 / .797 / .226	.728 / .662 / .163

The overlap statistic. For a client pair, a layer, and a projection we take orthonormal bases of each subspace—row space of A, column space of B, row space of $A - A_0$—and report the mean singular value of $Q_i^\top Q_j$, averaged over the three pairs and the 32 layers. Chance is the same statistic on 200 pairs of random Gaussian subspaces of matching shape. It is $.0367 \pm .0033$ in ambient dimension 4096 (all A and displacement subspaces, and query B) and $.0732 \pm .0067$ in dimension 1024 (key and value B). Table 14 leaves the B and displacement cells of the two low-drift regimes blank, but these regimes behave like the others. `S_fz` and `S_lr01` give B overlap .202–.261, and `S_lr01` gives displacement overlap .247–.258. Note that `S_fz` has no displacement, which is also why its A overlap is 1.000 by construction.

Disclosure: `I2` is not a matched seed. Its sentiment and headline clients were trained on the full training sets, while their `I` counterparts use a 10,000-row subsample. So they differ in data volume as well as in A_0, and volume drives drift (.969 and .797 vs. .428 and .312). The ner pair is matched (.222 and .226). Therefore `I2` is a replication of A at chance at two to three times the drift, and it is not a variance estimate. The confound runs in the direction that strengthens the claim, because A overlap stays at chance while drift reaches .97. No merges were built on `I2`.

The crossover is a property of the operator and not of A. Table 16 gives both sides at both sample sizes. The product-level rule is PolEq with Iso-CTS at the budget that a three-client pool supports (rank 24, k_c=12). At n=500, with task-balanced paired bootstraps over the six tasks, factor-wise aggregation minus that rule is $+.122$ $[+.101, +.142]$ at drift .046 (`S_lr01`), $-.117$ $[-.136, -.099]$ at drift .303 (`S`) and $-.052$ $[-.072, -.032]$ from independent starts (`I`). The win at `S_lr01` does not show that A holds much information there, since the energy ratio of the displacement of A is .006–.009 (Table 15). The premise of Prop. 3 can be measured. After the PolEq client step, the twelfth singular value of the sum, which is the last one the common block reaches, is .698 of the first at `I`, .120 at `S` and .018 at `S_lr01` (medians over matrices). So the low-drift pool is nearly degenerate. At zero drift the sign is not stable across sample sizes ($+.003$ at n=100, $-.023$ at n=500), so the crossover vanishes there and does not reverse. A public pool ships no A_0, so we cannot measure where it sits relative to the crossover. Its A overlap only tells us whether its clients shared a start.

Table 16: Factor-wise against product-level aggregation in the controlled initialization study. Each pool holds three clients we trained ourselves (sentiment, headline, NER) under one regime: `S_fz` = shared A_0, A frozen; `S_lr01` = shared A_0, A learning rate $\times 0.1$; `S` = shared A_0, full recipe; `I` = independent A_0, full recipe. Factor-wise is the average of the A and of the B factors (`avg_factors`); product-level is PolEq + Iso-CTS at the rank budget of a three-client pool (`isocts_white_s3`). Scores are the unweighted mean accuracy over the six classification tasks the three clients own. Difference is factor-wise minus product-level, bold where $|\cdot| \geq .1$. The last two columns add task arithmetic and TIES-trim under the scalar control where those merges were evaluated; a dash marks a merge not evaluated at that n. The n column is the number of items per task. Source: `n5_pool` and `n5_pool500`.

regime	n	factor-wise	product-level	difference	`concat`	`trim_g50`
`S_fz`	100	.498	.495	+.003	.465	.492
	500	.489	.512	−.023	–	–
`S_lr01`	100	.495	.360	**+.135**	.433	.377
	500	.481	.358	**+.122**	.428	.379
`S`	100	.403	.522	**−.118**	.417	.472
	500	.411	.528	**−.117**	.426	.489
`I`	100	.485	.548	−.063	.410	.495
	500	.500	.553	−.052	–	–

Flattening explains only part of the crossover. From `S` to `S_lr01`, PolEq with Iso-CTS gives up 16.2 points (n=100). TIES-trim under the scalar control, which does not flatten, gives up 9.5 (.472 → .377). In contrast, task arithmetic is flat (.417 → .433) and factor-wise aggregation rises (.403 → .495). Therefore roughly 7 points are specific to flattening. We report flattening as a contributor of that size, and not as the cause.

These accuracies are not comparable to the public pool. The controlled pool has three clients instead of eight, and it scores only the six classification tasks of those clients instead of all twelve. It also uses a 10,000-row subsample of sentiment and headline in the four primary regimes and trains at α=16. All of this is identical across regimes, so comparisons within the study hold. However, its numbers must not be set against public-pool numbers.

The diagnostic on public pools. Table 17 gives the main pool and the three further public pools of §5. Only the hub pool, whose clients were fine-tuned from one broadcast A_0, shows A overlap above chance. The three separately trained pools are at chance on A on three base models, while B is above chance on all four. This shows that spread depends on how a pool was assembled and not on its base model.

Table 17: The initialization diagnostic on public pools. $\mathrm{sim}\, A$ is the mean cross-client overlap of the A factors against a matched random baseline (chance); σ counts excess over that baseline, averaged over the attention projections the diagnostic reads for each pool. Spread is the median spread over those matrices; over all 224 adapted matrices of MM-MergeBench it is the $4.4\times$ used in Appendix I.2. *All 27 hosted files, including one byte-identical duplicate. On the 26 distinct clients (§7.3) the median spread is $52\times$, and the duplicate pair moves the mean A overlap by under .001.

pool	base model	n	r	spread	sim A (chance)	A excess	B excess
FINLORA (main pool)	Llama-3.1-8B	8	8	128×	0.045 (0.037)	$+3\sigma$	$+14\sigma$
LoRA Land hub*	Mistral-7B	27	8	51.8×	0.796 (0.037)	$+230\sigma$	$+10\sigma$
MergeMedBench	Qwen3-VL-2B	8	16	1.4×	0.077 (0.074)	$+1\sigma$	$+58\sigma$
MM-MergeBench	LLaVA-1.5-7B	8	16	4.8×	0.054 (0.053)	$+1\sigma$	$+12\sigma$

G FULL RESULTS

Every table here is generated from the saved metrics, generations, and bootstrap output by the scripts in the supplementary code, and uses the identifiers of Table 4. Scores are printed to four decimals

and rounded to three in the main text. Note that a difference is computed before rounding, so it can differ from the difference of two printed scores in the last place.

Every model at n=2000. Table 18 lists the 44 merges and two floors with a complete twelve-task record, together with the classification (class6) and extraction (xbrl6) halves, which trade against each other. For example, KnOTS-TIES under the scalar control and PolEq with Iso-CTS tie on retention (.6703, .6704) but have opposite profiles. OCap with Iso-CTS leads by .0033 but loses answer format. Below it, the five highest rows span .0038 at a pooled SE of at most .0045. All ten of their pairwise retention intervals include zero. On transfer, seven of ten do, and the three exceptions all involve the share cap (Table 19).

Table 18: All 46 models with a complete twelve-task record at n=2000, sorted by retention. Retention is the unweighted mean of the twelve task accuracies and SE its pooled binomial standard error. That SE counts each of xbrl_finer's four tag queries per prompt as an independent trial, so it is nominally optimistic; scoring that one task per prompt instead is the opposite extreme and raises the largest SE in the column only from .0049 to .0050. class6 and xbrl6 are the means of the six classification and the six extraction tasks; transfer is the unweighted mean of the three held-out suites and is printed only when all three are scored; ECTSum is ROUGE-Lsum on 495 documents and is printed where that suite was generated. The transfer suites carry 1147, 1490 and 2000 items (finqa, convfinqa, finexam10k). Source: the e1_pool2000 and e1_heldout2000 result directories.

identifier	retention	SE	class6	xbrl6	transfer	ECTSum
capmad_isocts	.6905	.0045	.5820	.7990	.5032	—
asym_cts_k56	.6872	.0045	.5769	.7974	.5177	.2166
isocts_cap25	.6868	.0045	.5731	.8005	.5075	.2178
asym_cts_k48	.6853	.0045	.5743	.7962	.5115	.2171
isoc_white	.6836	.0044	.5787	.7885	.5183	.2175
isocts_g60	.6834	.0045	.5767	.7901	.5165	.2186
isocts_g55	.6810	.0045	.5789	.7832	.5082	—
isocts_g50	.6704	.0046	.5800	.7609	.5049	.2155
isocts_white	.6704	.0045	.5467	.7940	.5123	.2240
knots_ties_g50	.6703	.0046	.6468	.6939	.4585	.1941
asym_cts	.6695	.0045	.5452	.7938	.5138	—
adaptive_mad	.6586	.0046	.5867	.7305	.4832	.2255
isocts_neq	.6555	.0046	.5215	.7895	.5088	—
adaptive	.6549	.0047	.5876	.7223	.4830	—
asym_shared_only	.6549	.0046	.5754	.7344	.5194	—
adaptive_mad_k2	.6523	.0047	.5812	.7233	.4862	—
isoc_g50	.6504	.0046	.5742	.7266	.5039	—
isocts_g70	.6409	.0046	.4971	.7848	.5085	—
asym_cts_k16	.6391	.0045	.4952	.7830	.5071	—
trim_g50	.6356	.0047	.5787	.6924	.4799	.1905
adaptive_t10	.6247	.0047	.5773	.6721	.4785	—
ties64_g50	.6176	.0048	.6233	.6120	.4553	.2084
tsvm_g50	.6045	.0046	.6971	.5119	.3631	.0015
knots_ta	.5705	.0047	.6228	.5181	.4467	—
adaptive_t1	.5702	.0048	.4647	.6757	.4926	—
concat	.5638	.0047	.6169	.5108	.4457	.2167
isocts_g100	.5538	.0046	.3678	.7398	.5047	—
capmad_concat	.5535	.0049	.5699	.5370	.4804	—
adaptive_mad_dcmerge	.5346	.0049	.4726	.5966	.4656	—
dcmerge_g50	.4760	.0048	.3677	.5842	.4614	—
capmed_concat	.4758	.0049	.3792	.5725	.4837	—
adaptive_mad_concat	.4714	.0049	.4152	.5275	.4699	—
capmad_dcmerge	.4664	.0048	.3601	.5727	.4607	—
isocts_neq_nm	.4657	.0048	.3478	.5837	.4751	—
concat_nm_capmad	.4567	.0048	.4396	.4738	.4821	—
concat_weighted	.4536	.0043	.5671	.3401	.1338	—
normeq_med	.4373	.0045	.2649	.6096	.5037	—
client_xbrl_term	.3935	.0049	.3702	.4168	.5015	.2210
domerging	.3722	.0045	.4915	.2529	.3411	—
base	.2876	.0041	.1296	.4455	.4678	.2243
knots_ties	.1337	.0030	.2371	.0302	.0254	—
adaptive_tinf	.0000	.0000	.0000	.0000	.0005	—
dcmerge	.0000	.0000	.0000	.0000	.0188	—
isoc	.0000	.0000	.0000	.0000	.0000	—
isocts	.0000	.0000	.0000	.0000	.0000	—
tsvm	.0000	.0000	.0000	.0000	.0000	—

