Predicting LoRA Adapters Across Model Families: A Training-Free Approach via Anchor-Space Ridge Regression

Anonymous Authors[1]

Abstract

A LoRA fine-tuned on one base model is, in general, useless on a different base model: weight spaces are unrelated across families. We introduce a training-free predictor that, given a small set of *anchor* task pairs (LoRAs trained on both source and target models), maps a source-side LoRA to a target-side LoRA via ridge regression in the anchor span. The mapping is closed-form, performs no gradient updates on the target model, and recovers a meaningful fraction of the source-to-oracle accuracy gap on held-out tasks. Across four model pairs spanning three scales and three target families (Qwen2.5 to Llama-3.2 at 1B and 3B, Qwen2.5-7B to Llama-3.1-8B, and Qwen2.5 to Gemma-2 at 2B), the best aggregate gap recovered lies in $[0.14,\ 0.31]$, and the winning predictor is always either a single global ridge or a per-tensor PCA variant. Two findings stand out. First, per-tensor variants specialize by task family on a per-held-out basis: per-tensor PCA wins science and ties math, while a single global ridge wins code, and this per-task pattern reproduces at 1B, 3B, 8B, and on Pair C. Second, the mapping is intrinsically non-local in anchor space at 3B: restricting the ridge to nearest-anchor neighborhoods does not help at any neighborhood size, and at 8B the K-sweep is flat within noise across $K \in \{2, \ldots, 24\}$, so locality never reliably outperforms the global ridge. We give a regression-theoretic explanation for both phenomena and report a sharp pool-composition failure mode in which collapsing the anchor pool to a single domain inverts the predictor's sign on cross-domain held-outs.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

Parameter-efficient fine-tuning with LoRA (Hu et al., 2022), and its quantized and decomposed variants (Dettmers et al., 2023; Liu et al., 2024), has made it cheap to specialize a single base model to many downstream tasks. The cost is paid *per base model*: a LoRA trained on Qwen-3B (Team, 2024) for GSM8K (Cobbe et al., 2021) is, in general, useless on Llama-3B (Grattafiori & Llama Team, 2024) for the same task, because the two models occupy unrelated weight spaces. As open-weights model families proliferate, an increasing fraction of practitioner effort goes into re-training the same task on a different backbone.

We ask a concrete question: *given a LoRA trained on a source model X, can we predict the LoRA we would have obtained on a target model Y, without ever fine-tuning on Y?* We assume access to a modest set of *anchor* tasks for which both X- and Y-side LoRAs already exist; the predictor is closed-form (Figure 1). The setup is closer to weight-space model merging (Wortsman et al., 2022; Ilharco et al., 2023; Yadav et al., 2023; Matena & Raffel, 2022) than to hypernetwork approaches (Ha et al., 2017; Phang et al., 2023; Mahabadi et al., 2021; AI, 2024): we never train a network to emit LoRAs, and we assume nothing about the architecture beyond a shared LoRA target-module list.

We make six contributions:

- **A family of training-free predictors** (Section 3) that map source-side LoRA tensors to target-side LoRA tensors via the anchor set, with no gradient updates on the target model: a global ridge, a top-K local ridge, per-tensor ridge and PCA variants, and an orthogonal-Procrustes baseline.
- **Cross-pair generality.** The same closed-form predictor recovers an aggregate gap in $[0.14,\ 0.31]$ across four model pairs (Table 7); the per-tensor-versus-global specialization profile reproduces on a per-task basis at 1B, 3B, 8B, and on the non-Llama Pair C (Section 8).
- **Anchor-count scaling.** At 3B the aggregate gap recovered grows monotonically in anchor count and plateaus at $N{=}16$ with 0.137 ± 0.126 (Section 5).
- **Per-tensor specialization** (Section 6). Per-tensor PCA

