ROTATION-PRESERVING SUPERVISED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning (SFT) improves in-domain performance but can degrade out-of-domain (OOD) generalization. Prior work relates this degradation to changes in dominant singular subspaces of pretrained weight matrices. Identifying dominant loss-sensitive subspaces from full Hessian or Fisher information is computationally expensive at LLM scale. In this work, we propose Rotation-Preserving Supervised Fine-Tuning (RPSFT), which anchors the dominant top-k product block in the pretrained SVD basis, motivated by the concentration of empirical-Fisher gradient energy in leading pretrained coordinates. This block-level anchor empirically reduces dominant-subspace rotation while preserving task adaptation. Across model families and sizes, RPSFT improves the in-domain/OOD trade-off over standard SFT and strong SFT baselines on math and non-math adaptation, and provides stronger initializations for downstream RL fine-tuning.

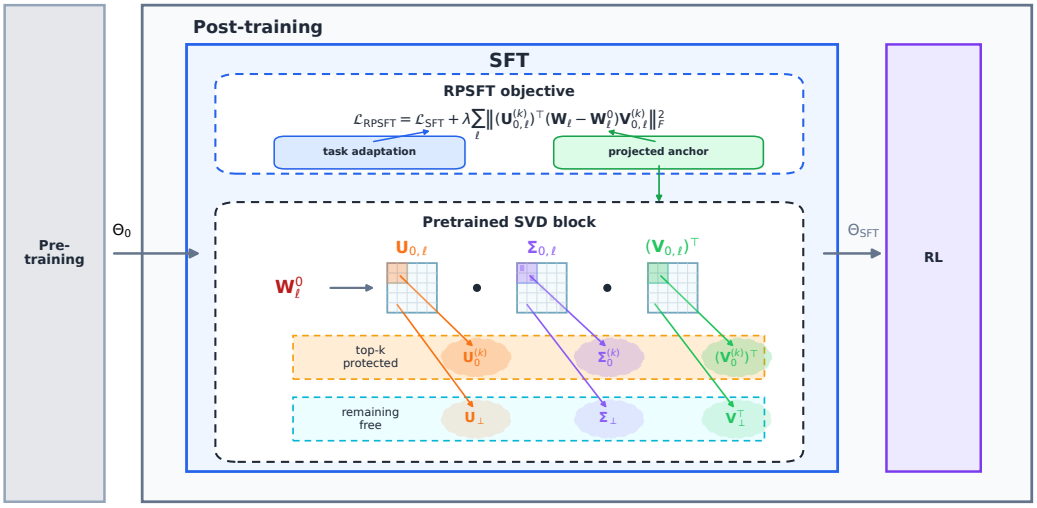

Figure 1: Method overview. RPSFT modifies SFT by adding a projected-block anchor of strength λ in the pretrained SVD basis while retaining full-parameter task adaptation. The top-k pretrained singular block is protected, complementary directions remain free, and the resulting θ_{SFT} initializes downstream reinforcement learning fine-tuning.

1 INTRODUCTION

Large language models are commonly post-trained with supervised fine-tuning (SFT) followed by reinforcement learning fine-tuning (RLFT). SFT improves tasks represented in the post-training data and can degrade out-of-domain (OOD) capabilities through excessive specialization (Zhu et al., 2025c; Wu et al., 2025). Prior work links this degradation to geometric drift, especially rotations of dominant singular directions in pretrained weight matrices (Jin et al., 2025; Zhu et al., 2025b). These observations motivate stabilizing the transformations associated with leading pretrained singular directions during task adaptation.

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.

This motivates our question: *Can supervised fine-tuning preserve OOD generalization by limiting unnecessary geometric drift while retaining task adaptation?* We introduce *Rotation-Preserving Supervised Fine-Tuning* (RPSFT), a simple regularization method motivated by the measured concentration of empirical-Fisher gradient energy in dominant pretrained product subspaces, to address this question. RPSFT softly anchors accumulated displacement in the pretrained top-k singular-vector block of each selected weight matrix. The pretrained weights supply its reference, and all model parameters remain trainable throughout adaptation.

Concretely, for each selected matrix $\mathbf{W}$, RPSFT precomputes the pretrained top-k left and right singular-vector bases $\mathbf{U}_0^{(k)}$ and $\mathbf{V}_0^{(k)}$, and adds the weighted squared Frobenius norm $\lambda\|(\mathbf{U}_0^{(k)})^\top(\mathbf{W}-\mathbf{W}^0)\mathbf{V}_0^{(k)}\|_F^2$ to the SFT loss. This formulation retains the full-parameter model class and adds a restoring gradient within the selected product block. Cross and complementary blocks remain available for adaptation. Figure 1 summarizes the overall RPSFT post-training workflow and Algorithm 1 summarizes the algorithm.

We evaluate full-parameter RPSFT across Llama (Grattafiori et al., 2024) and Qwen (Team, 2024) checkpoints on math and non-math adaptation. RPSFT improves the ID/OOD balance over SFT, leads the evaluated retention baselines on both non-math tasks, and provides stronger downstream DAPO and GRPO initializations. Protected-block drift analysis, local loss diagnostics, and weight- and representation-space measurements characterize its adaptation–retention behavior.

Our methodological contribution is soft anchoring of accumulated displacement in pretrained singular-vector product coordinates. Sculpting Subspaces (Nayak et al., 2025) projects instantaneous gradients; RPSFT instead permits task-driven changes in the protected block while penalizing their magnitude. Both approaches retain cross-block adaptation. Matched hard-block and random-block controls on non-math tasks test these two design choices: soft protection and pretrained direction selection.

2 PRELIMINARIES

Modern post-training for reasoning language models typically proceeds in two stages: supervised fine-tuning (SFT) on curated instruction data, followed by reinforcement learning fine-tuning (RLFT) on task-level rewards (DeepSeek-AI, 2025; OpenAI et al., 2024; Bai et al., 2023).

SFT and PEFT. Supervised fine-tuning adapts pretrained parameters θ_0 on labeled pairs $\mathcal{D} = \{(x, y)\}$ by minimizing the negative log-likelihood $\mathcal{L}_{\mathrm{SFT}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[-\log \pi_\theta(y \mid x)]$, where x is an input prompt, y is the target response, and $\pi_\theta(y \mid x)$ is the model distribution with parameters θ initialized from θ_0. In this work, the main setting is full-parameter SFT, where all model parameters are updated. We include vanilla LoRA (Hu et al., 2021) as a PEFT baseline in Appendix D.

RLFT and DAPO. After SFT, the model can be further optimized with reinforcement learning. We use DAPO (Yu et al., 2025) that samples a group of G responses from the rollout policy $\pi_{\theta_{\mathrm{old}}}(\cdot \mid x)$, computes a normalized group-relative advantage $\hat{A}_i = \frac{r_i - \frac{1}{G}\sum_{j=1}^{G} r_j}{\mathrm{std}(\{r_j\}_{j=1}^{G})+\varepsilon}$ with $r_i = r(x, y_i)$ and small numerical constant ε, and uses the token-level PPO ratio (Schulman et al., 2017) $\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(y_{i,t}|x,y_{i,<t})}$. The resulting surrogate objective is

$$\mathcal{J}_{\mathrm{DAPO}}(\theta) = \mathbb{E}_{x,\{y_i\}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i,t}\min\Big(\rho_{i,t}\hat{A}_i,\ \mathrm{clip}\big(\rho_{i,t}, 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}}\big)\hat{A}_i\Big)\right]. \tag{1}$$

Here i indexes the G sampled responses for each prompt, t ranges over the valid tokens of response y_i, $|y_i|$ is the response length, and $\epsilon_{\mathrm{low}}, \epsilon_{\mathrm{high}}$ are the PPO clipping thresholds. DAPO normalizes the clipped policy objective by the total number of valid tokens across sampled responses; GRPO uses per-response normalization.

3 Dominant Subspaces and Capability Retention

Post-training balances *rapid adaptation* to the target task and *capability retention* from the pretrained model (Lyle et al., 2023; Lu et al., 2025). Prior work studies connections between weight spectra and loss curvature (Haink, 2023) and associates dominant-subspace rotation with OOD degradation (Jin et al., 2025). We analyze this trade-off through local loss changes and sensitivity in pretrained product coordinates. Let $w_0 \in \mathbb{R}^d$ denote the vectorized pretrained parameters, let $w' = w_0 + \Delta w$, and let ℓ_{id} and ℓ_{ret} denote the target-task and retained-capability losses. For $\tau \in \{\mathrm{id}, \mathrm{ret}\}$, a second-order expansion gives

$$\ell_\tau(w') - \ell_\tau(w_0) = g_\tau^\top \Delta w + \frac{1}{2}\Delta w^\top H_\tau \Delta w + \mathcal{E}_\tau(\Delta w), \qquad g_\tau = \nabla \ell_\tau(w_0), \quad H_\tau = \nabla^2 \ell_\tau(w_0). \tag{2}$$

If the Hessian is locally Lipschitz, $\mathcal{E}_\tau(\Delta w) = O(\|\Delta w\|_2^3)$. For the retained loss, a negative first-order term indicates beneficial transfer and a positive term indicates interference. The quadratic term records signed directional curvature. Appendix D.8 evaluates this expansion along Qwen2.5-7B Base→SFT and SFT→RL directions at interpolation distances $t = \pm 0.1$. The quadratic predictor matches the loss-change sign in all eight tests with small absolute error in NLL.

For a positive-semidefinite measure of local model-distribution sensitivity, let $\mathbf{F}_{\mathrm{ret}}$ be the model Fisher at w_0 under a fixed retained-input distribution and define

$$Q_{\mathrm{ret}}(\Delta w) = \frac{1}{2}\Delta w^\top \mathbf{F}_{\mathrm{ret}} \Delta w. \tag{3}$$

This is the quadratic term of the expected forward KL divergence from the pretrained predictive distribution to the updated distribution (Martens, 2020). The retained-label loss expansion in Eq. (2) includes the corresponding first-order transfer and Hessian terms.

The Fisher identity equates the model Fisher with the expected negative log-likelihood Hessian when targets are drawn from the reference model and differentiation and expectation can be interchanged (Martens, 2020). For observed targets, the empirical Fisher is the sum or average of per-example gradient outer products and measures gradient energy (Kunstner et al., 2019). These quantities provide complementary perspectives on local sensitivity. RPSFT uses pretrained singular bases to select its protected coordinates; the gradient-energy diagnostic evaluates how these coordinates overlap with observed loss gradients.

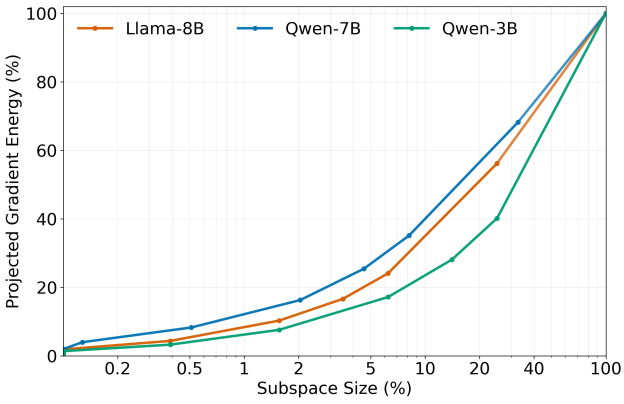

Figure 2: Fisher-projected total sum of squared gradient norms (gradient energy) captured by top-$r \times r$ SVD blocks. The x-axis reports $x(r) = \frac{r^2}{R^2}$, the strict top-$r \times r$ SVD-block size relative to the full $R \times R$ block, and the y-axis reports $y(r) = \frac{\sum_t \|\mathbf{U}_r^\top \mathbf{G}_t \mathbf{V}_r\|_F^2}{\sum_t \|\mathbf{G}_t\|_F^2}$, the Fisher-projected gradient-energy percentage, where $\mathbf{G}_t \in \mathbb{R}^{m \times n}$ denotes the gradient matrix of the t-th sample.

Figure 2 shows this overlap for the layer-1 attention query projection in Llama-8B, Qwen-7B, and Qwen-3B. A leading block occupying roughly 5% of the SVD-core coordinates captures about 20% of the measured gradient energy. This enrichment motivates leading-block protection. The main experiments use $k = 768$, approximately 20% of the singular-basis dimension of the principal 7B–8B matrices, and transfer the same rank cap to Qwen-3B. Appendix F.1 defines the diagnostic, and Section 5.4 tests soft protection and pretrained direction selection.

Let P denote the block-diagonal orthogonal projector onto the selected pretrained product blocks, with zero blocks for parameters outside the eligible set, and let $\delta = w^\star - w_0$. The RPSFT penalty is

$\lambda\|P\delta\|_2^2$. For $\lambda > 0$, at a stationary point of the regularized objective,

$$P\delta = -\frac{1}{2\lambda} P\nabla\mathcal{L}_{\text{task}}(w^\star). \tag{4}$$

Thus, protected displacement is tied directly to the projected task gradient. Its Fisher-weighted sensitivity satisfies

$$\frac{1}{2}(P\delta)^\top \mathbf{F}_{\text{ret}}(P\delta) \le \frac{\|P\mathbf{F}_{\text{ret}}P\|_2 \, \|P\nabla\mathcal{L}_{\text{task}}(w^\star)\|_2^2}{8\lambda^2}. \tag{5}$$

The total sensitivity combines this protected contribution, its interaction with $(I - P)\delta$, and the complementary contribution. Appendix C gives this decomposition, its connection to retained loss, and the protected-block dynamics. RPSFT therefore supplies a soft restoring force in selected pretrained coordinates, while full-parameter optimization jointly adapts all blocks. Principal-angle measurements quantify the resulting singular-subspace alignment.

4 METHOD

Rotation-Preserving SFT (RPSFT) anchors the transformation between the leading pretrained input and output singular subspaces, building on Section 3. All reported RPSFT results use full-parameter fine-tuning (FPFT) and sum the eligible-matrix penalties uniformly.

Setup and Notation. Let θ_0 be the pretrained model and let $\mathcal{M}$ contain every eligible two-dimensional weight, including attention projections, MLP projections, and embedding matrices. For $\mathbf{W}_\ell^0 \in \mathbb{R}^{m_\ell \times n_\ell}$, we set $k_\ell = \min(k, m_\ell, n_\ell)$, compute the pretrained SVD $\mathbf{W}_\ell^0 = \mathbf{U}_\ell^0 \mathbf{\Sigma}_\ell^0 (\mathbf{V}_\ell^0)^\top$, and cache $\mathbf{U}_{0,\ell}^{(k_\ell)}$ and $\mathbf{V}_{0,\ell}^{(k_\ell)}$. Thus, matrices smaller than the default $k = 768$ use their maximum valid rank.

Rotation-Preserving Regularization. For the current weight $\mathbf{W}_\ell$, define its dominant-to-dominant block in the fixed pretrained coordinates as

$$S_\ell(\mathbf{W}_\ell) = (\mathbf{U}_{0,\ell}^{(k_\ell)})^\top \mathbf{W}_\ell \mathbf{V}_{0,\ell}^{(k_\ell)}, \qquad S_\ell^{\text{ref}} = S_\ell(\mathbf{W}_\ell^0). \tag{6}$$

RPSFT anchors accumulated displacement in this fixed pretrained product block, while the task loss jointly optimizes the protected, cross, and complementary blocks. Every model weight remains trainable. Here, rotation preservation denotes the empirically reduced dominant left-subspace rotation, measured through principal angles; Appendix C.3 details how the jointly adapted blocks determine alignment.

This viewpoint also clarifies the boundary cases. At $k = 0$ or $\lambda = 0$, RPSFT recovers standard SFT. At maximum projected rank, the projected penalty equals full weight-space ℓ_2 anchoring for square matrices and the maximal SVD-core anchor for rectangular matrices; L2 Init (Kumar et al., 2024) applies the explicit full-weight anchor. Appendix C formalizes these cases.

In the full-parameter setting, the total loss is:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \lambda \sum_{\ell \in \mathcal{M}} \left\| S_\ell(\mathbf{W}_\ell) - S_\ell^{\text{ref}} \right\|_F^2, \tag{7}$$

where $\lambda \ge 0$ controls the anchoring strength. The penalty is zero at initialization and grows with displacement inside the pretrained top-k_ℓ product subspace. Appendix Algorithm 1 summarizes FPFT training, and Appendix A.2 gives an optional LoRA formulation.

5 EXPERIMENTS

Benchmarks and evaluation. The primary supervised stage is trained on OpenR1-Math (Hugging Face, 2025), and downstream DAPO is initialized from those checkpoints using the DAPO-Math-17k RL set. In-domain results cover AIME24, AIME25, AMC23, MATH-500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024), while out-of-domain

results cover GPQA (Rein et al., 2023), IFEval-loose (Zhou et al., 2023), MMLU-Pro (Wang et al., 2024b), SuperGPQA (Team et al., 2025), Safety Benchmark, and TruthfulQA (Lin et al., 2022). Figure 3 summarizes the pipeline; Appendix E.1 specifies subsets, scoring, and aggregation. Separate CommonsenseQA (Talmor et al., 2019) and ProofWriter (Tafjord et al., 2021) runs use the task-specific held-out set for ID and the IFEval/MMLU-Pro/SuperGPQA mean for OOD.

Base models and baselines. We evaluate Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-7B-Instruct, and Qwen2.5-3B-Instruct (Team, 2024), with DeepSeek-7B in Appendix D.11. Baselines include Base, full-parameter SFT, Dynamic Fine-Tuning (DFT) (Wu et al., 2025), importance-weighted SFT (IW) (Qin & Springenberg, 2025), and the retention methods EWC (Kirkpatrick et al., 2017), O-LoRA (Wang et al., 2023), Sculpting Subspaces (Nayak et al., 2025), Protected SFT (Jin et al., 2025), and principal-subspace preservation (PSP) (Franke et al., 2024). We focus the broader three-seed retention comparison on non-math (Table 4) to allocate compute efficiently given longer math training schedules; Table 5 provides a complementary math comparison. Appendices D.1 and D.4 give configurations; Figure 5 and Table 12 give rank and strength sweeps.

Training datasets Math SFT uses about 25k processed OpenR1-Math samples (Hugging Face, 2025). Non-math SFT uses CommonsenseQA (Talmor et al., 2019) and ProofWriter (Tafjord et al., 2021). DAPO uses DAPO-Math-17k and ID-validation checkpoint selection; Qwen2.5-7B additionally uses standard GRPO. Each RL run starts from its corresponding supervised checkpoint, isolating the SFT initializer.

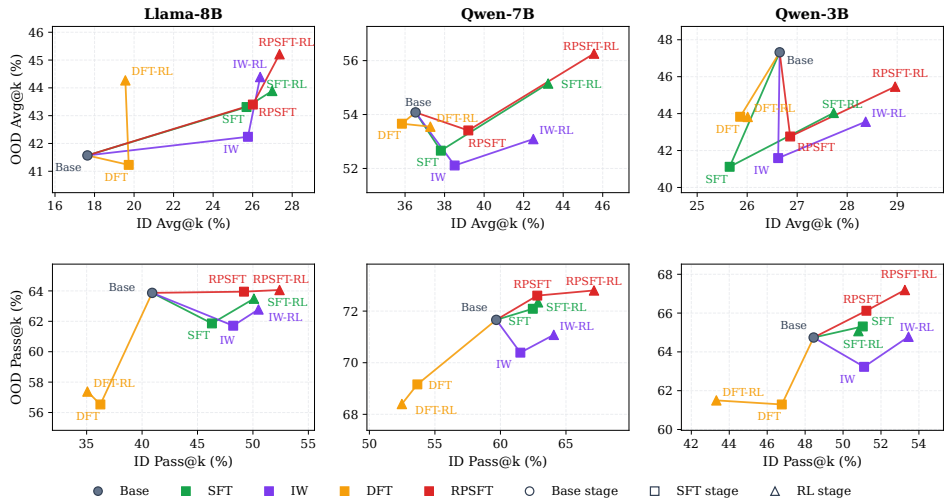

Figure 3: ID/OOD trade-off across SFT and RL stages for RPSFT and baselines. The top row compares ID Avg@k with OOD Avg@k, and the bottom row compares ID Pass@k with OOD Pass@k. Scores average the same six ID and six OOD benchmarks.

5.1 Results on Supervised Fine-tuning

RPSFT improves the adaptation–retention trade-off. Table 1 shows the highest tuned ID Avg@k and Pass@k summaries for RPSFT across all three models, with OOD Pass@k above Base. Its OOD Avg@k exceeds Base on Llama and SFT on Qwen. DFT has higher Qwen OOD Avg@k, while RPSFT has higher ID performance. Across three independently trained Qwen2.5-7B checkpoints, RPSFT improves six-benchmark ID and OOD Avg@k means over SFT by 1.18 and 0.73 percentage points (Appendix D.5). Appendix D.7 reports evaluation-seed stability.

First-order alignment identifies local transfer and interference. Figure 4 reports the layerwise gradient–update inner product $\langle g_{\mathbf{W}}, \Delta\mathbf{W} \rangle$ at the diagnostic reference checkpoint. Negative values indicate a local descent direction for the corresponding dataset, and positive values indicate local interference. The per-layer averages summarize the signed contributions of the selected matrices; the full local loss expansion also includes curvature and higher-order terms. Appendix F.2 defines the reference-point notation and aggregation.

Table 1: **SFT results on in-domain and out-of-domain benchmarks.** "Δ vs. Base" rows show change relative to the base model. Bold marks the strongest tuned result in the same block, and summary-average rows or columns are marked as Avg. For the in-domain benchmarks, we use the same benchmark-specific k for both Avg@k and Pass@k: $k = 16$ for AIME24, AIME25, and AMC23, $k = 4$ for MATH-500 and OlympiadBench, and $k = 8$ for Minerva Math. For out-of-domain benchmarks, we use $k = 4$ and report both Avg@k and Pass@k.