Table 19: The top of the retention ranking: all 10 pairwise contrasts among the five rules ranked below OCap with Iso-CTS, $A - B$, on retention (twelve tasks) and on transfer (three suites). 10 of 10 retention intervals and 7 of 10 transfer intervals include zero. Paired bootstrap at n=2000, 2000 resamples: items are resampled inside each task (retention) or suite (transfer) and the per-task or per-suite mean differences are averaged with equal weights; intervals are 95% percentile intervals. Intervals marked † include zero. Source: the saved output of `scripts/task_bootstrap.py` and `scripts/heldout_bootstrap.py` on the `e1_pool2000` and `e1_heldout2000` generations.

		retention		transfer	
A	B	Δ	95% interval	Δ	95% interval
asym_cts_k56	isocts_cap25	+.000	[−.003, +.004]†	+.010	[+.002, +.018]
asym_cts_k56	isoc_white	+.004	[−.000, +.008]†	−.001	[−.008, +.006]†
asym_cts_k56	isocts_g60	+.004	[−.001, +.008]†	+.001	[−.007, +.009]†
isocts_cap25	isoc_white	+.003	[−.001, +.007]†	−.011	[−.019, −.003]
isocts_cap25	isocts_g60	+.003	[−.001, +.008]†	−.009	[−.017, −.001]
asym_cts_k48	asym_cts_k56	−.002	[−.006, +.002]†	−.006	[−.012, +.001]†
asym_cts_k48	isocts_cap25	−.002	[−.005, +.002]†	+.004	[−.004, +.012]†
asym_cts_k48	isoc_white	+.002	[−.002, +.006]†	−.007	[−.014, +.000]†
asym_cts_k48	isocts_g60	+.002	[−.003, +.006]†	−.005	[−.013, +.003]†
isocts_g60	isoc_white	−.000	[−.005, +.005]†	−.002	[−.010, +.007]†

Table 21 gives the paired intervals of PolCap with Iso-CTS, which has the highest retention among safe rules, against every rule it is compared with in §7. It also gives the transfer interval of every rule against the base. The paired contrasts of OCap itself are those of §7.2 and Table 20.

Summarization. On ECTSum (ROUGE-Lsum, the last column of Table 18; paired bootstrap over the 495 documents, computed by `scripts/make_appendix_main_tables.py`) no merge significantly beats the unmerged base. PolCap is highest with $+.0012$ $[-.0044, +.0069]$. Task arithmetic loses .0075 $[.0006, .0143]$, KnOTS-TIES and TIES-trim under the scalar control lose .030 and .034, and TSV-M under the scalar control collapses (.0015). The transfer gains of the main text are specific to extractive and answer-span QA.

Capability. Table 22 gives 36 models (35 merges and the base) at harness limit 500, a limit-2000 check of PolEq against the base (.637 against .711 on GSM8K strict, as at limit 500), the rsLoRA sibling pool, and the limit-200 runs that include the single adapters.

The whole grid. Table 23 lists all 114 rules once, at the largest n at which each has a complete twelve-task record. The pooled SE of a retention mean is at most .0128 at n=100, so the lowest tier is read for direction only and not for order.

Table 20: **No-regression table: the twenty-nine rules measured on all three axes at the largest draw, and the $\tau{=}\infty$ ablation (cap removed, last row), against the unmerged base.** Task-balanced paired bootstrap (2000 resamples) on retention (12 tasks, $n{=}2000$) and transfer (3 suites, $n{=}2000$); paired bootstrap over the 500 GSM8K items under both lm-eval answer filters. The other twelve of the forty-one rules measured on all three axes have capability only at harness limit 200 or retention only at $n \leq 500$. Rows are ordered by **worst** z: the least favourable of the four measures in units of its own standard error, $(\mathrm{hi} - \mathrm{lo})/3.92$ of its interval. Bold marks an interval strictly below zero, and ✓ a rule with none. The eleven checked rules are the same under a Bonferroni correction across the four measures; correcting across the 116 tests of the twenty-nine rules ($z^*{=}3.52$) admits two more, PolEq+asym-CTS at $k_c{=}16$ and PolEq+Iso-CTS (-3.17, -3.25). The ordering at the top depends on the measure set: on retention, transfer and GSM8K strict, the highest worst z are PolCap with Iso-CTS at 2 MADs ($+2.78$), task arithmetic shrunk to OCap's norm ($+2.69$) and OCap with task arithmetic ($+1.47$); with the flexible filter in place of the strict one, OCap with task arithmetic is first ($+0.51$) and clipping at the median second (0.00). Rows are generated by `scripts/no_regression.py`.

rule	retention	transfer	GSM8K strict	GSM8K flexible	worst z
OCap+task arithmetic (ours) ✓	$+.266$ [$+.257, +.274$]	$+.013$ [$+.003, +.023$]	$+.030$ [$-.010, +.070$]	$+.008$ [$-.024, +.038$]	$+0.51$
clip at median+task arithmetic ✓	$+.188$ [$+.180, +.197$]	$+.016$ [$+.008, +.024$]	$+.018$ [$-.016, +.054$]	$+.000$ [$-.032, +.030$]	$+0.00$
PolCap+Iso-CTS, 2 MADs ✓	$+.365$ [$+.356, +.373$]	$+.018$ [$+.009, +.029$]	$+.054$ [$+.016, +.092$]	$-.002$ [$-.034, +.032$]	-0.12
Iso-CTS, full equalization, TA norm ✓	$+.178$ [$+.171, +.185$]	$+.007$ [$-.000, +.015$]	$+.020$ [$-.012, +.054$]	$-.002$ [$-.030, +.024$]	-0.15
PolCap+task arithmetic ✓	$+.184$ [$+.175, +.192$]	$+.002$ [$-.005, +.010$]	$+.028$ [$-.004, +.062$]	$-.002$ [$-.030, +.024$]	-0.15
task arithmetic, shrunk to OCap's norm ✓	$+.169$ [$+.161, +.177$]	$+.014$ [$+.006, +.023$]	$+.044$ [$+.012, +.076$]	$-.002$ [$-.030, +.024$]	-0.15
PolCap+DC-Merge ✓	$+.247$ [$+.238, +.255$]	$-.002$ [$-.011, +.006$]	$+.020$ [$-.012, +.054$]	$+.012$ [$-.016, +.040$]	-0.52
task arithmetic, full equalization ✓	$+.150$ [$+.142, +.158$]	$+.036$ [$+.026, +.045$]	$+.006$ [$-.030, +.042$]	$-.010$ [$-.040, +.018$]	-0.68
PolCap+Iso-CTS ✓	$+.371$ [$+.363, +.379$]	$+.015$ [$+.005, +.026$]	$+.028$ [$-.012, +.068$]	$-.016$ [$-.048, +.016$]	-0.98
DC-Merge, $\gamma{=}.5$ ✓	$+.188$ [$+.181, +.196$]	$-.006$ [$-.015, +.002$]	$+.034$ [$+.000, +.070$]	$+.000$ [$-.028, +.030$]	-1.54
OCap+DC-Merge ✓	$+.179$ [$+.171, +.187$]	$-.007$ [$-.016, +.001$]	$+.018$ [$-.016, +.054$]	$+.000$ [$-.028, +.028$]	-1.66
PolEq+asym-CTS, $k_c{=}16$	$+.352$ [$+.343, +.360$]	$+.039$ [$+.028, +.050$]	$\mathbf{-.068}$ [$-.110, -.026$]	$-.004$ [$-.036, +.026$]	-3.17
PolEq+Iso-CTS	$+.383$ [$+.374, +.391$]	$+.044$ [$+.033, +.056$]	$\mathbf{-.068}$ [$-.108, -.026$]	$-.032$ [$-.062, +.000$]	-3.25
Iso-CTS, full equalization	$+.368$ [$+.359, +.376$]	$+.041$ [$+.030, +.052$]	$\mathbf{-.128}$ [$-.170, -.086$]	$\mathbf{-.034}$ [$-.064, -.004$]	-5.97
Iso-CTS, $\gamma{=}.6$	$+.396$ [$+.387, +.404$]	$+.049$ [$+.037, +.060$]	$\mathbf{-.186}$ [$-.230, -.138$]	$\mathbf{-.040}$ [$-.074, -.006$]	-7.92
Iso-CTS, $\gamma{=}.55$	$+.393$ [$+.385, +.402$]	$+.040$ [$+.028, +.053$]	$\mathbf{-.216}$ [$-.266, -.166$]	$\mathbf{-.046}$ [$-.082, -.008$]	-8.47
PolEq+asym-CTS, $k_c{=}32$	$+.382$ [$+.373, +.390$]	$+.046$ [$+.035, +.057$]	$\mathbf{-.196}$ [$-.238, -.150$]	$\mathbf{-.046}$ [$-.080, -.012$]	-8.73
task arithmetic	$+.276$ [$+.267, +.285$]	$\mathbf{-.022}$ [$-.033, -.011$]	$\mathbf{-.256}$ [$-.306, -.210$]	$\mathbf{-.098}$ [$-.136, -.058$]	-10.45
Iso-CTS, $\gamma{=}.5$	$+.383$ [$+.374, +.391$]	$+.037$ [$+.025, +.050$]	$\mathbf{-.314}$ [$-.364, -.266$]	$\mathbf{-.058}$ [$-.094, -.022$]	-12.56
Iso-CTS, share cap $\frac{1}{4}$	$+.399$ [$+.390, +.408$]	$+.040$ [$+.029, +.051$]	$\mathbf{-.314}$ [$-.362, -.266$]	$\mathbf{-.042}$ [$-.076, -.008$]	-12.82
PolEq+asym-CTS, $k_c{=}48$	$+.398$ [$+.389, +.406$]	$+.044$ [$+.031, +.055$]	$\mathbf{-.356}$ [$-.404, -.310$]	$\mathbf{-.036}$ [$-.070, -.002$]	-14.85
OCap+Iso-CTS	$+.403$ [$+.394, +.411$]	$+.035$ [$+.024, +.047$]	$\mathbf{-.400}$ [$-.452, -.350$]	$\mathbf{-.074}$ [$-.112, -.036$]	-15.37
PolEq+asym-CTS, $k_c{=}56$	$+.400$ [$+.391, +.408$]	$+.050$ [$+.038, +.061$]	$\mathbf{-.400}$ [$-.446, -.352$]	$\mathbf{-.062}$ [$-.100, -.026$]	-16.68
PolEq+Iso-C	$+.396$ [$+.387, +.404$]	$+.051$ [$+.039, +.062$]	$\mathbf{-.456}$ [$-.506, -.406$]	$\mathbf{-.078}$ [$-.114, -.042$]	-17.88
DO-Merging, $\lambda{=}1/n^2$	$+.085$ [$+.075, +.094$]	$\mathbf{-.127}$ [$-.141, -.112$]	$\mathbf{-.614}$ [$-.656, -.570$]	$\mathbf{-.380}$ [$-.426, -.332$]	-27.97
TIES, $\gamma{=}.5$	$+.330$ [$+.322, +.339$]	$\mathbf{-.012}$ [$-.024, -.001$]	$\mathbf{-.672}$ [$-.714, -.628$]	$\mathbf{-.126}$ [$-.168, -.082$]	-30.63
TIES-trim, $\gamma{=}.5$	$+.348$ [$+.339, +.356$]	$+.012$ [$+.001, +.024$]	$\mathbf{-.662}$ [$-.702, -.618$]	$\mathbf{-.128}$ [$-.168, -.086$]	-30.89
TSV-M, $\gamma{=}.5$	$+.317$ [$+.308, +.326$]	$\mathbf{-.105}$ [$-.118, -.091$]	$\mathbf{-.710}$ [$-.748, -.670$]	$\mathbf{-.502}$ [$-.548, -.454$]	-35.68
KnOTS-TIES, $\gamma{=}.5$	$+.383$ [$+.374, +.391$]	$-.009$ [$-.022, +.003$]	$\mathbf{-.712}$ [$-.750, -.672$]	$\mathbf{-.330}$ [$-.376, -.280$]	-35.78
PolCap+Iso-CTS, $\tau{=}\infty$ (cap removed)	$\mathbf{-.288}$ [$-.296, -.280$]	$\mathbf{-.467}$ [$-.481, -.454$]	$\mathbf{-.712}$ [$-.750, -.672$]	$\mathbf{-.784}$ [$-.818, -.748$]	-69.01

