GEOMETRIC CALIBRATION OF SUPERVISED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies investigate the weight geometry through its spectral structure, which characterizes how the weights transform different input directions. Within this view, supervised fine-tuning (SFT) is shown to induce dense updates that perturb the spectral geometry of pre-SFT weights along their principal directions, potentially disrupting the model's prior knowledge. These observations motivate using the geometry of pre-SFT weights as a reference for refining learned SFT updates. We study this perspective as *Geometric Calibration* and develop SPACE (**S**ubspace **P**rojection for **A**dapted **C**heckpoint **E**nhancement), a lightweight post-hoc method that selectively reverts SFT update components within the principal subspace of the pre-SFT weights while preserving those in its orthogonal complement. SPACE is a data-free, training-free method that requires only the pre- and post-SFT checkpoints. We evaluate SPACE on Qwen2.5, Qwen3, Qwen3.5, Llama-3.1, and Olmo-3 models ranging from 2B to 14B, across mathematical, coding, and medical SFT, as well as several SFT variants. Overall, SPACE mitigates out-of-domain (OOD) capability regression while further improving in-domain (ID) performance over the uncalibrated SFT checkpoints. For math SFT, SPACE yields average ID/OOD gains of 2.85/2.26 points on Qwen2.5-7B, and 2.41/1.52 points on Qwen3-14B-Base, respectively. These results highlight how the geometric relationship between SFT updates and pre-SFT weights shapes the adapted model's capabilities and, more broadly, motivate geometry-aware parameter updates for continual model improvement.

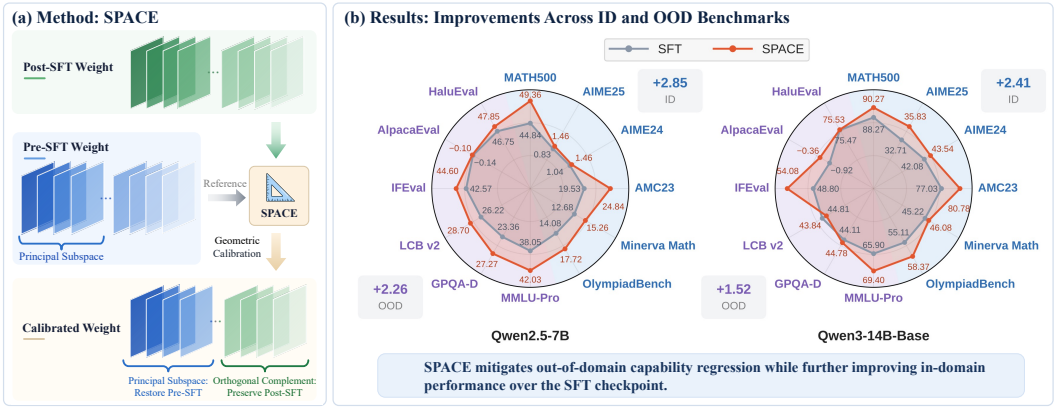


Figure 1: **Overview of SPACE and results. (a) Method.** SPACE uses the principal input subspace of the pre-SFT weight as a geometric reference for calibrating the post-SFT weights. The pre-SFT weight is visualized as rank-one SVD components ordered by singular value, while the post-SFT weight is decomposed along the same pre-SFT input directions. Blue and green denote pre- and post-SFT components, respectively, with color intensity reflecting the corresponding pre-SFT singular value. SPACE restores the pre-SFT transformation along the selected principal directions while retaining the post-SFT transformation along the remaining directions. **(b) Results.** After regular SFT on mathematical data, SPACE improves both in-domain (ID) and out-of-domain (OOD) performance over the uncalibrated SFT checkpoints.

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.

1 INTRODUCTION

Supervised fine-tuning (SFT) is a standard approach for adapting large language models (LLMs) to target tasks, particularly for improving reasoning capabilities (Ho et al., 2023; Ye et al., 2025; Lai et al., 2025). However, gains on the target task can be accompanied by regression in capabilities beyond the training domain (Chu et al., 2025; Huan et al., 2026; Lai et al., 2026; Shenfeld et al., 2026), a phenomenon commonly studied as forgetting. To gain deeper insight into this problem, we believe it is important to understand how SFT changes the geometry of the model weights.

Recent work studies the weight geometry of LLMs through the spectral structure, which captures how different input directions are transformed (Zhu et al., 2025; Shen et al., 2026; Zhang et al., 2026). Under this view, SFT tends to induce dense updates that perturb this geometry along principal directions. These principal directions typically correspond to singular vectors associated with large singular values, which together span a principal subspace. This subspace has been associated with important pre-existing or transferable capabilities, and perturbations within it can contribute to forgetting (Luo et al., 2026; Wang et al., 2025a; Liu et al., 2026; Jin et al., 2026).

These insights lead to a direct question: *can an SFT checkpoint be improved by removing the learned updates within these principal subspaces?* More broadly, this motivates using pre-SFT weight geometry as a reference for refining completed SFT updates. We formalize this perspective as *Geometric Calibration* and instantiate it with SPACE (**S**ubspace **P**rojection for **A**dapted **C**heckpoint **E**nhancement), a lightweight post-hoc method for refining SFT checkpoints (Figure 1(a)). For each pre-SFT weight matrix, SPACE defines a principal input subspace using its leading right singular vectors. It then projects the SFT update onto the orthogonal complement of this subspace, restoring the pre-SFT transformation along the principal directions while retaining the learned update along the remaining directions. Unlike existing post-hoc methods that determine refinement through parameter interpolation (Wortsman et al., 2022b; Wang et al., 2025b) or properties of the learned update itself (Abdollahpoorrostam et al., 2026; Abro & Tahir, 2026), SPACE explicitly conditions the refinement on the geometric relationship between the SFT update and the pre-SFT weights. Moreover, SPACE operates directly on the pre- and post-SFT checkpoints in a data- and training-free manner.

We evaluate SPACE across five model families—Qwen2.5, Qwen3, Qwen3.5, Llama-3.1, and Olmo-3—ranging from 2B to 14B parameters, and across mathematical, coding, and medical SFT. Across these settings, SPACE consistently yields positive average gains over the corresponding SFT checkpoints on both ID and OOD evaluations. For the two mathematical SFT settings in Figure 1(b), SPACE improves average ID/OOD performance by 2.85/2.26 points on Qwen2.5-7B and 2.41/1.52 points on Qwen3-14B-Base. The improvements further extend across different SFT variants, supporting the broad applicability of SPACE beyond standard SFT. It also outperforms a range of representative post-hoc model refinement baselines, highlighting the benefit of accounting for the relationship between SFT updates and pre-SFT geometry. Overall, these results support the use of pre-SFT weight geometry as a reference for refining completed SFT updates and motivate the development of more targeted and effective geometric calibration methods for SFT.

2 RELATED WORK

Generalization and forgetting in SFT. Recent work has increasingly focused on understanding the generalization and forgetting behaviors of LLMs during SFT (Wu et al., 2026b; Huan et al., 2026; Ren et al., 2026). Comparisons between SFT and reinforcement learning reveal distinct generalization and forgetting patterns (Chu et al., 2025; Shenfeld et al., 2026). To mitigate capability degradation, existing efforts mainly improve SFT through changes to the training data or objectives. One line of work improves the selection and use of training data (Ye et al., 2025; Kopiczko et al., 2026). Another modifies the supervised objective through token reweighting or regularization (Wu et al., 2026b; Li et al., 2025; Zhu et al., 2026; Jin et al., 2026). These methods primarily intervene during SFT to control how the updates are learned. In contrast, we study whether a completed SFT update can be refined post hoc without data or additional training.

Weight geometry in model adaptation. Recent studies analyze the geometry of LLM weights during post-training through their spectral structure (Zhu et al., 2025; Shen et al., 2026; Zhang et al., 2026). These studies reveal that SFT typically produces dense updates along principal directions,

which can distort the pretrained spectral structure. In contrast, reinforcement learning tends to induce sparse and off-principal updates, resulting in less disruption to the spectral structure (Liu et al., 2025b; Mukherjee et al., 2026). The role of principal directions has also been emphasized in studies of parameter-efficient fine-tuning (PEFT) and continual model merging. In these lines of work, principal subspaces of pretrained weights are often associated with prior or transferable capabilities. Accordingly, many PEFT methods organize adaptation around principal or complementary singular subspaces (Meng et al., 2024; Wang et al., 2025a; Luo et al., 2026; Xiong & Xie, 2026), while continual model merging methods leverage structured subspaces to reduce interference with previously acquired capabilities (Ilharco et al., 2023; Yu et al., 2024; Sun et al., 2026; Li et al., 2026; Shi et al., 2026; Qiu et al., 2026). Together, these studies suggest that the geometry of pretrained weights provides a useful reference for controlling how new updates interact with existing capabilities. Our work builds on this perspective but uses the principal subspace of the pre-SFT weights to refine a single completed SFT update post hoc.

Post-hoc model refinement. Our work is closely related to post-hoc model merging/editing/refinement methods that improve a trained model by directly modifying its parameters. Existing approaches refine completed checkpoints through weight interpolation (Wortsman et al., 2022b; Wang et al., 2025b), update pruning (Yu et al., 2024; Huang et al., 2025), model souping (Wortsman et al., 2022a; Abdollahpoorrostam et al., 2026), or spectral filtering (Abro & Tahir, 2026). Their refinement strategies are typically determined by parameter interpolation or by properties of the learned update or the fine-tuned weights themselves. Although some methods use the pre-SFT checkpoint, they generally do not explicitly consider its geometric relationship with the learned update when determining which update components should be modified. Motivated by the importance of pre-SFT weight geometry discussed above, we instead study this relationship directly and use it to refine completed SFT checkpoints post hoc.