		Llama-3.1-8B-Instruct					Qwen2.5-7B-Instruct					Qwen2.5-3B-Instruct				
Domain	Benchmark	Base	SFT	IW	DFT	RPSFT	Base	SFT	IW	DFT	RPSFT	Base	SFT	IW	DFT	RPSFT
ID Avg@k	AIME24	3.13	**5.00**	3.13	4.17	4.38	12.08	**15.21**	14.17	12.29	14.38	4.79	3.96	**6.67**	2.71	6.46
	AIME25	0.83	5.00	**6.88**	1.88	6.67	7.71	**19.38**	18.54	9.38	18.75	2.29	5.00	**6.04**	4.79	5.00
	AMC23	21.25	32.19	**33.59**	19.53	31.56	52.03	50.47	52.50	48.91	**55.31**	37.03	35.00	**36.25**	34.06	34.84
	MATH-500	43.65	59.45	58.50	46.40	**60.45**	72.85	72.85	72.60	73.35	**74.95**	63.35	60.95	61.05	62.05	**63.35**
	Minerva	22.33	26.88	26.75	20.17	**27.48**	37.09	31.53	**34.05**	33.73	33.82	26.24	22.75	23.71	**26.56**	23.85
	Olympiad	14.63	25.59	25.71	**26.22**	25.52	37.41	37.41	**39.19**	37.41	38.00	26.22	26.26	26.00	25.00	**27.67**
	Avg	17.64	25.69	25.76	19.73	**26.01**	36.53	37.81	38.51	35.84	**39.20**	26.65	25.65	26.62	25.86	**26.86**
	Δ vs. Base	—	↑8.05	↑8.12	↑2.09	↑**8.37**	—	↑1.28	↑1.98	↓0.69	↑**2.67**	—	↓1.00	↓0.03	↓0.79	↑**0.21**
ID Pass@k	AIME24	20.00	**20.00**	16.67	13.33	**20.00**	36.67	**50.00**	46.67	23.33	43.33	20.00	30.00	26.66	23.33	**33.33**
	AIME25	13.33	13.33	**26.67**	16.67	20.00	36.67	36.67	36.67	33.33	**43.33**	20.00	23.33	**30.00**	23.33	23.33
	AMC23	72.50	77.50	80.00	62.50	**85.00**	90.00	87.50	85.00	82.50	**90.00**	80.00	**82.50**	**82.50**	77.50	80.00
	MATH-500	65.00	**78.60**	76.20	61.20	77.20	84.60	87.60	**89.00**	84.20	87.40	79.20	**80.60**	79.60	75.80	79.80
	Minerva	44.85	47.06	48.16	37.50	**50.74**	55.15	**58.09**	55.51	48.53	56.62	50.00	**47.43**	45.22	42.28	47.06
	Olympiad	29.78	41.33	41.63	26.22	**42.22**	54.96	54.96	**56.30**	50.07	56.15	41.48	42.37	42.67	38.37	**43.85**
	Avg	40.91	46.30	48.22	36.24	**49.19**	59.68	62.47	61.52	53.66	**62.81**	48.45	51.04	51.11	46.77	**51.23**
	Δ vs. Base	—	↑5.39	↑7.31	↓4.67	↑**8.28**	—	↑2.79	↑1.85	↓6.02	↑**3.13**	—	↑2.59	↑2.66	↓1.68	↑**2.78**
OOD Avg@k	GPQA	27.62	32.70	**34.32**	26.79	33.48	32.76	**35.66**	34.82	30.47	34.21	27.18	27.73	**29.41**	26.45	28.13
	IFEval	32.30	**32.58**	31.70	31.84	31.93	64.23	54.48	55.91	**61.18**	58.32	58.09	47.97	47.92	**52.59**	49.72
	MMLU-Pro	50.71	**56.07**	52.86	50.00	52.86	62.50	64.64	64.64	66.07	**68.21**	50.36	40.71	42.14	40.71	**43.93**
	SuperGPQA	19.88	**24.38**	22.28	17.72	23.77	25.62	**28.46**	25.99	25.25	27.84	21.36	19.81	19.38	19.26	**21.67**
	Safety	59.43	57.83	**62.08**	60.45	60.55	73.00	70.30	69.80	**72.35**	70.45	65.90	59.98	60.00	**66.03**	60.83
	TruthfulQA	59.50	56.29	50.18	**60.56**	57.82	66.37	62.43	61.48	**66.63**	61.40	61.04	50.55	50.69	**57.93**	52.30
	Avg	41.57	43.31	42.24	41.23	**43.40**	54.08	52.66	52.11	**53.66**	53.41	47.32	41.12	41.59	**43.83**	42.76
	Δ vs. Base	—	↑1.73	↑0.66	↓0.35	↑**1.83**	—	↓1.42	↓1.97	↓**0.42**	↓0.67	—	↓6.20	↓5.73	↓**3.49**	↓4.56
OOD Pass@k	GPQA	61.38	65.40	**69.20**	57.14	68.08	59.82	**65.63**	64.29	58.04	65.18	54.46	61.61	**65.40**	55.13	62.05
	IFEval	35.86	35.49	35.30	34.57	**35.67**	76.71	69.50	69.50	73.01	**73.38**	70.61	63.03	62.85	**65.62**	65.25
	MMLU-Pro	81.43	75.71	77.14	75.71	**81.43**	81.43	80.00	78.57	80.00	**84.29**	72.86	70.00	68.57	60.00	**72.86**
	SuperGPQA	48.15	**54.81**	49.14	38.02	53.58	49.38	53.83	51.11	45.93	**54.57**	49.63	45.68	43.70	44.94	**48.89**
	Safety	76.60	67.80	**73.10**	64.60	68.60	84.10	**83.30**	80.80	78.30	79.80	69.10	**80.20**	69.70	68.20	76.20
	TruthfulQA	79.82	71.93	66.37	69.15	**76.32**	78.51	**80.26**	78.07	79.68	78.36	71.78	71.35	69.15	**73.83**	71.49
	Avg	63.87	61.86	61.71	56.53	**63.95**	71.66	72.09	70.39	69.16	**72.60**	64.74	65.31	63.23	61.29	**66.12**
	Δ vs. Base	—	↓2.02	↓2.17	↓7.34	↑**0.07**	—	↑0.43	↓1.27	↓2.50	↑**0.94**	—	↑0.57	↓1.51	↓3.45	↑**1.38**

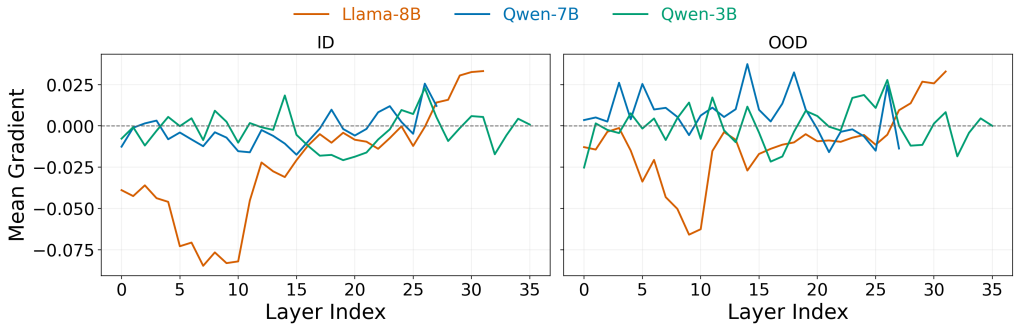

Figure 4: Layerwise first-order signal on the in-domain (left) and out-of-domain (right) distributions across Llama-8B, Qwen-7B, and Qwen-3B.

5.2 Results on RL Fine-Tuning

Stronger SFT initializers transfer better to downstream RL. Table 2 shows that RPSFT gives the best final ID Avg@k and Pass@k on Llama-3.1-8B and Qwen2.5-7B. On Qwen2.5-3B, RPSFT still gives the best final Avg@k, while IW is marginally higher on Pass@k (53.45 vs. 53.26). Thus, the improved SFT-stage ID/OOD trade-off carries into the final DAPO checkpoints across the evaluated model families.

RPSFT gives the strongest final OOD RL performance. Table 3 shows that RPSFT achieves the best final OOD Avg@k and Pass@k summary averages across three models. IW often has a larger

Table 2: **Downstream RL (DAPO) results on the six ID math benchmarks**, following Table 1's notation. Avg@k and Pass@k blocks use init→DAPO entries; bold marks the highest final value among methods.

Metric	Method	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	$\overline{\text{Avg}}$
Llama-3.1-8B-Instruct								
Avg@k	SFT	5.00→**6.46**	5.00→5.42	32.19→**34.38**	59.45→**61.05**	26.88→**29.09**	25.59→25.56	25.69→26.99
	IW	3.13→5.21	6.88→6.46	33.59→32.81	58.50→59.60	26.75→28.77	25.71→25.44	25.76→26.38
	DFT	4.17→3.13	1.88→0.83	19.53→24.53	46.40→48.50	20.17→22.98	26.22→17.41	19.73→19.56
	RPSFT	4.38→6.25	6.67→**7.92**	31.56→33.59	60.45→60.80	27.48→28.54	25.52→**27.03**	26.01→**27.36**
Pass@k	SFT	20.00→20.00	13.33→26.67	77.50→82.50	78.60→77.80	47.06→51.47	41.33→42.07	46.30→50.09
	IW	16.67→23.33	26.67→**30.00**	80.00→77.50	76.20→78.40	48.16→51.84	41.63→41.93	48.22→50.50
	DFT	13.33→10.00	16.67→10.00	62.50→60.00	61.20→63.40	37.50→38.60	26.22→28.44	36.24→35.07
	RPSFT	20.00→**30.00**	20.00→23.33	85.00→**85.00**	77.20→**79.40**	50.74→**52.94**	42.22→**43.70**	49.19→**52.40**
Qwen2.5-7B-Instruct								
Avg@k	SFT	15.21→18.75	19.38→20.83	50.47→58.12	72.85→79.35	31.53→38.97	37.41→43.44	37.81→43.24
	IW	14.17→17.29	18.54→20.00	52.50→60.00	72.60→77.95	34.05→36.76	39.19→42.96	38.51→42.49
	DFT	12.29→11.04	9.38→12.29	48.91→53.13	73.35→74.50	33.73→35.94	37.41→36.81	35.84→37.28
	RPSFT	14.38→**21.25**	18.75→**22.71**	55.31→**62.19**	74.95→**80.50**	33.82→**39.94**	38.00→**46.78**	39.20→**45.56**
Pass@k	SFT	50.00→46.67	36.67→33.33	87.50→90.00	87.60→89.60	58.09→58.09	54.96→59.55	62.47→62.87
	IW	46.67→46.66	36.67→**43.33**	85.00→92.50	89.00→88.00	55.51→55.14	56.30→58.81	61.52→64.07
	DFT	23.33→23.33	33.33→23.33	82.50→82.50	84.20→84.20	48.53→52.94	50.07→48.59	53.66→52.48
	RPSFT	43.33→**53.33**	43.33→**43.33**	90.00→**95.00**	87.40→**90.20**	56.62→**59.19**	56.15→**61.78**	62.81→**67.14**
Qwen2.5-3B-Instruct								
Avg@k	SFT	3.96→4.79	5.00→5.42	35.00→37.50	60.95→**65.90**	22.75→25.32	26.26→27.44	25.65→27.73
	IW	6.67→6.25	6.04→**6.25**	36.25→**40.94**	61.05→64.65	23.71→24.54	26.00→27.59	26.62→28.37
	DFT	2.71→2.50	4.79→4.38	34.06→39.53	62.05→60.70	26.56→23.67	25.00→25.30	25.86→26.01
	RPSFT	6.46→**6.46**	5.00→6.04	34.84→40.00	63.35→65.75	23.85→**26.47**	27.67→**29.04**	26.86→**28.96**
Pass@k	SFT	30.00→23.33	23.33→**33.33**	82.50→75.00	80.60→**81.00**	47.43→47.79	42.37→44.44	51.04→50.81
	IW	26.66→**33.33**	30.00→**33.33**	82.50→**85.00**	79.60→80.40	45.22→45.96	42.67→42.66	51.11→**53.45**
	DFT	23.33→13.33	23.33→26.67	77.50→72.50	75.80→70.80	42.28→39.71	38.37→36.89	46.77→43.32
	RPSFT	33.33→30.00	23.33→30.00	80.00→82.50	79.80→80.00	47.06→**50.37**	43.85→**46.67**	51.23→53.26

Table 3: **Downstream RL (DAPO) results on the six OOD benchmarks**, following Table 1's notation. Avg@k and Pass@k blocks use init→DAPO entries with $k = 4$; final checkpoints are selected by ID performance.

Metric	Method	GPQA	IFEval	MMLU-Pro	SuperGPQA	Safety	TruthfulQA	$\overline{\text{Avg}}$
Llama-3.1-8B-Instruct								
Avg@k	SFT	32.70→35.10	32.58→**31.79**	56.07→51.07	24.38→26.17	57.83→58.85	56.29→60.34	43.31→43.89
	IW	34.32→**36.72**	31.70→31.15	52.86→55.00	22.28→24.94	62.08→**65.13**	50.18→53.40	42.24→44.39
	DFT	26.79→28.96	31.84→31.01	50.00→**57.86**	17.72→18.77	60.45→64.50	60.56→**64.55**	41.23→44.27
	RPSFT	33.48→34.54	31.93→31.56	52.86→55.71	23.77→**27.41**	60.55→59.25	57.82→62.79	43.40→**45.21**
Pass@k	SFT	65.40→68.08	35.49→**35.12**	75.71→**80.00**	54.81→**56.79**	67.80→66.40	71.93→74.56	61.86→63.49
	IW	69.20→**68.75**	35.30→**35.12**	77.14→77.14	49.14→51.85	73.10→**74.40**	66.37→69.44	61.71→62.78
	DFT	57.14→55.36	34.57→34.01	75.71→72.86	38.02→39.75	64.60→69.20	69.15→73.10	56.53→57.38
	RPSFT	68.08→67.86	35.67→34.75	81.43→78.57	53.58→54.57	68.60→71.40	76.32→**77.19**	63.95→**64.06**
Qwen2.5-7B-Instruct								
Avg@k	SFT	35.66→37.39	54.48→52.82	64.64→71.07	28.46→32.28	70.30→71.90	62.43→65.42	52.66→55.15
	IW	34.82→34.26	55.91→53.37	64.64→67.50	25.99→28.51	69.80→72.05	61.48→62.83	52.11→53.09
	DFT	30.47→33.54	61.18→**59.10**	66.07→64.29	25.25→24.88	72.35→**73.18**	66.63→66.34	53.66→53.55
	RPSFT	34.21→**38.00**	58.32→54.07	68.21→**72.86**	27.84→**33.33**	70.45→71.93	61.40→**67.36**	53.41→**56.26**
Pass@k	SFT	65.63→64.73	69.50→70.06	80.00→84.29	53.83→56.05	83.30→80.40	80.26→78.51	72.09→72.34
	IW	64.29→60.49	69.50→68.39	78.57→**85.71**	51.11→54.32	80.80→**81.10**	78.07→76.46	70.39→71.08
	DFT	58.04→61.61	73.01→**71.35**	80.00→78.57	45.93→45.43	78.30→77.10	79.68→76.32	69.16→68.40
	RPSFT	65.18→**66.52**	73.38→70.43	84.29→82.86	54.57→**58.02**	79.80→79.70	78.36→**79.24**	72.60→**72.80**
Qwen2.5-3B-Instruct								
Avg@k	SFT	27.73→27.12	47.97→46.77	40.71→50.71	19.81→**22.84**	59.98→63.63	50.55→53.18	41.12→44.04
	IW	29.41→29.13	47.92→47.92	42.14→48.93	19.38→20.93	60.00→61.25	50.69→53.22	41.59→43.56
	DFT	26.45→26.90	52.59→**52.73**	40.71→41.07	19.26→19.94	66.03→**63.95**	57.93→**58.33**	43.83→43.82
	RPSFT	28.13→**31.64**	49.72→49.40	43.93→**51.43**	21.67→22.41	60.83→63.15	52.30→54.75	42.76→**45.46**
Pass@k	SFT	61.61→57.37	63.03→63.59	70.00→71.43	45.68→**48.64**	80.20→**78.90**	71.35→70.47	65.31→65.07
	IW	65.40→59.60	62.85→63.22	68.57→**75.71**	43.70→47.41	69.70→73.00	69.15→69.73	63.23→64.78
	DFT	55.13→52.45	65.62→**67.84**	60.00→64.29	44.94→42.47	68.20→66.80	73.83→**75.15**	61.29→61.50
	RPSFT	62.05→**65.18**	65.25→63.40	72.86→**75.71**	48.89→**48.64**	76.20→77.10	71.49→73.10	66.12→**67.19**

init→DAPO gain. RPSFT starts from a stronger OOD checkpoint and remains best after RL; across three independently trained Qwen2.5-7B DAPO runs, it retains higher final ID and OOD Avg@k and Pass@k macro means (Appendix D.6).

The advantage transfers to GRPO. On Qwen2.5-7B, RPSFT-GRPO leads SFT-GRPO on four of six ID metrics with two ties and on five of six OOD metrics. Appendix D.10 gives the full comparison.

5.3 Results beyond Math Adaptation

We train Qwen2.5-7B for two epochs on CommonsenseQA and ProofWriter with three independent training seeds. Table 4 reports the final checkpoints; Appendix D.1 gives benchmark-level results. RPSFT achieves the highest ID and OOD-macro means on both tasks, improving over SFT by 2.05 ID and 3.69 OOD-macro points on CommonsenseQA, and by 0.96 ID and 5.30 OOD-macro points on ProofWriter.

Table 4: **Non-math adaptation at the final two-epoch checkpoints.** Values are mean ± standard error over three training seeds. OOD macro averages IFEval-loose, MMLU-Pro, and SuperGPQA within each seed. Base is fixed; bold marks the highest tuned mean per task.

Adaptation task	Method	ID	OOD macro
CommonsenseQA	Base	79.69	46.24
	SFT	81.46 ± 0.76	34.92 ± 0.40
	EWC	81.08 ± 0.79	35.60 ± 1.27
	Sculpting Subspaces	82.91 ± 0.48	36.02 ± 1.24
	Protected SFT	82.77 ± 0.43	34.34 ± 0.77
	PSP	82.39 ± 0.52	36.68 ± 0.41
	RPSFT	$\mathbf{83.51 \pm 0.46}$	$\mathbf{38.61 \pm 1.11}$
ProofWriter	Base	72.51	46.24
	SFT	93.13 ± 0.54	34.13 ± 0.31
	EWC	92.55 ± 0.21	33.11 ± 0.34
	Sculpting Subspaces	89.88 ± 4.56	38.01 ± 0.90
	Protected SFT	92.95 ± 0.86	34.70 ± 0.78
	PSP	93.20 ± 0.40	37.13 ± 0.19
	RPSFT	$\mathbf{94.09 \pm 0.12}$	$\mathbf{39.43 \pm 1.26}$

Direct retention baselines. RPSFT leads EWC, Sculpting Subspaces, Protected SFT, and PSP on ID and OOD macro for both non-math tasks. The complementary OpenR1-Math comparison under a shared training schedule gives higher RPSFT means than EWC, O-LoRA, and Sculpting Subspaces on 11/12, 10/12, and 11/12 metrics, respectively (Table 5; Appendix D.4).

Table 5: **Direct Qwen2.5-7B/OpenR1-Math baselines.** Benchmark rows report mean ± sample SD over three stochastic evaluation seeds from one trained checkpoint per method. $\overline{\text{Avg}}$ averages the three benchmark means; bold marks each row maximum.

Domain	Metric	Benchmark	SFT	EWC	O-LoRA	Sculpting	RPSFT
ID	Avg@16	AIME24	14.03 ± 1.15	14.10 ± 0.64	10.00 ± 0.63	13.54 ± 0.72	$\mathbf{14.38 \pm 1.46}$
		AIME25	18.61 ± 0.84	14.51 ± 1.15	8.54 ± 0.75	17.78 ± 1.39	$\mathbf{18.89 \pm 0.64}$
		AMC23	49.37 ± 0.98	$\mathbf{55.73 \pm 0.59}$	50.94 ± 0.72	51.51 ± 0.59	52.66 ± 1.02
		$\overline{\text{Avg}}$	27.34	28.11	23.16	27.61	**28.64**
	Pass@16	AIME24	42.22 ± 1.92	35.56 ± 1.92	26.67 ± 3.33	$\mathbf{43.33 \pm 3.33}$	40.00 ± 3.33
		AIME25	37.78 ± 1.92	35.56 ± 3.85	25.56 ± 3.85	32.22 ± 1.92	$\mathbf{41.11 \pm 1.92}$
		AMC23	87.50 ± 2.50	85.83 ± 3.82	85.83 ± 3.82	87.50 ± 2.50	$\mathbf{88.33 \pm 1.44}$
		$\overline{\text{Avg}}$	55.83	52.32	46.02	54.35	**56.48**
OOD	Avg@4	IFEval	54.24 ± 0.22	54.14 ± 0.36	$\mathbf{58.01 \pm 1.29}$	53.79 ± 0.73	56.62 ± 1.53
		MMLU-Pro	65.47 ± 1.44	57.98 ± 0.82	66.31 ± 1.65	49.76 ± 2.68	$\mathbf{67.74 \pm 0.55}$
		SuperGPQA	28.87 ± 0.40	26.54 ± 0.55	26.01 ± 0.18	17.14 ± 0.89	$\mathbf{29.18 \pm 1.81}$
		$\overline{\text{Avg}}$	49.53	46.22	50.11	40.23	**51.18**
	Pass@4	IFEval	69.50 ± 0.19	69.13 ± 0.49	70.86 ± 1.48	67.84 ± 0.74	$\mathbf{71.66 \pm 1.66}$
		MMLU-Pro	82.38 ± 2.18	79.52 ± 2.18	$\mathbf{83.81 \pm 4.12}$	78.57 ± 2.47	81.91 ± 2.18
		SuperGPQA	53.42 ± 0.72	52.02 ± 1.51	52.59 ± 0.65	39.42 ± 1.11	$\mathbf{54.98 \pm 1.17}$
		$\overline{\text{Avg}}$	68.43	66.89	69.09	61.94	**69.52**

5.4 Design Controls and Geometric Evidence

Soft protection and pretrained direction selection. We compare RPSFT with Hard Top, which fixes the leading pretrained product block after each optimizer update, and Random Block, which applies the same soft penalty in random orthonormal coordinates. The controls share protection rank $k = 768$ and training hyperparameters; Random Block also uses the same penalty coefficient. RPSFT achieves the highest mean ID and OOD-macro accuracy on both tasks (Table 6). Its OOD-macro gains over Hard Top and Random Block are 4.51 and 3.62 points on CommonsenseQA, and 10.00 and 6.34 points on ProofWriter. Appendix D.2 reports benchmark-level results.

Table 6: Matched subspace controls on Qwen2.5-7B at the final two-epoch checkpoints. Values are mean ± sample SD across three training seeds. OOD macro averages IFEval-loose, MMLU-Pro, and SuperGPQA. RPSFT repeats Table 4 with SD instead of standard error; bold marks the highest mean.

