GenSI

Spectral Rewiring for Exploration, Purification, and Model Merging

Zhilong Zhang[‡,1,2,*], Hongli Yu[1,2,*], Huan-ang Gao[1,2], Hanlin Wu[1,2], Yuxuan Song[1,2], Wei-Ying Ma[1,2], Ya-Qin Zhang[1,2], Hao Zhou[†,1,2]

[1]SIA-Lab of Tsinghua AIR and ByteDance Seed
[2]Institute for AI Industry Research (AIR), Tsinghua University

[*]Equal Contribution, [†]Corresponding Authors, [‡]Project Lead

arXiv:2607.03065v1 [cs.LG] 3 Jul 2026

Abstract

Reinforcement learning has become a standard post-training recipe for large language models, but dense full-parameter updates create two deployment-relevant bottlenecks: suppressed reasoning performance, often reflected by premature saturation of test-time scaling, and interference when consolidating multiple capabilities through multi-domain training or model merging. We show that the reasoning-effective component of these updates is largely concentrated in the base model's spectral space, motivating Subspace-Aligned Rewiring (SAR), a post-hoc editing method that retains this spectral core while removing orthogonal components. SAR therefore preserves reasoning gains and filters residual update directions that suppress performance or amplify cross-domain interference. Across several model families and scales, SAR **extracts** compact reasoning cores using as little as $\sim$0.58% of total parameters: it preserves over 99% of post-training performance and improves high-k exploration in mathematical reasoning, and generalizes to agentic coding by improving six of seven open benchmarks on an in-house model. SAR also **purifies** mixed-domain training updates by releasing suppressed coding capability while maintaining math reasoning and instruction following. It further enables **model merging** across experts, yielding cross-domain generalization that surpasses previous merging baselines and even the best single-domain experts. Overall, SAR shows that extracting reasoning-effective updates from parameter geometry can serve as a training-free mechanism to improve reasoning and multi-domain performance.

1 Introduction

Reinforcement learning has become a central post-training paradigm for improving large language models. By optimizing models against task-level feedback–such as mathematical answers, code execution results, and success signals in agentic workflows–recent systems can improve capabilities in mathematics, coding, and broader interactive settings [1–4].

Despite its empirical success, the "black-box" nature of full-parameter RL post-training offers limited control over the induced parameter update, creating two practical bottlenecks for large-scale reasoning models. *First*, it can induce **suppressed reasoning performance**, often visible as saturation of test-time scaling: reward optimization may concentrate the policy around a narrow set of high-reward trajectories, reducing the diversity of candidate solutions and limiting the ability to solve harder instances [5–8]. *Second*, it can induce **cross-domain interference** during capability consolidation: updates from multiple domains, such as mathematics,