3 Problem Formulation

Supervised fine-tuning. SFT adapts an LLM to target tasks by learning from labeled training data. Prior work has shown that SFT can induce non-trivial parameter updates (Yuan et al., 2025; Mukherjee et al., 2026). While such updates can improve performance on the target task, they may also lead to capability regression beyond the training domain (Wu et al., 2026a; Shenfeld et al., 2026; Huan et al., 2026). These observations suggest that some components of the learned SFT updates may be redundant or detrimental to the adapted model. We therefore investigate whether identifying and calibrating such components can mitigate capability regression while preserving or further improving target-domain performance.

Geometric calibration. Given a pre-SFT checkpoint θ_0 and its corresponding post-SFT checkpoint θ_{SFT}, the SFT-induced parameter update is defined as

$$\Delta\theta_{\mathrm{SFT}} = \theta_{\mathrm{SFT}} - \theta_0. \tag{1}$$

Rather than treating the post-SFT checkpoint as the endpoint of adaptation, we explore further refinement of the learned update to form the final adapted model. We approach this refinement through *Geometric Calibration*, using the geometric structure of the pre-SFT weights as a reference.

Formally, we define a calibration operator $\mathcal{C}$ that maps the learned SFT update to a calibrated update conditioned on the pre-SFT parameters:

$$\Delta\theta_{\mathrm{cal}} = \mathcal{C}\left(\Delta\theta_{\mathrm{SFT}}; \theta_0\right), \tag{2}$$

where $\Delta\theta_{\mathrm{SFT}}$ is the update being calibrated and θ_0 serves as the geometric reference. We seek a calibrated checkpoint $\theta_{\mathrm{cal}} = \theta_0 + \Delta\theta_{\mathrm{cal}}$ with improved overall capabilities.

4 Method

We develop SPACE as a post-hoc implementation for Geometric Calibration. For a pre-SFT weight $W_0 \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$ and its post-SFT counterpart W_{SFT}, let $\Delta W = W_{\mathrm{SFT}} - W_0$. The central idea is to refine the completed update ΔW using the principal subspace of W_0 as a geometric reference.

4.1 MOTIVATION

Principal directions and capability preservation. A common view in existing work is to characterize the geometry of model weights through their singular directions, where directions associated with large singular values are referred to as principal directions and their span defines the principal subspace (Zhu et al., 2025; Wang et al., 2025a; Shen et al., 2026). Prior work suggests that this principal subspace is closely associated with important prior or broadly transferable knowledge, and that limiting changes to these directions can help preserve existing capabilities during adaptation (Wang et al., 2025a; Luo et al., 2026; Sun et al., 2026; Jin et al., 2026). At the same time, SFT tends to induce dense updates that perturb the pre-SFT weight geometry within the principal subspace (Zhu et al., 2025; Shen et al., 2026). Together, these findings motivate using the principal subspace of the pre-SFT weights as a reference for refining the learned SFT update. In particular, a natural calibration objective is to preserve the pre-SFT transformation within this subspace while retaining the post-SFT transformation along the remaining directions.

Preserving the pre-SFT transformation. For a layer input representation $h \in \mathbb{R}^{d_{\text{in}}}$, we seek a calibrated transformation W_{cal} that follows the pre-SFT transformation on the component associated with the principal subspace of W_0, while retaining the post-SFT transformation on the remaining component. Let P_0 be a rank-k orthogonal projector onto the principal subspace, where k denotes its dimension. The desired behavior is

$$W_{\text{cal}}h = W_0P_0h + W_{\text{SFT}}(I - P_0)h. \tag{3}$$

To determine P_0, we seek a projection that retains as much of the pre-SFT transformation as possible. Specifically, among all rank-k orthogonal projectors, we choose P_0 such that W_0P_0 best approximates W_0 in Frobenius norm:

$$P_0 \in \underset{\substack{P=P^\top=P^2 \\ \text{rank}(P)=k}}{\arg\min} \|W_0 - W_0P\|_F^2 . \tag{4}$$

Since P is an orthogonal projector, minimizing this error is equivalent to maximizing $\text{tr}(PW_0^\top W_0)$. Let $W_0 = U_0\Sigma_0V_0^\top$ be the SVD of W_0. The Ky Fan maximum principle (Fan, 1949) gives a solution

$$P_0 = V_{0,k}V_{0,k}^\top, \tag{5}$$

where $V_{0,k}$ contains the leading k right singular vectors of W_0. A complete derivation from equation 4 to equation 5 is provided in Appendix A. With this choice, $W_0P_0 = U_{0,k}\Sigma_{0,k}V_{0,k}^\top$ is a best rank-k approximation of W_0. Thus, P_0h selects the component of the input representation associated with this principal part of the pre-SFT transformation. We refer to $\text{span}(V_{0,k})$ as the *principal input subspace*, and use P_0 for calibration.

4.2 SPACE: SUBSPACE PROJECTION FOR ADAPTED CHECKPOINT ENHANCEMENT

Subspace Projection. Requiring equation 3 to hold for any layer input representation h gives $W_{\text{cal}} = W_0P_0 + W_{\text{SFT}}(I - P_0)$. Using $W_{\text{SFT}} = W_0 + \Delta W$, this can be equivalently written as

$$W_{\text{cal}} = W_0 + \Delta W(I - P_0). \tag{6}$$

Applying this calibration to the selected weight matrices yields the model-level calibrated update $\Delta\theta_{\text{cal}}$ defined in Section 3. SPACE therefore removes the SFT update within the principal input subspace while preserving it on the orthogonal complement.

Adaptive subspace dimension. Since weight matrices in LLMs can differ substantially in both their dimensions and spectral concentration, using a fixed k across all matrices may capture very different fractions of their singular-value spectra. We therefore choose the subspace dimension k adaptively according to the singular-value spectrum of each weight matrix. Specifically, let $\sigma_{0,1} \geq \cdots \geq \sigma_{0,r} \geq 0$ denote the singular values of W_0. We select the smallest k such that the leading singular values account for at least a fraction ρ of the total singular-value mass:

$$k = \min\left\{ m : \frac{\sum_{i=1}^{m} \sigma_{0,i}}{\sum_{i=1}^{r} \sigma_{0,i}} \geq \rho \right\}. \tag{7}$$

Table 1: Main experimental results on mathematical reasoning SFT. We compare ID and OOD performance across different post-hoc methods. "Avg. (Δ vs. SFT)" denotes the average score change relative to the SFT checkpoint. **Bold** denotes the best score excluding the Base model.

Model	Benchmark	Base	SFT	WiSE-FT	TIES	LiNeS	DARE	POME	FAPM	DG-Hard	MonoSoup	SPACE
Qwen2.5-7B	*In-domain (ID)*											
	MATH500	28.79	44.84	45.14	45.70	43.89	44.13	46.23	46.83	47.50	46.43	**49.36**
	AIME25	0.42	0.83	0.83	0.42	0.63	0.42	0.83	**1.46**	0.63	1.04	**1.46**
	AIME24	1.65	1.04	1.04	0.63	0.83	0.83	1.04	**1.67**	1.46	1.46	1.46
	OlympiadBench	7.69	14.08	17.06	14.62	14.34	13.82	15.75	17.44	15.39	15.06	**17.72**
	Minerva Math	9.26	12.68	11.60	12.82	10.62	11.53	14.27	11.79	13.26	12.68	**15.26**
	AMC23	15.65	19.53	22.03	19.38	20.00	18.28	20.94	**24.84**	22.97	20.00	**24.84**
	Avg. (Δ vs. SFT)	-	-	+0.78	+0.10	-0.45	-0.67	+1.01	+1.84	+1.37	+0.61	**+2.85**
	Out-of-domain (OOD)											
	MMLU-Pro	40.33	38.05	40.85	38.12	41.90	38.11	39.41	38.36	35.16	39.99	**42.03**
	GPQA-D	26.77	23.36	26.77	23.96	26.42	22.73	24.37	25.95	24.34	25.44	**27.27**
	LCB v2	28.96	26.22	**29.03**	26.74	28.83	26.48	27.92	26.61	28.64	28.25	28.70
	IFEval	42.57	42.57	45.56	42.57	43.88	41.25	42.81	**45.68**	41.85	45.44	44.60
	AlpacaEval	-1.13	-0.14	-0.20	-0.13	-0.12	-0.11	-0.15	-0.16	-0.20	-0.16	**-0.10**
	HaluEval	46.86	46.75	47.80	46.71	47.67	46.56	46.36	46.49	47.15	47.23	**47.85**
	Avg. (Δ vs. SFT)	-	-	+2.17	+0.19	+1.96	-0.30	+0.65	+1.02	+0.02	+1.56	**+2.26**
Qwen3-14B-Base	*In-domain (ID)*											
	MATH500	55.07	88.27	88.60	88.60	89.47	88.20	89.40	88.20	88.93	89.20	**90.27**
	AIME25	5.21	32.71	29.58	36.25	33.96	34.79	34.58	31.25	33.33	**36.67**	35.83
	AIME24	6.25	42.08	38.54	41.46	**45.21**	42.92	44.38	35.83	42.92	41.88	43.54
	OlympiadBench	27.31	55.11	55.85	56.74	57.33	55.36	57.14	55.56	54.72	56.25	**58.37**
	Minerva Math	22.92	45.22	42.40	45.71	43.87	45.71	44.24	42.52	44.98	44.49	**46.08**
	AMC23	38.59	77.03	78.59	79.84	**80.78**	80.31	79.38	78.28	80.00	80.16	**80.78**
	Avg. (Δ vs. SFT)	-	-	-1.14	+1.36	+1.70	+1.15	+1.45	-1.46	+0.74	+1.37	**+2.41**
	Out-of-domain (OOD)											
	MMLU-Pro	61.80	65.90	66.30	69.20	66.20	67.00	68.10	67.00	**69.40**	67.80	**69.40**
	GPQA-D	44.61	44.11	45.29	**45.62**	44.28	42.93	40.24	43.94	42.59	44.95	44.78
	LCB v2	46.71	44.81	44.88	44.10	44.55	45.08	**45.47**	45.01	43.77	44.81	43.84
	IFEval	65.35	48.80	48.20	51.80	47.48	52.16	52.04	45.08	50.00	51.20	**54.08**
	AlpacaEval	0.42	-0.92	**-0.18**	-0.50	-0.95	-1.01	-0.78	-0.27	-0.79	-0.55	-0.36
	HaluEval	53.97	75.47	73.10	75.53	75.70	75.80	**76.87**	72.07	75.93	75.83	75.53
	Avg. (Δ vs. SFT)	-	-	-0.10	+1.26	-0.15	+0.63	+0.63	-0.89	+0.46	+0.98	**+1.52**