	CommonsenseQA		ProofWriter	
Method	ID	OOD macro	ID	OOD macro
Hard Top	82.06 ± 0.25	34.10 ± 0.29	92.84 ± 0.85	29.43 ± 0.42
Random Block	82.20 ± 0.48	34.99 ± 0.27	92.66 ± 0.87	33.09 ± 1.03
RPSFT	$\mathbf{83.51 \pm 0.80}$	$\mathbf{38.61 \pm 1.92}$	$\mathbf{94.09 \pm 0.20}$	$\mathbf{39.43 \pm 2.19}$

RPSFT reduces dominant left-singular-subspace rotation across the three evaluated checkpoints. Appendix Figures 6 and 7 report layerwise averages across attention and MLP projections and rankwise results for one representative matrix per model. The hidden-state analysis in Appendix Figures 11, 12, and 13 supplies a complementary representation-space view: Qwen2.5-3B shows the clearest centroid separation, while the larger models exhibit more distributed drift. The PCA plots place RPSFT closer to the Base centroid than standard SFT in many benchmark panels. Appendix D.9 gives a complementary leading-versus-random direction-restoration intervention.

Empirical computation costs. Table 7 reports the original training costs. RPSFT uses only 6.0% more GPU-hours than SFT (165.7 vs. 156.3 GPU-hours), while remaining cheaper than IW and L2 Init. Appendix D.12 gives the analytical memory and FLOP costs, and Appendix Table 11 compares non-math training overheads across methods.

Table 7: Empirical computation and memory cost for RPSFT and baselines. Samples/s and Steps/s report training throughput; Peak alloc. and Peak reserved report maximum allocated and CUDA-reserved GPU memory.

Metric	SFT	DFT	IW	L2 Init	RPSFT
GPU-hours	156.3	159.0	181.8	174.1	165.7
Samples/s	4.368	4.294	3.755	3.922	4.119
Steps/s	0.273	0.268	0.235	0.245	0.257
Peak alloc.	32.4	40.0	59.3	60.9	38.5
Peak reserved	125.4	128.1	137.0	136.6	135.2

6 Conclusion

We presented RPSFT, a supervised fine-tuning regularizer that softly anchors accumulated displacement in the pretrained top-k product block. Across math and non-math adaptation, RPSFT improves the ID/OOD balance over SFT and provides stronger RLFT initializations. It leads the evaluated retention baselines on both non-math tasks. Matched hard-block and random-block controls demonstrate the value of soft protection and pretrained direction selection. The analysis characterizes protected-block drift and local sensitivity, while empirical-Fisher diagnostics and principal-angle measurements connect leading-coordinate anchoring with reduced dominant left-subspace rotation. Together, these results establish projected-block anchoring as an effective approach to capability retention with full-parameter adaptation.

Scope and future work. The experiments cover math and non-math adaptation on 3B–8B checkpoints, with reinforcement learning fine-tuning as the downstream stage. Coding, safety alignment, multilingual adaptation, larger checkpoints, and additional preference-optimization objectives form the next evaluation axes. Joint selection of rank, anchoring strength, and stopping time offers another route to improving the adaptation–retention trade-off. The layer-1 query-projection diagnostic motivates adaptive protection across layers and matrix types, consistent with evidence that fine-tuning effects concentrate in particular layers (Shi et al., 2025; Zhao et al., 2026).

AI USE STATEMENT

An AI assistant supported editing, reported-data arithmetic checks, and LaTeX formatting.

REPRODUCIBILITY STATEMENT

The method and training setup are described in Sections 4 and 5. Appendix E gives datasets, hyperparameters, evaluation and checkpoint-selection procedures, and Appendix C details the analytical results. A code package has been uploaded with this submission.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. URL https://arxiv.org/abs/2309.16609.

Wesley Chung, Lynn Cherif, David Meger, and Doina Precup. Parseval regularization for continual reinforcement learning, 2024. URL https://arxiv.org/abs/2412.07224.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in large language models via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Mohamed Elsayed, Qingfeng Lan, Clare Lyle, and A. Rupam Mahmood. Weight clipping for deep continual and reinforcement learning, 2024. URL https://arxiv.org/abs/2407.01704.

Jörg K.H. Franke, Michael Hefenbrock, and Frank Hutter. Preserving principal subspaces to reduce catastrophic forgetting in fine-tuning. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. URL https://openreview.net/forum?id=XoWtroECJU.

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.

David Haink. Hessian eigenvectors and principal component analysis of neural network weight matrices, 2023. URL https://arxiv.org/abs/2311.00452.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024. URL https://arxiv.org/abs/2402.14008.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning, 2025. URL https://arxiv.org/abs/2507.00432.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Hangzhan Jin, Sitao Luan, Sicheng Lyu, Guillaume Rabusseau, Reihaneh Rabbany, Doina Precup, and Mohammad Hamdaqa. Rl fine-tuning heals ood forgetting in sft, 2025. URL https://arxiv.org/abs/2509.12235.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization, 2024. URL https://arxiv.org/abs/2308.11958.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems 32*, pp. 4158–4169, 2019.

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Min Xie, Qingfu Zhang, Hongbin Liu, Gaofeng Meng, and Fei Zhu. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training, 2025. URL https://arxiv.org/abs/2507.05386.

Alex Lewandowski, Michal Bortkiewicz, Saurabh Kumar, Andras Gyorgy, Dale Schuurmans, Mateusz Ostaszewski, and Marlos C. Machado. Learning continually by spectral regularization, 2024. URL https://arxiv.org/abs/2406.06811.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, Yufan Guo, Kai Zhong, Weiqi Zhang, Sujay Sanghavi, Changyou Chen, Hyokun Yun, and Lihong Li. Sft doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in llms, 2025. URL https://arxiv.org/abs/2509.20758.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022. URL https://arxiv.org/abs/2109.07958.

Aojun Lu, Hangjie Yuan, Tao Feng, and Yanan Sun. Rethinking the stability-plasticity trade-off in continual learning from an architectural perspective, 2025. URL https://arxiv.org/abs/2506.03951.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks, 2023. URL https://arxiv.org/abs/2303.01486.

James Martens. New insights and perspectives on the natural gradient method, 2020. URL https://arxiv.org/abs/1412.1193.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models, 2025. URL https://arxiv.org/abs/2404.02948.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models, 2025. URL https://arxiv.org/abs/2505.11711.

Nikhil Shivakumar Nayak, Krishnateja Killamsetty, Ligong Han, Abhishek Bhandwaldar, Prateek Chanda, Kai Xu, Hao Wang, Aldo Pareja, Oleg Silkin, Mustafa Eyceoz, and Akash Srivastava. Sculpting subspaces: Constrained full fine-tuning in llms for continual learning, 2025. URL https://arxiv.org/abs/2504.07097.

Tianwei Ni, Allen Nie, Sapana Chaudhary, Yao Liu, Huzefa Rangwala, and Rasool Fakoor. Offline learning and forgetting for reasoning with large language models, 2025. URL https://arxiv.org/abs/2504.11364.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Chongli Qin and Jost Tobias Springenberg. Supervised fine tuning on curated data is reinforcement learning (and can be improved), 2025. URL https://arxiv.org/abs/2507.12856.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less, 2025. URL https://arxiv.org/abs/2509.04259.

Guangyuan Shi, Zexin Lu, Xiaoyu Dong, Wenlong Zhang, Xuanyu Zhang, Yujie Feng, and Xiao-Ming Wu. Understanding layer significance in llm alignment, 2025. URL https://arxiv.org/abs/2410.17875.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shawn Gavin, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi Li, Yunwen Li, David Ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tyshawn Hsing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, Yuelin Bai, Xingyuan Bu, Chenglin Cai, Liang Chen, Yifan Chen, Chengtuo Cheng, Tianhao Cheng, Keyi Ding, Siming Huang, Yun Huang, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Tianyang Pang, Zhongyuan Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Shanghaoran Quan, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jinyang Zhang, Xiyue Zhang, Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, Chenghua Zhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

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

Chenxu Wang, Yilin Lyu, Zicheng Sun, and Liping Jing. Continual gradient low-rank projection fine-tuning for llms, 2025a. URL https://arxiv.org/abs/2507.02503.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. Milora: Harnessing minor singular components for parameter-efficient llm finetuning, 2025b. URL https://arxiv.org/abs/2406.09044.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application, 2024a. URL https://arxiv.org/abs/2302.00487.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning, 2023. URL https://arxiv.org/abs/2310.14152.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024b. URL https://arxiv.org/abs/2406.01574.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of sft: A reinforcement learning perspective with reward rectification, 2025. URL https://arxiv.org/abs/2508.05629.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Qinghua Zhao, Xueling Gong, Xinyu Chen, Zhongfeng Kang, and Xinlu Li. A layer-wise analysis of supervised fine-tuning, 2026. URL https://arxiv.org/abs/2604.11838.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Jinwon Lee, David Z. Pan, Zhangyang Wang, Yuandong Tian, and Kai Sheng Tai. Why rl updates look sparse: An implicit compass drives optimization bias, 2025a. URL https://openreview.net/forum?id=Q4mF4tLGbf. Submitted to ICLR 2026.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, David Z. Pan, Zhangyang Wang, Yuandong Tian, and Kai Sheng Tai. The path not taken: Rlvr provably learns off the principals, 2025b. URL https://arxiv.org/abs/2511.08567.

Wenhong Zhu, Ruobing Xie, Rui Wang, Xingwu Sun, Di Wang, and Pengfei Liu. Proximal supervised fine-tuning, 2025c. URL https://arxiv.org/abs/2508.17784.

APPENDIX CONTENTS

A Algorithms **17**

A.1 SVD Notation 17

A.2 LoRA Objective 17

B Related Work **18**

B.1 Traditional Continual-Learning Regularization 18

B.2 LLM Post-Training and Reasoning Generalization 18

B.3 Comparison with Regularization Methods 18

C Theory: Properties of RPSFT Regularization **19**

C.1 Setup and Notation 19

C.2 Rank Boundary Cases 19

C.3 Block Decomposition Interpretation 20

C.4 Optimality Condition and Protected-Block Drift Bound 20

C.5 Rank Selection: Boundary and Threshold Rule 21

C.6 Gradient-Flow View: Exponentially Damped Task-Induced Drift 24

D Additional Results **25**

D.1 Non-Math Adaptation Across Training Seeds 25

D.2 Matched Hard and Random Subspace Controls 25

D.3 Anchoring-Strength Sweep Against L2 Init 27

D.4 Direct Subspace and Importance Baselines 27

D.5 Training-Seed Robustness on Math Adaptation 28

D.6 Post-DAPO Robustness Across Training Seeds 28

D.7 Stochastic Evaluation Stability on Math and OOD Benchmarks 28

D.8 Local Quadratic Validation Along Checkpoint Directions 29

D.9 Dominant-Direction Restoration with Rank-Matched Controls 29

D.10 GRPO from SFT and RPSFT Initializations 30

D.11 Transfer to the DeepSeekMath Model Family 31

D.12 Optimized Memory–Throughput Trade-offs 31

D.13 Rank Sensitivity and OOD Pass@1 Results 32

E Experiment Details **35**

E.1 Evaluation Subsets, Scoring, and Aggregation 36

E.2 Individual Training-Run Results 37

F Metrics for Analysis Figures **37**

F.1 Strict SVD Subspace Energy 37

F.2 Layerwise First-Order Signal 38

F.3 Rotation Metrics 39
F.4 Entropy Metric 40
F.5 Auxiliary Entropy Plots Across Model Families 41
F.6 Base-Model Drift in Hidden Representations 41

A ALGORITHMS

Algorithm 1 Rotation-Preserving SFT (RPSFT)

Require: Pretrained weights θ_0, dataset $\mathcal{D}$, eligible matrix set $\mathcal{M}$, maximum rank k, anchoring strength $\lambda \geq 0$

Precompute: for each $\ell \in \mathcal{M}$, set $k_\ell \leftarrow \min(k, m_\ell, n_\ell)$ and compute $(\mathbf{U}_{0,\ell}^{(k_\ell)}, \mathbf{V}_{0,\ell}^{(k_\ell)})$ from the SVD of $\mathbf{W}_\ell^0$; set $S_\ell^{\mathrm{ref}} \leftarrow (\mathbf{U}_{0,\ell}^{(k_\ell)})^\top \mathbf{W}_\ell^0 \mathbf{V}_{0,\ell}^{(k_\ell)}$
Initialize $\theta \leftarrow \theta_0$
for supervised training step $t = 1, 2, \ldots$ **do**
  Sample minibatch $(x, y) \sim \mathcal{D}$
  Compute the standard SFT loss $\mathcal{L}_{\mathrm{SFT}}$
  $\mathcal{L}_{\mathrm{reg}} \leftarrow 0$
  for each $\ell \in \mathcal{M}$ **do**
    $S \leftarrow (\mathbf{U}_{0,\ell}^{(k_\ell)})^\top \mathbf{W}_\ell \mathbf{V}_{0,\ell}^{(k_\ell)}$
    $\mathcal{L}_{\mathrm{reg}} \leftarrow \mathcal{L}_{\mathrm{reg}} + \|S - S_\ell^{\mathrm{ref}}\|_F^2$
  end for
  $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{SFT}} + \lambda \mathcal{L}_{\mathrm{reg}}$
  Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
end for
return θ

Illustrative per-step PyTorch implementation. This per-step sketch implements Eq. (7) by caching the pretrained singular bases once and adding the projected-block penalty to the SFT loss:

```
# Precompute once for each selected pretrained weight W0
svd_cache = []
with torch.no_grad():
    for W0 in pretrained_layers:
        U, _, Vh = torch.linalg.svd(W0.float(), full_matrices=False)
        Uk = U[:, :k].to(device=W0.device, dtype=W0.dtype)
        Vk = Vh[:k].T.to(device=W0.device, dtype=W0.dtype)
        S0 = Uk.T @ W0 @ Vk
        svd_cache.append((Uk, Vk, S0))

# During SFT, reg_strength is the coefficient lambda
reg_loss = task_loss.new_zeros(())
for W, (Uk, Vk, S0) in zip(model_layers, svd_cache):
    S = Uk.T @ W @ Vk
    reg_loss = reg_loss + (S - S0).pow(2).sum()

loss = task_loss + reg_strength * reg_loss
loss.backward()
```

A.1 SVD NOTATION

For a weight matrix $\mathbf{W} \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$, write its economy-size singular value decomposition as $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, with $p = \min(d_{\mathrm{out}}, d_{\mathrm{in}})$, $\mathbf{U} \in \mathbb{R}^{d_{\mathrm{out}} \times p}$, $\mathbf{V} \in \mathbb{R}^{d_{\mathrm{in}} \times p}$, and $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$. The columns of $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and the diagonal entries of $\mathbf{\Sigma}$ are nonnegative and sorted in descending order. This convention includes zero singular values and supports every protected rank $0 \leq k \leq p$. The leading directions define the pretrained product subspaces used by RPSFT.

A.2 LORA OBJECTIVE

RPSFT can also be applied to LoRA by adding the same projected-block drift penalty to the effective weight. The reported experiments use full-parameter RPSFT; the LoRA rows in Appendix D are

vanilla LoRA baselines with adapter rank $r = 32$.

$$\mathcal{L}(A, B) = \mathcal{L}_{\text{task}}(A, B) + \lambda \sum_{\ell \in \mathcal{M}'} \left\| S_\ell(\mathbf{W}_\ell^0 + \Delta \mathbf{W}_\ell) - S_\ell^{\text{ref}} \right\|_F^2, \tag{8}$$

where $\Delta \mathbf{W}_\ell = B_\ell A_\ell$. This keeps the regularizer aligned with FPFT: it penalizes drift inside the pretrained dominant subspace even when only low-rank parameters are trained.

B Related Work

B.1 Traditional Continual-Learning Regularization

Continual-learning surveys commonly separate replay, architecture expansion, and regularization-based approaches (Wang et al., 2024a). Our work is closest to the regularization family, which modifies the optimization objective or update rule to control parameter drift. A classical example is Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017), which uses a Fisher-weighted quadratic penalty to discourage movement of important parameters. A simpler related baseline is anchoring or regenerative regularization, which penalizes uniform Euclidean movement from a reference model, typically through a term proportional to $\|\theta - \theta_0\|_2^2$ (Kumar et al., 2024). Other recent continual-learning regularizers control the weight geometry more directly: spectral regularization keeps the maximum singular value of each layer near one (Lewandowski et al., 2024), weight clipping bounds weight magnitudes after each update (Elsayed et al., 2024), and Parseval regularization encourages near-orthogonal weight matrices (Chung et al., 2024). These methods target global or parameterwise geometry; RPSFT anchors the dominant pretrained product coordinates studied by our gradient-energy and geometric analyses.

B.2 LLM Post-Training and Reasoning Generalization

Recent LLM studies show that reasoning post-training can either generalize or over-specialize depending on the learning signal and update geometry. Other work shows that math-only post-training often transfers imperfectly: SFT can induce representation and output drift, whereas RL tends to preserve broader capabilities (Huan et al., 2025). This is consistent with findings that RL fine-tuning mitigates forgetting through reward-variance scaling (Lai et al., 2025), that online RL is biased toward KL-minimal solutions (Shenfeld et al., 2025), and with analyses of sparse, spectrum-preserving, and off-principal RL updates (Mukherjee et al., 2025; Zhu et al., 2025a;b). Offline reasoning studies also show that naive SFT can harm search capability (Ni et al., 2025; Lin et al., 2025). These results motivate our focus on the SFT stage: RPSFT explicitly controls SFT-induced movement in dominant pretrained singular subspaces, combining task adaptation with an explicit anchor on pretrained product coordinates.

Several LLM fine-tuning methods share this goal. Proximal SFT constrains policy drift with PPO-style updates (Zhu et al., 2025c); Dynamic SFT and importance-weighted SFT rescale token or example losses (Wu et al., 2025; Qin & Springenberg, 2025). Protected SFT preserves designated singular directions through projected updates (Jin et al., 2025); PSP preserves principal subspaces (Franke et al., 2024); and Sculpting Subspaces estimates protected ranks and projects instantaneous gradients (Nayak et al., 2025). Gradient low-rank projection follows the same hard-update family (Wang et al., 2025a), while O-LoRA constrains an adapter subspace orthogonal to prior task directions (Wang et al., 2023). Spectral PEFT methods such as PiSSA and MiLoRA use singular directions for efficient adaptation (Meng et al., 2025; Wang et al., 2025b). For fixed bases, the product-block projection in Sculpting Subspaces takes the form $G \mapsto G - P_U G P_V$, where $P_U = \mathbf{U}_k \mathbf{U}_k^\top$ and $P_V = \mathbf{V}_k \mathbf{V}_k^\top$ (Nayak et al., 2025). This update and RPSFT both retain cross-block adaptation. RPSFT adds the restoring term $2\lambda P_U(\mathbf{W} - \mathbf{W}_0)P_V$ to the task gradient, making its protection proportional to accumulated displacement. Table 5 reports method-level results under the configurations in Appendix D.4.

B.3 Comparison with Regularization Methods

Let θ_0 denote the pretrained parameters, and let $\mathbf{W}_\ell^0 = \mathbf{U}_\ell^0 \mathbf{\Sigma}_\ell^0 (\mathbf{V}_\ell^0)^\top$ be the SVD of layer ℓ. Table 8 compares methods whose update rules or objectives are explicit enough to make a direct theoretical comparison. The table focuses on papers with explicit regularizers or update rules.

Table 8: Comparison with related regularization and subspace-preservation methods.

Method	Objective or update	Extra compute	Extra memory	Ref.
EWC	$\mathcal{L}_{\text{task}} + \frac{\lambda}{2}\sum_i F_i(\theta_i - \theta_{0,i})^2$	$O(TP)$ Fisher estimation; $O(P)$ per step	$O(P)$	(Kirkpatrick et al., 2017)
L2 anchor	$\mathcal{L}_{\text{task}} + \lambda\|\theta - \theta_0\|_2^2$	$O(P)$ per step	$O(P)$	(Kumar et al., 2024)
Weight clipping	$\theta_i \leftarrow \text{clip}(\theta_i, -\epsilon, \epsilon)$	$O(P)$ per step	$O(1)$	(Elsayed et al., 2024)
Hard subspace projection	$g_\ell \leftarrow (\mathbf{I} - \mathbf{P}_\ell)g_\ell$	$O\big(\sum_\ell m_\ell n_\ell k_\ell\big)$ per step	$O\big(\sum_\ell (m_\ell + n_\ell)k_\ell\big)$	(Jin et al., 2025; Franke et al., 2024; Nayak et al., 2025)
O-LoRA	$\mathcal{L}_{\text{task}} + \lambda\|A^\top A_{\text{ref}}\|_F^2$ on adapter factors	$O\big(\sum_\ell d_\ell r^2\big)$ per step	$O\big(\sum_\ell d_\ell r\big)$	(Wang et al., 2023)
RPSFT (ours)	$\mathcal{L}_{\text{task}} + \lambda\sum_\ell \|S_\ell - S_\ell^0\|_F^2$, $S_\ell = (\mathbf{U}_{0,\ell}^{(k)})^\top \mathbf{W}_\ell \mathbf{V}_{0,\ell}^{(k)}$	$O\big(\sum_\ell m_\ell n_\ell k_\ell / s\big)$	$O\big(\sum_\ell ((m_\ell + n_\ell)k_\ell + k_\ell^2)\big)$	–

C Theory: Properties of RPSFT Regularization

C.1 Setup and Notation

Consider a weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$ with pretrained initialization $\mathbf{W}_0$. Let $p = \min(m, n)$ and write the economy-size SVD of $\mathbf{W}_0$ as

$$\mathbf{W}_0 = \mathbf{U}_0 \mathbf{\Sigma}_0 \mathbf{V}_0^\top, \tag{9}$$

where $\mathbf{U}_0 \in \mathbb{R}^{m\times p}$ and $\mathbf{V}_0 \in \mathbb{R}^{n\times p}$ have orthonormal columns and $\mathbf{\Sigma}_0 \in \mathbb{R}^{p\times p}$ contains the singular values, including zeros, in descending order. Fix these bases and a rank $0 \le k \le p$. Let $\mathbf{U}_k \in \mathbb{R}^{m\times k}$ and $\mathbf{V}_k \in \mathbb{R}^{n\times k}$ denote the top-k left and right singular vectors of $\mathbf{W}_0$. Define the linear operator

$$\mathcal{P}_k(\mathbf{W}) \triangleq \mathbf{U}_k^\top \mathbf{W} \mathbf{V}_k \in \mathbb{R}^{k\times k}. \tag{10}$$

RPSFT adds the penalty

$$\mathcal{R}_k(\mathbf{W}) \triangleq \|\mathcal{P}_k(\mathbf{W}) - \mathcal{P}_k(\mathbf{W}_0)\|_F^2 = \left\|\mathbf{U}_k^\top (\mathbf{W} - \mathbf{W}_0)\mathbf{V}_k\right\|_F^2. \tag{11}$$

Given a task loss $f(\mathbf{W})$ (e.g., the SFT objective) and a fixed regularization strength $\lambda \ge 0$, the RPSFT objective is

$$F_k(\mathbf{W}) \triangleq f(\mathbf{W}) + \lambda \mathcal{R}_k(\mathbf{W}). \tag{12}$$

At $\lambda = 0$, the objective reduces to standard SFT for every rank.