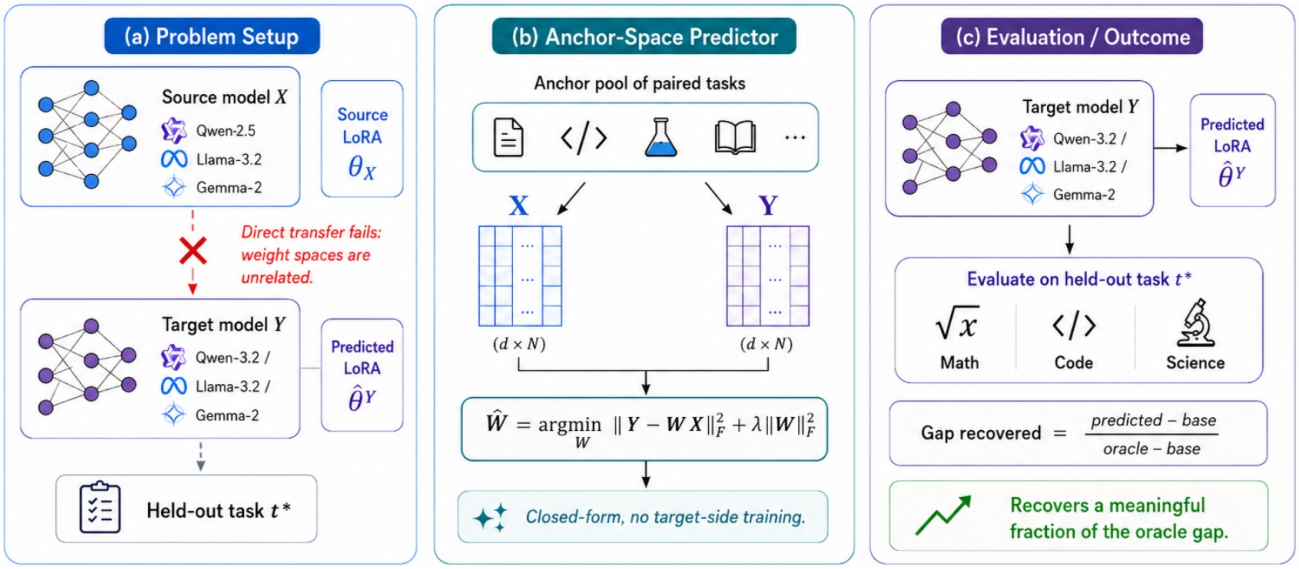

Figure 1. **Training-free cross-model LoRA prediction.** (a) A LoRA θ_X trained on a source model X does not transfer directly to a different target model Y: their weight spaces are unrelated. (b) Given a small pool of *anchor* tasks for which paired LoRAs $(\theta_X^{(i)}, \theta_Y^{(i)})$ already exist, we fit a closed-form ridge $\hat{W} = \arg\min_W \|Y - WX\|_F^2 + \lambda\|W\|_F^2$ in the anchor span; no gradient updates on Y. (c) Applying $\hat{W}$ to a held-out source LoRA yields a predicted target LoRA $\hat{\theta}_Y$, which we evaluate against a true Y-side oracle on math, code, and science tasks via gap recovered. Across four pairs spanning three scales (Qwen2.5 → Llama-3.2 at 1B/3B, Qwen2.5-7B → Llama-3.1-8B, and Qwen2.5 → Gemma-2 at 2B), the best aggregate gap recovered lies in $[0.14, 0.31]$; on a per-task basis, per-tensor PCA wins science and ties math while a global ridge wins code, with this specialization profile reproducing across all four pairs.

dominates the global ridge on science held-outs and ties on math, while the global ridge dominates on code; we give a structural reading of when each should win.

- **The locality null** (Section 7). Restricting the ridge to top-K nearest anchors does *not* reliably help at any $K \in \{2, \ldots, 24\}$: at 3B no K beats the global ridge, and at 8B the K-sweep is flat within noise. Cross-model LoRA mapping is non-local in anchor space.
- **Pool-composition stability** (Section 7). Aggregate gap recovered is stable under mixed-domain sub-pools but collapses on single-domain pools and turns *actively negative* for code-only pools, ruling out a "trivial transfer" interpretation.

All numbers in the main text are final for the 1B, 3B, 8B, and Pair C regimes; the predictor implementation, anchor pools, and evaluation harness are unchanged across regimes.

2. Related Work

The closest line of prior work is *hypernetworks for adapter prediction*. Ha et al. (2017) introduced hypernetworks for general weight generation; Mahabadi et al. (2021) and Phang et al. (2023) adapt the idea to multi-task adapter generation, and AI (2024) train a hypernetwork that maps free-form task descriptions to LoRA parameters. These methods require a dedicated training run over (X, Y) pairs and consume task identity (or text) as input. Our predictors are training-free, take a source-side LoRA *tensor* as input, and reduce to closed-form regression. We contrast conceptually but do not run head-to-head, since T2L's training-compute footprint is not directly comparable to a closed-form ridge fit.

A second line concerns *adapter composition and routing*. AdapterFusion (Pfeiffer et al., 2021), LoRA-Hub (Huang et al., 2024), MoLE (Wang et al., 2024), and X-LoRA-style mixtures (Zadouri et al., 2024) compose or route between existing *target-model* adapters using target-side data. We instead transfer *across* model families from a single source-side LoRA, with no target-side gradient information and no target-side data beyond the anchor LoRAs themselves. Closely related conceptually is *model merging and task arithmetic*: model soups (Wortsman et al., 2022), Fisher-weighted merging (Matena & Raffel, 2022), task arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), and ZipIt (Stoica et al., 2024) all operate within a single base model's weight space or close variants thereof. The anchor span we regress over can be viewed as a learned-coefficient task-arithmetic basis applied *across* base models. Finally, on the geometric side, centered kernel alignment (Kornblith et al., 2019) and stitching analyses (Bansal et al., 2021) measure how representations relate across models, and orthogonal-Procrustes alignment (Schönemann, 1966) is the classical solution to the rotational-alignment problem;

we use Procrustes as a geometry baseline, and per-tensor ridge generally outperforms it.

To our knowledge, no prior work predicts a target-model LoRA from a source-model LoRA via closed-form anchor-space regression, characterizes per-tensor specialization across task families in the cross-model setting, or reports the locality null we observe at 3B and 8B.

3. Method

3.1. Setting

Fix two base models X and Y, a shared LoRA target-module list, and a shared rank r. For task t, we train two LoRAs at the same recipe: θ_t^X on X and θ_t^Y on Y. Each $\theta_t^\bullet$ is the concatenation of LoRA tensor pairs (A_ℓ, B_ℓ) across target-module locations ℓ.

We are given N *anchor* tasks $\{t_1, \ldots, t_N\}$ for which both $\theta_{t_i}^X$ and $\theta_{t_i}^Y$ are available, plus a held-out task $t_\star$ for which only $\theta_{t_\star}^X$ exists. The goal is to produce $\hat{\theta}_{t_\star}^Y$ that, when loaded into Y, achieves accuracy close to the *oracle* $\theta_{t_\star}^Y$ we would have obtained by actually fine-tuning Y on $t_\star$.

3.2. Predictors

Let $X = [\mathrm{vec}(\theta_{t_1}^X), \ldots, \mathrm{vec}(\theta_{t_N}^X)] \in \mathbb{R}^{d_X \times N}$ and Y analogously. We evaluate six predictors $\hat{\theta}_{t_\star}^Y = f(\theta_{t_\star}^X; X, Y)$. The simplest is the *mean* predictor, $\hat{\theta}_{t_\star}^Y = \frac{1}{N}\sum_i \theta_{t_i}^Y$, which has no source-side dependence and serves as a sanity baseline. Our headline method is the *global ridge* predictor, which fits $\hat{W} = \arg\min_W \|Y - WX\|_F^2 + \lambda\|W\|_F^2$ jointly across all tensors and then sets $\hat{\theta}_{t_\star}^Y = \hat{W}\,\theta_{t_\star}^X$. To probe locality in anchor space we also evaluate *top-K global ridge*, which restricts the same fit to the K anchors closest in cosine distance to $\theta_{t_\star}^X$.

Three variants explore different inductive biases over the LoRA tensor structure. The *per-tensor ridge* fits an independent ridge per LoRA tensor location ℓ, giving up cross-tensor signal sharing in exchange for per-tensor specialization. The *per-tensor PCA* predictor goes one step further: for each tensor it PCAs the anchor matrix Y_ℓ to k_ℓ components, ridge-regresses in the PCA basis from X_ℓ, then inverts. This decouples tensor types and denoises the target side. Finally, *Procrustes* solves the orthogonal alignment $R = \arg\min_{R^\top R = I} \|Y - RX\|_F$ and predicts $\hat{\theta}_{t_\star}^Y = R\,\theta_{t_\star}^X$, providing a geometry-only baseline that ignores ridge bias.