This allows the selected subspace to capture a comparable fraction of the dominant singular structure across different weight matrices. In practice, we use $\rho = 0.5$ and apply the projection to all two-dimensional weight matrices by default, as these choices work well across the evaluated settings.

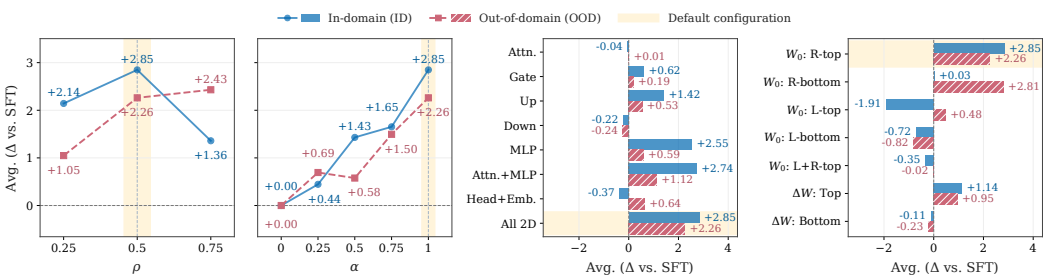
5 Experiments

5.1 Main Experiments on Mathematical Reasoning SFT

We first evaluate SPACE in the setting of mathematical reasoning SFT, in which base models are fine-tuned on long chain-of-thought (CoT) mathematical reasoning data.

Training recipe. We conduct full-parameter SFT on two base models: Qwen2.5-7B (Yang et al., 2025b) and Qwen3-14B-Base (Yang et al., 2025a). Following prior studies (Zhu et al., 2026; Ren et al., 2026), we use pretrained checkpoints without instruction tuning. For Qwen2.5-7B, following Wu et al. (2026b); Zhu et al. (2026), we use 100K training examples sampled from NuminaMath-CoT (Li et al., 2024). For Qwen3-14B-Base, we follow Ren et al. (2026) and use Math-CoT-20k, which contains 20K examples. Additional training details are provided in Appendix B.1.

Benchmarks and evaluation protocol. We evaluate ID performance on six mathematical benchmarks: MATH-500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME 2024 (Zhang & Math-AI, 2024), AIME 2025 (Zhang & Math-AI, 2025), and AMC 2023 (Mathematical Association of America, 2023). Following the benchmark suite used by Ren et al. (2026), we further evaluate OOD performance on six benchmarks: MMLU-Pro (Wang et al., 2024), GPQA-Diamond (GPQA-D) (Rein et al., 2024), LiveCodeBench v2 (LCB v2) (Jain et al., 2025), IFEval (Zhou et al., 2023), AlpacaEval (RM) (Dubois et al., 2024), and HaluEval (Li et al., 2023). Further evaluation details are provided in Appendix B.2.



(a) Subspace threshold (b) Calibration strength (c) Matrix selection (d) Subspace selection

Figure 2: Ablation study on Qwen2.5-7B. "Avg. (Δ vs. SFT)" denotes the average changes relative to the SFT checkpoint over the six ID mathematical benchmarks and the six OOD benchmarks. In (c), "Attn." denotes attention projection matrices; "Gate", "Up", and "Down" denote the corresponding MLP matrices; "MLP" includes all MLP projection matrices; "Head+Emb." denotes the output head and embedding matrices; and "All 2D" denotes all two-dimensional weight matrices. In (d), "R" and "L" denote the right and left singular vectors, while "top" and "bottom" denote the leading and trailing singular subspaces with their dimensions determined using the default threshold $\rho = 0.5$. "W_0" and "ΔW" indicate that the subspace is derived from the pre-SFT weights and the SFT update, respectively. Shading indicates the default SPACE configuration in each panel.

Baselines. We compare SPACE with eight representative post-hoc model editing/merging baselines: WiSE-FT (Wortsman et al., 2022b), TIES (Yadav et al., 2023), LiNeS (Wang et al., 2025b), DARE (Yu et al., 2024), POME (Liu et al., 2025a), FAPM (Huang et al., 2025), DG-Hard (Abro & Tahir, 2026), and MonoSoup (Abdollahpoorrostam et al., 2026). See Appendix B.3 for details.

Main results. Table 1 shows that SPACE yields the strongest overall improvements among all baselines, achieving the highest "Avg. (Δ vs. SFT)" in both ID and OOD performance for both models. On Qwen2.5-7B, SPACE improves every evaluated benchmark over SFT and even surpasses the Base model on five of the six OOD benchmarks, suggesting that reasoning capabilities acquired during mathematical SFT may generalize to OOD tasks after calibration. On Qwen3-14B-Base, SPACE also yields substantial additional gains despite the already strong mathematical reasoning performance after SFT. Importantly, the ID gains suggest that SPACE does more than recover capabilities degraded by SFT. They also suggest that reverting updates within the principal subspace can improve generalization to unseen ID examples. The comparison with WiSE-FT further highlights the benefit of selective calibration: global interpolation can recover OOD performance in some settings, but often provides smaller ID gains or degrades the adapted model, whereas SPACE improves both. Finally, SPACE's advantage over spectral post-hoc baselines such as DG-Hard and MonoSoup, which operate on the learned update itself, suggests that the geometry of the pre-SFT weights provides additional information for post-hoc refinement. Overall, these results suggest that effective post-hoc refinement depends not simply on shrinking or restructuring the learned update, but on strategically calibrating it with respect to the geometry of the pre-SFT weights.

5.2 Ablation Study

We conduct an ablation study on Qwen2.5-7B to examine the sensitivity of SPACE to its key hyperparameters and design choices. All training and evaluation settings follow those in Section 5.1. We report the average change relative to the SFT checkpoint separately over the six ID mathematical benchmarks and the six OOD benchmarks.

Hyperparameter sensitivity. We first study the sensitivity of SPACE to the subspace threshold ρ and calibration strength α. For the latter, we generalize equation 6 as $W_{\text{cal}} = W_0 + \Delta W(I - \alpha P_0)$, where $\alpha \in [0, 1]$ controls the fraction of the SFT update reverted within the selected principal subspace. Thus, $\alpha = 1$ corresponds to the default SPACE. Figure 2(a) shows that $\rho = 0.5$ provides a good balance between ID and OOD performance. Using a smaller subspace weakens the gains, while a larger one further improves OOD performance at the cost of reduced ID improvement. We therefore use $\rho = 0.5$ as the default. Figure 2(b) shows that performance generally improves as

Table 2: Experimental results of training Qwen2.5-7B on coding and medical data. "(M)" denotes the medical subset. **Bold** indicates the best result excluding the Base model.