C.2 Rank Boundary Cases

We record the boundary cases for the protected rank k.

Case $k = 0$. $\mathcal{R}_0(\mathbf{W}) \equiv 0$, hence RPSFT is exactly standard SFT.

Case $k = \min(m, n)$ (maximum projected rank). Let $p = \min(m, n)$, $\Delta = \mathbf{W} - \mathbf{W}_0$, and let $\mathbf{U}_p, \mathbf{V}_p$ denote the fixed economy-size singular bases. At $k = p$,

$$\mathcal{R}_p(\mathbf{W}) = \begin{cases} \|\Delta\|_F^2, & m = n = p, \\ \|\mathbf{U}_p \mathbf{U}_p^\top \Delta\|_F^2, & m > n, \\ \|\Delta \mathbf{V}_p \mathbf{V}_p^\top\|_F^2, & n > m. \end{cases} \tag{13}$$

The projected objective spans the full matrix for square weights and the maximal SVD-coordinate block for rectangular weights. The L2 Init endpoint applies $\lambda\|\mathbf{W} - \mathbf{W}_0\|_F^2$ directly to every matrix.

C.3 Block Decomposition Interpretation

Let $\mathbf{U} = [\mathbf{U}_k \;\; \mathbf{U}_\perp] \in \mathbb{R}^{m\times m}$ and $\mathbf{V} = [\mathbf{V}_k \;\; \mathbf{V}_\perp] \in \mathbb{R}^{n\times n}$ be orthogonal matrices extending $\mathbf{U}_k, \mathbf{V}_k$. Define the rotated coordinates

$$\widetilde{\mathbf{W}} \triangleq \mathbf{U}^\top \mathbf{W}\mathbf{V} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \tag{14}$$

where $A \in \mathbb{R}^{k\times k}$. Similarly $\widetilde{\mathbf{W}}_0 = \mathbf{U}^\top \mathbf{W}_0 \mathbf{V}$ has top-left block A_0. Then

$$\mathcal{R}_k(\mathbf{W}) = \|A - A_0\|_F^2. \tag{15}$$

Thus, RPSFT directly controls the top-left $k \times k$ block A, while B, C, D retain adaptation capacity. With complete pretrained singular bases, the initialization has $A_0 = \mathbf{\Sigma}_{0,k}$ and zero cross blocks. The left and right singular subspaces of the updated weight are eigenspaces of the Gram matrices

$$\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^\top = \begin{pmatrix} AA^\top + BB^\top & AC^\top + BD^\top \\ CA^\top + DB^\top & CC^\top + DD^\top \end{pmatrix}, \quad \widetilde{\mathbf{W}}^\top\widetilde{\mathbf{W}} = \begin{pmatrix} A^\top A + C^\top C & A^\top B + C^\top D \\ B^\top A + D^\top C & B^\top B + D^\top D \end{pmatrix}.$$

These identities express the roles of the anchored block, cross-block coupling, and spectral separation in singular-subspace alignment. For example, for $k \geq 1$, $B = C = 0$ and $\sigma_{\min}(A) > \|D\|_2$ ensure that the leading k left and right singular subspaces equal the pretrained spans. RPSFT anchors A throughout joint optimization of all four blocks; Appendix F.3 measures the resulting left-subspace alignment.

C.4 Optimality Condition and Protected-Block Drift Bound

We characterize protected-block displacement and its contribution to a positive-semidefinite local sensitivity measure.

Proposition C.1 (Stationary Condition)**.** *Assume f is differentiable. Any stationary point $\mathbf{W}^\star$ of F_k satisfies*

$$\nabla f(\mathbf{W}^\star) + 2\lambda\, \mathbf{U}_k\big(\mathbf{U}_k^\top(\mathbf{W}^\star - \mathbf{W}_0)\mathbf{V}_k\big)\mathbf{V}_k^\top = 0. \tag{16}$$

Proof. Using $\mathcal{R}_k(\mathbf{W}) = \|\mathbf{U}_k^\top(\mathbf{W} - \mathbf{W}_0)\mathbf{V}_k\|_F^2$ and standard matrix calculus, $\nabla_{\mathbf{W}}\mathcal{R}_k(\mathbf{W}) = 2\mathbf{U}_k(\mathbf{U}_k^\top(\mathbf{W} - \mathbf{W}_0)\mathbf{V}_k)\mathbf{V}_k^\top$. Setting $\nabla F_k(\mathbf{W}) = \nabla f(\mathbf{W}) + \lambda\nabla\mathcal{R}_k(\mathbf{W}) = 0$ yields Eq. (16). □

Proposition C.2 (Protected-Block Drift Bound)**.** *Let $\lambda > 0$ and let $\mathbf{W}^\star$ be a stationary point of F_k. Then*

$$\big\|\mathbf{U}_k^\top(\mathbf{W}^\star - \mathbf{W}_0)\mathbf{V}_k\big\|_F = \frac{\|\mathbf{U}_k^\top \nabla f(\mathbf{W}^\star)\mathbf{V}_k\|_F}{2\lambda} \leq \frac{\|\nabla f(\mathbf{W}^\star)\|_F}{2\lambda}. \tag{17}$$

More generally, at any $\mathbf{W}$ with $\|\mathbf{U}_k^\top \nabla F_k(\mathbf{W})\mathbf{V}_k\|_F \leq \varepsilon$,

$$\big\|\mathbf{U}_k^\top(\mathbf{W} - \mathbf{W}_0)\mathbf{V}_k\big\|_F \leq \frac{\|\mathbf{U}_k^\top \nabla f(\mathbf{W})\mathbf{V}_k\|_F + \varepsilon}{2\lambda}.$$

Proof. For every $\mathbf{W}$, projecting the objective gradient gives

$$\mathbf{U}_k^\top \nabla F_k(\mathbf{W})\mathbf{V}_k = \mathbf{U}_k^\top \nabla f(\mathbf{W})\mathbf{V}_k + 2\lambda\mathbf{U}_k^\top(\mathbf{W} - \mathbf{W}_0)\mathbf{V}_k. \tag{18}$$

At a stationary point, rearranging this identity and taking Frobenius norms yields Eq. (17). At an approximate stationary point, the triangle inequality gives the bound with ε. □

Implication. Protected-block displacement is determined by the balance between the projected task gradient and the restoring gradient. The approximate-stationarity bound quantifies the same balance with an explicit optimization residual.

Proposition C.3 (Projected Fisher Sensitivity Bound)**.** *Let*

$$\delta = \mathrm{vec}(\mathbf{W}^\star - \mathbf{W}_0), \qquad p_k = (\mathbf{V}_k\mathbf{V}_k^\top) \otimes (\mathbf{U}_k\mathbf{U}_k^\top), \qquad q_k = I - p_k,$$

and decompose the displacement as

$$a = p_k\delta, \qquad b = q_k\delta.$$

The projector satisfies

$$\|a\|_2 = \left\|\mathbf{U}_k^\top(\mathbf{W}^\star - \mathbf{W}_0)\mathbf{V}_k\right\|_F.$$

Let $\mathbf{F}_{\mathrm{ret}} \succeq 0$ be a fixed local sensitivity matrix, such as a retained-input model Fisher, a generalized Gauss–Newton matrix, or an empirical-Fisher gradient second moment, and define

$$\alpha = \|p_k\mathbf{F}_{\mathrm{ret}}p_k\|_2, \qquad \gamma = \|q_k\mathbf{F}_{\mathrm{ret}}q_k\|_2, \qquad \rho = \|p_k\mathbf{F}_{\mathrm{ret}}q_k\|_2.$$

Define $Q_{\mathrm{ret}}(\delta) = \frac{1}{2}\delta^\top\mathbf{F}_{\mathrm{ret}}\delta$. Then

$$Q_{\mathrm{ret}}(\delta) \le \frac{\alpha}{2}\|a\|_2^2 + \rho\|a\|_2\|b\|_2 + \frac{\gamma}{2}\|b\|_2^2. \tag{19}$$

Furthermore, define

$$G_k = \|p_k \operatorname{vec}(\nabla f(\mathbf{W}^\star))\|_2.$$

For $\lambda > 0$ and a stationary point $\mathbf{W}^\star$ of F_k, the projected stationary condition gives

$$a = -\frac{1}{2\lambda}p_k \operatorname{vec}(\nabla f(\mathbf{W}^\star)),$$

and hence

$$Q_{\mathrm{ret}}(\delta) \le \frac{\alpha G_k^2}{8\lambda^2} + \frac{\rho G_k}{2\lambda}\|b\|_2 + \frac{\gamma}{2}\|b\|_2^2. \tag{20}$$

Proof. Expanding $\delta = a + b$ gives

$$\delta^\top\mathbf{F}_{\mathrm{ret}}\delta = a^\top\mathbf{F}_{\mathrm{ret}}a + 2a^\top\mathbf{F}_{\mathrm{ret}}b + b^\top\mathbf{F}_{\mathrm{ret}}b.$$

Applying Cauchy–Schwarz and the operator-norm definitions of α, γ, and ρ gives Eq. (19). Substituting $\|a\|_2 = G_k/(2\lambda)$ gives Eq. (20). □

Interpretation. The three terms quantify sensitivity within the protected block, coupling to the complementary update, and sensitivity within the complement. For a complementary-displacement bound $\|b\|_2 \le B$, Eq. (20) gives $Q_{\mathrm{ret}}(\delta) \le \alpha G_k^2/(8\lambda^2) + \rho G_k B/(2\lambda) + \gamma B^2/2$. Thus, the total sensitivity depends jointly on protected displacement, complementary displacement, and cross-subspace coupling. An approximate projected-stationarity residual of size ε gives the same bound with G_k replaced by $G_k + \varepsilon$.

Connection to retained loss. The decomposition also applies to the full parameter vector with the block-diagonal projector P from Section 3. Let $H_{\mathrm{ret}} = \nabla^2\ell_{\mathrm{ret}}(w_0)$ and suppose that the retained-loss Hessian is L_H-Lipschitz along the segment from w_0 to $w_0 + \delta$. For a chosen positive-semidefinite $\mathbf{F}_{\mathrm{ret}}$, set $\varepsilon_H = \|H_{\mathrm{ret}} - \mathbf{F}_{\mathrm{ret}}\|_2$. Taylor's theorem yields

$$\ell_{\mathrm{ret}}(w_0 + \delta) - \ell_{\mathrm{ret}}(w_0) \le g_{\mathrm{ret}}^\top\delta + Q_{\mathrm{ret}}(\delta) + \frac{\varepsilon_H}{2}\|\delta\|_2^2 + \frac{L_H}{6}\|\delta\|_2^3.$$

Indeed, the quadratic term equals $Q_{\mathrm{ret}}(\delta) + \frac{1}{2}\delta^\top(H_{\mathrm{ret}} - \mathbf{F}_{\mathrm{ret}})\delta$, and the Lipschitz-Hessian assumption bounds the Taylor remainder. This formula separates signed first-order transfer, the chosen sensitivity measure, the Hessian–sensitivity discrepancy, and the higher-order remainder.

C.5 Rank Selection: Boundary and Threshold Rule

We analyze rank selection in a separable quadratic model written in the fixed pretrained singular basis. Let $\Delta = \mathbf{W} - \mathbf{W}_0$ and

$$\widetilde{\Delta} = \mathbf{U}^\top\Delta\mathbf{V}. \tag{21}$$

Index its entries by $s = (i, j)$, with coordinate value δ_s. For $0 \le k \le p = \min(m, n)$, define the nested protected set

$$S_k = \{(i, j) : 1 \le i \le k,\ 1 \le j \le k\}. \tag{22}$$

The ID loss-change surrogate is

$$\widehat{L}_{\mathrm{id}}(\delta) = -\sum_s g_s\delta_s + \frac{1}{2}\sum_s h_s\delta_s^2, \qquad h_s > 0, \tag{23}$$

where g_s is the ID learning drive (the negative ID-gradient coordinate) and h_s is a positive curvature coefficient. The RPSFT surrogate objective is

$$\widehat{L}_{\mathrm{id}}(\delta) + \lambda \sum_{s \in S_k} \delta_s^2. \tag{24}$$

For retained sensitivity, define the diagonal positive-semidefinite surrogate

$$\widehat{Q}_{\mathrm{ret}}(\delta) = \frac{1}{2} \sum_s c_s \delta_s^2, \qquad c_s \geq 0. \tag{25}$$

The coefficients c_s specify the sensitivity assigned to each product coordinate. The results below characterize these separable surrogates, with signed retained transfer incorporated after the threshold rule.

Proposition C.4 (Optimal local step)**.** *The unique minimizer of the regularized ID surrogate has coordinates*

$$\delta_s^\star(k) = \begin{cases} \dfrac{g_s}{h_s + 2\lambda}, & s \in S_k, \\ \dfrac{g_s}{h_s}, & s \notin S_k. \end{cases} \tag{26}$$

Its retained-sensitivity cost is

$$F_{\mathrm{ood}}(k) \coloneqq \widehat{Q}_{\mathrm{ret}}(\delta^\star(k)) = \frac{1}{2} \sum_{s \in S_k} \frac{c_s g_s^2}{(h_s + 2\lambda)^2} + \frac{1}{2} \sum_{s \notin S_k} \frac{c_s g_s^2}{h_s^2}. \tag{27}$$

Its gain on the unregularized ID surrogate is

$$G_{\mathrm{id}}(k) \coloneqq -\widehat{L}_{\mathrm{id}}(\delta^\star(k)) = \sum_{s \in S_k} \left(\frac{g_s^2}{h_s + 2\lambda} - \frac{1}{2} h_s \frac{g_s^2}{(h_s + 2\lambda)^2} \right) + \frac{1}{2} \sum_{s \notin S_k} \frac{g_s^2}{h_s}. \tag{28}$$

Proof. The objective is separable and strictly convex, with coordinate curvature $h_s + 2\lambda$ on S_k and h_s on its complement. Setting each coordinate derivative to zero gives $\delta_s^\star(k)$. Substitution into the two surrogates gives the stated expressions. □

Proposition C.5 (Existence of an optimal rank within the sensitivity support)**.** *Assume that* $q \in \{0, \ldots, p\}$ *satisfies*

$$c_s = 0 \qquad \textit{for all } s \notin S_q. \tag{29}$$

Define

$$\Phi(k) = G_{\mathrm{id}}(k) - \beta F_{\mathrm{ood}}(k), \qquad \beta > 0. \tag{30}$$

Then an optimal protected rank exists within $\{0, \ldots, q\}$*:*

$$\arg\max_{0 \leq k \leq p} \Phi(k) \ \cap \ \{0, \ldots, q\} \neq \varnothing. \tag{31}$$

If $\lambda > 0$ *and every* $k > q$ *adds a coordinate* $s \in S_k \setminus S_q$ *with* $g_s \neq 0$*, every maximizing rank satisfies* $k^\star \leq q$*.*

Proof. For $k \geq q$, all coordinates with positive retained-sensitivity coefficients are protected, so

$$F_{\mathrm{ood}}(k) = F_{\mathrm{ood}}(q). \tag{32}$$

The ID-gain difference is

$$G_{\mathrm{id}}(q) - G_{\mathrm{id}}(k) = \sum_{s \in S_k \setminus S_q} \frac{2\lambda^2 g_s^2}{h_s (h_s + 2\lambda)^2} \geq 0. \tag{33}$$

Consequently,

$$\Phi(q) - \Phi(k) = \sum_{s \in S_k \setminus S_q} \frac{2\lambda^2 g_s^2}{h_s (h_s + 2\lambda)^2} \geq 0. \tag{34}$$

A maximum over the finite rank set therefore has a representative at or below q. For $\lambda > 0$, the additional learning-drive condition makes the displayed difference strictly positive for every $k > q$, giving the final statement. At $\lambda = 0$, every rank gives the same surrogate solution and utility. □

Corollary C.6 (Coordinate and rank-increment threshold rules)**.** *For a currently unprotected coordinate s, define the ID cost of protection as*

$$\Delta_{\mathrm{ID},s} = \frac{1}{2}\frac{g_s^2}{h_s} - \left(\frac{g_s^2}{h_s+2\lambda} - \frac{1}{2}h_s\frac{g_s^2}{(h_s+2\lambda)^2}\right) = \frac{2\lambda^2 g_s^2}{h_s(h_s+2\lambda)^2}, \tag{35}$$

and the retained-sensitivity reduction as

$$\Delta_{\mathrm{OOD},s} = \frac{1}{2}c_s\frac{g_s^2}{h_s^2} - \frac{1}{2}c_s\frac{g_s^2}{(h_s+2\lambda)^2} = \frac{2\lambda c_s(h_s+\lambda)g_s^2}{h_s^2(h_s+2\lambda)^2}. \tag{36}$$

Protecting s increases the scalarized coordinate utility exactly when

$$\beta\Delta_{\mathrm{OOD},s} > \Delta_{\mathrm{ID},s}. \tag{37}$$

For $\lambda > 0$ and $g_s \neq 0$, this is equivalent to

$$c_s > \frac{\lambda h_s}{\beta(h_s+\lambda)}. \tag{38}$$

For $\lambda = 0$ or $g_s = 0$, both utility contributions equal zero. For the nested rank family, define the newly protected shell $T_{k+1} = S_{k+1} \setminus S_k$. Its exact utility increment is

$$\Phi(k+1) - \Phi(k) = \sum_{s\in T_{k+1}} \left(\beta\Delta_{\mathrm{OOD},s} - \Delta_{\mathrm{ID},s}\right).$$

Thus, a rank increment improves the trade-off exactly when this shell sum is positive.

Proof. Subtract the protected coordinate contributions from the unprotected contributions in Proposition C.4. For $\lambda > 0$ and $g_s \neq 0$, dividing by the positive factor $2\lambda g_s^2/[h_s^2(h_s+2\lambda)^2]$ gives the threshold. For $\lambda = 0$ or $g_s = 0$, direct substitution gives zero. Summing the coordinate changes over T_{k+1} gives the rank-increment identity. □

Including signed retained transfer. A retained-loss surrogate that also includes first-order transfer is

$$\widehat{L}_{\mathrm{ret}}(\delta) = \sum_s a_s\delta_s + \frac{1}{2}\sum_s c_s\delta_s^2,$$

where a_s is the retained-loss gradient coordinate at the reference model. The decrease in this surrogate produced by protecting coordinate s is

$$\Delta_{\mathrm{ret},s} = \frac{2\lambda a_s g_s}{h_s(h_s+2\lambda)} + \frac{2\lambda c_s(h_s+\lambda)g_s^2}{h_s^2(h_s+2\lambda)^2}.$$

For $\lambda > 0$, the first term is negative when $a_s g_s < 0$: shrinking a beneficial ID-driven direction reduces its first-order retained-task gain. The second term is the saved positive-semidefinite sensitivity cost. The corresponding scalarized utility $G_{\mathrm{id}}(k) - \beta\widehat{L}_{\mathrm{ret}}(\delta^\star(k))$ improves upon increasing the rank when the shell sum of $\beta\Delta_{\mathrm{ret},s} - \Delta_{\mathrm{ID},s}$ is positive. This extension expresses both transfer preservation and sensitivity control in the same coordinates.

Interpretation. The support proposition supplies an optimal-rank boundary for the diagonal sensitivity model, including the possibility of equal-utility ranks. The shell rule evaluates the combined adaptation cost and retention benefit of increasing the actual top-k product block. Incorporating signed transfer also accounts for the value of useful OOD movement. These expressions provide a local framework for interpreting the measured rank-dependent ID/OOD trade-offs in Appendix D.13.

Practical guidance. The experimental default is the fixed cap $k = 768$, with $k_\ell = \min(768, m_\ell, n_\ell)$. This is approximately 20% of the singular-basis dimension of the principal 7B–8B matrices; the protected fraction varies with matrix shape and model size. Figure 2 motivates the focus on leading coordinates, and Appendix D.13 evaluates the empirical trade-off around this default.

C.6 Gradient-Flow View: Exponentially Damped Task-Induced Drift

We study the continuous-time Euclidean gradient flow of the regularized objective

$$F_k(\mathbf{W}) = f(\mathbf{W}) + \lambda \|\mathbf{U}_k^\top (\mathbf{W} - \mathbf{W}_0)\mathbf{V}_k\|_F^2.$$

Assume that f has a locally Lipschitz gradient and consider an interval $[0, T]$ on which the gradient-flow solution exists. The dynamics are

$$\frac{d\mathbf{W}(t)}{dt} = -\nabla f(\mathbf{W}(t)) - 2\lambda\, \mathbf{U}_k\big(\mathbf{U}_k^\top (\mathbf{W}(t) - \mathbf{W}_0)\mathbf{V}_k\big)\mathbf{V}_k^\top. \tag{39}$$

Protected coordinates. Define the protected coordinates

$$A(t) \triangleq \mathbf{U}_k^\top (\mathbf{W}(t) - \mathbf{W}_0)\mathbf{V}_k,$$

which measure the deviation of the weights from the pretrained model within the top-k singular subspace of $\mathbf{W}_0$. Multiplying Eq. (39) on the left by $\mathbf{U}_k^\top$ and on the right by $\mathbf{V}_k$ yields

$$\frac{dA(t)}{dt} = -\,\mathbf{U}_k^\top \nabla f(\mathbf{W}(t))\mathbf{V}_k - 2\lambda A(t). \tag{40}$$

Integral representation. Along the gradient-flow trajectory, define

$$G(t) \triangleq \mathbf{U}_k^\top \nabla f(\mathbf{W}(t))\mathbf{V}_k,$$

so that $\dot{A}(t) + 2\lambda A(t) = -G(t)$. Applying the integrating factor $e^{2\lambda t}$ yields

$$A(t) = e^{-2\lambda t}A(0) - \int_0^t e^{-2\lambda(t-s)} G(s)\, ds, \tag{41}$$

or equivalently,

$$A(t) = e^{-2\lambda t}A(0) - \int_0^t e^{-2\lambda(t-s)} \big(\mathbf{U}_k^\top \nabla f(\mathbf{W}(s))\mathbf{V}_k\big)\, ds.$$

For $\lambda > 0$, the first term represents an exponentially decaying initial deviation from the pretrained weights, while the second term is an exponentially weighted accumulation of task-induced gradients in the protected subspace. If

$$G_{\max}(t) = \sup_{0 \le s \le t} \left\| \mathbf{U}_k^\top \nabla f(\mathbf{W}(s))\mathbf{V}_k \right\|_F,$$

then, for $\lambda > 0$, Eq. (41) gives

$$\|A(t)\|_F \le e^{-2\lambda t}\|A(0)\|_F + \frac{G_{\max}(t)}{2\lambda}\left(1 - e^{-2\lambda t}\right). \tag{42}$$

In particular, when training starts from $\mathbf{W}_0$, $A(0) = 0$ and $\|A(t)\|_F \le G_{\max}(t)/(2\lambda)$. At $\lambda = 0$, Eq. (41) becomes $A(t) = A(0) - \int_0^t G(s)\, ds$, with the continuous-limit bound $\|A(t)\|_F \le \|A(0)\|_F + tG_{\max}(t)$.