Table 21: PolCap against everything else, in two blocks. Top: `adaptive_mad` $-B$ on retention (twelve tasks) and on transfer (three suites), sorted by Δ retention; 14 of 14 retention and 13 of 14 transfer intervals exclude zero, and a dash marks a contrast not run on that axis. Bottom: $A -$ `base` on transfer for 26 further rules, the separation claim, sorted by Δ; 19 of the 26 intervals lie above zero. Paired bootstrap at n=2000, 2000 resamples: items are resampled inside each task (retention) or suite (transfer) and the per-task or per-suite mean differences are averaged with equal weights; intervals are 95% percentile intervals. Intervals marked † include zero. Source: the saved output of `scripts/task_bootstrap.py` and `scripts/heldout_bootstrap.py` on the `e1_pool2000` and `e1_heldout2000` generations.

	retention		transfer	
rule	Δ	95% interval	Δ	95% interval
`adaptive_mad` − the rule named in the row				
`base`	+.371	[+.363, +.379]	+.015	[+.005, +.026]
`client_xbrl_term`	+.265	[+.256, +.274]	−.018	[−.029, −.008]
`dcmerge_g50`	+.183	[+.174, +.191]	+.022	[+.012, +.032]
`concat`	+.095	[+.087, +.102]	+.038	[+.026, +.049]
`tsvm_g50`	+.054	[+.046, +.062]	+.120	[+.107, +.133]
`ties64_g50`	+.041	[+.034, +.048]	+.028	[+.017, +.040]
`trim_g50`	+.023	[+.016, +.029]	+.003	[−.007, +.014]†
`isocts_white`	−.012	[−.018, −.006]	−.029	[−.038, −.019]
`knots_ties_g50`	−.012	[−.019, −.004]	+.025	[+.014, +.036]
`isoc_white`	−.025	[−.031, −.019]	−.035	[−.045, −.025]
`isocts_g60`	−.025	[−.031, −.019]	−.033	[−.043, −.023]
`asym_cts_k48`	−.027	[−.033, −.021]	−.028	[−.038, −.018]
`isocts_cap25`	−.028	[−.034, −.022]	−.024	[−.034, −.014]
`asym_cts_k56`	−.029	[−.035, −.023]	−.034	[−.044, −.024]
the rule named in the row − `base`, transfer only				
`asym_shared_only`	—	—	+.052	[+.040, +.063]
`isoc_white`	—	—	+.051	[+.039, +.062]
`asym_cts_k56`	—	—	+.050	[+.038, +.061]
`isocts_g60`	—	—	+.049	[+.037, +.060]
`asym_cts`	—	—	+.046	[+.035, +.057]
`asym_cts_k48`	—	—	+.044	[+.031, +.055]
`isocts_white`	—	—	+.044	[+.033, +.056]
`isocts_g70`	—	—	+.041	[+.030, +.052]
`isocts_neq`	—	—	+.041	[+.030, +.052]
`isocts_cap25`	—	—	+.040	[+.029, +.051]
`isocts_g55`	—	—	+.040	[+.028, +.053]
`asym_cts_k16`	—	—	+.039	[+.028, +.050]
`isocts_g100`	—	—	+.037	[+.027, +.047]
`isocts_g50`	—	—	+.037	[+.025, +.050]
`isoc_g50`	—	—	+.036	[+.024, +.049]
`client_xbrl_term`	—	—	+.034	[+.024, +.044]
`adaptive_t1`	—	—	+.025	[+.016, +.034]
`adaptive`	—	—	+.015	[+.005, +.026]
`trim_g50`	—	—	+.012	[+.001, +.024]
`adaptive_t10`	—	—	+.011	[−.002, +.023]†
`dcmerge_g50`	—	—	−.006	[−.015, +.002]†
`knots_ties_g50`	—	—	−.009	[−.022, +.003]†
`ties64_g50`	—	—	−.012	[−.024, −.001]
`concat`	—	—	−.022	[−.033, −.011]
`tsvm_g50`	—	—	−.105	[−.118, −.091]
`concat_weighted`	—	—	−.334	[−.348, −.320]

Table 22: Capability under the EleutherAI harness, 43 rows in four labelled blocks; every model is loaded on the same 8-bit base. The limit is the harness cap on items per task (per subject for MMLU); it leaves 1319 / 2000 / 14042 at limit 2000, 500 / 500 / 12173 at limit 500, 200 / 200 / 9183 at limit 200 items (GSM8K / HellaSwag / MMLU). MMLU (macro) is the unweighted mean over the 61 `mmlu_*` entries the harness reports (57 subjects and its category aggregates) and is the figure the main body quotes; MMLU (agg.) is the harness's own item-weighted aggregate. GSM8K is exact match under the strict and the flexible answer filter, HellaSwag is length-normalized accuracy. Rows are sorted by GSM8K strict inside each block. Source: the `e6_500`, `e6_2000`, `e6_500_rslora` and `e6` result directories.

	MMLU		GSM8K		
identifier	macro	agg.	strict	flex	HellaSwag
main pool, limit 500					
adaptive_mad_k2	.685	.696	.766	.782	.708
concat_nm_capmad	.688	.698	.756	.782	.716
dcmerge_g50	.680	.690	.746	.784	.696
adaptive_mad_ties	.679	.691	.744	.768	.714
capmad_concat	.687	.699	.742	.792	.714
adaptive_mad	.679	.689	.740	.768	.708
adaptive_mad_concat	.683	.696	.740	.782	.712
adaptive_mad_dcmerge	.680	.693	.732	.796	.696
isocts_neq_nm	.685	.697	.732	.782	.706
capmad_dcmerge	.680	.691	.730	.784	.704
capmed_concat	.685	.698	.730	.784	.706
normeq_med	.687	.698	.718	.774	.712
base	.680	.692	.712	.784	.706
adaptive_mad_k4	.683	.694	.682	.740	.710
asym_cts_k16	.686	.697	.644	.780	.718
isocts_white	.688	.698	.644	.752	.732
isocts_neq	.689	.700	.584	.750	.718
isocts_g60	.683	.693	.526	.744	.710
asym_cts	.693	.703	.516	.738	.718
isocts_g55	.679	.688	.496	.738	.712
concat	.670	.682	.456	.686	.688
adaptive_mad_knots	.681	.692	.420	.730	.700
isocts_cap25	.688	.698	.398	.742	.724
isocts_g50	.673	.683	.398	.726	.712
asym_cts_k48	.685	.696	.356	.748	.716
asym_cts_k56	.685	.696	.312	.722	.714
capmad_isocts	.683	.694	.312	.710	.718
isoc_white	.684	.695	.256	.706	.722
adaptive_mad_tsvm	.671	.682	.176	.730	.690
adaptive_mad_isoc	.686	.698	.162	.700	.722
domerging	.490	.493	.098	.404	.640
trim_g50	.681	.691	.050	.656	.688
ties64_g50	.667	.678	.040	.658	.662
tsvm_g50	.637	.648	.002	.282	.512
adaptive_tinf	.231	.228	.000	.000	.270
knots_ties_g50	.670	.681	.000	.454	.678
main pool, limit 2000					
base	.681	.680	.711	.785	.693
isocts_white	.685	.685	.637	.743	.710
rsLoRA sibling pool, limit 500					
rslora_concat	.685	.697	.738	.806	.708
rslora_isocts_white	.686	.697	.716	.784	.700
floors: the unmerged base and single adapters alone, limit 200					
client_xbrl_term	.687	.697	.720	.785	.710
base	.682	.694	.680	.745	.705
client_sentiment	.325	.327	.000	.000	.485

Table 23: The full grid: all 114 distinct merge rules evaluated on the main pool suite. Each rule is shown once, at the largest evaluation size at which it has a complete twelve-task record (44 rules at n=2000, 21 at n=500, 49 at n=100), sorted by tier and then by retention. n is the number of items requested per task; tasks with fewer test items use all of them. Transfer is measured at the matching tier (n=2000, 500 and 200 respectively) and is printed only when all three suites are scored. On the binomial convention of Table 18 the pooled SE of a retention mean is at most .0049, .0063 and .0128 in the three tiers; no claim in the main body rests on order inside the n=100 tier. Source: the `e1_pool2000`, `e1_pool500` and `e1_pool` result directories and their held-out twins.