Coding SFT						Medical SFT					
Benchmark	**Base**	**SFT**	**DG-Hard**	**MonoSoup**	**SPACE**	**Benchmark**	**Base**	**SFT**	**DG-Hard**	**MonoSoup**	**SPACE**
In-domain (ID), Coding						*In-domain (ID), Medical*					
HumanEval	77.44	**78.66**	75.61	76.83	**78.66**	MedMCQA	53.74	54.79	54.22	54.12	**55.25**
HumanEval+	71.34	71.95	70.73	71.34	**73.78**	MedQA	58.68	**65.12**	60.80	63.39	64.65
MBPP	75.40	78.04	77.78	**78.31**	**78.31**	PubMedQA	71.70	70.00	70.60	70.00	**73.20**
MBPP+	65.61	67.99	67.99	**68.78**	67.99	MMLU-Pro (M)	55.83	57.65	55.83	59.15	**59.54**
LCB v2	28.96	23.09	22.50	**24.53**	22.77	GPQA (M)	44.87	36.92	37.95	42.82	**47.69**
Avg. (Δ vs. SFT)	-	-	-1.02	+0.01	**+0.36**	**Avg. (Δ vs. SFT)**	-	-	-1.02	+1.00	**+3.17**
Out-of-domain (OOD)						*Out-of-domain (OOD)*					
MMLU-Pro	40.33	38.35	39.30	40.39	**42.32**	MMLU-Pro	40.33	**53.90**	53.41	53.68	52.47
GPQA-D	26.77	25.35	22.16	26.04	**27.27**	GPQA-D	26.77	24.91	**26.04**	25.22	25.88
						LCB v2	28.96	24.07	25.64	**27.14**	26.48
IFEval	42.57	54.08	53.00	**54.20**	53.60	IFEval	42.57	35.01	34.41	35.97	**36.09**
AlpacaEval	-1.13	-0.43	**-0.23**	-0.25	-0.39	AlpacaEval	-1.13	-2.30	-2.31	-1.99	**-1.73**
HaluEval	46.86	49.32	49.52	49.41	**50.31**	HaluEval	46.86	63.31	64.05	**65.39**	65.10
Avg. (Δ vs. SFT)	-	-	-0.58	+0.62	**+1.29**	**Avg. (Δ vs. SFT)**	-	-	+0.39	**+1.09**	+0.90

the calibration strength increases, with $\alpha = 1$ performing best. Together, these results suggest that SPACE is best applied as a hard projection, with ρ as the sole trade-off hyperparameter.

Matrix selection. We next examine which weight matrices should be calibrated. As shown in Figure 2(c), calibrating attention matrices alone has little effect, whereas jointly calibrating the MLP matrices yields substantially larger improvements. This is consistent with prior work suggesting that MLP modules serve as important sites for factual knowledge storage and recall (Geva et al., 2021; Dai et al., 2022; Meng et al., 2022). Among individual MLP projections, the Up projection provides the largest gains, while the Down projection alone is ineffective. Interestingly, attention contributes little on its own but provides additional gains when combined with MLP calibration, suggesting that MLPs drive the primary benefits while attention plays a complementary role. Finally, applying SPACE to all two-dimensional weight matrices gives the strongest overall results.

Subspace selection. Finally, we compare different choices of the calibration subspace. As shown in Figure 2(d), the leading right singular subspace of the pre-SFT weights, as used by SPACE, provides the best balance between ID and OOD improvements. In contrast, the bottom right singular subspace strongly favors OOD recovery but provides almost no ID gain, indicating a less balanced trade-off. Moreover, the weaker performance of subspaces derived from ΔW supports the central premise of Geometric Calibration: the pre-SFT weights provide a more informative reference for refining a completed SFT update. Overall, the clear advantage of the right over the left singular subspaces is consistent with the input-side formulation of SPACE.

5.3 Experiments on Different Target Domains

We next examine whether the benefits of SPACE extend beyond mathematical reasoning SFT by training on coding and medical data.

Setup. For medical SFT, we fine-tune Qwen2.5-7B on 23,493 high-quality medical reasoning examples from m23k (Huang et al., 2026), as used in prior work (Li et al., 2025). For coding SFT, following prior work (Zhu et al., 2026), we use 100K examples sampled from Magicoder-Evol-Instruct-110K (Wei et al., 2024) as training data, and evaluate ID performance using the EvalPlus library (Liu et al., 2023) on HumanEval, HumanEval+ (Chen et al., 2021), MBPP, and MBPP+ (Austin et al., 2021). LiveCodeBench v2 (Jain et al., 2025) is also included for ID evaluation. The maximum generation length is set to 4,096 tokens. For medical SFT, we use the evaluation toolkit provided by Li et al. (2025) and report results on MedMCQA (Pal et al., 2022), MedQA (Jin et al., 2021), PubMedQA (Jin et al., 2019), MMLU-Pro (Medical) (Wang et al., 2024), and GPQA (Medical) (Rein et al., 2024), with a maximum generation length of 8,192 tokens. Additionally, we include DG-Hard and MonoSoup as two strong baselines for comparison. All other training and evaluation settings follow the same protocol as in Section 5.1.

Table 3: Compatibility with different SFT methods. "Original" denotes the checkpoint produced by each training method. "Avg. (Δ vs. Original)" denotes the average score change relative to the original checkpoint. **Bold** indicates the best score within each method.

Benchmark	Base	DFT			ASFT			RPSFT		
		Original	MonoSoup	SPACE	Original	MonoSoup	SPACE	Original	MonoSoup	SPACE
In-domain (ID)										
MATH500	28.79	54.40	57.33	**58.64**	60.55	**61.56**	61.12	45.98	46.64	**48.69**
AIME25	0.42	**1.88**	1.46	1.25	1.67	**2.08**	1.88	**1.04**	0.63	0.83
AIME24	1.65	2.08	3.54	**3.75**	**5.83**	5.00	5.21	**1.46**	1.25	1.25
OlympiadBench	7.69	23.22	26.29	**27.18**	27.98	**28.81**	28.45	14.40	15.53	**17.03**
Minerva Math	9.26	**20.57**	20.22	20.24	**24.43**	22.43	22.73	13.01	13.81	**15.12**
AMC23	15.65	27.81	**29.06**	28.91	35.31	36.72	**38.28**	**21.72**	17.97	20.16
Avg. (Δ vs. Original)	-	-	+1.32	**+1.67**	-	+0.14	**+0.32**	-	-0.30	**+0.91**
Out-of-domain (OOD)										
MMLU-Pro	40.33	**33.66**	30.84	33.39	33.97	33.54	**34.45**	34.67	36.38	**41.06**
GPQA-D	26.77	27.27	**27.78**	26.14	26.48	**27.02**	25.88	24.15	25.69	**27.27**
LCB v2	28.96	30.85	32.22	**33.27**	27.14	**29.35**	28.90	29.03	29.35	**29.75**
IFEval	42.57	43.17	46.88	**48.20**	45.20	47.12	**48.44**	44.48	44.48	**45.56**
AlpacaEval	-1.13	-0.43	-0.17	**-0.13**	**-0.08**	-0.09	-0.09	-0.03	**-0.01**	-0.07
HaluEval	46.86	**48.04**	46.97	47.11	47.06	47.43	**48.62**	46.86	47.48	**48.68**
Avg. (Δ vs. Original)	-	-	+0.33	**+0.90**	-	+0.77	**+1.07**	-	+0.70	**+2.18**

Results. As shown in Table 2, SPACE remains effective when the training domain shifts to coding or medicine. For coding SFT, where the uncalibrated SFT checkpoint already performs well on most ID benchmarks, SPACE largely preserves the learned adaptation while improving OOD performance. For medical SFT, the ID gains are more pronounced, particularly on benchmarks where SFT causes regression relative to the Base model. This pattern suggests that geometric calibration is particularly beneficial when SFT introduces stronger interference with capabilities already encoded in the pre-SFT model. Overall, SPACE improves upon SFT across both domains and outperforms two strong baselines, DG-Hard and MonoSoup, demonstrating its applicability across different forms of task adaptation rather than relying on properties specific to mathematical reasoning.

5.4 Compatibility with Different SFT Methods

Recently, several variants of SFT have been proposed to improve performance and generalization. We further examine whether SPACE is compatible with these methods.

Setup. The comparison is conducted on Qwen2.5-7B trained with data from NuminaMath-CoT. Three SFT variants are considered: DFT (Wu et al., 2026b), which reweights the token-level objective; ASFT (Zhu et al., 2026), which augments DFT with distributional anchoring; and RPSFT (Jin et al., 2026), which introduces regularization to constrain weight updates during fine-tuning. For each method, we apply SPACE to its completed checkpoint using the same pre-SFT checkpoint as the geometric reference. Other training and evaluation settings remain consistent with Section 5.1.

Results. Table 3 shows that SPACE improves overall ID and OOD performance across all three SFT methods. The consistent gains across methods with different training objectives suggest that the utility of pre-SFT geometry is not tied to a particular SFT update formation process. Notably, the gains are particularly pronounced on RPSFT, even though it already constrains weight geometry during training, suggesting that training-time regularization and post-hoc geometric calibration address complementary aspects. Moreover, SPACE consistently achieves larger aggregate gains than the strong baseline MonoSoup across all three SFT variants. Appendix C.1 shows that these benefits also extend to LoRA fine-tuning. Overall, these results show that SPACE can effectively complement different training objectives by refining their completed updates post hoc.

5.5 Generalization Across Model Families and Scales

We further evaluate whether SPACE generalizes across different model families and scales, covering Qwen3.5, Qwen3, Olmo-3, and Llama-3.1 models from 2B to 8B parameters.

Table 4: Generalization of SPACE across different model families and scales.