Correspondence: zhouhao@air.tsinghua.edu.cn

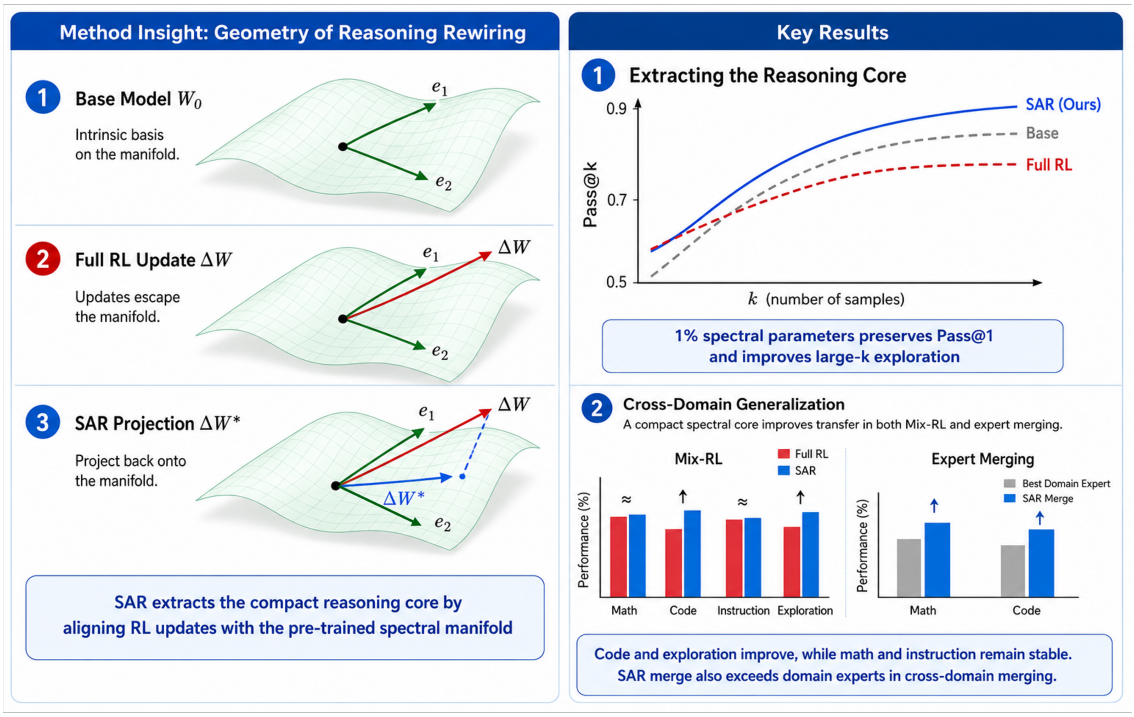


Figure 1 Overview of Subspace-Aligned Rewiring (SAR). SAR identifies a compact reasoning core by projecting reinforcement learning updates back onto the pretrained model's spectral manifold. This alignment preserves the intrinsic reasoning basis of the base model while removing off-manifold update components introduced by full RL. The resulting spectral update retains Pass@1 performance, improves large-k exploration, and transfers more robustly across domains, including Mix-RL and expert merging.

competitive programming, and instruction following, may occupy incompatible directions in parameter space, so joint RL or weight-space merging can improve one capability while suppressing another [9–11]. This motivates a key question: can we identify a coordinate system that separates reasoning-effective update components from residual directions, thereby recovering suppressed reasoning performance and improving multi-domain generalization?

We propose a coordinate system for studying reasoning updates from a geometric perspective. Outcome-reward RL often elicits knowledge and skills that already exist in the base model, so the base model and the RL model should remain **closely related in an appropriate geometric manifold**. We propose that this manifold is given by the base model's SVD coordinates. In this view, the singular vectors of the base weights represent pretrained "atomic skills," while the diagonal singular-value matrix specifies how these skills are originally connected. RL keeps this pretrained basis largely fixed, but changes the middle matrix from a diagonal map into a non-diagonal **rewiring matrix**. This rewiring reconnects atomic skills that were previously isolated, allowing the model to compose existing knowledge into new reasoning circuits.

This perspective leads to Subspace-Aligned Rewiring (SAR), a post-hoc model editing method that projects a dense reasoning update onto the pretrained spectral manifold. The resulting spectral update supports three application-facing uses. **Extraction**: SAR isolates a compact reasoning-effective component that preserves post-training performance across 1.5B–32B public models, improves high-k exploration, and improves open agentic coding benchmarks on an in-house model. **Purification**: SAR filters residual directions in Mix-RL updates, reducing domain conflict and releasing suppressed cross-domain capability. **Merging**: SAR places expert updates in a common pretrained coordinate system, improving multi-domain model merging beyond previous baselines and even the best single-domain experts. Our contributions are organized as four takeaways:

- **Compact Spectral Rewiring Extracts Effective Post-Training Signal.** Across 1.5B–32B reasoning models, SAR preserves full-RL-level math performance with as little as ∼**0.58%** of total parameters and **more than 99% of peak** $Pass@1$ **preserved**. Beyond math, the same top-1% spectral projection improves six of seven open agentic coding benchmarks on an in-house model, with a **+2.52% improvement on average**.
- **Spectral Projection Restores Useful Exploration.** SAR improves exploration beyond the base model and **surpasses unconstrained full-parameter RL baselines in Pass@k scaling**, crossing the full-RL curve

at small k and expanding the set of AIME problems reachable through repeated sampling.