identifier	n	retention	class6	xbrl6	transfer					
	identifier	n	retention	class6	xbrl6	transfer				
capmad_isocts	2000	.6905	.5820	.7990	.5032	adaptive_mad_ties	500	.5636	.6348	.4924	.4393
asym_cts_k56	2000	.6872	.5769	.7974	.5177	knots_white	500	.5620	.3801	.7439	.5107
isocts_cap25	2000	.6868	.5731	.8005	.5075	isocts_white_a05	500	.5554	.4042	.7066	—
asym_cts_k48	2000	.6853	.5743	.7962	.5115	isocts_white_a20	500	.5372	.5556	.5188	—
isoc_white	2000	.6836	.5787	.7885	.5183	ties_white	500	.4830	.2193	.7468	.5100
isocts_g60	2000	.6834	.5767	.7901	.5165	dcmerge_white	500	.4829	.2814	.6844	.4693
isocts_g55	2000	.6810	.5789	.7832	.5082	domerging_l1n	500	.0010	.0021	.0000	—
isocts_g50	2000	.6704	.5800	.7609	.5049	isocts_white_mean	500	.0000	.0000	.0000	—
isocts_white	2000	.6704	.5467	.7940	.5123	ties_split	100	.6275	.5750	.6800	.4850
knots_ties_g50	2000	.6703	.6468	.6939	.4585	trim_g50_clip10	100	.6188	.5567	.6808	.4900
asym_cts	2000	.6695	.5452	.7938	.5138	tiesmean_g50	100	.6100	.6167	.6033	.4467
adaptive_mad	2000	.6586	.5867	.7305	.4832	ties128_g50	100	.6040	.6167	.5913	.4683
isocts_neq	2000	.6555	.5215	.7895	.5088	ties64_g60	100	.6023	.5850	.6196	.4683
adaptive	2000	.6549	.5876	.7223	.4830	ties64_g40	100	.6008	.6667	.5350	.4350
asym_shared_only	2000	.6549	.5754	.7344	.5194	ties64_neq	100	.5942	.5200	.6683	.4950
adaptive_mad_k2	2000	.6523	.5812	.7233	.4862	clipcell10	100	.5788	.6283	.5292	.4800
isoc_g50	2000	.6504	.5742	.7266	.5039	clipcell3	100	.5706	.6083	.5329	.5100
isocts_g70	2000	.6409	.4971	.7848	.5085	isocts_g50_nm	100	.5615	.5167	.6063	.5250
asym_cts_k16	2000	.6391	.4952	.7830	.5071	normeq_g25	100	.5550	.5800	.5300	.5133
trim_g50	2000	.6356	.5787	.6924	.4799	svd_prod64	100	.5517	.6183	.4850	.4783
adaptive_t10	2000	.6247	.5773	.6721	.4785	concat_cap50	100	.5465	.5700	.5229	.5017
ties64_g50	2000	.6176	.6233	.6120	.4553	pico64	100	.5448	.5883	.5012	.5200
tsvm_g50	2000	.6045	.6971	.5119	.3631	proj_b32	100	.5433	.6183	.4683	.4533
knots_ta	2000	.5705	.6228	.5181	.4467	normeq_g50	100	.5365	.5550	.5179	.5083
adaptive_t1	2000	.5702	.4647	.6757	.4926	elect_g50	100	.5346	.5550	.5142	.4900
concat	2000	.5638	.6169	.5108	.4457	concat_l05	100	.5292	.5983	.4600	.5117
isocts_g100	2000	.5538	.3678	.7398	.5047	trim_cap50	100	.5281	.6783	.3779	.3367
capmad_concat	2000	.5535	.5699	.5370	.4804	isocts_g40	100	.5269	.5500	.5038	.4500
adaptive_mad_dcmerge	2000	.5346	.4726	.5966	.4656	proj_a16	100	.5173	.6033	.4312	.4783
dcmerge_g50	2000	.4760	.3677	.5842	.4614	normeq_g75	100	.5160	.5183	.5138	.4967
capmed_concat	2000	.4758	.3792	.5725	.4837	proj_b16	100	.5087	.6217	.3958	.4550
adaptive_mad_concat	2000	.4714	.4152	.5275	.4699	normeq_pico64	100	.4844	.4217	.5471	.4900
capmad_dcmerge	2000	.4664	.3601	.5727	.4607	svd_prod8	100	.4825	.5983	.3667	.4567
isocts_neq_nm	2000	.4657	.3478	.5837	.4751	concat_white	100	.4742	.4033	.5450	.5117
concat_nm_capmad	2000	.4567	.4396	.4738	.4821	asym_cts_aniso	100	.4665	.4033	.5296	.5033
concat_weighted	2000	.4536	.5671	.3401	.1338	concat_l2	100	.4665	.6200	.3129	.2067
normeq_med	2000	.4373	.2649	.6096	.5037	dare64	100	.4642	.5967	.3317	.4417
domerging	2000	.3722	.4915	.2529	.3411	normeq_geo	100	.4592	.3167	.6017	.5233
knots_ties	2000	.1337	.2371	.0302	.0254	avg_factors	100	.4346	.4500	.4192	.4867
adaptive_tinf	2000	.0000	.0000	.0000	.0005	asym_full	100	.4290	.3317	.5262	.4900
dcmerge	2000	.0000	.0000	.0000	.0188	dare64_neq	100	.4242	.2883	.5600	.5267
isoc	2000	.0000	.0000	.0000	.0000	isoc_white_nm	100	.4133	.3033	.5233	.4967
isocts	2000	.0000	.0000	.0000	.0000	dcmerge_neq	100	.4029	.2100	.5958	.4800
tsvm	2000	.0000	.0000	.0000	.0000	asym_cts_mean	100	.3975	.2867	.5083	.5100
isocts_white_kc40	500	.6820	.5670	.7970	.5107	asym_cts_eq	100	.3708	.2400	.5017	.4983
isocts_white_kc16	500	.6817	.5750	.7884	—	slao	100	.3090	.1900	.4279	.4817
asym_cts_b25_k56	500	.6814	.5666	.7962	.5073	concat_l4	100	.2608	.5033	.0183	.0467
adaptive_mad_tsvm	500	.6813	.6453	.7172	.4547	ties64_clip10	100	.2508	.4683	.0333	.0617
asym_cts_b75_k56	500	.6810	.5676	.7944	.5087	ties64	100	.1592	.3183	.0000	.0583
tsvm_white	500	.6799	.5676	.7923	.5080	concat_l6	100	.0242	.0483	.0000	.0017
adaptive_mad_isoc	500	.6755	.5640	.7871	.5087	dcmerge_nosmooth	100	.0192	.0383	.0000	.0000
isocts_white_a15	500	.6674	.5535	.7812	.5087	concat_l8	100	.0000	.0000	.0000	.0000
isocts_white_kc48	500	.6669	.5422	.7915	—	dcmerge_linear	100	.0000	.0000	.0000	.0000
adaptive_mad_knots	500	.6592	.6076	.7107	.4693	dcmerge_trim4	100	.0000	.0000	.0000	.0000
adaptive_mad_k4	500	.6559	.5771	.7347	.4740	isocts_cap33	100	.0000	.0000	.0000	.0000
isocts_white_kc24	500	.6516	.5188	.7845	—	isocts_cap50	100	.0000	.0000	.0000	.0000
isocts_white_a075	500	.6252	.4811	.7694	—	normeq_mean	100	.0000	.0000	.0000	.0000

H SENSITIVITY AND ABLATIONS

This appendix expands §7.2, and Figure 6 shows both parameter curves. We vary one constant at a time, because the setting supplies no validation data to search against. We give the tier of every number. The pooled SE of one merge's retention mean is at most .0049 at n=2000, .0063 at n=500 and .0128 at n=100.

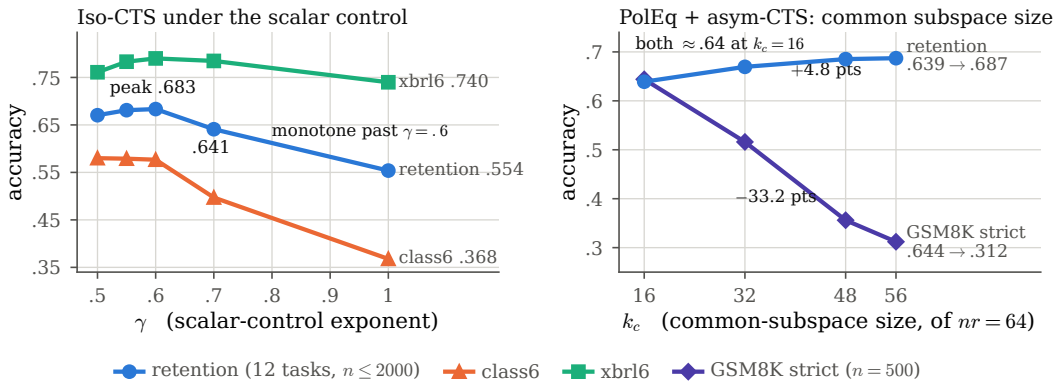


Figure 6: Left: the scalar control on Iso-CTS peaks at γ=.6 and falls monotonically beyond it, and almost all of the fall is in classification. Right: in the PolEq + asym-CTS family the size k_c of the common subspace trades the axes monotonically, since retention rises while GSM8K strict falls.

The scalar control needs its exponent chosen. At n=2000 the scalar control on Iso-CTS scores .6704, .6810, .6834, .6409 and .5538 at $\gamma = .5, .55, .6, .7, 1$ (Table 24). The optimum is interior, and the fall past it comes from classification. From γ=.6 to .7, `class6` loses .080 while `xbrl6` loses .005, so a server tuning on extraction alone would see nothing at stake. Full equalization without the minimum (`isocts_neq`, .6555) is above γ=.7 because it also amplifies the light clients. Note that it is a different transformation and not a point of this sweep. At its best γ the scalar control beats PolCap with Iso-CTS on retention by .025 but costs .186 of GSM8K strict against the base (Table 20). In addition, choosing γ requires the validation data that the setting lacks.

Table 24: The scalar control with exponent γ on Iso-CTS, with PolEq for reference. Retention, `class6`, `xbrl6` and transfer are as defined above, at n=2000 items per task. Column maxima are bold. Source: `e1_pool2000` and `e1_heldout2000`.

setting	identifier	retention	class6	xbrl6	transfer
γ=.5	`isocts_g50`	.6704	**.580**	.761	.5049
γ=.55	`isocts_g55`	.6810	.579	.783	.5082
γ=.6	`isocts_g60`	**.6834**	.577	.790	**.5165**
γ=.7	`isocts_g70`	.6409	.497	.785	.5085
γ=1	`isocts_g100`	.5538	.368	.740	.5047
full equalization (no minimum)	`isocts_neq`	.6555	.521	.789	.5088
PolEq + Iso-CTS (no γ)	`isocts_white`	.6704	.547	**.794**	.5123

Table 25: Common-subspace size k_c of the asym-CTS family (β=.5). Retention, `class6`, `xbrl6` and transfer are as defined above, at n=2000 items per task. GSM8K strict is exact match under the strict answer format from the evaluation harness at limit 500. Column maxima are bold. Source: `e1_pool2000`, `e1_heldout2000` and `e6_500`.

setting	identifier	retention	class6	xbrl6	transfer	GSM8K strict
k_c=16	`asym_cts_k16`	.6391	.495	.783	.5071	**.644**
k_c=32	`asym_cts`	.6695	.545	.794	.5138	.516
k_c=48	`asym_cts_k48`	.6853	.574	.796	.5115	.356
k_c=56	`asym_cts_k56`	**.6872**	**.577**	**.797**	**.5177**	.312

GSM8K strict measures format. Merges that lose it still solve many problems. Every merge in Table 22 with strict below .6, except the cap-removed ablation, scores .28–.75 under the flexible answer filter.

The size of the common subspace. In the asym-CTS family, k_c moves the axes in different directions (Table 25). Retention rises to .6872 and saturates (the $16 \rightarrow 32$ step is $+.030$ $[+.026, +.035]$, the $48 \rightarrow 56$ step $+.002$ $[-.002, +.006]$). Almost all of the rise is in classification, and almost all of that is in `headline`. Transfer stays within .0106. GSM8K strict falls $.644 \rightarrow .516 \rightarrow .356 \rightarrow .312$, while MMLU and HellaSwag stay within .008. A scalar objective would read this as a clean improvement and select k_c=56, which pays .332 of GSM8K strict for .048 of retention.