Benchmark	Qwen3.5-2B-Base			Qwen3-8B-Base			Olmo-3-7B-Instruct-SFT			Llama-3.1-8B-Instruct		
	Base	SFT	SPACE	Base	SFT	SPACE	Base	SFT	SPACE	Base	SFT	SPACE
In-domain (ID)												
MATH500	72.40	76.47	**78.00**	55.93	83.73	**84.87**	60.80	87.93	**89.33**	39.07	49.20	**53.33**
AIME25	15.63	17.29	**17.50**	4.58	23.96	**24.38**	3.33	**35.00**	34.38	0.21	**1.46**	1.25
AIME24	12.92	**16.67**	15.21	7.29	**22.71**	**22.71**	6.67	41.25	**41.46**	1.88	1.67	**2.29**
OlympiadBench	37.09	40.15	**41.38**	24.05	46.86	**48.79**	25.88	55.26	**56.79**	10.47	18.96	**21.04**
Minerva Math	27.08	39.95	**40.32**	19.49	**41.79**	40.44	25.74	33.95	**34.56**	15.56	18.01	**21.57**
AMC23	49.22	53.59	**55.16**	32.50	64.22	**67.34**	37.03	79.53	**82.19**	14.69	26.09	**30.16**
Avg. (Δ vs. SFT)	-	-	**+0.58**	-	-	**+0.88**	-	-	**+0.97**	-	-	**+2.38**
Out-of-domain (OOD)												
MMLU-Pro	46.40	52.60	**54.00**	56.70	**56.40**	56.20	42.70	44.40	**44.90**	43.50	40.80	**44.80**
GPQA-D	30.64	29.97	**32.66**	36.36	**29.46**	**29.46**	28.79	22.22	**25.93**	23.74	15.49	**19.87**
LCB v2	14.94	11.55	**13.31**	33.66	24.92	**29.29**	17.09	**22.44**	21.14	18.92	8.41	**12.20**
IFEval	57.67	45.32	**53.84**	54.08	35.85	**40.89**	83.57	79.26	**82.49**	81.89	29.14	**36.81**
AlpacaEval	0.21	-0.84	**-0.23**	-0.39	-2.32	**-1.10**	0.28	0.34	**0.38**	1.17	-1.87	**-0.38**
HaluEval	50.00	**59.97**	59.20	49.33	72.13	**72.30**	54.23	**55.47**	55.20	62.73	63.37	**63.97**
Avg. (Δ vs. SFT)	-	-	**+2.37**	-	-	**+1.77**	-	-	**+0.99**	-	-	**+3.66**

Setup. The experiments use Qwen3.5-2B-Base, Qwen3-8B-Base, Olmo-3-7B-Instruct-SFT (Olmo et al., 2025) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as starting checkpoints for mathematical SFT. Math-CoT-20k is employed as the training data, which contains longer reasoning trajectories and enables more substantial performance improvements through SFT. Other training and evaluation settings are kept the same as in Section 5.1.

Results. As shown in Table 4, SPACE yields positive average gains in both ID and OOD performance across all evaluated models. Importantly, the OOD improvements are accompanied by further ID gains across different families and scales, suggesting that geometric calibration refines rather than reverses the SFT adaptation. The gains are particularly pronounced on Llama-3.1-8B, where SFT causes substantial OOD capability regression; in this setting, SPACE improves every evaluated OOD benchmark while further strengthening ID performance. Even at the smallest 2B scale, SPACE still performs well, indicating that its benefits are not limited to larger models. Additional results in Appendix C.2 show consistent gains on Qwen3-4B-Base and larger average improvements than MonoSoup across all evaluated models. Overall, these results demonstrate the broad applicability of SPACE across model families and scales.

5.6 Additional Analyses

Further analyses show that SPACE is effective at intermediate SFT checkpoints (Appendix C.3), robust to SFT training hyperparameters (Appendix C.4), and more effective as a post-hoc calibration than as a training-time projection (Appendix C.5).

6 Conclusion

We study *Geometric Calibration*, a perspective for refining SFT updates using the geometry of the pre-SFT weights as a reference. We develop SPACE as a lightweight post-hoc implementation of this perspective, which removes SFT update components within the principal input subspace of the pre-SFT weights while preserving the remaining adaptation. SPACE operates directly on the pre- and post-SFT checkpoints, with no data or additional training required. Overall, across multiple model families, scales, training domains, and SFT variants, SPACE improves both ID and OOD performance over the SFT checkpoint and outperforms representative post-hoc baselines. Our ablations further support the principal right singular subspace of the pre-SFT weights as an effective calibration reference. Taken together, these results highlight the value of pre-SFT weight geometry for refining completed adaptations and, more broadly, motivate *Geometric Calibration* as a general principle for continued model improvement beyond SFT.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing and polishing, literature search and discovery, and software code assistance. All identified literature was independently reviewed and verified by the authors. Regarding code assistance, generative AI tools were used only to assist with code editing, completion, and debugging. All AI-assisted code was manually reviewed, tested, and validated. The research ideas, methodology, experimental design, code structure and implementation logic, and interpretation of the results were developed by the authors. All AI-assisted content was validated by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our study does not involve human-subject experiments or the collection of personal or sensitive data. The experiments are conducted using publicly available models, datasets, and standard evaluation benchmarks. Since SPACE operates on existing language-model checkpoints, the resulting models may retain safety risks associated with the underlying LLMs, including biased, inaccurate, or potentially harmful outputs. Appropriate validation, monitoring, and task-specific safeguards should therefore be maintained in downstream applications.

REPRODUCIBILITY STATEMENT

Detailed training settings, evaluation protocols, and baseline details are provided in Appendices B.1, B.2, and B.3. Additional experiments examine robustness across models, SFT settings, and training configurations. We will release the source code upon acceptance.

REFERENCES

Alireza Abdollahpoorrostam, Nikolaos Dimitriadis, Adam Hazimeh, and Pascal Frossard. MODEL SOUPS NEED ONLY ONE INGREDIENT. In *Forty-third International Conference on Machine Learning*, 2026.

Aarash Abro and Muhammad Tahir. Spectral unforgetting: Post-hoc recovery of damaged capabilities without retraining. *arXiv preprint arXiv:2605.20296*, 2026.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*, 2025.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 5484–5495, 2021.

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.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 14852–14882, 2023.

Maggie Ziyu Huan, Yuetai Li, Tianyu 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. In *Forty-third International Conference on Machine Learning*, 2026.

Wei Huang, Anda Cheng, and Yinggui Wang. Mitigating catastrophic forgetting in large language models with forgetting-aware pruning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21853–21867, 2025.

Xiaoke Huang, Juncheng Wu, Hui Liu, Xianfeng Tang, and Yuyin Zhou. m1: Unleash the potential of test-time scaling for medical reasoning in large language models. In *Machine Learning for Health 2025*, 2026.

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

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

Hangzhan Jin, Tianwei Ni, Lu Li, Pierre-Luc Bacon, Mohammad Hamdaqa, and Doina Precup. Rotation-preserving supervised fine-tuning. *arXiv preprint arXiv:2605.10973*, 2026.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019.

Dawid Jan Kopiczko, Sagar Vaze, Tijmen Blankevoort, and Yuki M Asano. Data repetition beats data scaling in long-cot supervised fine-tuning. In *Third Conference on Language Modeling*, 2026.

Hanyu Lai, Xiao Liu, Junjie Gao, Jiale Cheng, Zehan Qi, Yifan Xu, Shuntian Yao, Dan Zhang, Jinhua Du, Zhenyu Hou, et al. A survey of post-training scaling in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2771–2791, 2025.

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Qingfu Zhang, Hongbin Liu, Gaofeng Meng, and Fei Zhu. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. In *Forty-third International Conference on Machine Learning*, 2026.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Gaotang Li, Ruizhong Qiu, Xiusi Chen, Heng Ji, and Hanghang Tong. Beyond log likelihood: Probability-based objectives for supervised fine-tuning across the model capability continuum. *arXiv preprint arXiv:2510.00526*, 2025.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 2024.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 6449–6464, 2023.

Longhua Li, Lei Qi, Qi Tian, and Xin Geng. Model merging in the essential subspace. *arXiv preprint arXiv:2602.20208*, 2026.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Qingyuan Liu, Jia-Chen Gu, Yunzhi Yao, Hong Wang, and Nanyun Violet Peng. Energy-regularized sequential model editing on hyperspheres. In *International Conference on Learning Representations*, volume 2026, pp. 149084–149112, 2026.

Yong Liu, Di Fu, Yang Luo, Zirui Zhu, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Pome: Post optimization model edit via muon-style projection. *arXiv preprint arXiv:2510.06627*, 2025a.

Zihang Liu, Tianyu Pang, Oleg Balabanov, Chaoqun Yang, Tianjin Huang, Lu Yin, Yaoqing Yang, and Shiwei Liu. LIFT the veil for the truth: Principal weights emerge after rank reduction for reasoning-focused supervised fine-tuning. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=XHHIZNgrho.

Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Min-Ling Zhang, and Tong Wei. Keeplora: Continual learning with residual gradient adaptation. In *International Conference on Learning Representations*, volume 2026, pp. 136017–136040, 2026.

Mathematical Association of America. AMC 2023 competition problems, 2023.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *Advances in Neural Information Processing Systems*, 38:132119–132138, 2026.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pp. 248–260. PMLR, 2022.

Zihuan Qiu, Lei Wang, Yang Cao, Runtong Zhang, Bing Su, Yi Xu, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Null-space filtering for data-free continual model merging: Preserving stability, promoting plasticity. In *International Conference on Learning Representations*, volume 2026, pp. 128492–128518, 2026.

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. In *First Conference on Language Modeling*, 2024.

Qihan Ren, Wang Peng, Ruikun Cai, Shuai Shao, Dadi Guo, Yuejin Xie, Yafu Li, Quanshi Zhang, Xia Hu, Jing Shao, and Dongrui Liu. Rethinking generalization in reasoning SFT: A conditional analysis on optimization, data, and model capability. In *Third Conference on Language Modeling*, 2026.

Zhennan Shen, Yanshu Li, Qingyu Yin, Chak Tou Leong, Zhilin Wang, Yanxu Chen, Rongduo Han, Sunbowen Lee, and Yi R Fung. On the geometry of on-policy distillation. *arXiv preprint arXiv:2606.07082*, 2026.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026.