- **Spectral Projection Improves Mix-RL.** In Mix-RL, SAR reveals **suppressed cross-domain capability**. On OLMo-3.1-32B-Think, SAR improves competitive coding (+1.48% on LiveCodeBench v5 and +0.95% on v6), improves mathematical exploration (AIME 25 $Pass@16$ from 88.33% to 91.71%), and maintains strong instruction-following performance.
- **Expert Merging Surpasses Domain Experts.** In math-code expert merging, SAR produces merged models that **surpass the best single-domain experts** at both 1.5B and 14B scales, improving AIME and LiveCodeBench simultaneously.

2 The Geometric Framework of Reasoning Elicitation

In this section, we construct a first-principles geometric framework to formalize how outcome-reward RL elicits reasoning capabilities from a pretrained model. We first establish the spectral subspace of the base model as a functional basis for reasoning. We then introduce Subspace-Aligned Rewiring (SAR), a projection-based algorithm designed to extract reasoning-effective updates geometrically. Finally, we provide a mechanistic analysis of how reasoning can emerge through spectral rewiring.

2.1 The Functional Basis: SVD in Forward Propagation

Consider a linear layer within a pretrained base model, parameterized by a weight matrix $W_0 \in \mathbb{R}^{d_{out} \times d_{in}}$. The forward propagation for an input representation x is defined as $y_{base} = W_0 x$. To geometrically dissect this transformation, we apply singular value decomposition (SVD) to the weight matrix:

$$y_{base} = W_0 x = U \Sigma V^\top x = \sum_{i=1}^{r} \sigma_i u_i (v_i^\top x) \tag{1}$$

where r denotes the rank, while $V = [v_1, \ldots, v_r]$ and $U = [u_1, \ldots, u_r]$ represent the right and left singular vectors of W_0, respectively.

As shown in Equation 1, the transformation decomposes into two phases: (1) **Information Read-in**: the right singular vectors v_i act as principal feature detectors, extracting activation scores $(v_i^\top x)$ from the input; and (2) **Information Read-out**: the left singular vectors u_i define the principal output directions, weighted by the singular values σ_i and $(v_i^\top x)$.

Based on this analysis, these singular vectors constitute the **latent skills** inherent in the base model, functioning as the geometric basis to **encode input signals** and **decode output representations**. Furthermore, extensive literature in model compression and spectral analysis [12–14] suggests that the vast majority of a pretrained model's functional capacity is bottlenecked within this principal spectral subspace. Consequently, we define the **pretrained spectral manifold** of a matrix space spanned by this coordinate system:

$$\mathcal{S}_r(W_0) = \mathrm{span}(U \otimes V), \tag{2}$$

where U_r, V_r are the singular vectors of W_0 with $r = \min(d_{out}, d_{in})$. This subspace provides a compact coordinate system for the model's functional library. In the following section, we leverage this geometric foundation to introduce our core insights into the mechanisms of reasoning elicitation.

2.2 The Geometric Principle of Reasoning and Spectral Projection

We propose a **first-principles geometric hypothesis** for reasoning elicited by outcome-reward RL: **the reasoning-effective component of a full reasoning update should be recoverable within the pre-existing spectral subspace of the base model.**

Let W_{RL} denote a model obtained by full-parameter outcome-reward RL and define the empirical update

$$\Delta W = W_{RL} - W_0. \tag{3}$$

The update ΔW is trained without geometric constraints, so it can be orthogonally decomposed into two components:

$$\Delta W = \Delta W^* + \Delta W_\perp, \tag{4}$$

where $\Delta W^* \in \mathcal{S}_k(W_0)$ is the subspace-aligned update associated with measured reasoning gains, and $\Delta W_\perp$ is the residual outside the pretrained spectral manifold.

Using the projection operators $P_U = UU^\top$ and $P_V = VV^\top$, we formalize the extraction of the reasoning component as:

$$\begin{aligned} \Delta W^* &= P_U \Delta W P_V \\ &= (UU^\top)\Delta W(VV^\top) \\ &= U(U^\top \Delta W V)V^\top \\ &= UMV^\top \end{aligned} \tag{5}$$

where

$$M = U^\top \Delta W V \in \mathbb{R}^{r \times r}$$

is the **Rewiring Matrix**. Unlike the diagonal matrix Σ, M is generically dense, capturing the **cross-dimensional interactions** between pretrained singular vectors.