The cap multiplier. Table 26 changes the 3 in $m + 3\,\mathrm{MAD}$ to 2 and 4. Retention and transfer do not move beyond their intervals. GSM8K strict falls as the cap loosens, because a looser cap lets more of the heavy clients' scale through. At k=2 GSM8K strict is significantly above the base, and at k=4 GSM8K flexible is significantly below it. We report 3, which was fixed before this sweep. A tighter cap would have served the capability axis better.

Table 26: The cap multiplier k in $m + k\,\mathrm{MAD}$, PolCap with Iso-CTS on FINLORA: difference to the base and paired-bootstrap 95% interval on retention and transfer (n=500) and GSM8K (limit 500). Bold: interval below zero.

k	retention	transfer	GSM8K strict	GSM8K flexible
2	$+.371$ $[+.359, +.383]$	$+.021$ $[+.003, +.040]$	$+.054$ $[+.016, +.092]$	$-.002$ $[-.034, +.032]$
3 (default)	$+.372$ $[+.360, +.384]$	$+.023$ $[+.005, +.041]$	$+.028$ $[-.012, +.068]$	$-.016$ $[-.048, +.016]$
4	$+.378$ $[+.366, +.391]$	$+.019$ $[-.000, +.037]$	$-.030$ $[-.072, +.012]$	$\mathbf{-.044}$ $[-.078, -.010]$

The pool step's constants. Table 27 varies the Iso-CTS constants under the PolEq client step at n=500. PolCap inherits the same defaults. We find no monotone structure in k_c (range .0304, two non-default values above the default), and we report this without tuning. The output multiplier α scores .6671 and .6674 at 1 and 1.5 and falls off on both sides (.5554 at .5, .5372 at 2). We ran no point between 1 and 1.5. Of the two anchors for PolEq, the median scores .6671 and the mean **.0000** on all twelve tasks. This is because the mean of the eight client norms is $4.2\times$ above their median at the median weight matrix (and about $10^3\times$ at the first query projection), so anchoring to it inflates the six lightest clients by up to $18\times$. Two constants elsewhere in the grid are less forgiving. The asym-CTS weight β is flat where we swept it densely (.0043 over .25/.5/.75 at k_c=56, n=500) but not across its endpoints at k_c=32 (n=100: β=.5 scores .6562, β=0 .6327, β=1 .4290). The share cap has a cliff. At n=100, $\frac{1}{4}$ scores .6683, while $\frac{1}{3}$ and $\frac{1}{2}$ both score .0000.

Operator and client step. Table 28 crosses six operators with four magnitude treatments. Under PolCap every operator leaves zero at n=500 (Iso-C .6755, TSV-M .6813, Iso-CTS .6498, DC-Merge .5195, KnOTS-TIES .6592, TIES .5636), but the operators differ on answer format. At harness limit 500, GSM8K strict is .162, .176, and .420 for Iso-C, TSV-M, and KnOTS-TIES, so the retention they add comes at the cost of the capability axis. It is .732, .744, and .740 for DC-Merge, TIES, and Iso-CTS, and of these Iso-CTS has the highest retention. Under the PolEq client step, the three operators that do not select entries stay within .0158 of each other (Iso-C .6829, TSV-M .6799, Iso-CTS .6671), while the three that do drop (KnOTS-TIES .5620, TIES .4830, DC-Merge .4829). This failure is active, because KnOTS-TIES and TIES score *lower* under the client step than under the scalar control (.5620 vs. .6651, .4830 vs. .6135). After the client step every singular value of every client is equal. This removes the two rankings that these operators use, namely which entries hold most of a client's mass and which client has more mass. The trim still acts, but it no longer selects anything the client learned. PolCap keeps the magnitudes of the non-outlier clients, and with them the ranking the trim acts on, so KnOTS-TIES recovers from .5620 under PolEq to .6592. This boundary holds for retention only. Under the client step the same entry-selecting operators *gain* transfer (KnOTS-TIES .4400 $\rightarrow$ .5107, TIES .4440 $\rightarrow$ .5100).

A pre-registered test of when to flatten. Before evaluating four further operators, we wrote down which of them need the polar step, using only their definitions. Our criterion was that an operator needs it if it equalizes the singular values of the merged update. Iso-C does, so the polar step should raise GSM8K strict. TSV-M (which rebuilds each client from its own singular values), TIES (trim and sign election) and KnOTS-TIES (TIES in a shared SVD basis) do not, so it should not. The

Table 27: Constants of PolEq, one constant varied at a time. Retention, `class6`, `xbrl6` and transfer are as defined above, at n=500 items per task. A dash in the transfer column marks a variant that was not evaluated on the transfer suites; the default row is repeated in each block. The Asym-CTS private-direction weight β at k_c=56 block is not repeated here: its three rows read .6814, .6853 and .6810 on retention, within one standard error of each other, and the grid (Table 23) lists their largest-draw values. Source: `e1_pool500` and `e1_heldout500`.

value	identifier	retention	class6	xbrl6	transfer
common-subspace size k_c (default 32)					
16	`isocts_white_kc16`	.6817	.575	.788	–
24	`isocts_white_kc24`	.6516	.519	.784	–
32 (default)	`isocts_white`	.6671	.540	.794	.4993
40	`isocts_white_kc40`	.6820	.567	.797	.5107
48	`isocts_white_kc48`	.6669	.542	.791	–
output scale α (default 1)					
.5	`isocts_white_a05`	.5554	.404	.707	–
.75	`isocts_white_a075`	.6252	.481	.769	–
1 (default)	`isocts_white`	.6671	.540	.794	.4993
1.5	`isocts_white_a15`	.6674	.553	.781	.5087
2	`isocts_white_a20`	.5372	.556	.519	–
client-norm anchor (default: pool median)					
median (default)	`isocts_white`	.6671	.540	.794	.4993
mean	`isocts_white_mean`	.0000	.000	.000	–

Table 28: Six published operators under four magnitude treatments: the operator's published defaults, the scalar control γ=.5, and the two polar client steps, PolEq (every flattened client set to the pool median m_F) and PolCap (OCap after the polar step, §6). Each cell is retention (unweighted mean of the twelve pool tasks) at the largest evaluation tier the model was run at; the superscript is that tier's nominal n per task, not a uniform count; the conventions above give the tasks that are scored at another count. Tiers are mixed within rows and columns, so differences below the tier effect should not be read: over the eight models of this table that were run at both n=500 and n=2000 the two retentions differ by at most .0151. Four cells of the two polar columns are printed at a higher tier; at the common n=500 tier `isoc_white` scores .6829, `isocts_white` scores .6671, `adaptive_mad` scores .6498 and `adaptive_mad_dcmerge` scores .5195. Identifiers are printed beside each cell. Source: `e1_pool2000`, `e1_pool500` and `e1_pool`.

operator	published defaults	scalar control γ=.5	PolEq	PolCap
Iso-C	$.0000^{2000}$ `isoc`	$.6504^{2000}$ `isoc_g50`	$.6836^{2000}$ `isoc_white`	$.6755^{500}$ `adaptive_mad_isoc`
Iso-CTS	$.0000^{2000}$ `isocts`	$.6704^{2000}$ `isocts_g50`	$.6704^{2000}$ `isocts_white`	$.6586^{2000}$ `adaptive_mad`
TSV-M	$.0000^{2000}$ `tsvm`	$.6045^{2000}$ `tsvm_g50`	$.6799^{500}$ `tsvm_white`	$.6813^{500}$ `adaptive_mad_tsvm`
DC-Merge	$.0000^{2000}$ `dcmerge`	$.4760^{2000}$ `dcmerge_g50`	$.4829^{500}$ `dcmerge_white`	$.5346^{2000}$ `adaptive_mad_dcmerge`
KnOTS-TIES	$.1337^{2000}$ `knots_ties`	$.6703^{2000}$ `knots_ties_g50`	$.5620^{500}$ `knots_white`	$.6592^{500}$ `adaptive_mad_knots`
TIES	$.1592^{100}$ `ties64`	$.6176^{2000}$ `ties64_g50`	$.4830^{500}$ `ties_white`	$.5636^{500}$ `adaptive_mad_ties`

criterion, the measure (PolCap minus OCap on GSM8K strict, paired over the same 500 items) and what would count as a failure were committed first. Iso-CTS, DC-Merge, and task arithmetic, which suggested the criterion, are not part of the test. Table 29 reports every outcome.

Table 29: Pre-registered test: GSM8K strict (harness limit 500) for OCap and PolCap under four operators not used to form the criterion, with the paired difference PolCap − OCap. Base: .712.

operator	equalizes merged spectrum	prediction	OCap	PolCap	PolCap − OCap	outcome
Iso-C	yes	polar helps	.110	.162	+.052 [+.024, +.082]	confirmed
TSV-M	no	no help	.002	.176	+.174 [+.142, +.206]	**failed**
TIES	no	no help	.006	.744	+.738 [+.698, +.776]	**failed**
KnOTS-TIES	no	no help	.004	.420	+.416 [+.372, +.462]	**failed**

Reading. The criterion held for Iso-C and failed for the other three. Without the polar step, TSV-M, KnOTS-TIES, and TIES fall to .002–.006 GSM8K strict, and with it they score .176, .420 and .744. The prediction was too narrow, although its direction was right. Every operator we test, except task arithmetic and DC-Merge (which flattens each client itself), loses answer format under OCap without the polar step. We report the broader rule in §7.2 as a finding made after the test, and not as its confirmation. The polar step is necessary for these operators, but it is not sufficient. Of those that need it, only TIES and Iso-CTS then stay level with the base.

Both levels are needed. Table 30 crosses client steps with pool steps. Without a client step, the isotropic pool step scores .0000 on every retention task and every transfer suite. OCap alone raises retention but loses answer format (GSM8K strict .312). Adding the polar step (PolCap) restores it, and $\tau{=}\infty$ (cap removed) returns the pipeline to zero. Without an isotropic pool step, the polar client step loses retention against task arithmetic alone. Finally, dropping the task-specific directions raises retention but takes GSM8K strict from .644 to .256.