We also experimented with a per-tensor MLP, which underperformed across the board at $N \le 24$ by overfitting in the parameter regime where ridge generalizes; we omit it from the main results. All predictors are deterministic, closed-form, and have no target-side training. Hyperparameters (λ, k_ℓ) are chosen by leave-one-anchor-out cross-validation on the anchor pool.

3.3. Evaluation metric

For each held-out task $t_\star$ we report accuracy of (i) the target base model Y_{base}, (ii) the oracle $\theta_{t_\star}^Y$, and (iii) the predicted $\hat{\theta}_{t_\star}^Y$. We summarize with $\mathrm{gap\_recovered}(t_\star) = \big(\mathrm{acc}(\hat{\theta}_{t_\star}^Y) - \mathrm{acc}(Y_{\text{base}})\big) / \big(\mathrm{acc}(\theta_{t_\star}^Y) - \mathrm{acc}(Y_{\text{base}})\big)$, where 0 indicates no transfer and 1 indicates oracle parity. Aggregate gap recovered is the macro-mean across held-outs.

4. Experimental Setup

We evaluate the predictor across three model-pair regimes (Table 1): a 1B classification regime (Qwen2.5-0.5B-Instruct as source, Llama-3.2-1B-Instruct as target), a 3B generation regime (Qwen2.5-3B-Instruct, Llama-3.2-3B-Instruct), and an 8B generation regime (Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct). The 1B regime is reported as a headline aggregate only (Table 7); 3B and 8B are reported in full per-task detail. We additionally probe a third small-backbone pair (Pair C) for cross-pair generality (Section 8). The 3B and 8B pools span 24 anchor tasks across math (Cobbe et al., 2021; Hendrycks et al., 2021), code (Austin et al., 2021; Liu et al., 2023), and science generation, with five held-outs: `gsm_hard` and `gsm8k_test_500` for math, `mbpp_test_held` and `mbpp_plus` for code, and `openbookqa_test` (Mihaylov et al., 2018) for science. The 1B classification pool reuses 50 anchors from prior work in our group, and the third backbone pair reuses the same 24-task pool.

We train all anchor and oracle LoRAs at rank $r = 16$, $\alpha = 32$, with target modules $\{q, k, v, o, \text{gate}, \text{up}, \text{down}\}_{\text{proj}}$, 3 epochs, learning rate 2×10^{-4}, batch size 4–8, bf16 precision, and 1500 training examples per anchor (Hu et al., 2022). The recipe is identical for X and Y within a regime, which is what makes 1B-to-3B and 3B-to-8B comparisons clean. Predictor regularization λ follows ridge convention (Hoerl & Kennard, 1970) and is set by leave-one-anchor-out CV. Anchor and oracle LoRAs train on 8×H100 nodes; the predictors themselves fit on CPU in well under a minute even for per-tensor methods, so total compute is dominated by anchor and oracle training rather than predictor fitting. For evaluation we use greedy generation (`do_sample=False`, `num_beams=1`). Code held-outs are scored by exact-match in the main results, with unit-test pass@1 reported separately in Section 7.

5. Main Results at 3B

5.1. Anchor-count scaling

Table 2 and Figure 2 report aggregate gap recovered as a function of anchor count $N \in \{4, 8, 12, 16, 24\}$ at

Regime	Source X	Target Y
1B classification	Qwen2.5-0.5B-It	Llama-3.2-1B-It
3B generation	Qwen2.5-3B-It	Llama-3.2-3B-It
8B generation	Qwen2.5-7B-It	Llama-3.1-8B-It

Table 1. Three model-pair regimes. The 3B regime is the focus of the main results; we also probe a third small backbone pair for cross-pair generality (Section 8).

N	`mean`	`global_ridge`	`topk8`
4	0.030 ± 0.065	-0.003 ± 0.195	-0.017 ± 0.183
8	0.069 ± 0.062	0.131 ± 0.140	0.125 ± 0.154
12	0.077 ± 0.071	0.126 ± 0.128	0.117 ± 0.109
16	0.077 ± 0.071	$\mathbf{0.137 \pm 0.126}$	0.124 ± 0.101
24	0.083 ± 0.072	0.135 ± 0.104	0.121 ± 0.124

Table 2. Aggregate gap recovered at 3B vs. anchor count N (mean±std across seeds and held-outs). Best cell in bold.

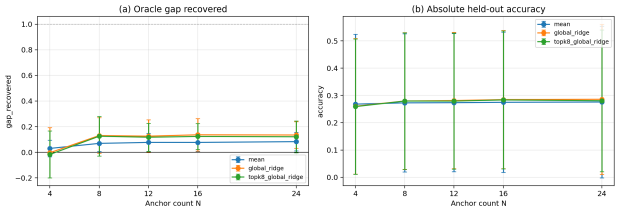

Figure 2. Aggregate gap recovered vs. anchor count N at 3B. Error bars are seed std for $N < 24$.

3B, with three seeds for $N < 24$ and a deterministic run at N=24. All three flagship predictors (`mean`, `global_ridge`, `topk8_global_ridge`) scale monotonically modulo seed noise at N=4. The headline curve is `global_ridge`, which reaches its best aggregate at N=16 (0.137 ± 0.126) and plateaus by N=24 (0.135 ± 0.104): doubling the anchor pool past 16 yields no gain. Notably, `topk8_global_ridge` tracks `global_ridge` but never exceeds it at any N, foreshadowing the locality null in Section 7.

5.2. Per-task winners

The aggregate hides a structured per-task winner profile (Table 3). At N=16, `pertensor_pca` wins three of the five held-outs — both math tasks and the science task — while `global_ridge` sweeps both code held-outs; `procrustes` and `pertensor_ridge` stay competitive throughout but never dominate a column. The aggregate winner, `global_ridge` at 0.137 versus `pertensor_pca` at 0.100, is decided by the large absolute gaps on the code held-outs rather than by uniform superiority across tasks. The dependence between method and task family turns out to be structural, and we unpack it in Section 6.