This process, summarized in Algorithm 1, extracts a compact reasoning-effective representation ΔW^* while discarding the residual component $\Delta W_\perp$. In practice, we first extract the **top-k low-rank component of** ΔW and then project it onto the pretrained SVD subspace, thereby combining parameter compression with spectral alignment to identify the compact geometry responsible for reasoning. Appendix C specifies the exact parameter groups, precision settings, and evaluation configuration used in our implementation.

Algorithm 1 Subspace-Aligned Rewiring (SAR)

Input: Base model W_0, RL model W_{RL}, target rank k
Output: Projected reasoning model W_{SAR}

1: **Spectral Extraction:** Compute top-k SVD of $W_0 = U\Sigma V^\top$
2: **Isolate Update:** $\Delta W \leftarrow W_{RL} - W_0$
3: **Low-Rank Extraction:** Extract the top-k component ΔW_k from ΔW
4: **Geometric Projection:** Extract rewiring matrix $M \leftarrow U^\top \Delta W_k V$
5: **Reconstruction:** $\Delta W^* \leftarrow UMV^\top$ {Aligns the compact update with the pretrained spectral subspace}
6: **return** $W_{SAR} \leftarrow W_0 + \Delta W^*$

2.3 Mechanistic View: Reasoning as Rewiring

To analyze how the extracted rewiring matrix M can capture complex reasoning behavior, we examine the forward propagation of the SAR-projected model:

$$W^* = W_0 + \Delta W^* = U(\Sigma + M)V^\top$$

For an input x, the rewired forward pass is:

$$\begin{aligned} y_{rewired} &= U(\Sigma + M)V^\top x \\ &= \sum_{i=1}^{r} \left[\underbrace{(\sigma_i + M_{ii})(v_i^\top x)}_{\text{Rescaling}} + \underbrace{\sum_{j \neq i} M_{ij}(v_j^\top x)}_{\text{Rewiring}} \right] u_i \end{aligned} \tag{6}$$

Equation 6 reveals the dual nature of RL updates within the spectral subspace:

- **Diagonal Elements (M_{ii}):** These represent the rescaling of isolated, pre-existing capabilities. While they amplify certain skills, they do not establish new logical connections.
- **Off-Diagonal Elements (M_{ij}):** These provide a geometric mechanism for reasoning. They enable an output direction u_i to be synthesized by **multiple, distinct input features** v_j **($j \neq i$)**. This transition from **one-to-one associative mapping to many-to-one logical synthesis** allows the model to perform multi-conditional deductions.

The algebraic distinction between diagonal and off-diagonal dynamics in M provides a structural analogy to relational reasoning [15–17]. Diagonal scaling M_{ii} resembles unary associative memory: a single latent cue $v_i^\top x$ activates an isolated output concept along u_i. Off-diagonal entries M_{ij}, by contrast, instantiate cross-component integration, routing evidence from one latent premise direction v_j toward a different conclusion direction u_i. When multiple such terms are active, the network no longer performs simple point-to-point retrieval; it **composes independent latent premises into a shared output representation**. This offers a mechanistic interpretation of why off-diagonal spectral rewiring can support reasoning elicitation: ΔW^* preserves the geometric structure associated with relational integration, while removing update components outside this spectral computation. Appendix B provides an illustrative example.

Takeaway.

- The pretrained SVD basis provides a functional coordinate system for reasoning updates, turning a dense weight change into an editable spectral object.
- In this coordinate system, post-training can be viewed as changing how pretrained skill directions are connected, rather than replacing the model's learned basis.
- This perspective gives a practical interface for downstream operations: compression, filtering, and merging can be applied to the rewiring matrix instead of the full parameter update.

In the subsequent sections, we empirically validate these theoretical insights through extensive experiments, demonstrating that SAR preserves peak reasoning performance, improves useful exploration and open agentic coding performance on an in-house model, and strengthens cross-domain generalization.