Ningyuan Shi, Zhipeng Zhou, Hao Wang, Chunyan Miao, and Peilin Zhao. Pact: Preserving anchored cores in task-vectors for model merging. *arXiv preprint arXiv:2606.18627*, 2026.

Wenju Sun, Qingyong Li, Tiancheng Li, Yangliao Geng, and Boyang Li. Revisiting the role of pretrained weights in model merging: On near-optimality within the core subspace. In *Forty-third International Conference on Machine Learning*, 2026.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4823–4836, 2025a.

Ke Wang, Nikos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, Francois Fleuret, and Pascal Frossard. Lines: Post-training layer scaling prevents forgetting and enhances model merging. In *International Conference on Learning Representations*, volume 2025, pp. 27100–27133, 2025b.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and LINGMING ZHANG. Magicoder: Empowering code generation with OSS-instruct. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=XUeoOBid3x.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. Pmlr, 2022a.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7949–7961. IEEE, 2022b.

Chao-Chung Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Vivian Chen, Shao-Hua Sun, and Hung-yi Lee. Mitigating forgetting in llm fine-tuning via low-perplexity token learning. *Advances in Neural Information Processing Systems*, 38:1708–1744, 2026a.

Yongliang Wu, Yizhou Zhou, Ziheng Zhou, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, et al. On the generalization of sft: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations*, volume 2026, pp. 27550–27571, 2026b.

Yifeng Xiong and Xiaohui Xie. Oplora: Orthogonal projection lora prevents catastrophic forgetting during parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34088–34096, 2026.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025b.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is more for reasoning. In *Second Conference on Language Modeling*, 2025.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Xiangchi Yuan, Xiang Chen, Tong Yu, Dachuan Shi, Can Jin, Wenke Lee, and Saayan Mitra. Mitigating forgetting between supervised and reinforcement learning yields stronger reasoners. *arXiv preprint arXiv:2510.04454*, 2025.

Jiaying Zhang, Lei Shi, Jiguo Li, Jun Xu, Jiuchong Gao, Jinghua Hao, and Renqing He. Geora: Geometry-aware low-rank adaptation for rlvr. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24207–24221, 2026.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, et al. The path not taken: Rlvr provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

He Zhu, Junyou Su, Peng Lai, Ren Ma, Wenjia Zhang, Linyi Yang, and Guanhua Chen. Anchored supervised fine-tuning. In *International Conference on Learning Representations*, volume 2026, pp. 110364–110385, 2026.

Appendix Contents

A Derivation of the Principal Input Projector **15**

B Detailed Experimental Settings **16**

B.1 Detailed Training Settings of the Main Experiments 16

B.2 Detailed Evaluation Settings of the Main Experiments 16

B.3 Baseline Details 16

C Additional Experiments **17**

C.1 Compatibility with LoRA Fine-Tuning 17

C.2 Additional Results on Model Families and Scales 18

C.3 Performance at Intermediate SFT Checkpoints 18

C.4 Robustness to SFT Training Hyperparameters 19

C.5 Training-Time vs. Post-Hoc Projection 19

A Derivation of the Principal Input Projector

We derive the solution to the projection problem in equation 4. Specifically, we solve the following optimization problem:

$$P_0 \in \underset{\substack{P=P^\top=P^2 \\ \mathrm{rank}(P)=k}}{\arg\min} \|W_0 - W_0 P\|_F^2.$$

Since $P = P^\top = P^2$,

$$\begin{aligned} \|W_0 - W_0 P\|_F^2 &= \|W_0(I-P)\|_F^2 \\ &= \mathrm{tr}\big((I-P)W_0^\top W_0 (I-P)\big) \\ &= \mathrm{tr}(W_0^\top W_0) - \mathrm{tr}(P W_0^\top W_0). \end{aligned}$$

Therefore, minimizing equation 4 is equivalent to maximizing

$$\mathrm{tr}(P W_0^\top W_0)$$

over rank-k orthogonal projectors.

Writing

$$W_0 = U_0 \Sigma_0 V_0^\top,$$

we have

$$W_0^\top W_0 = V_0 \Sigma_0^2 V_0^\top.$$

Any rank-k orthogonal projector can be written as

$$P = QQ^\top, \qquad Q^\top Q = I_k.$$

Therefore,

$$\mathrm{tr}(P W_0^\top W_0) = \mathrm{tr}(Q^\top W_0^\top W_0 Q).$$

By the Ky Fan maximum principle (Fan, 1949),

$$\max_{Q^\top Q = I_k} \mathrm{tr}(Q^\top W_0^\top W_0 Q) = \sum_{i=1}^{k} \sigma_{0,i}^2,$$

and the maximum is attained when the columns of Q span the eigenspace associated with the k largest eigenvalues of $W_0^\top W_0$. Since these eigenvectors are precisely the leading k right singular vectors of W_0, we may choose

$$Q = V_{0,k}.$$

Hence, an optimal projector is

$$P_0 = V_{0,k} V_{0,k}^\top.$$

B Detailed Experimental Settings

B.1 Detailed Training Settings of the Main Experiments

Unless otherwise specified, we use AdamW with a global batch size of 256 and a learning rate of 5×10^{-5}. The learning rate follows a cosine decay schedule with a warm-up ratio of 0.1, and all models are trained for one epoch. We use standard next-token prediction over the response tokens as the SFT objective. All experiments are conducted using four NVIDIA H200 GPUs.

B.2 Detailed Evaluation Settings of the Main Experiments

Following prior studies (Wu et al., 2026b; Zhu et al., 2026), for the mathematical benchmarks (MATH-500, Minerva Math, OlympiadBench, AIME 2024, AIME 2025, and AMC 2023), we adopt the Qwen2.5-Math (Yang et al., 2024) evaluation pipeline and use the default CoT prompt template with a temperature of 1.0 and top-p of 1.0. Following the benchmark suite used by Ren et al. (2026), we further consider OOD benchmarks. In particular, we use MMLU-Pro (Wang et al., 2024) for broad knowledge-intensive reasoning, GPQA-Diamond (GPQA-D) (Rein et al., 2024) for graduate-level scientific reasoning, LiveCodeBench v2 (LCB v2) (Jain et al., 2025) for code reasoning, IFEval (Zhou et al., 2023) for instruction following, AlpacaEval (RM) (Dubois et al., 2024) for open-ended response quality, and HaluEval (Li et al., 2023) for hallucination detection. For these evaluations, we follow the prompt templates and evaluation settings of Ren et al. (2026), using a temperature of 0.6 and top-p of 0.95.

The remaining evaluation settings differ slightly between Qwen2.5-7B and Qwen3-14B-Base, as their training data have substantially different maximum CoT lengths (approximately 3K vs. 16K tokens), leading to different inference costs. For Qwen2.5-7B, we set the maximum generation length to 4,096 tokens. For Qwen3-14B-Base, we set the maximum generation length to 16,384 tokens. Moreover, for Qwen2.5-7B, we report avg@16 on the mathematical benchmarks and GPQA-Diamond, and evaluate MMLU-Pro and HaluEval on their full test sets. For Qwen3-14B-Base, following Ren et al. (2026), we report avg@3 on MATH-500, Minerva Math, OlympiadBench, and GPQA-Diamond, and avg@16 on the remaining mathematical benchmarks. Also, MMLU-Pro and HaluEval are evaluated on subsets of 1K and 3K examples. For both models, we report pass@1 for MMLU-Pro and HaluEval, strict instruction-level accuracy for IFEval, and avg@3 for LiveCodeBench v2. For AlpacaEval (RM), we generate one response for each instruction and report the average reward score from the `allenai/Llama-3.1-8B-Instruct-RM-RB2` reward model.[1]

B.3 Baseline Details

We compare SPACE with the following representative post-hoc model editing/merging baselines.

- WiSE-FT (Wortsman et al., 2022b) linearly interpolates the pre-SFT and SFT checkpoints.
- TIES (Yadav et al., 2023) applies task-vector trimming and sign-aware merging.
- LiNeS (Wang et al., 2025b) applies depth-dependent layer-wise scaling to the fine-tuning update.
- DARE (Yu et al., 2024) sparsifies the fine-tuning update through random dropping and rescaling.
- POME (Liu et al., 2025a) refines the fine-tuning update through Muon-style projection.

[1] `https://huggingface.co/allenai/Llama-3.1-8B-Instruct-RM-RB2`

Table 5: Compatibility of SPACE with LoRA fine-tuning on Qwen2.5-7B ($r = 64$, $\alpha = 128$). "Original" denotes the LoRA fine-tuned checkpoint. "Avg. (Δ vs. Original)" denotes the average score change after applying MonoSoup or SPACE relative to Original within each benchmark group. **Bold** indicates the best score.

Benchmark	LoRA		
	Original	MonoSoup	SPACE
In-domain (ID)			
MATH500	45.30	46.06	**46.34**
AIME25	0.21	**1.25**	0.63
AIME24	0.83	0.83	**2.50**
OlympiadBench	14.82	14.95	**15.40**
Minerva Math	12.94	13.03	**13.63**
AMC23	**20.16**	17.97	18.91
Avg. (Δ vs. Original)	-	-0.03	**+0.53**
Out-of-domain (OOD)			
MMLU-Pro	38.20	37.28	**43.28**
GPQA-D	22.35	22.85	**27.84**
LCB v2	27.98	**28.25**	27.98
IFEval	42.81	44.00	**48.44**
AlpacaEval	-0.08	**-0.06**	-0.12
HaluEval	46.75	46.87	**51.22**
Avg. (Δ vs. Original)	-	+0.20	**+3.44**

- FAPM (Huang et al., 2025) performs forgetting-aware pruning of the fine-tuning update.
- DG-Hard (Abro & Tahir, 2026) performs spectral filtering of the fine-tuning update.
- MonoSoup (Abdollahpoorrostam et al., 2026) decomposes and reweights spectral components of the fine-tuning update.