Table 30: Client step × pool step (twelve of the combinations). Δ is against task arithmetic at the same tier (.5573 at $n{=}500$, .5540 at $n{=}100$, .5638 at $n{=}2000$); intervals are computed for the $n{=}500$ rows only, and at $n{=}100$ transfer is scored at $n{=}200$. GSM8K strict is at harness limit 500 (base: .712 at limit 500, .680 at 200). A dash marks a rule with no capability run.

client step	pool step	identifier	n	retention (Δ)	transfer	GSM8K strict
PolCap	Iso-CTS	`adaptive_mad`	500	.6498 (+.092 [+.082, +.103])	.4780	.740
PolCap	Iso-C	`adaptive_mad_isoc`	500	.6755 (+.118 [+.108, +.129])	.5087	.162
$\tau{=}\infty$ (cap removed)	Iso-CTS	`adaptive_tinf`	2000	.0000 (−.564)	.0005	.000
OCap	Iso-CTS	`capmad_isocts`	500	.6840 (+.127 [+.116, +.137])	.4947	.312
PolEq	Iso-CTS	`isocts_white`	500	.6671 (+.110 [+.099, +.121])	.4993	.644
PolEq	Iso-C	`isoc_white`	500	.6829 (+.126 [+.116, +.135])	.5120	.256
PolEq	uniform mean	`concat_white`	100	.4742 (−.080)	.5117	—
full equalization	Iso-CTS	`isocts_neq`	500	.6506 (+.093 [+.082, +.104])	.5047	.584
full equalization	uniform mean	`normeq_med`	500	.4334 (−.124 [−.136, −.112])	.4880	.718
none	Iso-CTS	`isocts`	2000	.0000 (−.564)	.0000	—
none	Iso-C	`isoc`	2000	.0000 (−.564)	.0000	—
none	uniform mean	`concat`	500	.5573 (—)	.4453	.456

I Generality and Limits

I.1 A Second Dominated Pool: LoRA Land

Pool. The LoRA Land authors host 27 task adapters for Mistral-7B-v0.1 (Zhao et al., 2024; Jiang et al., 2023). Each has rank 8 on the query and value projections of all 32 layers, i.e., 64 adapted matrices. Two files, `glue_mnli` and `glue_mrpc`, are byte-identical, so the pool has 26 distinct clients, and we keep `glue_mrpc`. Client magnitudes have a mean spread of 251×, a median spread of 52×, and a median per-matrix ratio of 42×. One client (`dbpedia`) is 81× above the pool median, and three more (`covid`, `drop`, `wikisql`) are 3.3–4.8× above it. The PolCap cap binds on those four in 63–64 of 64 matrices and on four other clients in at most nine (median 4 clients per matrix). OCap binds on the same four in all 64 matrices and on one other client in one matrix. It reduces the mean spread from 251× to 6.9×.

Tasks and prompts. We score a client's own task only where that client, run alone, validates the prompt, i.e., it must beat both the base and the majority label by at least .10. Eight of nine candidates pass: SST-2, MRPC, QNLI, CoLA, and QQP from GLUE (Wang et al., 2019), DBpedia and AG News (Zhang et al., 2015), and HellaSwag (Zellers et al., 2019). MNLI fails because its adapter is the MRPC file. Prompts are each adapter card's sample input, verbatim. We use $n{=}200$ items per task (seed 42, identical for every model), greedy decoding, and the base in 8-bit. Accuracy is exact match of the first output line after lowercasing and stripping punctuation. The base scores .000 on SST-2 and AG News because those templates end without an answer cue, and at $k_c{=}32$ no merged model emits AG News's JSON label. We keep both tasks in every mean.

Merging. We use rank budget $nr{=}208$ and every constant as on FinLoRA. This includes Iso-CTS's $k_c{=}32$ and $(nr - k_c)/n{=}6$ task-specific directions per client for every Iso-CTS rule, the

published one included. The lower block of Table 31 raises k_c to $nr/2$. This is a sensitivity check we added after seeing the upper block.

Reading. Every published operator and both DO-Merging settings fall far below the base, and every between-client control lifts Iso-CTS and DC-Merge off zero. With Iso-CTS, capping beats equalizing in all four paired comparisons (with and without the polar step, both k_c; $+.037$ to $+.144$). DC-Merge under the scalar control is best ($+.214$ $[.189, .240]$), followed by OCap with DC-Merge ($+.185$ $[.161, .211]$) and PolCap with DC-Merge ($+.159$ $[.134, .183]$). PolCap with Iso-CTS is below the base at k_c=32 ($-.071$) and above it at $nr/2$ ($+.081$), where OCap ($+.116$) and Iso-CTS under the scalar control ($+.145$) do better still. With task arithmetic, OCap is above the base ($+.108$ $[.086, .130]$), as is clipping at the median ($+.101$ $[.077, .123]$), and the two are within noise of each other ($+.007$ $[-.002, .017]$). Compared with full equalization before task arithmetic, OCap is $.018$ $[.008, .028]$ less accurate. Full equalization before task arithmetic is above the base ($+.126$ $[.104, .147]$) but below PolCap with DC-Merge ($-.033$), OCap with DC-Merge ($-.059$), and DC-Merge under the scalar control ($-.089$). Anchored to task arithmetic's norm, it is level with the base ($-.001$). PolCap with task arithmetic is above the base ($+.036$ $[.021, .053]$) and above the norm-anchored rule, but below full equalization before task arithmetic ($-.089$). On GSM8K the base scores .392 strict. On 100 items under the same harness, a single client adapter scores .35 (SST-2) and .47 (the pool's own GSM8K client). The rules that keep the merged update small stay at or above the base. These are the two equalizing rules (.396 and .410), OCap with task arithmetic (.406, $+.014$ $[-.022, .050]$), clipping at the median (.412), task arithmetic shrunk to OCap's norm (.432, $+.040$ $[.010, .070]$), and PolCap with task arithmetic (.428, $+.036$ $[.002, .070]$). Every other merged rule falls to .000–.062 with degenerate digit strings, for any client step. The flexible filter lifts PolCap with DC-Merge to .242, which is still far below the base.

Table 31: LoRA Land, 26 distinct adapters for Mistral-7B-v0.1: accuracy on eight client tasks (n=200 items each, identical for every model, greedy, 8-bit base), the mean, its paired-bootstrap difference to the base (bold: 95% interval below zero), and GSM8K strict (lm-eval, limit 500). Every constant is as on FINLORA; the lower block raises Iso-CTS's common subspace to half the rank budget. Source: `results/loraland`.

model	SST-2	MRPC	QNLI	CoLA	QQP	DBpedia	AG News	HellaSwag	mean	Δ base	GSM8K
expert on its own task	.980	.840	.915	.865	.875	.995	.915	.930	.914		—
unmerged base	.000	.680	.420	.670	.395	.090	.000	.360	.327	—	.392
task arithmetic	.000	.000	.070	.000	.345	.020	.000	.025	.058	**−.269**	—
Iso-CTS (published)	.000	.000	.000	.000	.000	.000	.000	.000	.000	**−.327**	—
DC-Merge (published)	.005	.050	.025	.000	.010	.225	.000	.055	.046	**−.281**	—
DO-Merging, λ=$1/n^2$	.000	.000	.000	.000	.000	.000	.000	.000	.000	**−.327**	—
DO-Merging, λ=$1/n$	.000	.000	.000	.000	.000	.000	.000	.000	.000	**−.327**	—
full equalization + Iso-CTS	.820	.295	.110	.315	.255	.080	.000	.320	.274	**−.053**	.022
full equalization + task arithmetic	.000	.775	.755	.850	.825	.080	.000	.335	.453	+.126	.396
full equalization + Iso-CTS, TA norm	.000	.680	.385	.670	.400	.090	.000	.385	.326	−.001	.410
PolEq + Iso-CTS	.000	.220	.075	.205	.195	.190	.000	.220	.138	**−.189**	.002
OCap + Iso-CTS	.695	.565	.280	.320	.365	.050	.000	.215	.311	−.016	.002
Iso-CTS, γ=.5	.545	.015	.175	.270	.130	.050	.000	.130	.164	**−.163**	.004
DC-Merge, γ=.5	.505	.790	.600	.800	.780	.505	.000	.350	.541	+.214	.000
PolCap + Iso-CTS	.795	.160	.070	.490	.105	.100	.000	.325	.256	**−.071**	.020
PolCap + DC-Merge	.000	.825	.800	.810	.810	.230	.000	.410	.486	+.159	.010
PolCap + task arithmetic	.000	.690	.490	.760	.565	.090	.000	.310	.363	+.036	.428
clip at the median + task arithmetic	.000	.745	.665	.830	.760	.090	.000	.330	.427	+.101	.412
task arithmetic, shrunk to OCap's norm	.005	.690	.545	.805	.705	.090	.000	.330	.396	+.069	.432
OCap + DC-Merge	.000	.830	.770	.800	.800	.610	.000	.285	.512	+.185	.002
OCap + task arithmetic (ours)	.000	.745	.690	.835	.775	.080	.000	.355	.435	+.108	.406
sensitivity: $k_c = nr/2 = 104$ *instead of* 32											
Iso-CTS (published)	.000	.000	.000	.000	.000	.000	.000	.050	.006	**−.321**	—
full equalization + Iso-CTS	.870	.220	.340	.665	.555	.000	.000	.430	.385	+.058	—
PolEq + Iso-CTS	.495	.295	.225	.750	.065	.060	.000	.220	.264	**−.063**	—
OCap + Iso-CTS	.890	.500	.665	.740	.350	.020	.015	.360	.443	+.116	.062
Iso-CTS, γ=.5	.665	.520	.785	.610	.820	.010	.000	.365	.472	+.145	—
PolCap + Iso-CTS	.135	.810	.385	.715	.655	.050	.070	.445	.408	+.081	.040

I.2 A POOL WITHOUT DOMINANCE: MM-MERGEBENCH

Protocol. MM-MergeBench (Zeng et al., 2025) releases eight rank-16 LoRA adapters for LLaVA-1.5-7B (Liu et al., 2024), one per seen task with 224 adapted matrices each, plus four unseen tasks. Per-client median $\|\Delta W_i\|_F$ ranges from 0.62 (ScienceQA) to 2.72 (OCRVQA), a median spread of 4.4×. We merge all eight at rank budget nr=128, carry every other constant over untuned (so Iso-CTS keeps k_c=32), and decode greedily on n=500 items per task. ImageNet is merged but not scored because its validation images are access-gated, so seen means are over seven tasks. The upstream scorer compares strings exactly (*strict*), so "`Yes .`" does not match "`Yes`". We also report a *normalized* score that ignores case, surrounding quotes, whitespace, and trailing punctuation. The two scores agree to the digit for the base, seven of the eight clients, and task arithmetic, TIES, and DC-Merge at published settings. Elsewhere they differ by up to 13.3 points, so we report both (Table 32).

Result. DC-Merge is the best seen-task operator under either scorer. It scores 56.7 as published, 56.9 under the scalar control, and 56.8 (39.2 unseen) with PolCap's client step, which binds too rarely here to change it. None of the three loses anything to the strict scorer. OCap with task arithmetic binds on 0.20 of 8 clients per matrix and scores 54.3 seen and 37.0 unseen, the same as task arithmetic alone, also losing nothing to the strict scorer. Published Iso-CTS scores 54.5 normalized and 41.9 strict, OCap with Iso-CTS 54.4 and 41.8, and DO-Merging 54.5 and 54.3. PolCap with Iso-CTS scores 56.0 normalized and 48.3 strict, so it loses less format than Iso-CTS itself. Its strict loss is answer format on two tasks, VQAv2 (21.4 strict, 64.0 normalized, vs. DC-Merge's 64.4) and OCRVQA (39.8 strict, 51.0 normalized, vs. 56.0). On the unseen tasks PolCap and DC-Merge tie at 39.2 normalized (38.7 and 39.2 strict), compared with 37.0 for task arithmetic and 33.0 for the base. PolEq, which amplifies every light client, scores 53.0 normalized and 41.9 strict, and the strict score is below the base's 47.5. No merge approaches the home experts (63.9 over the seven seen tasks).