3 Extracting the Reasoning Core: Single-Domain Validation

In this section, we demonstrate that SAR can extract a compact spectral update that preserves full post-training performance on mathematical reasoning tasks, improves high-k exploration, and boosts open agentic coding benchmark performance on an in-house model. Together, these results show that extraction is not merely a compression result, but a practical post-hoc editing operation for improving downstream utility.

3.1 Spectral Rewiring Preserves Reasoning Gains

To assess the compactness of spectral rewiring, we apply SAR to extract the reasoning-effective component ΔW^* from open-source RL models and compare the projected models with their full RL counterparts. We perform a rank sweep and report the smallest retained spectral rank that preserves full-RL performance within evaluation variance. Appendix E and Appendix F further compare SAR with no-projection and alternative projection controls.

Experimental Setup. We evaluate a diverse set of model families and scales, ranging from 1.5B to 32B parameters. Specifically, we consider DeepScaleR (1.5B) [18], POLARIS (4B) [19], and OLMo-3.1-32B-Think [20] as representative strong RL reasoning models. We also evaluate OLMo-3-7B-Base and its RL-trained counterpart to study reasoning elicitation from a base model not explicitly specialized for mathematical reasoning [20]. We use AIME 2024 and AIME 2025 as challenging mathematical reasoning benchmarks, following common practice in recent reasoning-model evaluations [2, 18, 21]. For each prompt, we sample 32 responses and report AVG@32 and Pass@32 [22]. Additional evaluation details are provided in Appendix C.

Table 1 Reasoning performance on AIME 2024 and AIME 2025 across different model scales and training recipes. SAR reports the smallest retained spectral rank found by rank sweep that preserves full-RL-level performance within evaluation variance.

Model Size	Method	AIME 24		AIME 25	
		AVG@32	Pass@32	AVG@32	Pass@32
The SOTA Training Recipes					
1.5B	Base: Deepseek-distill-qwen2.5	31.67%	80.00%	23.96%	53.33%
	RL: DeepscaleR	**40.31%**	**76.67%**	**29.58%**	**60.00%**
	Projected (Ours): 1%	40.21%	**76.67%**	28.96%	**60.00%**
4B	Base: Qwen3 4B	72.50%	90.00%	63.44%	86.67%
	RL: Polaris	**78.75%**	**93.33%**	**75.21%**	90.00%
	Projected (Ours): 10%	78.02%	**93.33%**	72.21%	**93.33%**
32B	Base: OLMo3-32B-think-dpo	74.48%	93.33%	70.83%	90.00%
	RL: OLMo-3.1-32B-Think	**79.56%**	**93.33%**	**75.76%**	90.00%
	Projected (Ours): 1%	78.88%	**93.33%**	75.12%	**93.33%**
Emergent Reasoning Ability from the Base Model					
7B	Base: OLMo3-7B-Base	22.08%	70.00%	19.58%	60.00%
	RL: OLMo-3.1-7B-RL-Zero-Math	**50.21%**	76.67%	**36.88%**	**76.67%**
	Projected (Ours): 30%	49.02%	**80.00%**	36.56%	**76.67%**

Results. As detailed in Table 1, SAR preserves nearly all of the measured reasoning gains obtained by full RL across model families and scales. Across AIME 2024 and AIME 2025, the projected models match their corresponding full RL models within a small margin under AVG@32, while often matching or slightly improving Pass@32. These results suggest that the evaluated RL-induced reasoning improvements are **largely concentrated in the pretrained spectral subspace**, rather than requiring the full unconstrained parameter update. The required rank also reflects the role of the starting model: strong post-trained or distilled reasoning recipes require only a small retained rank, whereas eliciting reasoning directly from OLMo3-7B-Base requires a larger 30% rank, suggesting that RL must modify a broader portion of the pretrained spectral coordinates when the starting model is less specialized for reasoning.