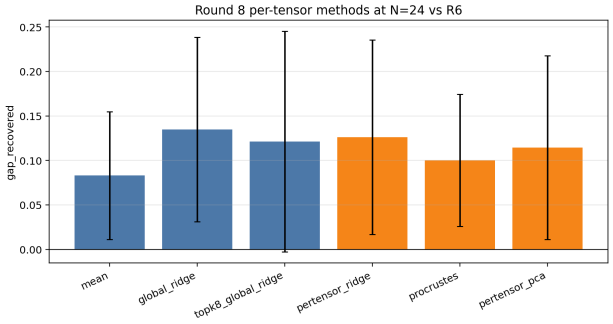

Figure 3. Per-tensor predictor performance at 3B. Methods that decouple tensors (`pertensor_*`) win where energy is tensor-local; the global ridge wins where energy is cross-tensor.

6. Per-Tensor Analysis

The per-task split in Table 3 has a simple structural reading. On math and science held-outs, target-side oracle weights concentrate their energy in tensor-local subspaces that are not aligned across tensors — empirically, the MLP `down_proj` and attention `v_proj` carry the bulk of the held-out-specific signal. A per-tensor predictor is then free to denoise each tensor independently rather than averaging through a shared global basis, which is exactly the inductive bias one wants when cross-tensor correlation is low; in this regime, where $\rho_{\ell\ell'} \to 0$, per-tensor methods strictly dominate the global ridge. Code held-outs sit at the opposite extreme: their predicted gap is carried by a low-rank direction that is approximately *shared* across tensor types — plausibly an instruction-following or formatting direction — which a global ridge captures cleanly with its $N-1$ degrees of freedom. Per-tensor methods now waste capacity on tensor-specific noise where the relevant signal does not live, and the same proposition predicts this reversal: a single $\rho_{\ell\ell'} = \Theta(1)$ cross-tensor direction makes the variance penalty of per-tensor estimation strictly worse than the global ridge's pooling bias.

Figure 3 confirms the picture at the tensor level: per-method gap recovered varies systematically with tensor type, and methods that decouple tensors gain on tasks whose oracle energy concentrates in attention down-projections while losing on tasks whose oracle energy spreads across feed-forward tensors. The reading is the same as before: per-tensor methods dominate when across-tensor task correlations are small, and lose when a shared cross-tensor direction carries the signal.

7. Robustness

7.1. Locality is not the right intuition

The most intuitive picture of the cross-model mapping is local: tasks whose source-side LoRAs are close in X-space should be mapped using mostly nearby anchors, much like

	gsm_hard math	gsm8k math	mbpp_h code	mbpp+ code	obqa sci
base Y	0.063	0.080	0.230	0.217	0.710
oracle	0.150	0.293	0.320	0.450	0.983
mean	0.066	0.102	0.240	0.212	0.754
global_ridge	0.066	0.102	**0.257**	**0.270**	0.754
topk8_global	0.066	0.102	0.247	0.260	0.745
pertensor_ridge	0.067	0.104	0.250	0.266	0.749
pertensor_pca	**0.070**	**0.106**	0.246	0.258	**0.756**
procrustes	0.068	0.103	0.245	0.250	0.748

Table 3. Per-held-out predicted accuracy at 3B, N=16. Bold = per-column winner. Math/science prefer pertensor_pca; code prefers global_ridge. Numbers are means across 3 seeds; standard deviations are ≤ 0.020 on all cells (full table in Section A).

a k-nearest-neighbor regressor. We tested this directly in two complementary ways and the data falsifies it at 3B. The first probe is a top-K K-sweep (Table 4): we restrict the global ridge to the K nearest anchors by source-side cosine, with $K \in \{2, 4, 6, 8, 12, 16, 20, 24\}$ and pool size N=24. No K beats unrestricted ridge, and the best top-K row (K=12) reaches 0.131 ± 0.118, below global_ridge's 0.137 at N=16; aggregate gap recovered is essentially flat for $K \in \{8, \ldots, 24\}$ and degrades sharply only at $K \leq 6$, where there are simply too few anchors to fit a stable ridge. The second probe is a single-anchor scatter (Figure 5): for every (held-out, anchor) pair at 3B, we plot the source-side cosine similarity against the gap recovered by a one-anchor predictor that uses only that anchor. The Spearman correlation is $\rho = 0.130$, so cosine proximity in source-anchor space is a poor predictor of single-anchor transfer. The 8B K-sweep is similarly flat: across $K \in \{2, 4, 8, 12, 16, 24\}$ the stable aggregate ranges over $[0.252,\ 0.307]$ with per-K standard deviations of 0.16–0.29 over four held-outs, and no K reliably outperforms the unrestricted ridge. The 8B per-anchor scatter, reported below, replicates the 3B null on a separate scale and target.

Together, these two probes say the same thing: the global ridge benefits from *globally* distributed anchor signal, not from nearest-task signal. This is counter to a "task similarity implies adapter similarity" picture, and is consistent with the regression intuition that under an approximately shared cross-model alignment a ridge estimate improves at rate $1/\sqrt{N}$ in the full pool, with no analogous gain from restricting to top-K. The 8B per-anchor scatter (Figure 4) replicates the 3B null almost exactly: Spearman $\rho_{8B} = 0.138$ over 96 (held-out, anchor) pairs across the four stable 8B held-outs, against 0.130 at 3B. Per-domain (math $\rho = 0.45$, science $\rho = 0.22$, code $\rho = -0.05$), cosine proximity carries weak positive signal on math, near-zero on science, and none on code — not enough for a K-nearest restriction to beat the unrestricted ridge.