Interpretation. For $\lambda > 0$, Equations (41) and (42) show that Euclidean RPSFT gradient flow exponentially weights the realized task gradients in the protected product block. A projected-gradient contribution at time s has weight $e^{-2\lambda(t-s)}$, and initialization at $\mathbf{W}_0$ gives $\|A(t)\|_F \le G_{\max}(t)/(2\lambda)$. Complementary coordinates follow the task gradient, whose value depends on all jointly evolving blocks. This identity describes the restoring dynamics associated with the objective. The trained-checkpoint principal-angle measurements in Appendix F.3 provide the empirical rotation analysis.

D Additional Results

D.1 Non-Math Adaptation Across Training Seeds

We evaluate full-parameter SFT on CommonsenseQA and ProofWriter to test adaptation beyond the mathematical domain. CommonsenseQA contains 1,221 evaluation examples and ProofWriter contains 2,048 evaluation examples. Each method is trained for two epochs with three independent training seeds (42, 43, and 44). Table 9 reports the final checkpoints: step 1200 for CommonsenseQA and step 2000 for ProofWriter. Values are mean ± standard error, computed as $\mathrm{SEM} = s/\sqrt{3}$ from the sample standard deviation s. The OOD macro averages IFEval-loose, MMLU-Pro, and SuperGPQA within each seed before aggregation. The Base row is a fixed pretrained checkpoint. For these non-math runs, EWC uses a penalty coefficient of 1, Sculpting Subspaces uses a maximum-rank ratio of 0.1 and a target-retention ratio of 0.8, Protected SFT uses protected rank $k = 768$, and PSP preserves 40% of the principal subspace.

Table 9: Final-checkpoint non-math results on Qwen2.5-7B after two epochs. Values are mean ± SEM (%) over three training seeds. OOD macro averages IFEval-loose, MMLU-Pro, and SuperGPQA within each seed. Bold marks the highest tuned mean per task.

Method	ID	IFEval-loose	MMLU-Pro	SuperGPQA	OOD macro
CommonsenseQA (step 1200)					
Base	79.69	61.92	60.00	16.79	46.24
SFT	81.46±0.76	39.56±0.75	50.48±1.72	14.73±0.22	34.92±0.40
EWC	81.08±0.79	40.79±0.91	50.95±4.15	15.06±0.57	35.60±1.27
Sculpting Subspaces	82.91±0.48	42.08±0.22	51.43±3.78	14.56±0.25	36.02±1.24
Protected SFT	82.77±0.43	39.25±0.53	49.05±1.72	14.73±1.05	34.34±0.77
PSP	82.39±0.52	41.22±0.85	53.33±1.72	**15.47±0.16**	36.68±0.41
RPSFT	**83.51±0.46**	**44.73±0.65**	**55.71±2.86**	15.39±0.30	**38.61±1.11**
ProofWriter (step 2000)					
Base	72.51	61.92	60.00	16.79	46.24
SFT	93.13±0.54	34.26±2.15	53.33±2.52	14.81±0.29	34.13±0.31
EWC	92.55±0.21	35.67±1.07	50.48±1.26	13.17±0.08	33.11±0.34
Sculpting Subspaces	89.88±4.56	38.44±2.03	**59.05±1.26**	16.54±0.79	38.01±0.90
Protected SFT	92.95±0.86	35.61±1.63	52.86±2.97	15.64±1.01	34.70±0.78
PSP	93.20±0.40	38.26±1.40	56.19±0.95	**16.95±0.58**	37.13±0.19
RPSFT	**94.09±0.12**	**46.52±1.37**	55.71±2.47	16.05±1.37	**39.43±1.26**

RPSFT achieves the highest ID and OOD-macro means on both tasks. Its OOD-macro gains over SFT are 3.69 points on CommonsenseQA and 5.30 points on ProofWriter. It also exceeds the strongest competing OOD macro by 1.93 points on CommonsenseQA (PSP) and 1.42 points on ProofWriter (Sculpting Subspaces).

D.2 Matched Hard and Random Subspace Controls

We compare RPSFT with two controls on Qwen2.5-7B using the same training hyperparameters on CommonsenseQA and ProofWriter, with protection rank $k = 768$. Table 10 compares the two-epoch checkpoints across three independent training seeds. For a pretrained matrix $\mathbf{W}_0 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, write $\Delta\mathbf{W} = \mathbf{W} - \mathbf{W}_0$ and let $\mathbf{U}_k, \mathbf{V}_k$ be its leading singular-vector bases.

Hard Top. This control enforces the exact product-block constraint $\mathbf{U}_k^\top \Delta\mathbf{W}\mathbf{V}_k = \mathbf{0}$ by removing the protected component after each optimizer update:

$$\Delta\mathbf{W} \leftarrow \Delta\mathbf{W} - \mathbf{U}_k\big(\mathbf{U}_k^\top \Delta\mathbf{W}\mathbf{V}_k\big)\mathbf{V}_k^\top. \tag{43}$$

The constraint holds the leading pretrained product block fixed, testing exact protection against RPSFT's soft anchoring.

Random Block. This control uses fixed, seeded random orthonormal bases $\mathbf{Q}_L, \mathbf{Q}_R$ of matching sizes. At matched block size and penalty scale, it tests direction selection by replacing each matrix's

unweighted penalty in Eq. (7) with

$$\mathcal{R}_{\text{random}}(\mathbf{W}) = \left\|\mathbf{Q}_L^\top \Delta\mathbf{W}\mathbf{Q}_R\right\|_F^2 . \tag{44}$$

The objective uses the same coefficient λ on this random-block penalty.

Table 10: Matched subspace controls for non-math adaptation on Qwen2.5-7B. Values are mean $\pm$ sample SD (%) across three training seeds at two epochs. OOD macro averages IFEval, MMLU-Pro, and SuperGPQA within each seed. RPSFT repeats the final-checkpoint results with SD in place of SEM. Bold marks the highest mean within each task.

Method	ID	IFEval	MMLU-Pro	SuperGPQA	OOD macro
CommonsenseQA					
Hard Top	82.06±0.25	40.05±2.34	46.19±2.97	**16.05±0.89**	34.10±0.29
Random Block	82.20±0.48	40.42±1.23	48.57±2.47	15.97±1.51	34.99±0.27
RPSFT	**83.51±0.80**	**44.73±1.12**	**55.71±4.95**	15.39±0.51	**38.61±1.92**
ProofWriter					
Hard Top	92.84±0.85	31.92±0.83	42.38±1.65	13.99±2.07	29.43±0.42
Random Block	92.66±0.87	32.90±1.11	50.00±5.15	**16.38±1.87**	33.09±1.03
RPSFT	**94.09±0.20**	**46.52±2.38**	**55.71±4.29**	16.05±2.38	**39.43±2.19**

RPSFT achieves the highest ID and OOD-macro means on both tasks. Relative to Hard Top and Random Block, its OOD gains are 4.51 and 3.62 points on CommonsenseQA, and 10.00 and 6.34 points on ProofWriter, supporting soft anchoring of the leading pretrained directions.

Non-math training overhead. Table 11 compares memory use and training throughput on both adaptation tasks. Throughput loss is $100(1 - T/T_{\text{SFT}})$, where T is samples per second.

Table 11: Training costs on CommonsenseQA and ProofWriter. Memory is in GiB; throughput loss is relative to SFT and computed before display rounding. Bold marks the best retention-method value per task; SFT provides the reference.

Method	GPU allocated	GPU reserved	CPU batch MaxRSS	Samples/s	Steps/s	Throughput loss (%)
CommonsenseQA						
SFT	21.95	22.78	383.80	8.54	0.53	0.00
EWC	50.44	54.35	**382.38**	**8.42**	**0.53**	**1.49**
Sculpting Subspaces	30.78	31.84	383.12	7.87	0.49	7.89
Protected SFT	**23.17**	**24.01**	390.35	8.12	0.51	4.99
PSP	28.16	29.40	389.76	5.16	0.32	39.65
RPSFT	28.60	37.70	384.00	8.31	0.52	2.73
ProofWriter						
SFT	22.03	23.62	382.98	8.49	0.53	0.00
EWC	50.52	54.57	382.36	**8.45**	**0.53**	**0.49**
Sculpting Subspaces	30.94	32.89	383.15	8.08	0.51	4.80
Protected SFT	**23.25**	**24.84**	382.42	7.87	0.49	7.30
PSP	28.25	30.16	**381.71**	5.20	0.33	38.70
RPSFT	28.60	38.30	383.00	8.27	0.52	2.57

RPSFT has the lowest throughput loss among the subspace-based methods, at 2.73% on CommonsenseQA and 2.57% on ProofWriter. Among the retention methods, EWC alone has lower throughput loss, while RPSFT retains 3.01 and 6.32 more OOD-macro points, respectively (Table 9). RPSFT also uses 43.30% and 43.38% less allocated GPU memory than EWC, combining stronger retention with lower memory use.

D.3 Anchoring-Strength Sweep Against L2 Init

We sweep $\lambda \in \{0.1, 0.2, 0.5, 2, 5, 10\}$ for RPSFT and L2 Init on ProofWriter, evaluating each run at step 500 across three training seeds. RPSFT uses $k = 768$ and the projected-block objective in Eq. (7); L2 Init applies $\mathcal{L}_{\mathrm{SFT}} + \lambda\|\theta - \theta_0\|_2^2$. The sweep OOD macro averages MMLU-Pro, SuperGPQA, and SafetyOOD within each seed, followed by the mean and sample SD across seeds. SafetyOOD uses the safer-response selection metric defined in Appendix E.1.

Table 12: ProofWriter anchoring-strength sweep at step 500: mean ± sample SD (%) over three training seeds. Bold marks the higher mean at each λ.

λ	Method	ProofWriter ID	MMLU-Pro	SuperGPQA	SafetyOOD	OOD macro
0.1	RPSFT	**83.56±2.73**	55.71±3.78	**17.45±1.75**	67.50±0.44	46.89±0.55
	L2 Init	77.56±2.04	**56.67±2.18**	17.37±1.59	**69.57±0.12**	**47.87±1.24**
0.2	RPSFT	**85.14±1.33**	56.19±3.60	**17.37±1.22**	67.57±0.60	47.04±1.16
	L2 Init	74.14±0.84	**57.62±0.82**	16.05±0.89	**70.80±0.46**	**48.16±0.06**
0.5	RPSFT	**81.02±1.17**	57.14±1.43	**17.37±0.87**	67.80±0.20	47.44±0.45
	L2 Init	76.79±1.98	**58.10±4.36**	15.31±1.13	**70.87±1.10**	**48.09±1.22**
2	RPSFT	**86.08±1.90**	**57.62±4.59**	**16.79±1.13**	67.60±0.79	**47.34±1.26**
	L2 Init	75.76±1.10	53.81±0.82	15.14±1.40	**70.77±0.35**	46.57±0.76
5	RPSFT	**84.46±1.91**	**57.14±4.29**	**17.28±1.73**	68.07±0.83	**47.50±0.59**
	L2 Init	74.72±0.50	54.76±2.18	15.72±1.45	**70.83±0.75**	47.11±1.24
10	RPSFT	**85.71±1.46**	**57.62±2.18**	**16.87±0.62**	67.93±0.45	**47.47±0.51**
	L2 Init	74.12±0.69	57.14±1.43	14.65±1.24	**70.47±0.55**	47.42±0.26

RPSFT achieves higher ID accuracy at every coefficient, with gains of 4.23–11.59 points over L2 Init. Across the sweep endpoints, its ID accuracy rises from 83.56 to 85.71, while L2 Init falls from 77.56 to 74.12, with intermediate fluctuations. The OOD macro remains within 46.89–47.50 for RPSFT and 46.57–48.16 for L2 Init. Thus, projected-block anchoring sustains stronger task adaptation across a 100-fold coefficient range while retaining stable OOD performance.

D.4 Direct Subspace and Importance Baselines

We compare RPSFT with EWC, O-LoRA, and Sculpting Subspaces on Qwen2.5-7B using the OpenR1-Math training data. Every method uses the same starting checkpoint, optimization schedule, checkpoint-selection rule, and evaluation protocol. The uncertainty in Table 5 is the sample standard deviation over three stochastic evaluation seeds from one fixed trained checkpoint per method. This experiment measures evaluation stability under shared training conditions.

Baseline configurations. EWC performs full-model training with a diagonal empirical Fisher estimated from squared per-example gradients on 64 deterministic training examples; its Fisher and reference tensors are cached in bfloat16. O-LoRA uses the Qwen2.5-7B Base checkpoint as its reference, a rank-32 Base-to-Instruct weight-change subspace, LoRA rank 32, alpha 64, zero dropout, and adapters on the query and value projections. Sculpting Subspaces uses `mrr=0.1`, `trr=0.8`, and `importance_samples=64`. It estimates adaptive protected ranks from 64 deterministic training examples and projects gradients for the attention query/key/value/output projections and the MLP gate/up/down projections during full-model training.

RPSFT has a higher mean than EWC on 11 of 12 metrics, O-LoRA on 10 of 12 metrics, and Sculpting Subspaces on 11 of 12 metrics. The strongest OOD separations appear on MMLU-Pro and SuperGPQA, while O-LoRA is competitive on IFEval and MMLU-Pro Pass@4. These results characterize the adaptation–retention profile of the evaluated full-model, adapter-based, Fisher-weighted, and projected-gradient configurations.

Objective scope. The layer-1 query-projection diagnostic measures gradient-energy enrichment in leading pretrained coordinates. The main RPSFT configuration regularizes every eligible two-dimensional attention, MLP, and embedding matrix while retaining full-parameter trainability.

D.5 Training-Seed Robustness on Math Adaptation

We combine the Qwen2.5-7B SFT and RPSFT runs reported in Table 1 with two additional independently trained checkpoints at step 19,200. Appendix E.2 gives the individual run scores and their machine-readable supplement.

Table 13: **Three-training-seed robustness on Qwen2.5-7B.** Values are mean ± sample SD across three independently trained checkpoints. Bold marks the higher mean within each metric block. ID uses benchmark-specific k from Table 1; OOD uses $k = 4$.

Metric	Method	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	$\overline{\text{Avg}}$
ID Avg@k	SFT	**14.17±1.04**	**18.20±1.03**	50.89±0.48	73.35±1.04	32.46±0.97	37.53±0.21	37.77±0.16
	RPSFT	13.20±1.15	18.06±0.64	**54.32±1.71**	**74.62±0.99**	**33.35±0.98**	**40.14±1.85**	**38.95±0.49**
ID Pass@k	SFT	40.00±8.82	33.33±3.34	85.83±5.20	**87.87±0.31**	56.62±1.47	54.72±0.22	59.73±2.55
	RPSFT	**42.22±5.09**	**40.00±3.33**	**88.33±1.44**	86.93±0.42	**57.23±1.06**	**58.42±1.97**	**62.19±1.10**
Metric	Method	GPQA	IFEval	MMLU-Pro	SuperGPQA	Safety	TruthfulQA	$\overline{\text{Avg}}$
OOD Avg@k	SFT	34.15±1.51	55.58±1.07	66.31±1.49	**28.42±0.19**	70.05±0.28	**61.91±0.65**	52.74±0.41
	RPSFT	**35.29±0.93**	**57.04±1.11**	**68.57±0.36**	28.21±0.75	**70.28±0.15**	61.46±0.46	**53.47±0.11**
OOD Pass@k	SFT	64.44±1.13	70.79±1.15	83.33±2.97	53.74±1.11	**82.73±0.90**	78.80±1.27	72.31±0.20
	RPSFT	**66.15±1.87**	**71.59±1.55**	**84.76±0.82**	**54.32±0.65**	80.73±1.53	**79.09±0.92**	**72.78±0.59**

RPSFT has the higher $\overline{\text{Avg}}$ in all four ID/OOD Avg/Pass blocks, confirming that its aggregate advantage persists across independently trained checkpoints.

D.6 Post-DAPO Robustness Across Training Seeds

We combine the final Qwen2.5-7B SFT→DAPO and RPSFT→DAPO results in Tables 2 and 3 with two additional independently trained DAPO runs for each initialization. The individual run scores appear in Appendix E.2.

Table 14: **Post-DAPO training-seed robustness on Qwen2.5-7B.** Values are mean ± sample SD across the original run and two additional independently trained DAPO runs. Bold marks the higher mean within each metric block. ID uses benchmark-specific k from Table 2; OOD uses $k = 4$. $\overline{\text{Avg}}$ SD is computed across the three per-run six-benchmark macros.

Metric	Initialization	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	$\overline{\text{Avg}}$
ID Avg@k	SFT→DAPO	18.20±1.34	20.97±0.43	58.59±0.95	79.30±0.78	38.14±0.73	44.12±0.67	43.22±0.63
	RPSFT→DAPO	**20.07±1.69**	**22.01±0.73**	**60.89±1.33**	**79.85±0.80**	**39.48±0.65**	**46.09±0.64**	**44.73±0.96**
ID Pass@k	SFT→DAPO	44.45±3.85	35.55±3.85	90.00±0.00	**89.67±0.90**	57.48±0.56	60.05±0.52	62.87±0.13
	RPSFT→DAPO	**47.78±6.94**	**40.00±3.33**	**92.50±2.50**	89.40±1.06	**59.68±0.85**	**61.58±0.48**	**65.16±1.85**
Metric	Initialization	GPQA	IFEval	MMLU-Pro	SuperGPQA	Safety	TruthfulQA	$\overline{\text{Avg}}$
OOD Avg@k	SFT→DAPO	**37.76±0.36**	53.10±0.40	71.79±0.95	31.21±0.97	72.61±0.67	65.45±0.84	55.32±0.46
	RPSFT→DAPO	37.67±0.34	**54.84±0.76**	**72.74±0.55**	**32.67±0.88**	**73.03±1.17**	**66.28±0.96**	**56.20±0.44**
OOD Pass@k	SFT→DAPO	64.14±0.68	68.76±1.40	**85.71±1.43**	54.65±1.27	**81.23±0.97**	79.19±0.61	72.28±0.17
	RPSFT→DAPO	**65.03±1.29**	**70.43±0.56**	85.24±2.18	**57.28±1.07**	81.03±1.53	**79.58±0.99**	**73.10±0.37**

RPSFT has the higher mean on 20 of 24 benchmark metrics and the higher $\overline{\text{Avg}}$ in all four ID/OOD Avg/Pass blocks after DAPO.

D.7 Stochastic Evaluation Stability on Math and OOD Benchmarks

We evaluate one fixed SFT checkpoint and one fixed RPSFT checkpoint per model with three stochastic evaluation seeds. Table 15 reports mean ± sample SD. Avg@q averages correctness across generated responses and Pass@q records whether any response is correct; $q = 16$ for the three ID benchmarks and $q = 4$ for the three OOD benchmarks. Appendix E.1 gives the exact aggregation and distinguishes evaluation-seed SD from training-seed SD and standard error.

Table 15: Evaluation stability for Qwen2.5-7B and Llama-3.1-8B. Values are mean ± sample SD over three stochastic evaluation seeds from one fixed trained checkpoint per method.

Model	Domain	Benchmark	SFT Avg@q	RPSFT Avg@q	SFT Pass@q	RPSFT Pass@q
Qwen2.5-7B	ID	AIME24	14.03 ± 1.15	14.38 ± 1.46	42.22 ± 1.92	40.00 ± 3.33
	ID	AIME25	18.61 ± 0.84	18.89 ± 0.64	37.78 ± 1.92	41.11 ± 1.92
	ID	AMC23	49.37 ± 0.98	52.66 ± 1.02	87.50 ± 2.50	88.33 ± 1.44
	OOD	IFEval	54.24 ± 0.22	56.62 ± 1.53	69.50 ± 0.19	71.66 ± 1.66
	OOD	MMLU-Pro	65.47 ± 1.44	67.74 ± 0.55	82.38 ± 2.18	81.91 ± 2.18
	OOD	SuperGPQA	28.87 ± 0.40	29.18 ± 1.81	53.42 ± 0.72	54.98 ± 1.17
Llama-3.1-8B	ID	AIME24	4.93 ± 0.12	4.24 ± 1.27	18.89 ± 1.92	17.78 ± 1.92
	ID	AIME25	5.35 ± 0.43	5.56 ± 1.26	18.89 ± 5.09	23.33 ± 5.77
	ID	AMC23	31.04 ± 1.48	32.08 ± 1.48	76.67 ± 1.44	80.83 ± 1.44
	OOD	IFEval	32.12 ± 0.40	31.75 ± 0.18	34.94 ± 0.67	35.30 ± 0.37
	OOD	MMLU-Pro	54.64 ± 1.29	54.05 ± 1.03	76.19 ± 0.83	80.00 ± 5.15
	OOD	SuperGPQA	22.74 ± 1.43	22.78 ± 0.91	48.48 ± 5.65	49.05 ± 3.97

RPSFT has the higher mean on 18 of 24 metrics: five of six Qwen ID metrics, five of six Qwen OOD metrics, four of six Llama ID metrics, and four of six Llama OOD metrics. The clearest separations relative to evaluation variability occur on Qwen AMC23 Avg, IFEval Avg and Pass, MMLU-Pro Avg, and Llama AMC23 Pass. The independent-training-seed results in Appendices D.1 and D.5 provide complementary training robustness.