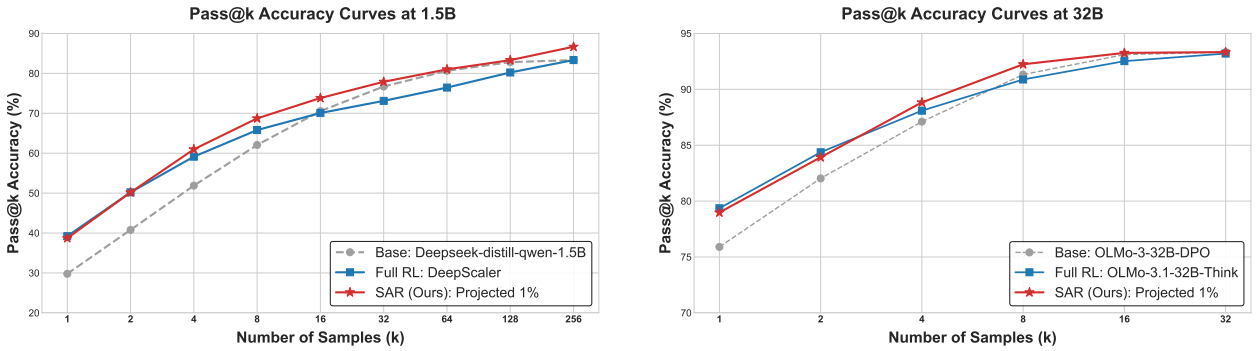
The Density of Reasoning-Effective Parameters. The rank sweep identifies a small saturation point for the strong reasoning recipes: retaining only 1% to 10% of the pretrained spectral rank recovers full-RL-level reasoning performance for the evaluated models. As shown in Table 2, the corresponding rewiring matrix M is much smaller than both the full model and the reconstructed update ΔW^*, with storage ratios as low as $\sim$0.58% and $\sim$0.64% of total parameters for the 1.5B and 32B models, respectively. This indicates that the reasoning-effective part of the RL update is highly concentrated in a compact spectral representation. Rather than suggesting that all discarded parameters are useless in general, these results show that a large fraction of

the unconstrained RL update is unnecessary for preserving the measured AIME reasoning gains.

Table 2 Parameter efficiency and compression ratios of SAR across different model scales. The intrinsic rewiring matrix M is highly compact relative to the total parameter count.

Model Name	Total Params	Projected Rank	Parameter Update (ΔW^*)	Rewiring Matrix (M)	Compression Ratio (M / Total)
DeepScaleR	1.54B	1%	16.1M	9.0M	∼0.58%
POLARIS	4.00B	10%	552.0M	310.0M	∼7.75%
OLMo-3.1-32B-Think	32.00B	1%	311.0M	204.0M	∼0.64%

3.2 Spectral Rewiring Improves High-k Exploration



(a) DeepScaleR-1.5B **(b)** OLMo-3.1-32B-Think

Figure 2 Pass@k scaling curves on AIME 2024. Unconstrained RL baselines exhibit early saturation. In contrast, SAR models extract a compact reasoning-effective update, improving useful exploration and scaling momentum.

RL post-training often improves Pass@1 but can saturate under larger sampling budgets, limiting the benefit of test-time scaling. Under our spectral view, this suggests that the full RL update may contain both reasoning-effective directions and residual directions that narrow useful exploration. SAR tests this hypothesis by retaining the compact spectral rewiring component while removing the orthogonal residual update.

Experimental Setup. We evaluate this effect on two model families: DeepScaleR (1.5B) and OLMo-3.1-32B-Think. For each AIME 2024 problem, we generate 256 rollouts and compute Pass@k for $k \in [1, 128]$ using the same decoding configuration across all compared models [7, 22]. To complement the Pass@k curves, we also report a coverage-based metric following prior reasoning-RL analyses: a problem is counted as covered if at least one rollout among a fixed sampling budget produces the correct final answer. This metric measures useful exploration, since it counts only diversity that leads to a verified solution [23].

Results. Figure 2 shows the Pass@k scaling curves on AIME 2024. The full RL baselines exhibit early saturation: their performance improves at small k but gains diminish at larger sampling budgets. In contrast, SAR-projected models continue to benefit from additional samples and overtake the corresponding full RL baselines at small-to-moderate k values. For DeepScaleR and OLMo-3.1-32B-Think, SAR crosses the full RL curve at $k = 2$ and $k = 4$, respectively, and remains better at larger k.