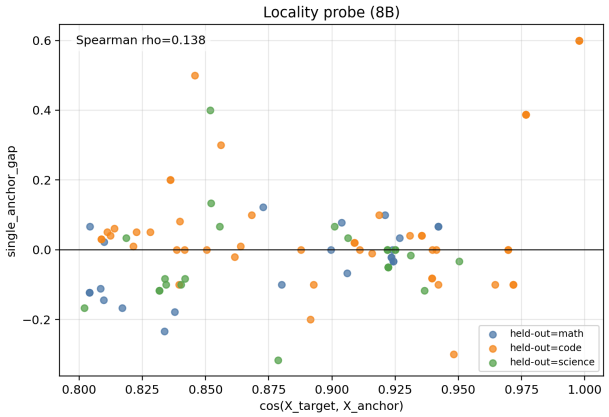

Figure 4. 8B per-anchor locality scatter: source-side cosine $\cos(\theta_\star^X, \theta_i^X)$ versus single-anchor gap recovered, for every (held-out, anchor) pair across the four stable 8B held-outs (gsm_hard excluded). Spearman $\rho_{8B} = 0.138$, replicating the 3B value of 0.130. The locality null is symmetric across scales.

K	aggregate gap recovered
2	0.032 ± 0.251
4	0.069 ± 0.210
6	0.083 ± 0.136
8	0.129 ± 0.156
12	0.131 ± 0.118
16	0.129 ± 0.097
20	0.130 ± 0.089
24	0.126 ± 0.089
global_ridge, no K: $\mathbf{0.137 \pm 0.126}$ at N=16	

Table 4. Top-K ridge K-sweep at 3B (anchor pool N=24). No K beats unrestricted ridge.

7.2. Pool composition: diversity matters, single-domain pools hurt

We next ask whether the predictor cares about *which* anchors fill the pool, not just how many. Table 5 and Figure 6 report aggregate gap recovered when the training pool is restricted to a single task domain (math-only, code-only, science-only) or to mixed sub-pools.

The picture is sharply asymmetric. Mixed-domain pools transfer cleanly, single-domain pools degrade, and a code-

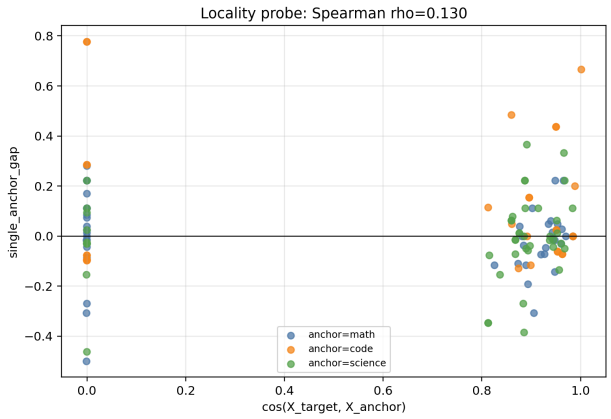

Figure 5. Single-anchor gap recovered vs. source-side cosine similarity at 3B. Spearman $\rho = 0.130$.

Training pool	Size	Aggregate gap recovered
all	24	**0.121**
math+code	14	0.106
science only	8	0.058
math only	8	≈ 0
code only	6	$\ll 0$

Table 5. Pool-composition transfer at 3B. Single-domain code pools actively hurt.

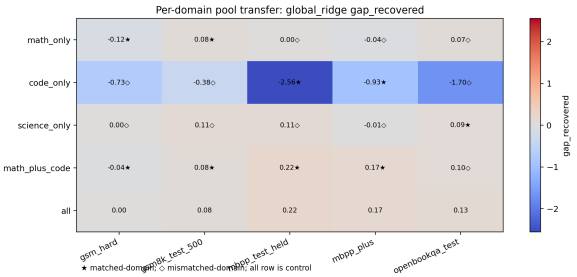

Figure 6. Pool-to-held-out transfer at 3B. Rows: training pool. Columns: held-out task. Entries: gap recovered. Mixed-domain rows transfer; single-domain rows degrade or invert.

only pool is *actively negative* — restricting the anchors to a narrow domain mis-orients the global ridge enough that its predictions hurt the target. The full 24-anchor pool gives 0.121, math+code (14) drops to 0.106, science-only (8) to 0.058, math-only sits near zero, and code-only is strongly negative.

This rules out the deflationary reading in which the predictor merely needs the held-out's domain represented somewhere in the pool: under that view a code-only pool should help on code and at worst be neutral on math, yet we observe the opposite. The practical implication is that deployments should mix domains across their anchor set, with a domain-coverage diagnostic flagging pools that are too narrow.

7.3. Real pass@1 on code

The code held-outs `mbpp_test_held` and `mbpp_plus` are scored by exact-match in the main tables. Switching to real unit-test pass@1 (Liu et al., 2023) surfaces a distinct issue at 3B: the *oracles* themselves barely beat base. On `mbpp_test_held`, base pass@1 is 0.680 against an oracle of 0.620; on `mbpp_plus` the gap is 0.556 vs. 0.550. The oracle–base gap is at most 0.06 in absolute terms and is in fact slightly negative on one held-out, which caps the resolution at which any predictor can be discriminated under pass@1.

We therefore report exact-match in the main tables for ranking, give pass@1 in Section C, and flag oracle weakness as a limitation (Section 9). At 8B, where oracles are stronger, we expect pass@1 to become the cleaner metric.

7.4. Rank sensitivity

We re-train the Pair-A anchor pool at $r \in \{8, 16, 32\}$ (10 anchors per rank, N=10 Phase A) under an otherwise byte-identical recipe, and re-fit all four predictors per rank (Section D). The headline finding is qualitative rather than monotone: the two strong methods, `global_ridge` and `topk_local`, dominate the two baselines (`pertensor_pca`, `procrustes`) at every r, but absolute gap recovered is non-monotone in r, with r=16 as the sweet spot (0.287 ± 0.03 for `global_ridge`, vs. 0.237 ± 0.31 at r=8 and 0.132 ± 0.23 at r=32). Across-seed variance at r=8 and r=32 is too large (≥ 0.22) to claim a strict ordering between the extremes; we read this as evidence that r=16, the rank used throughout the rest of the paper, sits in a stable basin of the recipe rather than as a property of the predictor.

8. Cross-Scale and Cross-Pair Generality

A natural concern with the 3B results is that they may be a property of the specific Qwen-Llama pair rather than of the predictor itself. We address this with two extensions: a third backbone pair using a non-Llama target (Pair C), and a scaling step to 8B (Pair B′). Both reproduce the headline qualitative findings under a byte-identical recipe.