Implementation details. We follow the recommended settings in the original papers or their released implementations, with the following adaptations. For TIES, we use a single SFT update, so the method reduces to global magnitude trimming. For FAPM, the default setting performs poorly, likely due to excessive update pruning in our setting. With the scoring rule unchanged, we evaluate several retention ratios and choose to retain 90% of entries per tensor, which provides good overall performance.

Comparison with SPACE. WiSE-FT and LiNeS scale updates globally or by layer depth. TIES, DARE, and FAPM prune individual update entries using magnitude-based scores or random masking. Although some of these methods incorporate pre-SFT weight information, they do not explicitly account for the geometry of the pre-SFT weights. POME, DG-Hard, and MonoSoup exploit spectral structure, but define their transformations using the learned update's own singular components. SPACE instead calibrates post-SFT checkpoints against the geometry of the pre-SFT weights, removing update components within their principal input subspaces while preserving the update on the orthogonal complements.

C Additional Experiments

C.1 Compatibility with LoRA Fine-Tuning

We further examine whether SPACE is compatible with parameter-efficient fine-tuning. Specifically, we fine-tune Qwen2.5-7B with LoRA on NuminaMath-CoT using rank $r = 64$ and scaling factor $\alpha = 128$, while keeping the remaining training and evaluation settings consistent with Section 5.1. We then apply SPACE to the resulting LoRA checkpoint using the same pre-SFT checkpoint as the geometric reference. As shown in Table 5, SPACE improves average ID and OOD performance over the original LoRA checkpoint, with particularly pronounced OOD gains. It also achieves larger aggregate gains than MonoSoup. These results suggest that the applicability of SPACE extends beyond full-parameter SFT to parameter-efficient adaptation.

Table 6: Generalization of SPACE across Qwen models at different scales (2B, 4B, and 8B).

Benchmark	Qwen3.5-2B-Base				Qwen3-4B-Base				Qwen3-8B-Base			
	Base	SFT	MonoSoup	SPACE	Base	SFT	MonoSoup	SPACE	Base	SFT	MonoSoup	SPACE
In-domain (ID)												
MATH500	72.40	76.47	**78.07**	78.00	40.13	78.73	80.53	**81.33**	55.93	83.73	84.67	**84.87**
AIME25	15.63	17.29	17.29	**17.50**	2.29	17.50	**20.00**	18.13	4.58	23.96	23.96	**24.38**
AIME24	12.92	**16.67**	14.79	15.21	5.63	17.71	16.25	**18.54**	7.29	22.71	**23.33**	22.71
OlympiadBench	37.09	40.15	**41.93**	41.38	19.56	40.89	**43.26**	42.72	24.05	46.86	48.49	**48.79**
Minerva Math	27.08	39.95	**40.81**	40.32	16.79	37.99	36.76	**39.22**	19.49	41.79	**42.28**	40.44
AMC23	49.22	53.59	54.53	**55.16**	22.03	56.41	57.19	**59.69**	32.50	64.22	65.47	**67.34**
Avg. (Δ vs. SFT)	-	-	+0.55	**+0.58**	-	-	+0.79	**+1.73**	-	-	+0.82	**+0.88**
Out-of-domain (OOD)												
MMLU-Pro	46.40	52.60	51.80	**54.00**	37.10	47.30	48.10	**49.20**	56.70	56.40	**56.50**	56.20
GPQA-D	30.64	29.97	**32.83**	32.66	18.86	21.04	20.54	**21.72**	36.36	**29.46**	28.96	**29.46**
LCB v2	14.94	11.55	11.35	**13.31**	31.83	15.33	17.87	**18.33**	33.66	24.92	27.66	**29.29**
IFEval	57.67	45.32	51.80	**53.84**	46.76	30.46	31.65	**32.61**	54.08	35.85	37.17	**40.89**
AlpacaEval	0.21	-0.84	-0.52	**-0.23**	-2.33	-3.08	-2.83	**-2.71**	-0.39	-2.32	-1.75	**-1.10**
HaluEval	50.00	**59.97**	59.87	59.20	48.50	55.13	55.70	**57.17**	49.33	72.13	**74.00**	72.30
Avg. (Δ vs. SFT)	-	-	+1.43	**+2.37**	-	-	+0.81	**+1.69**	-	-	+1.02	**+1.77**

Table 7: Generalization of SPACE across Olmo and Llama model families.

Benchmark	Olmo-3-7B-Instruct-SFT				Llama-3.1-8B-Instruct			
	Base	SFT	MonoSoup	SPACE	Base	SFT	MonoSoup	SPACE
In-domain (ID)								
MATH500	60.80	87.93	88.67	**89.33**	39.07	49.20	52.80	**53.33**
AIME25	3.33	**35.00**	30.83	34.38	0.21	**1.46**	1.04	1.25
AIME24	6.67	41.25	38.33	**41.46**	1.88	1.67	**2.50**	2.29
OlympiadBench	25.88	55.26	55.75	**56.79**	10.47	18.96	19.46	**21.04**
Minerva Math	25.74	33.95	34.19	**34.56**	15.56	18.01	20.47	**21.57**
AMC23	37.03	79.53	80.94	**82.19**	14.69	26.09	27.50	**30.16**
Avg. (Δ vs. SFT)	-	-	-0.70	**+0.97**	-	-	+1.40	**+2.38**
Out-of-domain (OOD)								
MMLU-Pro	42.70	44.40	43.20	**44.90**	43.50	40.80	44.40	**44.80**
GPQA-D	28.79	22.22	25.59	**25.93**	23.74	15.49	18.86	**19.87**
LCB v2	17.09	**22.44**	21.98	21.14	18.92	8.41	11.61	**12.20**
IFEval	83.57	79.26	78.90	**82.49**	81.89	29.14	34.65	**36.81**
AlpacaEval	0.28	0.34	0.34	**0.38**	1.17	-1.87	-0.83	**-0.38**
HaluEval	54.23	55.47	**55.83**	55.20	62.73	63.37	63.77	**63.97**
Avg. (Δ vs. SFT)	-	-	+0.29	**+0.99**	-	-	+2.85	**+3.66**

C.2 Additional Results on Model Families and Scales

We provide additional results for the model generalization experiments in Section 5.5. In addition to the models reported in the main text, we include Qwen3-4B-Base and compare SPACE with MonoSoup across all evaluated models. Tables 6 and 7 show that SPACE consistently improves the average ID and OOD performance of the corresponding SFT checkpoints across all evaluated models. The gains remain consistent across the evaluated Qwen scales and extend to the Olmo and Llama model families. Moreover, SPACE achieves larger average improvements than MonoSoup on both ID and OOD benchmarks for every evaluated model, further demonstrating its applicability.

C.3 Performance at Intermediate SFT Checkpoints

Setup. We further examine whether SPACE remains effective when applied before the end of SFT. For the Qwen2.5-7B mathematical SFT experiment, one epoch corresponds to approximately 390 optimization steps. We save intermediate checkpoints at 100, 200, and 300 steps along the same training trajectory, in addition to the final 390-step checkpoint. SPACE is applied independently to each checkpoint using the same default configuration as in the main experiments. We evaluate all checkpoints on the same six ID and six OOD benchmarks described in Section 5.1. For clarity, we report the average performance change relative to the final 390-step SFT checkpoint.

Results. As shown in Figure 3, the benefits of SPACE emerge early during SFT. After only 100 training steps, SPACE already outperforms the final 390-step SFT checkpoint by 1.28 points on

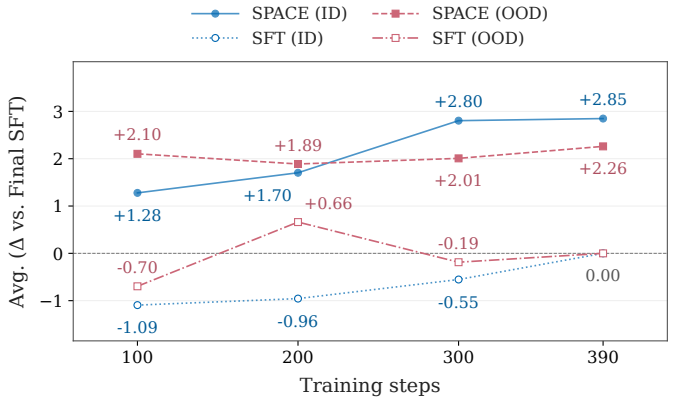


Figure 3: Performance of SFT and SPACE at different stages of the Qwen2.5-7B training trajectory. We report the average change over the six ID and six OOD benchmarks relative to the final 390-step SFT checkpoint. The 390-step checkpoint corresponds to one epoch of SFT.

ID benchmarks and 2.10 points on OOD benchmarks, whereas the corresponding 100-step SFT checkpoint remains 1.09 and 0.70 points below the final SFT checkpoint, respectively. As training proceeds, the ID improvement from SPACE further increases, reaching 2.85 points at 390 steps, while the OOD improvement remains relatively stable throughout training. These results show that SPACE does not rely on a fully trained SFT checkpoint and can provide substantial gains even at an early stage of adaptation.