Importantly, SAR preserves the average reasoning gains of RL while achieving **stronger large-k scaling**. The same trend appears at the problem-coverage level: with 256 rollouts on AIME 2024, SAR covers 26/30 problems, compared with 25/30 for the full RL model (Appendix Table 5). Diagnostic controls further show that this gain is not explained by low-rank extraction alone, as no-projection, random, diagonal-only, and off-diagonal-only variants perform worse than SAR (Appendix E and Appendix F).

3.3 Additional In-House Agentic Coding Validation

To evaluate SAR as an application-oriented editing method beyond public mathematical reasoning benchmarks, we further test it on the in-house model using open agentic coding benchmarks, as shown in Table 3. Starting from the same coding post-training delta, the top-1% spectral-rank projection improves six of seven open agentic coding benchmarks, with a **+2.52% improvement on average** and gains up to 25.10%. This result shows that spectral rewiring can extract useful post-training signal and transfer to applied coding workflows.

Table 3 Additional single-domain validation on the in-house model using open agentic coding benchmarks. SAR uses a top-1% spectral-rank projection of the coding post-training delta.

Benchmark	Agent	Baseline (In-house)	SAR (1% Rank)	Relative Change
SWE-Bench-Pro[24]	CodeAct[25]	0.297	**0.327**	**+10.21%**
SWE-Bench[26]	ClaudeCode	0.580	**0.590**	**+1.72%**
SWE-Bench[26]	CodeAct[25]	0.566	**0.574**	**+1.41%**
MSWE-Bench[27]	ClaudeCode	0.255	**0.313**	**+22.86%**
MSWE-Bench[27]	CodeAct[25]	0.271	0.270	-0.33%
TerminalBench 2.0[28]	Terminus[28]	0.194	**0.243**	**+25.10%**
TerminalBench 1.0[28]	Terminus[28]	0.323	**0.344**	**+6.73%**

Takeaway.

- SAR preserves full-RL-level math reasoning across model scales using a highly compact spectral update, showing that the reasoning-effective component of RL is strongly concentrated in the pretrained spectral geometry.
- The same compact update also improves high-k reasoning behavior without retraining the model, changing the reward objective, or adding new rollouts.
- The validation on the in-house model shows that SAR can also improve applied coding performance beyond public math benchmarks.
- The main results and ablations show that the reasoning-effective parameters primarily rewire the pretrained spectral space, providing parameter-level geometric evidence that RL incentivizes capabilities already latent in the base model.
- These results suggest a training-free route to improve useful exploration after RL, and point toward future training recipes that optimize directly within the reasoning-effective spectral geometry.

4 Cross-Domain Generalization via Spectral Reasoning Rewiring

Having shown that SAR preserves single-domain reasoning gains and improves large-k exploration, we next study cross-domain generalization. Training or merging models across mathematics, coding, and instruction following can introduce domain-specific update components that interfere with each other. Under our geometric view, the fundamental reasoning component should lie in the base spectral manifold; SAR therefore projects reasoning updates onto this manifold to retain shared structure and filter incompatible residual directions. We evaluate this principle in two settings: jointly trained multi-domain RL models (§4.1) and post-hoc merging of independently trained domain experts (§4.2).

4.1 Purification: Reducing Domain Interference in Mix-RL

We first evaluate this projection principle in a jointly trained Mix-RL model, where multiple capabilities are optimized within the same full-parameter update.