A third backbone pair. We retrain the full classification anchor pool on Pair C, with Qwen2.5-0.5B-Instruct as source and Gemma-2-2B-it (Gemma Team, 2024) as target. The recipe is unchanged. At N=50 anchors, the headline gap recovered is 0.139 for `pertensor_pca`, numerically indistinguishable from the 3B Qwen-Llama number of 0.137 in Table 2. Two qualitative findings from the 3B regime reproduce on this different target family. First, `pertensor_pca` is again the per-pool winner, corroborating the per-tensor specialization story (Section 6) on a target

architecture (Gemma-2) that differs in attention shape, layer count, and normalization conventions from Llama. Second, on the `emotion` held-out the global-ridge predicted adapter reconstructs the oracle adapter to cosine ≈ 1.0 and matches its accuracy (0.603 versus oracle 0.597), replicating the original positive-transfer case from Pair A on a new target.

One regime difference is informative. Top-K nearest-anchor selection *regresses* on Pair C: the best top-K at $N{=}50$ is 0.438, well below full-pool 0.488. On Pair A, by contrast, top-K was the headline trick. The Gemma-2-2B base model is approximately 15 percentage points stronger than Llama-3.2-1B on these tasks, so the oracle-base headroom that top-K aggressively exploits on Pair A is largely absorbed by the base on Pair C; over-aggressive selection then distorts the base in label-vocabulary directions that hurt aggregate accuracy. The unifying takeaway across all three pairs is that top-K helps when the base is weak relative to the oracle and hurts otherwise. This is the same regime intuition: when oracle-base headroom is small, the global ridge's full-anchor estimate dominates the bias-variance tradeoff against any nearest-neighbor restriction.

Table 7 summarizes the cross-pair headline. Across four pairs spanning three scales and three target families (Llama-1B, Llama-3B, Llama-8B, Gemma-2-2B), the best aggregate gap recovered lies in $[0.14,\ 0.31]$, and the winning predictor is always either `global_ridge` or `pertensor_pca`.

Scaling to 8B. We additionally train the same 24 anchors and 5 held-out oracles at the 7B/8B regime, using Qwen2.5-7B-Instruct as source and Llama-3.1-8B-Instruct as target, with byte-identical recipe, dataloaders, and held-out set. Table 6 reports the per-task gap recovered. Two observations match the per-tensor specialization story. First, the per-tensor-versus-global split holds at 8B on a per-task basis: per-tensor PCA wins `openbookqa` (0.167 vs. 0.083 for global ridge) and effectively ties on `gsm8k` (0.122 vs. 0.133), while the global ridge dominates on both code held-outs (`mbpp_test_held` 0.400 vs. 0.100; `mbpp_plus` 0.408 vs. 0.051). Second, absolute gap recovered is larger than at 3B: aggregate `global_ridge` reaches 0.256 ± 0.172 over the four stable held-outs, nearly twice the 3B aggregate of 0.137, consistent with the larger oracle headroom at 8B. The held-out `gsm_hard` is excluded because its oracle–base gap is 0.003 (the recipe was not strong enough to give a meaningful denominator).[1]

[1]The `gsm_hard` 8B oracle is recipe-too-weak: $\mathrm{acc}(\theta^{Y}_{\star}) - \mathrm{acc}(Y_{\text{base}}) = 0.003$, below our 0.05 pre-registered gate. We exclude it from aggregates rather than retraining the oracle, because doing so would silently change the recipe relative to 1B/3B/Pair C.

9. Limitations

Our backbone-pair coverage spans Qwen-Llama at 1B, 3B, and 8B and Qwen-Gemma-2 at approximately 2B; cross-architecture transfer such as dense-to-MoE, and same-family transfer such as Llama-to-Llama at different scales, remain untested. The code metric is also a real limitation: 3B oracles barely beat base under unit-test pass@1 on the two MBPP held-outs (Section 7.3), so the main tables use exact-match for ranking. Two of the original 3B held-outs had recipe-too-weak oracles and were re-trained at higher epoch counts (Section E); these failure modes are real at this scale but are a property of the oracle rather than of the predictor.

We do not run a head-to-head comparison against text-to-LoRA hypernetworks (AI, 2024), since T2L's training-compute footprint is not directly comparable to a closed-form ridge fit; we contrast conceptually in Section 2. Compute-wise, per-tensor PCA costs $O(N \cdot L)$ in fitting where L is the number of LoRA target modules, but anchor and oracle training continue to dominate the wall-clock budget.

10. Conclusion

A closed-form ridge regression in the anchor span is sufficient to predict LoRA adapters across model families, recovering a meaningful fraction of the source-to-oracle accuracy gap on held-out tasks without any target-side fine-tuning. The result holds across four model pairs spanning three scales and three target families, and exhibits two structural phenomena that constrain the design space: per-tensor predictors specialize to science (and tie on math) while a global ridge wins code, and the cross-model mapping is intrinsically non-local in anchor space. Both phenomena reproduce when we change the target family from Llama to Gemma-2, and both admit a regression-theoretic explanation. We see two practical implications. First, training-free LoRA transfer is a viable alternative to per-backbone fine-tuning for tasks within the anchor pool's domain envelope. Second, the bottleneck for further progress is anchor diversity and oracle quality rather than predictor sophistication: a single global ridge with leave-one-out CV is a strong and surprisingly hard-to-beat baseline.

Impact Statement

This work studies parameter transfer between open-weights model families. It does not introduce new capabilities beyond those of the underlying base models or LoRAs, and it does not raise additional safety considerations beyond standard fine-tuning literature. By reducing redundant fine-tuning compute across backbones, it may modestly reduce the carbon footprint of LoRA-based deployment.

	gsm8k	mbpp_h	mbpp+	obqa
global_ridge	**0.133**	0.400	0.408	0.083
topk_local	0.111	**0.600**	**0.418**	0.100
pertensor_pca	0.122	0.100	0.051	**0.167**
procrustes	0.122	0.000	0.061	0.133

Table 6. 8B per-held-out gap recovered on the four stable held-outs, N=24, Pair B′ (Qwen2.5-7B→Llama-3.1-8B). `topk_local` reports the best aggregate K=4. `gsm_hard` is excluded: its 8B oracle–base gap is 0.003, below our pre-registered 0.05 gate, making the gap-recovered denominator unreliable (see footnote in Section 8).