D.8 Local Quadratic Validation Along Checkpoint Directions

We validate the local second-order model along the actual Base→SFT and SFT→RL checkpoint directions of Qwen2.5-7B. For starting and ending parameters θ_a and θ_b, define

$$\theta(t) = \theta_a + td, \qquad d = \theta_b - \theta_a, \qquad \ell(t) = \mathcal{L}_{\text{heldout}}(\theta(t)) \,. \tag{45}$$

We estimate the directional slope and curvature by central differences at $\delta = 0.05$,

$$\widehat{g} = \frac{\ell(\delta) - \ell(-\delta)}{2\delta}, \qquad \widehat{h} = \frac{\ell(\delta) - 2\ell(0) + \ell(-\delta)}{\delta^2}, \qquad q(t) = \ell(0) + t\widehat{g} + \tfrac{1}{2}t^2\widehat{h}, \tag{46}$$

and evaluate at the held-out distances $t \in \{-0.1, +0.1\}$. The loss is mean target-answer-token NLL. Parameter interpolation and logits use FP32. The ID split contains 64 Math-500 examples; the OOD split contains 32 MMLU-Pro and 32 TruthfulQA-MC examples.

The quadratic predictor matches the exact sign in all eight held-out tests, achieves 1.59% median relative error and 0.02918 NLL maximum absolute error, and removes at least 71.1% of the first-order prediction error. The relative error reaches 45.9% for the positive Base→SFT OOD displacement. These measurements evaluate local prediction at $t = \pm 0.1$, with the directional fit computed at $\delta = 0.05$.

Signed directional contributions. The symmetric predictions also determine the fitted slope and curvature. Writing $\Delta q(t) = q(t) - q(0)$, the Base→SFT OOD entries give, using the rounded table values,

$$\widehat{g} = \frac{\Delta q(0.1) - \Delta q(-0.1)}{0.2} \approx 1.64, \qquad \widehat{h} = \frac{\Delta q(0.1) + \Delta q(-0.1)}{0.01} \approx -26.0.$$

At $t = 0.1$, the fitted first-order contribution is approximately $+0.164$ NLL and the quadratic contribution is approximately -0.130 NLL. This direction therefore exhibits first-order interference partially offset by negative curvature. The experiment validates the signed local expansion in Eq. (2); the positive-semidefinite Fisher quantity in Eq. (3) describes local model-distribution sensitivity.

D.9 Dominant-Direction Restoration with Rank-Matched Controls

We perform a direct weight-space intervention on Llama-3.1-11B adapted to the original card-game rule and evaluated on 200 examples from a rule-changed game, and on Qwen2.5-7B adapted to

Table 16: Held-out validation of the local quadratic approximation. Exact change is $\ell(t) - \ell(0)$ and quadratic change is $q(t) - q(0)$. Error metrics are reconstructed from the displayed three-decimal loss changes.

Direction	Split	t	Exact change	Quadratic change	Absolute error	Relative error	First-order error removed
Base→SFT	ID	-0.1	$+0.170$	$+0.175$	0.005	2.9%	76.7%
Base→SFT	ID	$+0.1$	-0.117	-0.122	0.005	4.3%	84.1%
Base→SFT	OOD	-0.1	-0.281	-0.294	0.013	4.6%	88.9%
Base→SFT	OOD	$+0.1$	$+0.064$	$+0.034$	0.030	46.9%	70.0%
SFT→RL	ID	-0.1	-0.226	-0.227	0.001	0.4%	83.3%
SFT→RL	ID	$+0.1$	$+0.212$	$+0.213$	0.001	0.5%	87.5%
SFT→RL	OOD	-0.1	-0.216	-0.216	$0.000^{\dagger}$	$0.0\%^{\dagger}$	$100.0\%^{\dagger}$
SFT→RL	OOD	$+0.1$	$+0.201$	$+0.202$	0.001	0.5%	87.5%

Definitions. Let $d_t = \ell(t) - \ell(0)$ and $p_t = q(t) - q(0)$. For each direction–split pair, the first-order contribution is reconstructed as $b_t = (p_t - p_{-t})/2$. The absolute errors of the first-order and quadratic predictors are $E_1 = |d_t - b_t|$ and $E_2 = |d_t - p_t|$, respectively. Relative error is $100E_2/|d_t|$, and first-order error removed is $100(1 - E_2/E_1)$; both are reported as percentages. †The displayed loss changes coincide in this row, yielding zero reconstructed error at three-decimal resolution.

OpenR1-Math and evaluated on instruction following. Llama Base and SFT OOD accuracies are 17.00% and 6.92%; Qwen Base and SFT OOD accuracies are 55.82% and 41.59%. The intervention is applied simultaneously to every eligible two-dimensional matrix, including embedding matrices.

For each $m \times n$ weight matrix, let $r = \min(m, n)$ and write the economy-size Base and SFT decompositions as $W^B = \sum_{i=1}^{r} \sigma_i^B u_i^B (v_i^B)^\top$ and $W^S = \sum_{i=1}^{r} \sigma_i^S u_i^S (v_i^S)^\top$. Set $\kappa = \min(k, r)$. The leading-direction intervention reuses every SFT-derived coefficient, substitutes the leading κ Base direction pairs, and retains the SFT components at all remaining indices:

$$W_{\mathrm{top}}(k) = \sum_{i=1}^{\kappa} \sigma_i^S u_i^B (v_i^B)^\top + \sum_{i=\kappa+1}^{r} \sigma_i^S u_i^S (v_i^S)^\top. \tag{47}$$

For a uniformly sampled rank-matched index set $\mathcal{I}$ with $|\mathcal{I}| = \kappa$, the random control is

$$W_{\mathrm{random}}(\mathcal{I}) = \sum_{i \in \mathcal{I}} \sigma_i^S u_i^B (v_i^B)^\top + \sum_{i \notin \mathcal{I}} \sigma_i^S u_i^S (v_i^S)^\top. \tag{48}$$

Random mean averages five independent random-index seeds at each k, matching the number of substituted components per matrix. The hybrid combines Base direction pairs at the selected indices with SFT direction pairs at the remaining indices. Its geometry, perturbation norm, and conditioning depend on the selected components. The retained σ_i^S are reconstruction coefficients in this generally nonorthogonal hybrid basis. The Fisher block and energy columns characterize the leading $\kappa \times \kappa$ product block associated with the Base bases; the intervention substitutes κ paired singular components. At full per-matrix rank, both constructions equal $\mathbf{U}^B \mathbf{\Sigma}^S (\mathbf{V}^B)^\top$.

At $k = 768$, leading restoration improves the unmodified SFT checkpoint by 5.05 points on Llama and 2.64 points on Qwen. Its advantage over the random control is 4.49 points on Llama and approximately 15.4 points on Qwen. On Qwen, the reported leading-block diagnostic at $k = 1024$ occupies 8.16% of the SVD core and captures 35.23% of the empirical-Fisher energy. Leading restoration reaches 61.37% OOD accuracy at this rank and 62.29% at $k = 2048$, with random-control accuracies of 10.65% and 12.31%. These results identify leading-component selection as an effective reconstruction strategy within the evaluated intervention family.

D.10 GRPO FROM SFT AND RPSFT INITIALIZATIONS

We train the standard GRPO objective on Qwen2.5-7B from the corresponding SFT and RPSFT checkpoints. Table 18 reproduces each reported standard deviation exactly.

Table 17: Dominant-direction restoration and rank-matched random controls. Recovered OOD is post-intervention accuracy. Random mean averages five independent random-rank seeds. The Llama evaluation uses three stochastic evaluation seeds; the table reports aggregate means.

Model/task	Recovery	k	Top-k Fisher block (%)	Top-k Fisher energy (%)	Recovered OOD (%)
Llama-3.1-11B / card game	Top	64	0.02	0.47	7.26
	Random mean	64	—	—	7.82
	Top	256	0.39	3.14	7.36
	Random mean	256	—	—	8.60
	Top	512	1.56	8.06	8.97
	Random mean	512	—	—	7.14
	Top	768	3.52	13.46	11.97
	Random mean	768	—	—	7.48
Qwen2.5-7B / instruction following	Top	64	0.03	0.98	38.46
	Random mean	64	—	—	30.00
	Top	256	0.51	5.93	42.31
	Random mean	256	—	—	32.69
	Top	512	2.04	13.27	40.39
	Random mean	512	—	—	27.69
	Top	768	4.59	21.30	44.23
	Random mean	768	—	—	28.85
	Top	1024	8.16	35.23	61.37
	Random mean	1024	—	—	10.65
	Top	2048	32.65	68.27	62.29
	Random mean	2048	—	—	12.31
	Top	3584 (full)	100.00	100.00	63.59
	Random mean	3584 (full)	—	—	63.59

Table 18: Qwen2.5-7B GRPO results. Entries are mean $\pm$ reported SD from three independent training seeds.

Domain	Benchmark	Metric	SFT-GRPO	RPSFT-GRPO
ID	AIME24	Avg@16	14.72 ± 0.64	16.60 ± 0.48
ID	AIME24	Pass@16	47.78 ± 3.85	47.78 ± 3.85
ID	AIME25	Avg@16	16.94 ± 1.05	18.12 ± 0.75
ID	AIME25	Pass@16	36.67 ± 5.77	37.78 ± 1.92
ID	AMC23	Avg@16	53.80 ± 1.21	55.99 ± 0.94
ID	AMC23	Pass@16	87.50 ± 0.00	87.50 ± 4.33
OOD	IFEval	Avg@4	51.63 ± 0.44	53.91 ± 0.85
OOD	IFEval	Pass@4	68.70 ± 0.53	69.69 ± 0.49
OOD	MMLU-Pro	Avg@4	69.76 ± 2.51	70.71 ± 2.23
OOD	MMLU-Pro	Pass@4	82.86 ± 3.78	84.76 ± 3.30
OOD	SuperGPQA	Avg@4	30.06 ± 0.86	29.84 ± 0.18
OOD	SuperGPQA	Pass@4	53.74 ± 1.51	54.49 ± 1.24

RPSFT-GRPO attains a higher mean on four of six ID metrics with two exact ties, and on five of six OOD metrics. Its Avg gains are 1.88 points on AIME24, 1.18 on AIME25, 2.19 on AMC23, 2.28 on IFEval, and 0.95 on MMLU-Pro. These results extend the initialization advantage from DAPO to the standard GRPO objective.

D.11 Transfer to the DeepSeekMath Model Family

We further evaluate RPSFT on DeepSeek-7B. Table 19 reports accuracy point estimates; RPSFT-768 uses protected rank $k = 768$.

RPSFT-768 exceeds SFT on seven of eight benchmarks and matches or exceeds the Base checkpoint on every benchmark. This result extends the observed adaptation–retention trade-off to the DeepSeekMath family.

D.12 Optimized Memory–Throughput Trade-offs

For each regularized matrix $\mathbf{W}_\ell^0 \in \mathbb{R}^{m_\ell \times n_\ell}$, we store $\mathbf{U}_{0,\ell}^{(k)}$, $\mathbf{V}_{0,\ell}^{(k)}$, and S_ℓ^{ref}:

$$\text{extra memory per layer} \;=\; \mathcal{O}\big((m_\ell + n_\ell)k_\ell + k_\ell^2\big)\,. \tag{49}$$

Table 19: DeepSeek-7B accuracy (%). Entries are point estimates.

Benchmark	Base	SFT	RPSFT-768
AIME24	0.00	16.67	**20.00**
AIME25	0.00	6.67	**10.00**
ARC-C	68.00	60.00	**72.00**
CommonsenseQA	57.69	46.15	**57.69**
IFEval-loose	23.33	20.00	**23.33**
MMLU-Pro	34.29	25.71	**35.71**
GPQA	24.00	**38.00**	28.00
SuperGPQA	6.00	16.00	**24.00**

For the truncated ranks used here, the dominant cache term scales linearly with protected rank. Our analytical implementation computes the projected residual with detached tensors and injects its closed-form gradient during backward. Table 7 reports the original training costs.

For each regularized matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, computing the projected block $(\mathbf{U}_0^{(k)})^\top \mathbf{W}\mathbf{V}_0^{(k)}$ adds $O(mnk)$ FLOPs per evaluation, with the same order for backpropagation. If applied every s optimization steps, the arithmetic-average projection cost becomes $O(mnk/s)$ per step. Relative training overhead depends on the token workload, sharding, communication, and optimizer implementation; Tables 7 and 20 give the measured end-to-end costs.

Table 20 isolates the measured non-math GPU-cache overhead against vanilla SFT.

Table 20: Measured non-mathematical training overhead.

Method	RPSFT cache location	Peak GPU allocated	Peak GPU reserved (single GPU)	CPU MaxRSS aggregate	Samples/s	Steps/s	Throughput overhead vs. SFT
Vanilla SFT	None	22.358 GiB	33.408 GiB	178.98 GiB	5.268	0.329	—
Analytical RPSFT (GPU cache)	Entire cache on GPU	28.569 GiB	37.734 GiB	165.17 GiB	5.060	0.316	3.96%

On this non-math workload, analytical GPU-cache RPSFT adds 6.211 GiB of peak allocated GPU memory (27.78% over SFT) and has 3.96% throughput overhead, reaching 5.060 samples/s versus 5.268 for SFT.

D.13 Rank Sensitivity and OOD Pass@1 Results

We study the sensitivity of RPSFT to the protected rank k on Llama-3.1-8B-Instruct as the empirical counterpart to the rank-boundary analysis in Appendix C.5. Here, SFT corresponds to $k = 0$, and our fixed default is $k = 768$. The LoRA row uses vanilla LoRA with adapter rank $r = 32$, and the L2 Init reference applies explicit full-weight anchoring around the pretrained model (Kumar et al., 2024). It is plotted at $k = 4096$, the principal hidden dimension, to mark the full-weight reference. Llama achieves its strongest ID/OOD operating points at intermediate projected ranks: ID Avg@k peaks at $k = 512$ and OOD Avg@1 peaks at $k = 768$, followed by lower averages at $k = 2048$ and for full-weight L2 Init. Appendix C.5 expresses the adaptation cost of protecting additional coordinates and the role of signed retained transfer. The negative Llama OOD alignments in Figure 4 motivate examining how rank selection preserves useful OOD movement.

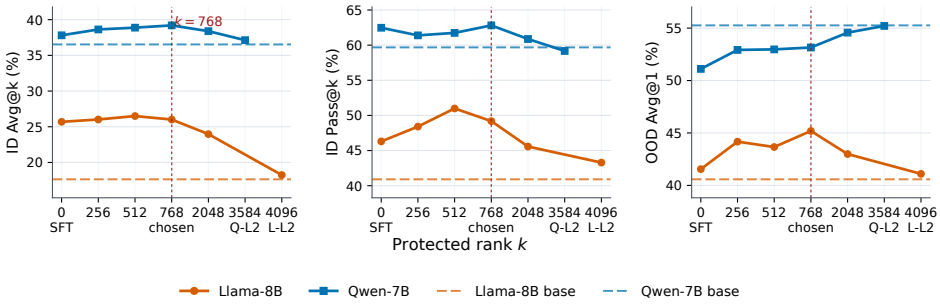

Figure 5: Rank sensitivity at $\lambda = 1$. The x-axis is the protected rank k and the y-axes report ID Avg@k, ID Pass@k, and OOD Avg@1. 0 is vanilla SFT, 768 is our fixed default, Q-L2/L-L2 use explicit full-weight L2 Init (Kumar et al., 2024), and dashed lines show base-model scores. Scores average six ID math benchmarks and six OOD benchmarks.

Table 21: Rank robustness on Llama-3.1-8B-Instruct across ID Avg@k, ID Pass@k, and OOD Avg@1 (%). The LoRA row is vanilla LoRA with adapter rank $r = 32$; RPSFT rows vary protected rank k. Base cells are left unmarked, and bold marks the best non-base value in each domain block and column.

Domain	Setting	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	$\overline{\text{Avg}}$
ID Avg@k	Base	3.13	0.83	21.25	43.65	22.33	14.63	17.64
	SFT	5.00	5.00	32.19	59.45	26.88	25.59	25.69
	Vanilla LoRA (r=32)	3.13	2.50	23.75	47.80	21.60	14.63	18.90
	RPSFT-256	**5.42**	6.46	29.84	59.05	**28.63**	26.70	26.02
	RPSFT-512	4.58	**7.50**	**34.06**	58.20	27.34	**27.34**	**26.50**
	RPSFT-768	4.38	6.67	31.56	**60.45**	27.48	25.52	26.01
	RPSFT-2048	4.17	5.21	28.44	56.70	25.64	23.63	23.97
	L2 Init (k=4096)	4.38	1.04	21.88	44.60	21.51	16.00	18.24
ID Pass@k	Base	20.00	13.33	72.50	65.00	44.85	29.78	40.91
	SFT	20.00	13.33	77.50	**78.60**	47.06	41.33	46.30
	Vanilla LoRA (r=32)	23.33	20.00	80.00	67.40	48.53	29.78	44.84
	RPSFT-256	20.00	**26.66**	75.00	75.40	**50.74**	42.67	48.41
	RPSFT-512	20.00	23.33	**87.50**	76.60	49.26	**49.26**	**50.99**
	RPSFT-768	20.00	20.00	85.00	77.20	**50.74**	42.22	49.19
	RPSFT-2048	16.67	20.00	72.50	76.60	47.79	39.85	45.57
	L2 Init (k=4096)	**26.66**	13.33	72.50	67.60	47.79	31.85	43.29
		GPQA	IFEval	MMLU-Pro	SuperGPQA	Safety	TruthfulQA	$\overline{\text{Avg}}$
OOD Avg@1	Base	25.89	32.53	47.14	18.77	61.10	58.04	40.58
	SFT	31.70	31.42	52.86	19.75	57.90	55.70	41.55
	Vanilla LoRA (r=32)	28.57	31.79	48.57	17.78	63.80	56.14	41.11
	RPSFT-256	33.93	30.87	**58.57**	21.98	63.40	56.29	44.17
	RPSFT-512	32.81	30.87	52.86	20.74	**68.70**	55.99	43.66
	RPSFT-768	**34.60**	**33.46**	**58.57**	**24.20**	61.70	58.63	**45.19**
	RPSFT-2048	33.26	31.05	54.29	23.95	58.80	56.58	42.99
	L2 Init (k=4096)	26.34	32.53	47.14	16.79	62.70	**61.11**	41.10

Qwen2.5-7B robustness sweep. Table 22 reports the corresponding sweep on Qwen2.5-7B-Instruct. SFT corresponds to $k = 0$, vanilla LoRA uses adapter rank $r = 32$, and the fixed RPSFT cap is $k = 768$. The explicit full-weight L2 Init reference is plotted at the principal hidden dimension 3584. Qwen ID Avg@k and ID Pass@k peak at $k = 768$, while the OOD Avg@1 average rises across the evaluated projected ranks and reaches its closest value to Base under full-weight L2 Init. These measurements expose the empirical adaptation–retention trade-off described by the local surrogate analysis in Appendix C.5. The evaluated rank-32 LoRA configuration has lower OOD Avg@1 than the RPSFT configurations in this sweep.

Table 22: Rank robustness on Qwen2.5-7B-Instruct across ID Avg@k, ID Pass@k, and OOD Avg@1 (%). The LoRA row is vanilla LoRA with adapter rank $r = 32$; RPSFT rows vary protected rank k. Base cells are left unmarked, and bold marks the best non-base value in each domain block and column.

Domain	Setting	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	$\overline{\text{Avg}}$
ID Avg@k	Base	12.08	7.71	52.03	72.85	37.09	37.41	36.53
	SFT	15.21	**19.38**	50.47	72.85	31.53	37.41	37.81
	Vanilla LoRA (r=32)	11.25	13.54	46.41	71.80	34.24	33.96	35.20
	RPSFT-256	**16.25**	17.50	54.38	73.25	32.81	37.56	38.63
	RPSFT-512	15.00	**19.38**	52.97	73.30	33.23	**39.37**	38.88
	RPSFT-768	14.38	18.75	**55.31**	**74.95**	33.82	38.00	**39.20**
	RPSFT-2048	14.17	17.08	51.72	74.70	33.50	39.15	38.39
	L2 Init (k=3584)	11.88	9.38	51.88	74.70	**37.55**	37.37	37.13
ID Pass@k	Base	36.67	36.67	90.00	84.60	55.15	54.96	59.68
	SFT	**50.00**	36.67	87.50	**87.60**	58.09	54.96	62.47
	Vanilla LoRA (r=32)	40.00	30.00	77.50	84.60	55.88	52.15	56.69
	RPSFT-256	46.67	36.67	**90.00**	86.60	54.04	54.37	61.39
	RPSFT-512	43.33	40.00	85.00	86.60	**58.46**	**57.04**	61.74
	RPSFT-768	43.33	**43.33**	**90.00**	87.40	56.62	56.15	**62.81**
	RPSFT-2048	43.33	36.67	85.00	86.60	57.35	56.29	60.87
	L2 Init (k=3584)	30.00	40.00	**90.00**	84.80	58.09	52.15	59.17
		GPQA	IFEval	MMLU-Pro	SuperGPQA	Safety	TruthfulQA	$\overline{\text{Avg}}$
OOD Avg@1	Base	33.93	61.37	67.14	27.65	74.50	66.96	55.26
	SFT	32.80	52.13	65.71	26.42	70.10	59.50	51.11
	Vanilla LoRA (r=32)	32.59	45.84	58.57	23.21	67.60	64.04	48.64
	RPSFT-256	34.82	56.38	67.14	29.38	70.00	59.80	52.92
	RPSFT-512	**36.38**	56.01	65.71	**29.63**	68.80	61.26	52.97
	RPSFT-768	34.60	54.34	70.00	26.42	**71.10**	62.43	53.15
	RPSFT-2048	33.71	57.67	**74.29**	27.41	70.90	63.45	54.57
	L2 Init (k=3584)	35.49	**63.03**	68.57	27.41	70.10	**66.67**	**55.21**

OOD Pass@1 results. Tables 23 and 24 report the earlier one-sample OOD evaluation. The main results use the four-sample OOD Avg@k and Pass@k evaluation in Tables 1 and 3.

Table 23: SFT-stage OOD Pass@1 results on the six OOD benchmarks. Bold marks the strongest tuned result within each model block.