Table 32: MM-MergeBench, LLaVA-1.5-7B, n=500 per task (`results/lmm_eval`), scores ×100. "Strict" is the upstream exact-match scorer; "norm." strips case and punctuation on the exact-match tasks before comparing and equals strict elsewhere. Means are over the seven seen and four unseen tasks every model has (ImageNet is absent: its images are gated). Rows: merges by normalized seen mean, then the eight single-task clients, then the unmerged base. Two columns: read down the left, continue at the top of the right.

	seen		unseen	
model	strict	norm.	strict	norm.
dcmerge_g50	56.9	56.9	39.3	39.3
adaptive_mad_dcmerge	56.8	56.8	39.2	39.2
dcmerge	56.7	56.7	39.2	39.2
trim_g50	55.3	56.4	39.3	39.3
adaptive_mad	48.3	56.0	38.7	39.2
adaptive	48.5	55.9	38.8	39.3
ties64	55.6	55.6	38.4	38.4
asym_cts	46.0	55.5	39.2	39.6
isocts_g50	41.7	55.0	39.1	39.8
asym_cts_k48	44.3	55.0	39.3	39.7
domerging	54.3	54.5	37.6	37.6
isocts_cap25	42.0	54.5	39.0	39.7
isocts	41.9	54.5	39.0	39.7
capmad_isocts	41.8	54.4	39.0	39.7
isocts_neq_nm	54.4	54.4	37.2	37.2
concat	54.3	54.3	37.0	37.0
capmad_concat	54.3	54.3	37.0	37.0
dcmerge_neq	48.8	54.0	39.9	39.9
isocts_neq	41.0	53.5	39.0	39.7
isocts_white	41.9	53.0	38.9	39.6
isoc_white	42.2	51.2	38.4	38.7
client_IconQA	51.4	51.4	32.1	32.1
client_ScienceQA	50.6	50.6	29.9	29.9
client_VizWiz	50.4	50.4	32.1	32.1
client_flickr30k	50.1	50.1	31.2	31.2
client_OCRVQA	47.5	47.5	31.4	31.4
client_VQAv2	47.2	47.2	31.2	31.2
client_REC	46.2	46.2	28.9	28.9
client_ImageNet	19.1	33.3	6.1	11.2
base	47.5	47.5	33.0	33.0

The format loss comes from one light client. The ImageNet adapter ends every answer on the exact-match tasks with a period. No other client, and not the base, does so on more than 0.2% of its OCRVQA or VQAv2 answers. ImageNet is a light client, and task arithmetic, TIES, and DC-Merge put its period into none of their answers on those tasks. Equalizing the pool scales this client up and the heaviest client, OCRVQA, down, and the period then appears. Over the twenty-one merges, the share of OCRVQA answers ending in a period ranges from 0 to 97.4% and has Spearman correlation −0.96 with strict OCRVQA accuracy. The share is 95.6% for PolEq (OCRVQA 1.4

strict), 33% for PolCap (39.8), and 0% for OCap with task arithmetic. Amplification is not the only cause, because the scalar control on Iso-CTS never amplifies a light client and still reaches 89.6%. Flattening therefore contributes too, and we do not attribute PolCap's smaller loss to the cap alone. Normalization does not fully repair OCRVQA, so the loss is not only a format loss. On its 192 yes/no items, PolEq answers yes or no 61% of the time, compared with 100% for the base and task arithmetic.

Reference numbers. We did not reproduce the benchmark's own results or run RobustMerge. Zeng et al. (2025) report, on full test sets under strict scoring, seen/unseen means from 43.4/25.2 zero-shot to 57.3/38.0 for RobustMerge. Note that our n=500 draws over seven seen tasks are not directly comparable with those figures.

I.3 Sibling Pool and Leave-One-Client-Out

rsLoRA sibling pool. We also merge FinLoRA's public rsLoRA versions of the same eight clients (Kalajdzievski, 2023) with the same code (Table 33). PolCap leads retention (.5654, +.101 over task arithmetic, n=100) and is third on transfer (.5450 at n=200), behind PolEq (.5567) and TIES-trim under the scalar control (.5550).

Table 33: rsLoRA sibling pool: FinLoRA's public rsLoRA versions of the same eight clients, merged by the same code. Retention, `class6` and `xbrl6` at n=100 (`results/rslora_pool`); transfer at n=200 (`results/rslora_heldout`). "—" means not measured: transfer was run for six merges. The eight lone-client rows are omitted; the best of them is quoted in the text. Capability for this pool (EleutherAI harness at limit 500, `results/e6_500_rslora`, two merges) is the rsLoRA block of Table 22.

	retention			
model	12 tasks	class6	xbrl6	transfer
`adaptive_mad`	.5654	.360	.771	.5450
`isocts_neq`	.5598	.343	.776	.5433
`isocts_white`	.5517	.328	.775	.5567
`isoc_white`	.5119	.255	.769	—
`trim_g50`	.5023	.267	.738	.5550
`concat`	.4640	.362	.566	.5167
`normeq_med`	.4385	.312	.565	.5233

Leave-one-client-out. Each cell of Table 34 merges the pool without one client, at the reduced pool's rank budget, and scores the merge on that client's own tasks. PolCap has the highest mean (.4549, vs. .4276 for TIES-trim and .4219 for PolEq) and is best in three of six columns. `headline` reaches .744 with no headline client in the merge. Four cells rest on a single task and the `formula` cell on 200 items, so we read the pattern and not single gaps.

Table 34: Leave-one-client-out, n=500 requested items per task. Each cell merges the pool *without* the named client, at the rank budget of the reduced pool, and scores the merge on that absent client's own tasks (mean over them); the last column is the unweighted mean of the six cells, and the first two rows give the tasks behind each cell and their scored items. Rows are sorted by the mean. Source: the `loo_pool500` result directory.

rule	`finer`	`formula`	`headline`	`ner`	`sentiment`	`xbrl_extract`	mean
tasks scored	1	1	1	1	4	4	—
items scored	2000	200	500	500	1735	2000	—
`adaptive_mad`	.1850	.3500	.7440	.4240	.4831	.5430	.4549
`trim_g50`	.2780	.3500	.7080	.2620	.4271	.5405	.4276
`isocts_white`	.1890	.3950	.4440	.5480	.4401	.5150	.4219
`asym_cts_k48`	.1935	.3850	.4840	.5020	.4692	.4615	.4159
`concat`	.2005	.3300	.6940	.2080	.4695	.4545	.3928
`normeq_med`	.2260	.3750	.3620	.3020	.2953	.5140	.3457

I.4 ALL-HEAVY SUB-POOLS

Two client orders. We order the eight clients from light to heavy by median $\|\Delta W_i\|_F$ (financebench, formula, xbrl_extract, xbrl_term, finer, ner, headline, sentiment). We then merge the first $k \in \{2, 3, 4, 6, 8\}$ of that order (*light-first*) or of its reverse (*heavy-first*) at rank budget kr. Each merge is scored at 200 items per task on the tasks its members own (Table 35). We use both orders because one order alone would confound pool size with composition. The two k=8 merges are byte-identical adapters and score .666 and .664, so this gap is generation noise.

Table 35: Client-count sweep on the FINLORA pool. The first k clients of a fixed order are merged. The light-first order is `financebench` → `formula` → `xbrl_extract` → `xbrl_term` → `finer` → `ner` → `headline` → `sentiment`; heavy-first is its reverse. Each merge is evaluated on the scored tasks its members own, so the task set grows with k: the subscript is its size. Two cells are comparable only when their subscripts match, which sharing a column does not guarantee. n=200 items per task. A dash marks a merge that was not built (`isocts_white_nm` light-first at k=2, 3, 4, 6, 8; `isocts_white_nm` heavy-first at k=8). †Scored on tasks the pool's members do not own. Light-first k=2 is also scored on `fpb`, `fiqa`, `tfns`, `nwgi` and `headline`; restricted to its owned tasks, `concat` scores $.385_1$, `adaptive_mad` scores $.385_1$ and `isocts_white` scores $.385_1$. Heavy-first k=6 is also scored on `formula`; restricted to its owned tasks, `isocts_white_nm` scores $.517_{11}$. Source: `nclients_pool` and the merge provenance files.

rule	identifier	k=2	k=3	k=4	k=6	k=8
light-first						
task arithmetic	`concat`	$.179^{\dagger}_{6}$	$.683_5$	$.632_5$	$.579_7$	$.561_{12}$
DC-Merge (default)	`dcmerge`	$.375_1$	$.476_5$	$.486_5$	$.516_7$	$.000_{12}$
PolCap + Iso-CTS	`adaptive_mad`	$.178^{\dagger}_{6}$	$.616_5$	$.683_5$	$.683_7$	$.655_{12}$
PolEq + Iso-CTS	`isocts_white`	$.171^{\dagger}_{6}$	$.632_5$	$.705_5$	$.729_7$	$.664_{12}$
PolEq + Iso-CTS, absolute anchor	`isocts_white_nm`	–	–	–	–	–
heavy-first						
task arithmetic	`concat`	$.617_5$	$.629_6$	$.601_7$	$.571_{11}$	$.560_{12}$
DC-Merge (default)	`dcmerge`	$.002_5$	$.003_6$	$.005_7$	$.002_{11}$	$.000_{12}$
PolCap + Iso-CTS	`adaptive_mad`	$.000_5$	$.000_6$	$.000_7$	$.005_{11}$	$.658_{12}$
PolEq + Iso-CTS	`isocts_white`	$.000_5$	$.000_6$	$.000_7$	$.000_{11}$	$.666_{12}$
PolEq + Iso-CTS, absolute anchor	`isocts_white_nm`	$.000_5$	$.000_6$	$.204_7$	$.505^{\dagger}_{12}$	–

A relative bound cannot help an all-heavy pool. The median and the bound have no absolute scale. On a heavy-first sub-pool the median is itself heavy, so no client is an outlier. With Iso-CTS, PolCap scores .000, .000, .000 and .005 at heavy-first $k = 2, 3, 4, 6$, and PolEq scores .000 at all four (5,794 of its 5,800 generations empty). In contrast, task arithmetic scores .571–.629. So, below the full pool, an all-heavy sub-pool is a regime in which the baseline is the better rule. OCap with task arithmetic was not run on these sub-pools. Light-first, PolCap scores .616, .683, .683 and PolEq .632, .705, .729 at $k = 3, 4, 6$, compared with task arithmetic's .683, .632, .579.

J NEGATIVE RESULTS AND WITHDRAWN CLAIMS

We built and evaluated everything below and kept it in the result tree. Each item gives its evaluation size, because several variants exist only at n=100 (400 labels for `xbrl_finer`).