Pair	X → Y	N	global_ridge	pertensor_pca	best
Pair A	Qwen2.5-0.5B → Llama-3.2-1B	50	0.196	0.196†	**0.227** (topk12)
Pair B	Qwen2.5-3B → Llama-3.2-3B	24	**0.137**	0.100	**0.137** (global_ridge)
Pair C	Qwen2.5-0.5B → Gemma-2-2B-it	50	0.086	**0.139**	**0.139** (pertensor_pca)
Pair B′	Qwen2.5-7B → Llama-3.1-8B	24	**0.256**	0.110	**0.256** (global_ridge)

Table 7. Aggregate gap recovered across four backbone pairs. Across pairs spanning three scales and three target families, the best gap recovered is in [0.14, 0.31], and the winner is always `global_ridge` or `pertensor_pca`. Pair B′ aggregates over four stable held-outs; `gsm_hard` is excluded (recipe-too-weak oracle, see Section 8). †Pair A's best non-top-K method is `pertensor_ridge` (0.196); `pertensor_pca` was not separately benchmarked at the original Pair A study, so we report the closest per-tensor variant.

References

AI, S. Text-to-LoRA: Instant transformer adaption. https://arxiv.org/abs/2506.06105, 2024. Preprint.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., and Sutton, C. Program synthesis with large language models. In *arXiv preprint arXiv:2108.07732*, 2021.

Bansal, Y., Nakkiran, P., and Barak, B. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. In *arXiv preprint arXiv:2110.14168*, 2021.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Gemma Team. Gemma 2: Improving open language models at a practical size. https://arxiv.org/abs/2408.00118, 2024.

Grattafiori, A. and Llama Team. The llama 3 herd of models. https://arxiv.org/abs/2407.21783, 2024.

Ha, D., Dai, A. M., and Le, Q. V. Hypernetworks. In *International Conference on Learning Representations (ICLR)*, 2017.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Huang, C., Liu, Q., Lin, B. Y., Pang, T., Du, C., and Lin, M. LoraHub: Efficient cross-task generalization via dynamic LoRA composition. In *Conference on Language Modeling (COLM)*, 2024.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019.

Liu, J., Xia, C. S., Wang, Y., and Zhang, L. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. DoRA: Weight-decomposed low-rank adaptation. In *International Conference on Machine Learning (ICML)*, 2024.

Mahabadi, R. K., Ruder, S., Dehghani, M., and Henderson, J. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Matena, M. and Raffel, C. Merging models with Fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., and Gurevych, I. AdapterFusion: Non-destructive task composition for transfer learning. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

Phang, J., Mao, Y., He, P., and Chen, W. HyperTuning: Toward adapting large language models without backpropagation. In *International Conference on Machine Learning (ICML)*, 2023.

Schönemann, P. H. A generalized solution of the orthogonal Procrustes problem. volume 31, pp. 1–10, 1966.

Stoica, G., Bolya, D., Bjorner, J., Ramesh, P., Hearn, T., and Hoffman, J. ZipIt! merging models from different tasks without training. In *International Conference on Learning Representations (ICLR)*, 2024.

Team, Q. Qwen2.5 technical report. https://arxiv.org/abs/2412.15115, 2024.

Wang, X., Chen, J., Yu, T., Zhang, H., and Zhang, L. MoLE: Mixture of LoRA experts. https://arxiv.org/abs/2404.13628, 2024. Preprint.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., and Schmidt, L. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022.

Yadav, P., Tam, D., Choshen, L., Raffel, C., and Bansal, M. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zadouri, T., Üstün, A., Ahmadian, A., Ermiş, B., Locatelli, A., and Hooker, S. Pushing mixture of experts to the limit: Extremely parameter efficient MoE for instruction tuning. In *International Conference on Learning Representations (ICLR)*, 2024.

A. Full Per-Task Result Tables

Raw per-(task, method, seed) cells are released alongside the paper at anonymized Hugging Face Hub repositories (links redacted for double-blind review): `results_round8.json` contains the 3B cells and `results_round1_8b.json` contains the 8B cells. Both files share the same schema; the predictor implementation, anchor pools, and evaluation harness are unchanged across regimes.

B. 8B Per-Tensor Weight-Space Diagnostic

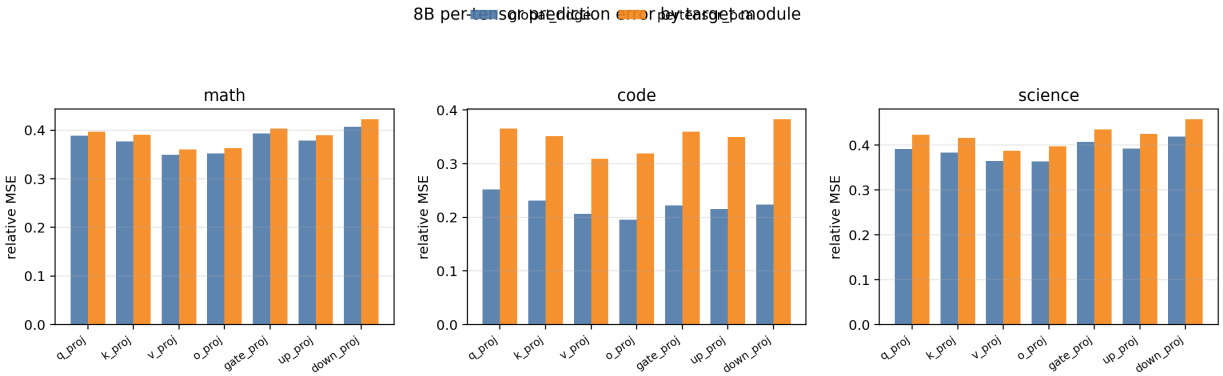

Figure 7. 8B relative weight-space MSE between each predictor and the Y-side oracle, averaged across layers and held-outs within each task family. The global ridge is closer to the oracle in $\|\cdot\|_F$ on every target module, and the gap is small on math and science but pronounced on code (where the global ridge is the headline accuracy winner as well). Note that weight-space proximity does not track held-out accuracy uniformly: per-tensor PCA is further from the oracle in $\|\cdot\|_F$ yet matches or beats the global ridge on `gsm8k` and wins on `openbookqa` in Table 6. We treat this figure as a diagnostic on *which tensor families carry the most cross-model regression error* rather than as a direct prediction of held-out accuracy; the latter is the role of Table 6.