Benchmark	**Llama-3.1-8B-Instruct**					**Qwen2.5-7B-Instruct**					**Qwen2.5-3B-Instruct**				
	Base	SFT	IW	DFT	RPSFT	Base	SFT	IW	DFT	RPSFT	Base	SFT	IW	DFT	RPSFT
GPQA	25.89	31.70	**37.72**	27.68	34.60	33.93	32.80	32.37	30.36	**34.60**	27.01	30.58	27.01	26.56	**31.70**
IFEval	32.53	31.42	32.53	31.98	**33.46**	61.37	52.13	54.34	**59.70**	54.34	54.16	47.32	46.03	51.20	**52.13**
MMLU-Pro	47.14	52.86	51.43	52.86	**58.57**	67.14	65.71	58.57	62.86	**70.00**	52.86	**45.71**	**45.71**	34.29	44.28
SuperGPQA	18.77	19.75	21.48	17.53	**24.20**	27.65	26.42	**26.91**	25.19	26.42	18.27	19.26	19.51	16.30	**21.48**
Safety	61.10	57.90	**62.20**	60.80	61.70	74.50	70.10	70.10	**71.90**	71.10	65.90	59.60	60.00	**65.80**	62.20
TruthfulQA	58.04	55.70	49.56	**60.82**	58.63	66.96	59.50	60.53	**65.06**	62.43	59.94	49.42	50.29	**57.31**	51.61
$\overline{\text{Avg}}$	40.58	41.55	42.49	41.95	**45.19**	55.26	51.11	50.47	52.52	**53.15**	46.36	41.99	41.43	41.91	**43.90**

Table 24: Downstream RL OOD Pass@1 results under DAPO. Entries are init→DAPO, and final checkpoints are selected by ID performance.

Method	GPQA	IFEval	MMLU-Pro	SuperGPQA	Safety	TruthfulQA	$\overline{\text{Avg}}$
Llama-3.1-8B-Instruct							
SFT	31.70→35.27	31.42→31.79	52.86→50.00	19.75→24.94	57.90→59.20	55.70→59.06	41.55→43.38
IW	37.72→**35.71**	32.53→31.24	51.43→58.57	21.48→24.69	62.20→60.50	49.56→54.82	42.49→44.25
DFT	27.68→27.23	31.98→31.05	52.86→**60.00**	17.53→19.75	60.80→58.60	60.82→**64.18**	41.95→43.47
RPSFT	34.60→35.27	33.46→**32.53**	58.57→54.29	24.20→**29.14**	61.70→**60.80**	58.63→62.28	45.19→**45.72**
Qwen2.5-7B-Instruct							
SFT	32.80→33.92	52.13→52.31	65.71→71.43	26.42→32.10	70.10→72.90	59.50→65.20	51.11→54.64
IW	32.37→37.28	54.34→56.38	58.57→**74.29**	26.91→32.59	70.10→73.50	60.53→61.99	50.47→**56.01**
DFT	30.36→31.25	59.70→**60.63**	62.86→54.29	25.19→26.17	71.90→**73.70**	65.06→**68.27**	52.52→52.38
RPSFT	34.60→**39.29**	54.34→54.53	70.00→71.43	26.42→**32.84**	71.10→72.60	62.43→63.74	53.15→55.74
Qwen2.5-3B-Instruct							
SFT	30.58→27.68	47.32→48.43	45.71→47.14	19.26→22.72	59.60→62.50	49.42→52.92	41.99→43.56
IW	27.01→27.90	46.03→46.58	45.71→48.57	19.51→**22.96**	60.00→62.30	50.29→51.32	41.43→43.27
DFT	26.56→25.67	51.20→**51.76**	34.29→44.29	16.30→17.78	65.80→**64.80**	57.31→**56.87**	41.91→43.53
RPSFT	31.70→**28.79**	52.13→48.61	44.28→**55.71**	21.48→20.99	62.20→63.70	51.61→53.95	43.90→**45.29**

E EXPERIMENT DETAILS

Precomputation. We compute the baseline SVD blocks immediately after loading the pretrained model. For every eligible two-dimensional matrix $\ell \in \mathcal{M}$, we set $k_\ell = \min(768, m_\ell, n_\ell)$, compute the corresponding truncated pretrained bases, and store $(\mathbf{U}_{0,\ell}^{(k_\ell)}, \mathbf{V}_{0,\ell}^{(k_\ell)}, S_\ell^{\text{ref}})$. The eligible set includes attention, MLP, and embedding matrices. Precomputation takes about three minutes for Llama-8B.

Hyperparameters. RPSFT uses the protected rank k (distinct from the generation count in Avg@k and Pass@k) and anchoring strength λ. As shown in Figure 5, performance is robust in intermediate ranks, while overly large ranks limit adaptation. Unless stated otherwise, we use $k = 768$ and $\lambda = 1$ across all three main models; the rank is approximately the top 20% singular rank of the principal 7B–8B matrices. The coefficient multiplies the sum of squared projected-block norms in Eq. (7). Appendix C.5 gives the rank-boundary analysis, with rank sweeps in Appendix D.13 and the coefficient sweep in Appendix D.3.

Training setup For OpenR1-Math supervised fine-tuning, we train Qwen2.5-7B and Qwen2.5-3B for 12 epochs and Llama-3.1-8B for 20 epochs. The main math comparison uses one trained checkpoint per method under the 8-H200 setup; Appendix D.5 additionally reports three independently trained Qwen2.5-7B SFT and RPSFT checkpoints, while Appendix D.7 reports stochastic evaluation stability. The CommonsenseQA and ProofWriter runs train Qwen2.5-7B for two epochs with three independent training seeds and checkpoints every half epoch. We report matched final checkpoints across methods. Shared optimization settings remain fixed within each comparison, with method-specific settings listed in Appendix D.4.

Table 25 lists the shared SFT hyperparameters used for all SFT-stage methods.

Table 25: Shared supervised fine-tuning hyperparameters.

Parameter	Value
Learning rate	1×10^{-6}
Epochs	12 for Qwen math; 20 for Llama math; 2 for non-math
Precision	`bf16`
LR scheduler	cosine
Warmup ratio	0.03
Weight decay	0.0

For downstream reinforcement learning, we use DAPO and initialize every run from the corresponding supervised checkpoint.

Table 26: Shared reinforcement-learning fine-tuning hyperparameters.

Parameter	Value
Actor learning rate	1×10^{-6}
Train batch size	256 prompts
PPO mini-batch size	32 prompts
Generation batch size	128 prompts
Responses per prompt	8
Max prompt length	2048 tokens
Max response length	10240 tokens
Temperature	1.0
Validation top-p	0.7
PPO-style clip low / high	0.2 / 0.28
Total training steps	100
Overlong buffer length	2048 tokens
Overlong penalty factor	1.0

Compute and artifacts. All SFT and RL training runs use 8 H200 GPUs. We provide the code, dataset preparation and evaluation artifacts, training configurations, and documentation as supplemental material for reproduction, and will make the artifact public on GitHub after the anonymous review period. We use VERL as our main training framework.

E.1 Evaluation Subsets, Scoring, and Aggregation

Benchmark identities. In-domain tasks: AIME24, AIME25, AMC23, MATH-500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024).

Non-math adaptation tasks use the CommonsenseQA held-out set (1,221 examples) and the ProofWriter held-out set (2,048 examples) for ID evaluation. Their OOD macro averages IFEval-loose, MMLU-Pro, and SuperGPQA.

Out-of-domain tasks: GPQA (Rein et al., 2023), IFEval-loose (Zhou et al., 2023), MMLU-Pro (Wang et al., 2024b), SuperGPQA (Team et al., 2025), Safety Benchmark, and TruthfulQA (Lin et al., 2022).

Safety evaluation. Safety measures two-choice safer-response selection accuracy on the 1,000-item `ThWu/safety_benchmark` train split. Each prompt presents the original user request and two candidate replies, with `response_a` labeled A and `response_b` labeled B. The supplied `safer_response_id` determines the correct option. The model returns an option letter, and automatic answer extraction and matching produce a binary score. This evaluates recognition of safer responses among supplied candidates. Safety Avg@4 is mean selection accuracy over four responses; Safety Pass@4 is the fraction of questions with at least one correct selection.

Response scoring. Evaluation uses a shared system prompt requesting step-by-step reasoning and a final answer in `\boxed{}`. The evaluator extracts boxed answers, with a final-answer-pattern fallback, and matches them to the benchmark target. Multiple-choice tasks score the extracted option letter. IFEval-loose uses the supplied local instruction checker: a response succeeds when all instructions associated with its prompt pass after response-wrapper processing. Its score is binary prompt-level success. The supplement identifies the preparation, prompting, and scoring source files.

Within-run and across-seed aggregation. Let $c_{sij} \in \{0, 1\}$ indicate correctness of response j to question i under run s, with N questions and exactly q responses per question. The evaluator computes percentages as

$$\text{Avg@}q_s = \frac{100}{Nq} \sum_{i=1}^{N} \sum_{j=1}^{q} c_{sij}, \qquad \text{Pass@}q_s = \frac{100}{N} \sum_{i=1}^{N} \mathbf{1}\left\{ \sum_{j=1}^{q} c_{sij} > 0 \right\}. \tag{50}$$

For a score x_s from each of $S = 3$ runs, the across-seed mean and sample standard deviation are

$$\bar{x} = \frac{1}{S} \sum_{s=1}^{S} x_s, \qquad \text{SD}(x) = \sqrt{\frac{1}{S-1} \sum_{s=1}^{S} (x_s - \bar{x})^2}. \tag{51}$$

Evaluation-seed SD holds the trained checkpoint fixed and varies stochastic generation; training-seed SD aggregates independently trained runs. Tables 4 and 9 report standard error, $\mathrm{SE} = \mathrm{SD}/\sqrt{S}$. For a domain summary, each run first averages its benchmark percentages with equal benchmark weight, then the same across-seed formulas give the macro mean and SD. The recomputation scripts retain precision through aggregation and round the displayed results to two decimals. The available training-run scores in Appendix E.2 are supplied to two-decimal precision.

E.2 Individual Training-Run Results

Tables 27 and 28 report the available per-run scores underlying Appendices D.5 and D.6. *Original* denotes the run reported in the main tables; 43 and 44 identify the two additional training runs.

Table 27: Supervised-stage per-run Qwen2.5-7B results. Each cell gives Avg@k/Pass@k (%); bold marks the higher value between SFT and RPSFT within the corresponding run. Original denotes the run in the main tables.

		Original		Seed 43		Seed 44	
Domain	Benchmark	SFT	RPSFT	SFT	RPSFT	SFT	RPSFT
ID	AIME24	**15.21/50.00**	14.38/43.33	**14.17**/36.67	13.13/**46.67**	**13.13**/33.33	12.08/**36.67**
	AIME25	**19.38**/36.67	18.75/**43.33**	17.50/30.00	**17.92/40.00**	**17.71**/33.33	17.50/**36.67**
	AMC23	50.47/87.50	**55.31/90.00**	51.41/**90.00**	**52.34**/87.50	50.78/80.00	**55.31/87.50**
	MATH-500	72.85/**87.60**	**74.95**/87.40	72.65/**87.80**	**73.50**/86.60	74.55/**88.20**	**75.40**/86.80
	Minerva	31.53/**58.09**	**33.82**/56.62	**32.40/56.62**	32.22/**56.62**	33.46/55.15	**34.01/58.46**
	OlympiadBench	37.41/54.96	**38.00/56.15**	37.41/54.67	**41.19/59.70**	37.78/54.52	**41.22/59.41**
	Avg	37.81/62.47	**39.20/62.81**	37.59/59.29	**38.38/62.85**	37.90/57.42	**39.25/60.92**
OOD	GPQA	**35.66/65.63**	34.21/65.18	32.65/63.39	**35.88/68.30**	34.15/64.29	**35.77/64.96**
	IFEval-loose	54.48/69.50	**58.32/73.38**	55.64/**71.72**	**56.52**/70.61	**56.61/71.16**	56.28/70.79
	MMLU-Pro	64.64/80.00	**68.21/84.29**	66.79/**85.71**	**68.93**/84.29	67.50/84.29	**68.57/85.71**
	SuperGPQA	**28.46**/53.83	27.84/**54.57**	28.21/52.59	**29.07/54.81**	**28.58/54.81**	27.72/53.58
	Safety	70.30/**83.30**	**70.45**/79.80	69.75/**83.20**	**70.20**/82.50	70.10/**81.70**	**70.18**/79.90
	TruthfulQA	**62.43/80.26**	61.40/78.36	**61.18**/78.22	61.04/**80.12**	**62.13**/77.92	61.95/**78.80**
	Avg	52.66/72.09	**53.41/72.60**	52.37/72.47	**53.61/73.44**	53.18/**72.36**	**53.41**/72.29

Table 28: Post-DAPO per-run Qwen2.5-7B results. Each cell gives Avg@k/Pass@k (%); bold marks the higher value between SFT and RPSFT within the corresponding run. Original denotes the run in the main tables.

		Original		Seed 43		Seed 44	
Domain	Benchmark	SFT	RPSFT	SFT	RPSFT	SFT	RPSFT
ID	AIME24	18.75/46.67	**21.25/53.33**	16.67/40.00	**20.83/50.00**	**19.17/46.67**	18.13/40.00
	AIME25	20.83/33.33	**22.71/43.33**	20.63/**40.00**	**22.08**/36.67	**21.46**/33.33	21.25/**40.00**
	AMC23	58.12/90.00	**62.19/95.00**	57.97/**90.00**	**60.94/90.00**	**59.69**/90.00	59.53/**92.50**
	MATH-500	79.35/89.60	**80.50/90.20**	78.50/88.80	**80.10/89.80**	**80.05/90.60**	78.95/88.20
	Minerva	38.97/58.09	**39.94/59.19**	37.59/56.99	**39.75/60.66**	37.87/57.35	**38.74/59.19**
	OlympiadBench	43.44/59.55	**46.78/61.78**	44.15/60.59	**45.96/61.93**	44.78/60.00	**45.52/61.04**
	Avg	43.24/62.87	**45.56/67.14**	42.59/62.73	**44.94/64.84**	**43.84**/62.99	43.69/**63.49**
OOD	GPQA	37.39/64.73	**38.00/66.52**	**37.78**/63.39	37.67/**64.29**	**38.11/64.29**	37.33/**64.29**
	IFEval-loose	52.82/70.06	**54.07/70.43**	52.91/67.28	**55.59/69.87**	53.56/68.95	**54.85/70.98**
	MMLU-Pro	71.07/**84.29**	**72.86**/82.86	71.43/**87.14**	**73.21**/85.71	**72.86**/85.71	72.14/**87.14**
	SuperGPQA	32.28/56.05	**33.33/58.02**	30.37/54.32	**33.02/57.78**	30.99/53.58	**31.67/56.05**
	Safety	71.90/**80.40**	**71.93**/79.70	72.70/81.00	**74.25/82.70**	**73.23/82.30**	72.90/80.70
	TruthfulQA	65.42/78.51	**67.36/79.24**	64.62/79.39	**65.94/80.70**	**66.30/79.68**	65.53/78.80
	Avg	55.15/72.34	**56.26/72.80**	54.97/72.09	**56.61/73.51**	**55.84**/72.42	55.74/**72.99**

F Metrics for Analysis Figures

F.1 Strict SVD Subspace Energy

Figure 2 measures the fraction of empirical-Fisher gradient energy captured by the strict top singular block. Let $\mathbf{G}_t \in \mathbb{R}^{m\times n}$ denote the gradient matrix of the t-th sample, where $t = 1, \ldots, N$ indexes

the N sampled gradients used in the diagnostic, and let

$$g_t = \mathrm{vec}(\mathbf{G}_t)$$

be its vectorized form. Define the unnormalized empirical-Fisher gradient second moment as

$$\widehat{\mathbf{F}}_{\mathrm{emp}} = \sum_{t=1}^{N} g_t g_t^\top.$$

Let $u_i \in \mathbb{R}^m$ and $v_j \in \mathbb{R}^n$ be the i-th left and j-th right singular vectors of the pretrained weight matrix, let r be the protected singular rank, and let $\mathbf{P}_{\mathrm{svd},r}$ denote the orthogonal projector onto the strict top-$r \times r$ singular-vector product subspace

$$\mathrm{span}\big\{\mathrm{vec}(u_i v_j^\top) : 1 \le i, j \le r\big\}.$$

Then the fraction of gradient energy captured by the strict top-$r \times r$ singular block is

$$\frac{\mathrm{tr}(\mathbf{P}_{\mathrm{svd},r}\widehat{\mathbf{F}}_{\mathrm{emp}})}{\mathrm{tr}(\widehat{\mathbf{F}}_{\mathrm{emp}})} = \frac{\sum_{t=1}^{N} \|\mathbf{P}_{\mathrm{svd},r} g_t\|_2^2}{\sum_{t=1}^{N} \|g_t\|_2^2}.$$

Equivalently, in matrix form, this can be written as

$$\frac{\mathrm{tr}(\mathbf{P}_{\mathrm{svd},r}\widehat{\mathbf{F}}_{\mathrm{emp}})}{\mathrm{tr}(\widehat{\mathbf{F}}_{\mathrm{emp}})} = \frac{\sum_{t=1}^{N} \big\|\mathbf{U}_r^\top \mathbf{G}_t \mathbf{V}_r\big\|_F^2}{\sum_{t=1}^{N} \|\mathbf{G}_t\|_F^2},$$

where $\mathbf{U}_r = [u_1, \ldots, u_r]$ and $\mathbf{V}_r = [v_1, \ldots, v_r]$.

Here, the denominator $\mathrm{tr}(\widehat{\mathbf{F}}_{\mathrm{emp}}) = \sum_{t=1}^{N} \|g_t\|_2^2$ is the total gradient energy across all samples, meaning the total sum of squared gradient norms, while the numerator $\mathrm{tr}(\mathbf{P}_{\mathrm{svd},r}\widehat{\mathbf{F}}_{\mathrm{emp}})$ is the portion of that energy lying in the strict top-$r \times r$ singular block. In the figure, the x-axis reports $x(r) = r^2/R^2$, where $R = \min(m, n)$ is the economy-SVD dimension, and the y-axis reports $\mathrm{tr}(\mathbf{P}_{\mathrm{svd},r}\widehat{\mathbf{F}}_{\mathrm{emp}})/\mathrm{tr}(\widehat{\mathbf{F}}_{\mathrm{emp}})$. We apply this diagnostic to the layer-1 attention `q_proj` weight for Llama-8B, Qwen-7B, and Qwen-3B.

For square matrices, r^2/R^2 is also the fraction of ambient matrix coordinates. For rectangular matrices, it is the fraction of the maximal SVD core; the ambient-coordinate fraction is $r^2/(mn)$. Dividing $\widehat{\mathbf{F}}_{\mathrm{emp}}$ by N gives the average empirical Fisher and leaves every reported energy ratio unchanged.

F.2 Layerwise First-Order Signal

For a checkpoint weight $\mathbf{W}_{\mathrm{ckpt}}$ and base-model weight $\mathbf{W}_{\mathrm{base}}$, define

$$\Delta\mathbf{W} = \mathbf{W}_{\mathrm{ckpt}} - \mathbf{W}_{\mathrm{base}}. \tag{52}$$

Let w_{ref} denote the parameter checkpoint at which the diagnostic gradient is evaluated. For a batch B from an ID or OOD dataset, define

$$g_{\mathbf{W}}(w_{\mathrm{ref}}) = \frac{1}{|B|} \sum_{(x,y)\in B} \nabla_{\mathbf{W}} \ell(\pi_w; x, y)\Big|_{w=w_{\mathrm{ref}}}. \tag{53}$$

For the set $\mathcal{P}_\ell$ of matrices assigned to layer ℓ, the plotted mean is

$$m_\ell = \frac{1}{|\mathcal{P}_\ell|} \sum_{\mathbf{W}\in\mathcal{P}_\ell} \langle g_{\mathbf{W}}(w_{\mathrm{ref}}), \Delta\mathbf{W}\rangle. \tag{54}$$

This is the signed local loss response, averaged over the selected matrices, to movement along the checkpoint-displacement direction at w_{ref}. Negative values indicate local descent and positive values indicate local interference. When $w_{\mathrm{ref}} = w_0$, the unaveraged sum over all parameter blocks equals the first-order term in Eq. (2). For the plotted matrix sets, the corresponding sum is $\sum_\ell |\mathcal{P}_\ell| m_\ell$.

F.3 ROTATION METRICS

For each layer ℓ and weight type

$$t \in \{\texttt{q_proj}, \texttt{k_proj}, \texttt{v_proj}, \texttt{o_proj}, \texttt{up_proj}, \texttt{down_proj}, \texttt{gate_proj}\},$$

let the base-model weight and tuned-model weight be

$$\mathbf{W}_{\text{base}}^{(\ell,t)}, \qquad \mathbf{W}_{m}^{(\ell,t)}.$$

Here m indexes the tuned method or checkpoint being compared. Let $d_{\text{out}}^{(\ell,t)}$ and $d_{\text{in}}^{(\ell,t)}$ denote the output and input dimensions of this matrix. We compute truncated SVD with

$$K_{\ell,t} = \min\bigl(512, d_{\text{out}}^{(\ell,t)}, d_{\text{in}}^{(\ell,t)}\bigr),$$

and write the corresponding rank-$K_{\ell,t}$ approximations as

$$\mathbf{W}_{\text{base},K}^{(\ell,t)} = \mathbf{U}_{\text{base}}^{(\ell,t)} \mathbf{\Sigma}_{\text{base}}^{(\ell,t)} {\mathbf{V}_{\text{base}}^{(\ell,t)}}^{\top}, \qquad \mathbf{W}_{m,K}^{(\ell,t)} = \mathbf{U}_{m}^{(\ell,t)} \mathbf{\Sigma}_{m}^{(\ell,t)} {\mathbf{V}_{m}^{(\ell,t)}}^{\top}.$$

where

$$\mathbf{U}_{\text{base}}^{(\ell,t)}, \mathbf{U}_{m}^{(\ell,t)} \in \mathbb{R}^{d_{\text{out}}^{(\ell,t)} \times K_{\ell,t}}, \qquad \mathbf{V}_{\text{base}}^{(\ell,t)}, \mathbf{V}_{m}^{(\ell,t)} \in \mathbb{R}^{d_{\text{in}}^{(\ell,t)} \times K_{\ell,t}}.$$

To measure U-space rotation, we compare the left-singular subspaces through

$$\mathbf{M}_{U}^{(\ell,t,m)} = \Bigl(\mathbf{U}_{\text{base}}^{(\ell,t)}\Bigr)^{\top} \mathbf{U}_{m}^{(\ell,t)}.$$

If the singular values of $\mathbf{M}_{U}^{(\ell,t,m)}$ are $\sigma_1^{(\ell,t,m)}, \ldots, \sigma_{K_{\ell,t}}^{(\ell,t,m)}$, the principal angles are

$$\theta_i^{(\ell,t,m)} = \arccos\Bigl(\sigma_i^{(\ell,t,m)}\Bigr).$$

The mean U-space rotation for layer/type pair (ℓ, t) is

$$r_{U}^{(\ell,t,m)} = \frac{180}{\pi} \cdot \frac{1}{K_{\ell,t}} \sum_{i=1}^{K_{\ell,t}} \theta_i^{(\ell,t,m)}.$$

The layerwise plot in Figure 6 uses `types-all`, so the value at layer ℓ averages these per-type rotations over the available matrix types:

$$y_m(\ell) = \frac{\sum_{t \in \mathcal{T}_\ell} c^{(\ell,t,m)}\, r_{U}^{(\ell,t,m)}}{\sum_{t \in \mathcal{T}_\ell} c^{(\ell,t,m)}},$$

where $\mathcal{T}_\ell$ is the set of available types in layer ℓ. In the current plotting script, $c^{(\ell,t,m)} = 1$ for each available entry, so this reduces to the simple mean across types:

$$y_m(\ell) = \frac{1}{|\mathcal{T}_\ell|} \sum_{t \in \mathcal{T}_\ell} \left(\frac{180}{\pi} \cdot \frac{1}{K_{\ell,t}} \sum_{i=1}^{K_{\ell,t}} \arccos\bigl(\sigma_i^{(\ell,t,m)}\bigr) \right).$$

Thus, each curve reports how much the top left-singular subspaces rotate away from the base model across layers after averaging over all selected matrix types.