- **Factor-level rules stay near the floor on an independent pool.** A one-shot port of SLAO's take-A/merge-B update (Qiao & Mahdavi, 2026) scores .3090 vs. .2750 for the base and .5540 for task arithmetic. Averaging A and B separately reaches .4346, and all of its margin comes from classification (extraction .4192 vs. the base's .4183). Calibrating B's over-shared directions in the manner of Pico (Tang & Yang, 2026) reaches .5448, level with task arithmetic. All three are at n=100. This is not a criticism of those methods in the shared-initialization regimes they were designed for (§5). The base entry is read from the expert-matrix run at the same size, so we read direction and not margins.

- **Switching rules by regime does not pay.** We cluster clients whose A row spaces overlap above τ, aggregate each cluster factor-wise, and merge the clusters at the product level. This loses to a fixed rule on every eight- and eleven-client pool we ran. At τ=.5 it scores .4660 vs. .4758 for the product level alone on a mixed eleven-client pool. At τ=.98 it scores .4233 and .4381 on two mixed pools, above both fixed choices it is assembled from but below task arithmetic (.4975, .4579). This is because a cluster whose members have drifted has no common frame for its mean A to represent.
- **Choosing the cap by minimizing spectral dispersion is degenerate.** Selecting the cap that minimizes the coefficient of variation of the singular values of the capped sum has no interior optimum. The objective is non-decreasing along the τ grid on all 96 weight matrices, so it selects the tightest cap on all of them. As a result, "no constant" would become "the smallest number on the grid".
- **FedAvg weighting does not correct dominance.** Weighting by data volume over-scales two heavy clients and starves the lightest. It gives retention .4536 and transfer .1338 at n=2000, vs. .5638 and .4457 uniform, because its weights follow row counts and not norms.
- **Five claims we withdrew.** *(i) That PolEq is the rule to recommend.* PolEq was our default until the capability axis showed it significantly below the base on GSM8K strict ($-.068$ $[-.108, -.026]$). We had set PolCap aside on retention alone (.6586 vs. .6704 at n=2000, $-.012$ $[-.018, -.006]$), but PolCap is never significantly below the base on any measure and is significantly above it on retention and transfer (§7.4). *(ii) That a trim-based rule wins retention.* TIES-trim under the scalar control led the n=100 grid when that grid held thirty-one rules. At n=2000 it is twentieth of 44 merges (.6356), .023 [.016, .029] below PolCap. *(iii) That the client step brings four operators within .05.* Under the PolEq client step three operators come within .016, but the operators that select entries do not follow, and the range over all six is .200. The experiment we ran to confirm the claim refuted it and gave the sharper statement of §7.2. *(iv) That the gain over the base is +.37 on retention.* The paired bootstrap does give $+.371$ $[+.363, +.379]$, but the base answers classification prompts in the wrong label format. Under its own chat template the base gains .170 and the margin is $+.202$ at n=500 (§7.5). No retention verdict against a baseline reverses under the template. On transfer, which the template lowers for every model, several do. *(v) That the polar step is part of the method.* PolCap was our method until a late ablation. In it, OCap with task arithmetic, i.e., the bound without the polar step, ranked first by worst z and beat PolCap with task arithmetic by .082 [.074, .091] retention and .072 LoRA Land accuracy at equal GSM8K. We kept the polar step only where the operator needs it (§7.2), and a pre-registered criterion for where that is failed on three of four operators (Table 29).

Three of these, PolEq, the trim rule, and PolCap, were at some point the intended contribution. Each was retired by an evaluation at larger n or on another axis, and not by an argument. A grid of 114 rules produces a leader regardless of the mechanism, so the rules that the pipeline rejected are part of the evidence for what it kept.

K Failure Modes, with Generations

Every row below is read directly from the saved generations, and none is transcribed by hand. Labels are cut to 18 characters for display. We observe that a spectral operator at its published settings either answers in the wrong label vocabulary or emits nothing.

A. A published operator at its published settings. Iso-C and Iso-CTS produce no output on almost every prompt (Iso-CTS on 17,314 of the 17,317 at n=2000), and TSV-M emits a repeated punctuation token. This explains their scores of .0000. DC-Merge (.0000) and KnOTS-TIES (.1337) degrade less uniformly.

model / task	gold	generated
isocts → headline	`No`	*(empty)*
	`No`	*(empty)*
n=2000, empty 2000/2000, distinct outputs 1		
tsvm → fpb	`neutral`	`~\n\n~\n\n~\n\n~\n\n~` … (truncated)
	`negative`	`~\n\n~\n\n~\n\n~\n\n~` … (truncated)
n=1212, empty 0/1212, distinct outputs 172		
dcmerge → fpb	`neutral`	`The`
	`negative`	*(empty)*
n=1212, empty 502/1212, distinct outputs 412		
knots_ties → headline	`No`	*(empty)*
	`No`	*(empty)*
n=2000, empty 351/2000, distinct outputs 1151		

B. The label-prior artefact on `nwgi`. Always predicting the majority label scores .383; Appendix D and Table 10 give the weighted F1 and majority baseline.

model / task	gold	generated
knots_ties_g50 → nwgi	`moderately positiv`	`moderately positive`
	`mildly positive`	`moderately positive`
n=2000, empty 0/2000, distinct outputs 10		
isocts_g50 → nwgi	`moderately positiv`	`mildly positive`
	`mildly positive`	`mildly positive`
n=2000, empty 0/2000, distinct outputs 9		
concat → nwgi	`moderately positiv`	`moderously positive`
	`mildly positive`	`moderately positive`
n=2000, empty 2/2000, distinct outputs 61		

L PROMPTS, DECODING, AND SCORING

Every prompt below is emitted by `scripts/make_prompt_appendix.py`, which calls the same task loaders as the evaluation runner (seed 42, first item of the draw). None is transcribed by hand. Prompts longer than 200 characters are elided in the middle, and the elision is marked. Each prompt is followed by the gold answer and, after `||`, PolCap's generation on that item.

Format. Client-task and held-out prompts are the benchmark's own context strings. They are given identically to every model, with no chat template, system prompt, or wrapper. There are two exceptions. The chat-format probe of §7.5 wraps the same strings in the base checkpoint's own chat template, and the adapters of the multimodal pool were trained with a conversation template. The benchmark's training configuration wraps the context in `[INST]` markers, but its evaluation path gives the tokenizer the raw context. As a result, every client adapter, public or ours, is evaluated without the wrapper it was trained inside.

Decoding. We use greedy decoding, left padding, left truncation, and bfloat16 compute under 8-bit quantization. Only the continuation past the prompt is decoded. For prompts ending in `Answer:`, this equals the upstream harness's split on a delimiter. For `xbrl_finer`, whose prompts end in `Output US GAAP tags:`, the upstream split returns the whole prompt, so decoding the continuation is a deliberate correction. Maximum new tokens are 10 for the six classification tasks, 20–30 for the four XBRL extraction tasks, 50 for `formula`, 100 for `xbrl_finer`, 32 for FinQA and ConvFinQA, 8 for FinExam-10K and 160 for ECTSum.

Sources. The client tasks are the test splits distributed with FINLORA (Wang et al., 2025a), which re-formats the Financial PhraseBank (Malo et al., 2014), FiQA task 1 (Maia et al., 2018), the Twitter financial news sentiment set (Costa, 2022), the FinGPT instruction splits (Yang et al., 2023), the gold-commodity headline set (Sinha & Khandait, 2021), the credit-agreement NER set (Salinas Alvarado et al., 2015), FiNER-139 (Loukas et al., 2022), the XBRL analysis sets of Han et al. (2024) and FinanceBench (Islam et al., 2023). The two BERTScore tasks (Zhang et al., 2020) are generated

but excluded from every mean. The held-out suites are FinQA (1147 items), ConvFinQA (1490), the public partition of FinExam-10K (5110) and ECTSum (495), re-ingested into one format. A 2026-09-15 audit found no training overlap with any client.

`fpb`, the shortest client prompt (205 characters, 5 elided).

```
Instruction: What is the sentiment of this news? Please choose an answer from {n
egative/neutral/positive}.
Input: L&T has also mad
[... 5 characters elided ...]
ommitment to redeem the remaining shares by the end of 2011 .
Answer:
gold: neutral   ||   rule: Neutral://
```

The answer-slot rule. Each held-out context is given exactly one answer cue. A context that already ends in a cue terminated by a colon (`Answer:`, `A:`, `Summary:`) keeps it, and any other context gets `\nAnswer:` appended. The rule matters because FinExam-10K ends with the instruction `Answer with the letter only.`, which is not a cue. Removing its appended slot cost 8–17 points uniformly across models. The per-task cue is stamped into every result record.

How lenient the scorer is. Label tasks use the benchmark's first-occurrence match against the full label inventory. We make it stricter in one respect by counting a label only at a token boundary. Upstream's substring search credited `no` inside `does not`, and this alone gave the base .51 on `headline` while it answered `0/1`. The convention matters more for some rules than for others, because both polar client steps with Iso-CTS append `://` to the label on 82% of classification answers, while the entry-wise and cover-space baselines almost never do. Table 36 scores the same generations three ways.

Table 36: The six classification tasks at n=2000 under three scorers, namely exact string match, the label surrounded only by non-word characters, and the first-occurrence rule used throughout.

model	exact	label ± punctuation	first occurrence	`://` rate
`base`	.000	.000	.130	.00
`client_xbrl_term`	.000	.007	.370	.00
`adaptive_mad` (PolCap)	.069	.553	.587	.82
`isocts_white` (PolEq)	.002	.466	.547	.82
`isoc_white`	.537	.577	.579	.04
`isocts_g60`	.541	.575	.577	.02
`asym_cts_k56`	.464	.570	.577	.15
`trim_g50`	.544	.572	.579	.04
`knots_ties_g50`	.647	.647	.647	.00
`concat`	.607	.617	.617	.00
`tsvm_g50`	.697	.697	.697	.00

Exact match is unusable here. The base and the single adapter score .000 under it because they answer with explanations. The middle scorer is the strictest one that still credits a correct bare answer. Under it, the entry-wise and cover-space baselines lose at most .007, PolCap loses .033, and PolEq loses .081 (before rounding). Other PolEq rules (`isoc_white`, `asym_cts_k56`) lose .002–.007, so the suffix comes from the client step combined with Iso-CTS's task-specific directions. We report first-occurrence scores throughout. Note that under a stricter scorer PolCap's classification deficit to task arithmetic widens.

Other pools, capability, training. MM-MergeBench uses the upstream `conv_vicuna_v1` template instead of the HuggingFace processor's, which omits the system prompt. LoRA Land uses each adapter card's sample input (Appendix I.1). Capability runs use `lm_eval 0.4.13` with transformers 4.57.6, with MMLU and HellaSwag zero-shot, GSM8K five-shot, up to 500 items per subtask, fixed seeds, and the same 8-bit base. The controlled study's adapters are trained on `[INST]` {`context`} `[/INST]`{`target`} with the loss on the target only, and they are evaluated on the unwrapped context.