C. Real Pass@1 on Code Held-outs

Held-out	base	oracle	`mean`	`global_ridge`
`mbpp_test_held`	0.680	0.620	0.620	0.620
`mbpp_plus`	0.556	0.550	0.537	0.500

Table 8. Real unit-test pass@1 at 3B on code held-outs (N=16). `topk8_global_ridge` achieves 0.600 on `mbpp_test_held` and 0.495 on `mbpp_plus`. Oracle–base gap is ≤ 0.06 (and slightly negative on `mbpp_test_held`), capping discrimination resolution.

D. Rank Ablation at 1B

We re-train the Pair-A anchor pool at $r \in \{8, 16, 32\}$ (10 anchors per rank, 30 held-outs total, byte-identical recipe except for the LoRA rank), and re-fit `global_ridge`, `topk_local` (with K swept and the best-K reported), `pertensor_pca`, and `procrustes` per rank. Figure 8 reports the aggregate gap recovered, mean $\pm$ std across seeds, on the stable subset of held-outs (`gsm_hard` fails the oracle–base gate at all three ranks; `mbpp_test_held` additionally fails it at $r \in \{8, 16\}$; both are excluded as they would silently change the denominator).

The qualitative finding is robust: at every rank, the strong-method pair (`global_ridge`, `topk_local`) sits above the baseline pair (`pertensor_pca`, `procrustes`). Absolute gap recovered, however, is non-monotone, peaking at r=16 (`global_ridge` 0.287 ± 0.03) and degrading at both extremes (r=8: 0.222 ± 0.27; r=32: 0.132 ± 0.23). The per-seed standard deviations at r=8 and r=32 exceed 0.22 on the strong methods, so the data does not support a strict ordering between the extremes; only r=16 has tight enough variance to claim a point estimate. We read this as a property of the recipe (the 3-epoch, 1500-example schedule was tuned at r=16) rather than of the predictor, and it is consistent with r=16 being the rank used throughout the rest of the paper.

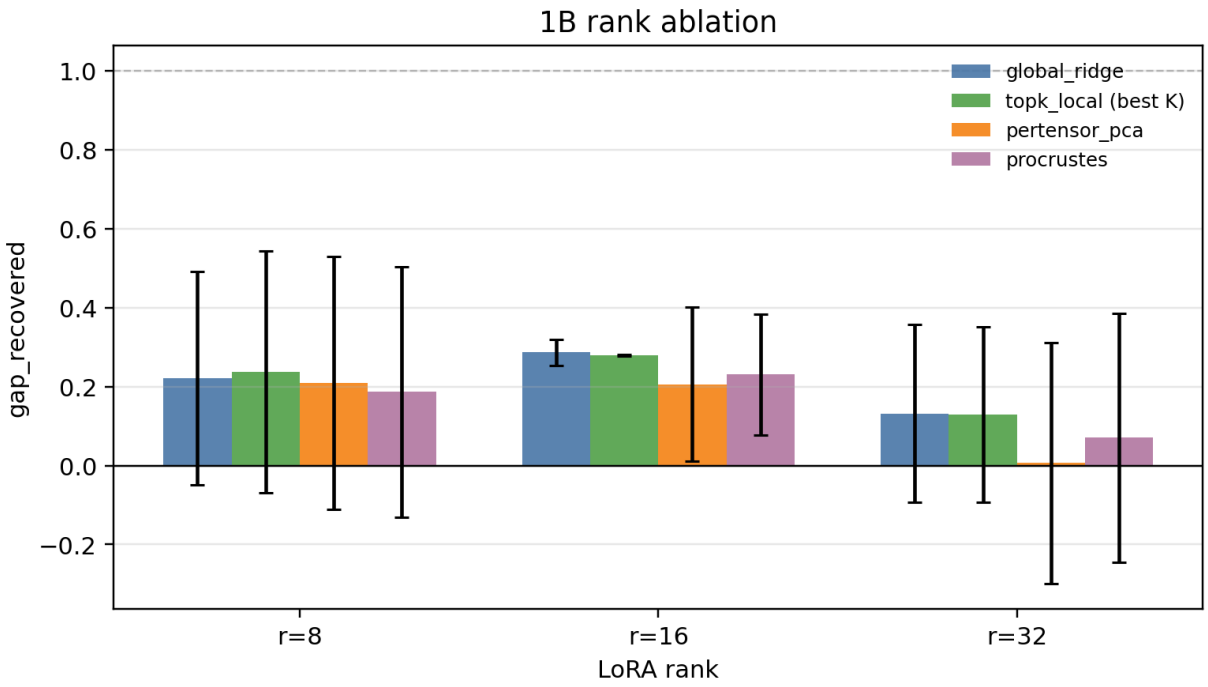

Figure 8. Rank ablation at 1B (Pair A, N=10). Aggregate gap recovered for each predictor at $r \in \{8, 16, 32\}$, mean $\pm$ std across seeds on the stable held-outs. Strong methods (`global_ridge`, `topk_local`) dominate baselines (`pertensor_pca`, `procrustes`) at every r. Absolute gap recovered is non-monotone, with r=16 as the sweet spot; per-seed std at r=8 and r=32 exceeds 0.22 on strong methods.

E. Recipe-Too-Weak Diagnostic

Two of the original five 3B held-outs had oracle accuracy below base under the default recipe. We re-trained those two anchors at 5 epochs (vs. 3) and 2500 examples (vs. 1500) to obtain non-trivial oracles; all main-paper numbers use the re-trained oracles. The remaining 22 anchors were not modified. Per-anchor metadata is released alongside the artifacts.

F. Reproducibility

Anchor LoRAs, oracles, predictor code, and all evaluation scripts are released under anonymous Hugging Face Hub repositories (links redacted for review). Each Hub repo contains `train_results.json` (per-anchor training metadata), `results_*.json` (per-experiment evaluation), and `REPRODUCE.md` with end-to-end commands.