For the rank-wise case-study plot in Figure 7, we fix one matrix case for each model and vary the prefix rank $1 \le r \le K_{\ell,t}$. At each r, we replace the full truncated bases by their first r columns, compute the singular values of

$$\mathbf{M}_{U}^{(r,m)} = \Bigl(\mathbf{U}_{\text{base}}^{(r)}\Bigr)^{\top} \mathbf{U}_{m}^{(r)},$$

where $\mathbf{U}_{\text{base}}^{(r)}$ and $\mathbf{U}_{m}^{(r)}$ are the first r columns of the truncated left-singular bases for the base and tuned matrices, and report the mean principal angle

$$y_m(r) = \frac{180}{\pi} \cdot \frac{1}{r} \sum_{i=1}^{r} \arccos\bigl(\sigma_i^{(r,m)}\bigr).$$

This keeps the same rotation definition and only changes the swept dimension: the layerwise figure varies ℓ after averaging over types, while the rankwise figure varies the prefix rank within one fixed case-study matrix.

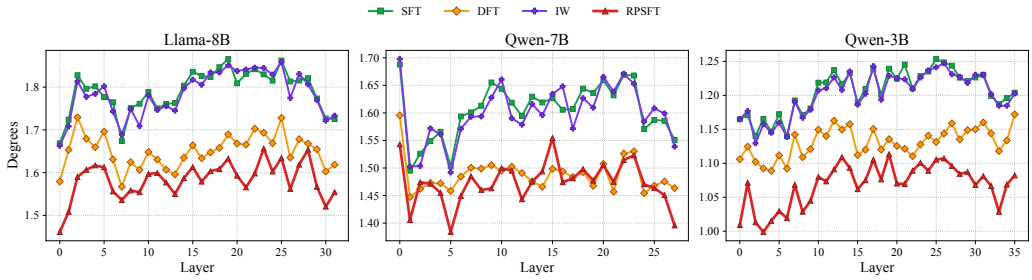

Figure 6: Layerwise U-space rotation across three models. Each panel averages principal-angle rotation over `q_proj`, `k_proj`, `v_proj`, `o_proj`, `up_proj`, `down_proj`, and `gate_proj`. RPSFT remains closest to the pretrained dominant subspaces across most layers.

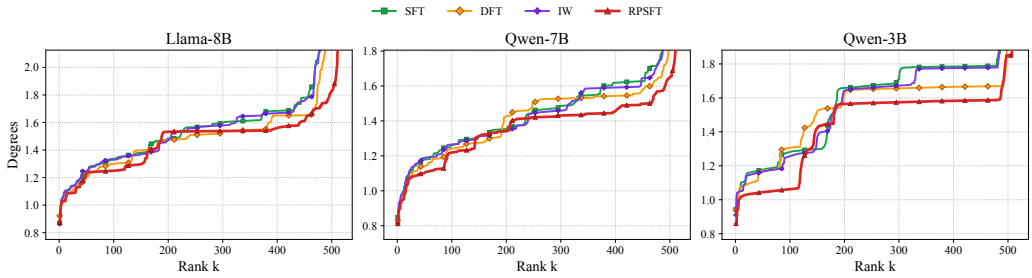

Figure 7: Rankwise U-space rotation for one representative weight matrix in each model. The x-axis shows positive prefix ranks up to 512, and the y-axis reports the mean principal-angle rotation of the corresponding left-singular subspace. RPSFT reduces rotation across the leading-to-middle rank range.

F.4 Entropy Metric

For prompt j, method m generates tokens

$$y^{(m,j)} = \left(y_1^{(m,j)}, \ldots, y_{T_{m,j}}^{(m,j)}\right),$$

with greedy decoding and $T_{m,j} \le 128$. At generation step t, let

$$p_t^{(m,j)}(v) = \mathrm{softmax}(z_t^{(m,j)})_v.$$

The token entropy is

$$h_t^{(m,j)} = -\sum_{v \in \mathcal{V}} p_t^{(m,j)}(v) \log p_t^{(m,j)}(v),$$

and the sample-level average entropy is

$$e_{m,j} = \frac{1}{T_{m,j}} \sum_{t=1}^{T_{m,j}} h_t^{(m,j)}.$$

Each figure plots a Gaussian KDE over $\{e_{m,j}\}_{j=1}^{N_m}$:

$$\hat{f}_m(x) = \frac{1}{N_m h_m} \sum_{j=1}^{N_m} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x - e_{m,j}}{h_m}\right)^2\right),$$

using the default bandwidth

$$h_m = 1.06\, s_m\, N_m^{-1/5},$$

where s_m is the sample standard deviation. We apply the same entropy definition to every ID and OOD benchmark panel.

F.5 Auxiliary Entropy Plots Across Model Families

The entropy plots characterize model- and benchmark-dependent differences in predictive uncertainty. They complement the principal-angle and hidden-state centroid analyses.

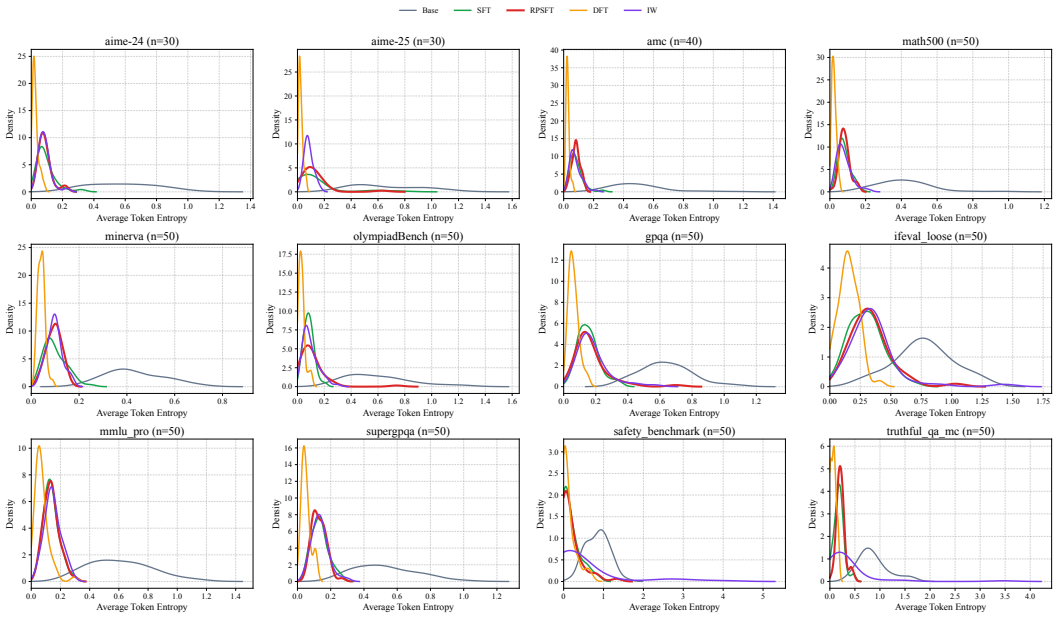

Figure 8: Average-token-entropy distributions across the twelve in-domain and out-of-domain benchmarks for Llama-3.1-8B-Instruct. The differences are modest and benchmark-dependent: DFT more often shifts toward lower entropy, while RPSFT generally stays in a similar range to the stronger tuned baselines.

F.6 Base-Model Drift in Hidden Representations

For dataset d with N_d prompts, let $h_i^{(m,d)} \in \mathbb{R}^p$ be the pooled hidden representation for prompt i under model $m \in \{\text{base}, \text{sft}, \text{RPSFT}, \text{dft}, \text{iw}\}$. The hidden-space centroid of model m on dataset d is

$$\mu^{(m,d)} = \frac{1}{N_d} \sum_{i=1}^{N_d} h_i^{(m,d)}.$$

We measure drift from the base model by the centroid distance

$$D_{\text{hidden}}^{(m,d)} = \left\| \mu^{(m,d)} - \mu^{(\text{base},d)} \right\|_2.$$

For visualization, we stack all prompt representations from the five models,

$$X^{(d)} = \begin{bmatrix} H^{(\text{base},d)} \\ H^{(\text{sft},d)} \\ H^{(\text{RPSFT},d)} \\ H^{(\text{dft},d)} \\ H^{(\text{iw},d)} \end{bmatrix} \in \mathbb{R}^{M_d \times p}, \qquad M_d = 5N_d,$$

center the matrix as $\tilde{X}^{(d)} = X^{(d)} - \mathbf{1}\,\bar{x}^{(d)\top}$, and project onto the first two PCA directions:

$$Z^{(d)} = \tilde{X}^{(d)} W_{1:2} \in \mathbb{R}^{M_d \times 2}.$$

If $z_i^{(m,d)} \in \mathbb{R}^2$ is the PCA coordinate of prompt i, the PCA-plane centroid is

$$c^{(m,d)} = \frac{1}{N_d} \sum_{i=1}^{N_d} z_i^{(m,d)},$$

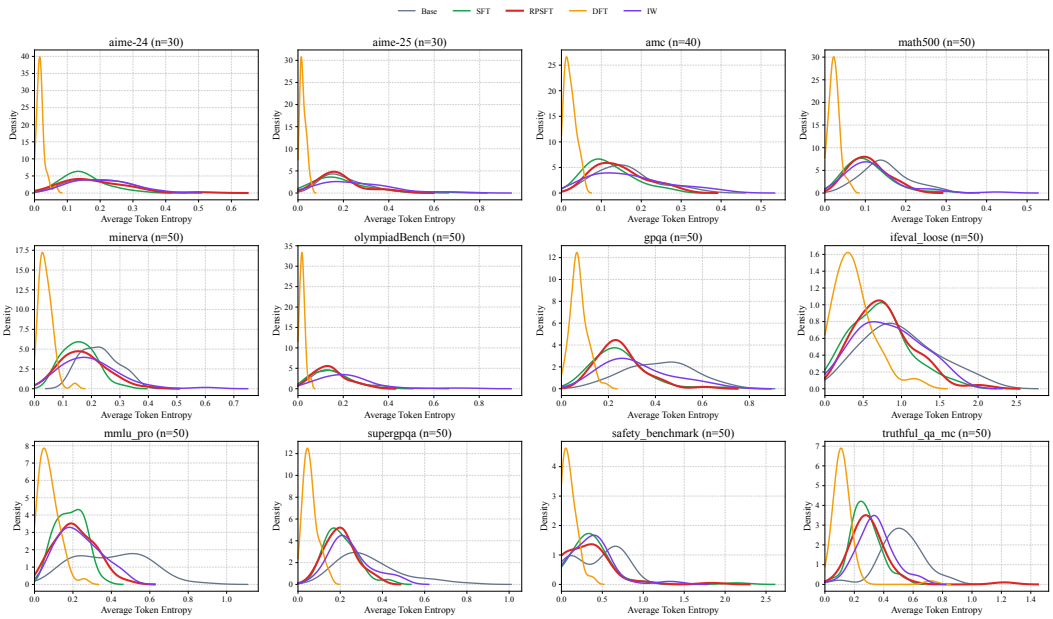

Figure 9: Average-token-entropy distributions across the same twelve benchmarks for Qwen2.5-7B-Instruct. RPSFT often remains above DFT and within the broader range of the tuned baselines; separation varies across benchmark panels.

and the plotted centroid shift is

$$D_{\mathrm{PCA}}^{(m,d)} = \left\| c^{(m,d)} - c^{(\mathrm{base},d)} \right\|_2 .$$

In the plots, each point is one prompt and each arrow starts at the base centroid $c^{(\mathrm{base},d)}$ and ends at the centroid of a tuned model. We use $D_{\mathrm{hidden}}^{(m,d)}$ for the quantitative drift measurement and $D_{\mathrm{PCA}}^{(m,d)}$ only for the 2D visualization.

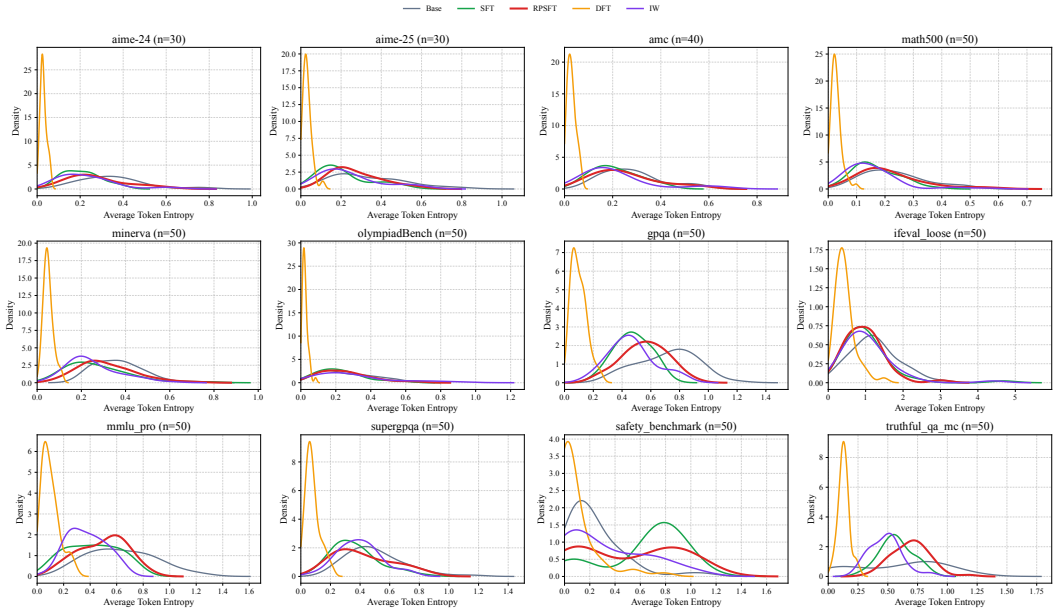

Figure 10: Average-token-entropy distributions across the same twelve benchmarks for Qwen2.5-3B-Instruct. RPSFT often shifts toward a broader entropy profile than standard SFT or DFT; the base model remains the highest-entropy reference and separation varies by benchmark.

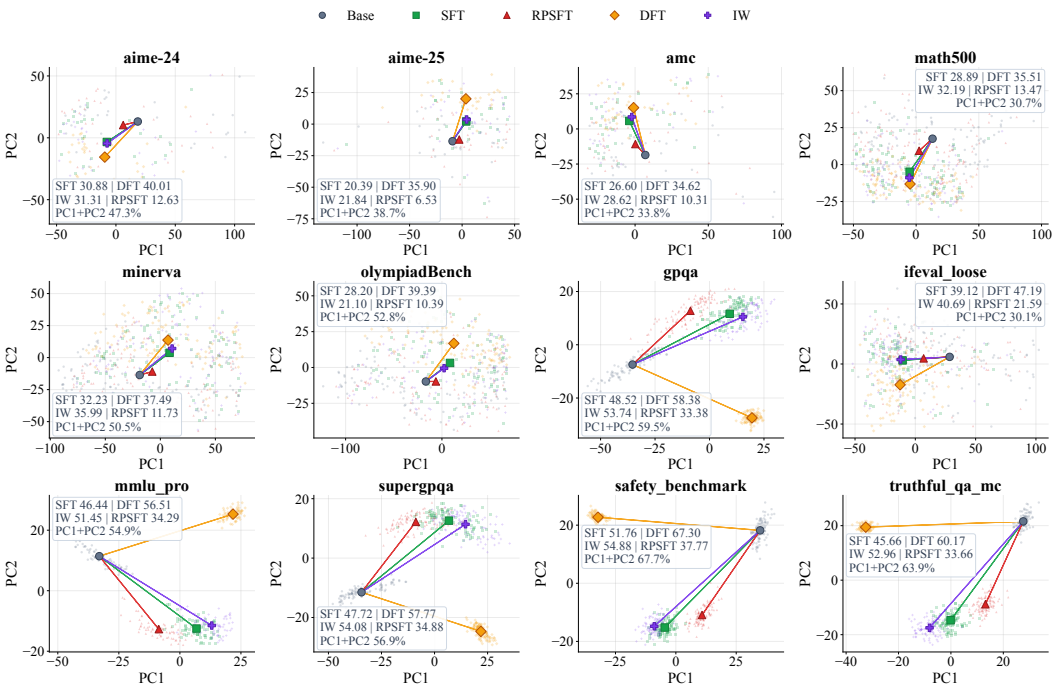

Figure 11: Hidden-state drift on Qwen2.5-3B-Instruct. Each panel shows centroid shifts away from the base model in a 2D PCA view. RPSFT stays among the closest tuned centroids to the base model, while SFT, IW, and especially DFT often shift farther.

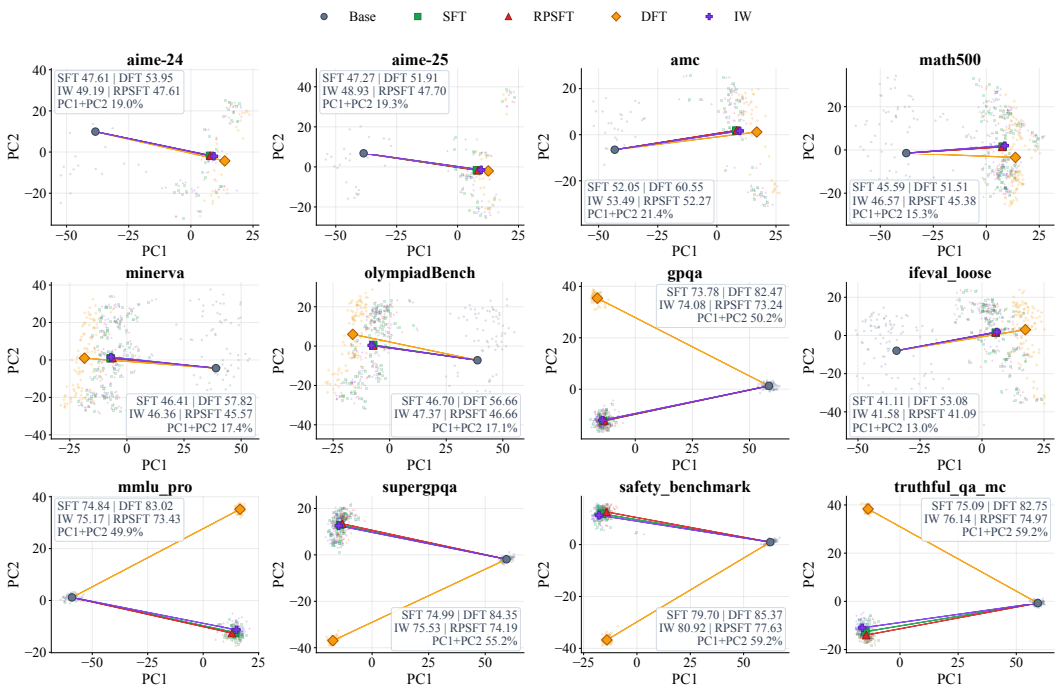

Figure 12: Hidden-state drift on Llama-3.1-8B-Instruct. Across the benchmark panels, RPSFT remains close to the base-model centroid and is competitive with the strongest tuned alternatives, which is consistent with reduced representation drift.

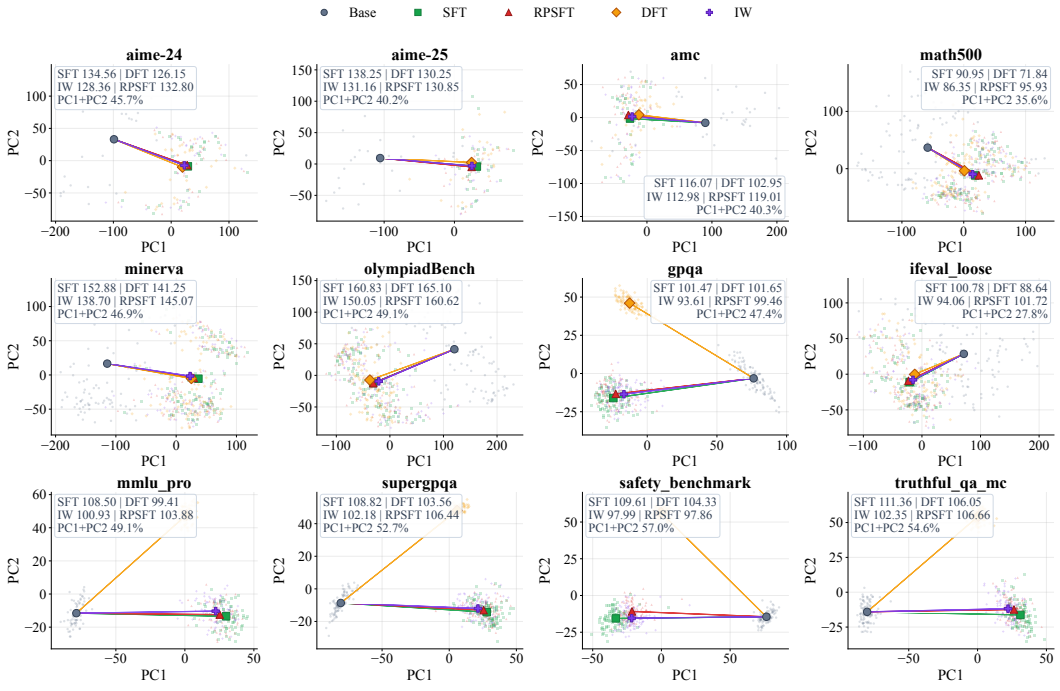

Figure 13: Hidden-state drift on Qwen2.5-7B-Instruct. The PCA view compares centroid shifts away from the base model across benchmark panels. RPSFT is usually among the closest tuned centroids and remains closer than standard SFT in most panels, which is consistent with reduced representation drift after supervised fine-tuning.