C.4 Robustness to SFT Training Hyperparameters

Setup. We further examine the robustness of SPACE to different SFT training hyperparameters. Using Qwen2.5-7B fine-tuned on NuminaMath-CoT, we vary the batch size, learning rate, and number of training epochs while keeping the remaining settings fixed. For all experiments, SPACE uses the same default configuration as in the main experiments.

Results. Tables 8, 9, and 10 show that SPACE yields positive average gains over the corresponding SFT checkpoints across all evaluated training hyperparameters. Although the magnitude of the gains varies across settings, the improvements consistently extend to both ID and OOD performance. These results suggest that SPACE is robust to variations in SFT optimization, rather than being specific to a particular choice of training hyperparameters.

C.5 Training-Time vs. Post-Hoc Projection

We further compare different ways of incorporating the projection used by SPACE into SFT. In addition to the default post-hoc SPACE, we consider a projection-based training loss that penalizes updates within the pre-SFT principal subspace, as well as an online variant that applies the SPACE projection every K optimization steps during training. We evaluate $K = 1$ and $K = 100$, and include RPSFT as a related training-time regularization baseline. RPSFT regularizes the update jointly within the leading left and right singular subspaces, whereas our projection loss only penalizes updates along the leading right singular subspace. All experiments use Qwen2.5-7B trained on NuminaMath-CoT with the same training and evaluation settings as the main experiments.

As shown in Table 11, the default post-hoc SPACE achieves the largest average gains on both ID and OOD benchmarks. Training-time projection loss improves OOD performance but provides little average ID gain, while online projection yields only modest improvements for both values of K. These results suggest that allowing SFT to learn its update first and calibrating it afterward is more effective than constraining or repeatedly projecting the update during optimization, which may interfere with the learning process.

Table 8: Performance of SPACE across SFT batch sizes for Qwen2.5-7B fine-tuned on NuminaMath-CoT. The learning rate is fixed at 5×10^{-5}, and training lasts one epoch. Projection uses $\rho = 0.5$, $\alpha = 1$, and all weight matrices. Bold marks the best score in each pair. † denotes the hyperparameters we used in the main experiment.

Benchmark	Batch size 128		Batch size 256†		Batch size 512	
	SFT	SPACE	SFT	SPACE	SFT	SPACE
In-domain (ID)						
MATH500	43.40	**48.88**	44.84	**49.36**	45.31	**48.45**
AIME25	0.83	**1.04**	0.83	**1.46**	0.42	**0.83**
AIME24	0.21	**2.08**	1.04	**1.46**	0.63	**1.25**
OlympiadBench	13.06	**18.49**	14.08	**17.72**	14.31	**17.70**
Minerva Math	12.39	**15.72**	12.68	**15.26**	12.71	**14.36**
AMC23	17.34	**25.31**	19.53	**24.84**	18.44	**22.34**
Avg. (Δ vs. SFT)	-	**+4.05**	-	**+2.85**	-	**+2.19**
Out-of-domain (OOD)						
MMLU-Pro	38.42	**43.28**	38.05	**42.03**	36.09	**39.30**
GPQA-D	25.19	**26.07**	23.36	**27.27**	23.74	**23.93**
LCB v2	24.92	**27.40**	26.22	**28.70**	25.51	**29.09**
IFEval	44.36	**46.04**	42.57	**44.60**	38.49	**41.97**
AlpacaEval	-0.26	**-0.25**	-0.14	**-0.10**	-0.02	**0.02**
HaluEval	46.60	**48.76**	46.75	**47.85**	46.72	**47.42**
Avg. (Δ vs. SFT)	-	**+2.01**	-	**+2.26**	-	**+1.87**

Table 9: Performance of SPACE across SFT learning rates for Qwen2.5-7B fine-tuned on NuminaMath-CoT. The batch size is fixed at 256, and training lasts one epoch. Projection uses $\rho = 0.5$, $\alpha = 1$, and all weight matrices. Bold marks the best score in each pair. † denotes the hyperparameters we used in the main experiment.

Benchmark	LR 2e-5		LR 5e-5†		LR 1e-4	
	SFT	SPACE	SFT	SPACE	SFT	SPACE
In-domain (ID)						
MATH500	**45.60**	44.80	44.84	**49.36**	39.41	**49.21**
AIME25	**0.42**	0.00	0.83	**1.46**	0.63	**1.25**
AIME24	**1.67**	1.46	1.04	**1.46**	1.04	**1.25**
OlympiadBench	**14.83**	14.29	14.08	**17.72**	10.98	**18.18**
Minerva Math	**12.98**	10.27	12.68	**15.26**	8.39	**14.89**
AMC23	17.50	**22.50**	19.53	**24.84**	16.09	**21.25**
Avg. (Δ vs. SFT)	-	**+0.05**	-	**+2.85**	-	**+4.92**
Out-of-domain (OOD)						
MMLU-Pro	35.26	**40.09**	38.05	**42.03**	38.18	**42.14**
GPQA-D	25.57	**28.16**	23.36	**27.27**	24.18	**25.32**
LCB v2	29.22	**30.53**	26.22	**28.70**	12.46	**19.96**
IFEval	48.08	**48.92**	42.57	**44.60**	33.33	**38.13**
AlpacaEval	**0.10**	-0.12	-0.14	**-0.10**	-0.92	**-0.60**
HaluEval	47.64	**48.66**	46.75	**47.85**	48.92	**49.38**
Avg. (Δ vs. SFT)	-	**+1.73**	-	**+2.26**	-	**+3.03**

Table 10: Performance of SPACE across SFT training epochs for Qwen2.5-7B fine-tuned on NuminaMath-CoT. The learning rate is fixed at 5×10^{-5}, and the batch size is fixed at 256. Projection uses $\rho = 0.5$, $\alpha = 1$, and all weight matrices. Bold marks the best score in each pair.

Benchmark	Epochs 2		Epochs 4		Epochs 8	
	SFT	SPACE	SFT	SPACE	SFT	SPACE
In-domain (ID)						
MATH500	53.06	**53.55**	**56.04**	**56.04**	55.49	**58.43**
AIME25	**1.88**	1.25	**2.29**	1.04	2.29	**2.50**
AIME24	2.08	**2.92**	**2.71**	**2.71**	**3.54**	3.33
OlympiadBench	19.06	**20.27**	22.77	**23.36**	23.43	**26.32**
Minerva Math	16.38	**17.03**	**19.65**	18.73	19.19	**20.68**
AMC23	26.41	**28.28**	28.12	**30.31**	28.44	**33.59**
Avg. (Δ vs. SFT)	-	**+0.74**	-	**+0.10**	-	**+2.08**
Out-of-domain (OOD)						
MMLU-Pro	47.17	**48.76**	48.86	**49.96**	43.66	**48.11**
GPQA-D	25.09	**28.03**	28.85	**30.84**	26.36	**30.93**
LCB v2	**31.70**	31.44	**30.01**	28.96	21.00	**27.59**
IFEval	46.04	**47.24**	44.84	**47.60**	**47.48**	47.36
AlpacaEval	0.15	**0.19**	**0.44**	0.40	0.36	**0.60**
HaluEval	51.22	**52.57**	**58.42**	57.30	56.33	**56.70**
Avg. (Δ vs. SFT)	-	**+1.14**	-	**+0.61**	-	**+2.68**

Table 11: Comparison of training-time and post-hoc projection on Qwen2.5-7B. "Proj. Loss" introduces projection-based regularization during SFT with weight $\lambda = 1$. "Online Proj." applies the projection every K optimization steps and at the end of training. SPACE denotes the default post-hoc projection. "Avg. (Δ vs. SFT)" denotes the average score change relative to SFT. **Bold** denotes the best score excluding the Base model.

Benchmark	Base	SFT	RPSFT ($\lambda = 1$)	Proj. Loss ($\lambda = 1$)	Online Proj. ($K = 1$)	Online Proj. ($K = 100$)	SPACE (post-hoc)
In-domain (ID)							
MATH500	28.79	44.84	45.98	45.66	46.08	45.54	**49.36**
AIME25	0.42	0.83	1.04	0.63	0.42	0.63	**1.46**
AIME24	1.65	1.04	**1.46**	1.04	0.83	1.25	**1.46**
OlympiadBench	7.69	14.08	14.40	13.89	14.54	14.73	**17.72**
Minerva Math	9.26	12.68	13.01	12.73	12.80	12.64	**15.26**
AMC23	15.65	19.53	21.72	19.38	19.69	20.78	**24.84**
Avg. (Δ vs. SFT)	-	-	+0.77	+0.06	+0.23	+0.43	**+2.85**
Out-of-domain (OOD)							
MMLU-Pro	40.33	38.05	34.67	37.50	37.59	37.83	**42.03**
GPQA-D	26.77	23.36	24.15	25.73	24.59	23.48	**27.27**
LCB v2	28.96	26.22	29.03	**29.75**	28.44	27.40	28.70
IFEval	42.57	42.57	44.48	**46.76**	43.41	44.00	44.60
AlpacaEval	-1.13	-0.14	-0.03	**0.05**	-0.14	-0.05	-0.10
HaluEval	46.86	46.75	46.86	47.42	45.64	46.62	**47.85**
Avg. (Δ vs. SFT)	-	-	+0.39	+1.73	+0.45	+0.41	**+2.26**