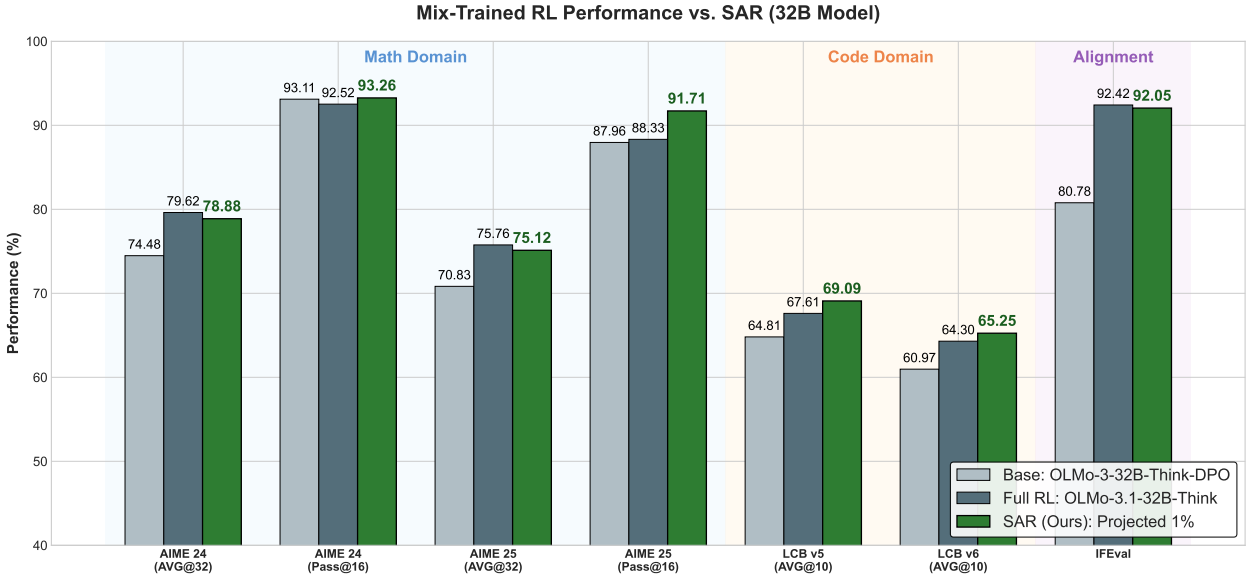


Figure 3 Cross-domain performance evaluation on the OLMo-3-32B scale. By extracting reasoning-effective parameters, SAR reduces cross-domain interference, improves LiveCodeBench v5/v6, maintains instruction following (IFEval) and math reasoning (AIME 24), and enhances exploration capability.

Experimental Setup. We evaluate the OLMo-3-32B series, which achieves state-of-the-art open-source performance. We use OLMo-3-32B-DPO as the base model and OLMo-3.1-32B-Think as the Mix-RL baseline, which is jointly trained on math, coding, instruction following, and chat. We apply SAR with the same top-1% spectral-rank setting used in the single-domain experiments. Evaluation spans rigorous mathematical reasoning (AIME 24/25), algorithmic coding (LiveCodeBench v5/v6) [29], and general instruction following (IFEval) [30].

Results. As illustrated in Figure 3, extracting the reasoning-effective parameters via SAR improves performance on competitive coding while maintaining capacity in math and instruction following.

- *Coding:* The projected model outperforms the full RL baseline on LCB v5 (from 67.61% to 69.09%) and v6 (from 64.30% to 65.25%). We observe consistent improvements in Pass@k on LCB v5 and v6; Appendix E provides an additional multi-domain control on LCB v5.
- *Math:* The peak math reasoning performance is preserved with a marginal variance of less than 1% in AVG@32, while exploration capability is significantly improved (e.g., from 88.33% to 91.71% on AIME 25 Pass@16).
- *Instruction Following:* General instruction-following capability is also retained, with IFEval changed only slightly from 92.42% to 92.05%.

With less than 1% of the model parameters represented in the rewiring matrix M, SAR improves LiveCodeBench, an execution-sensitive coding benchmark, while preserving instruction following and mathematical reasoning. SAR also improves math and coding Pass@k at larger sampling budgets, indicating that the projected model retains the accuracy gains of RL while recovering additional exploration capacity. These improvements are difficult to explain as lossy compression alone: reducing the update would typically be expected to preserve or degrade downstream performance, rather than improve a challenging held-out domain. We therefore interpret the LiveCodeBench gain as evidence that the full Mix-RL update contains domain-specific residual directions that interfere across tasks, while SAR exposes a compact spectral rewiring that is more compatible across domains.

4.2 Merging: Improving Cross-Domain Expert Merging

We next evaluate whether spectral rewiring can improve cross-domain model merging, focusing on math and code experts. Directly merging independently trained domain experts often degrades performance because their raw task deltas contain incompatible residual directions. SAR instead filters one expert through the pretrained spectral geometry before merging it with another.

Experimental Setup. We consider a scenario involving two domain experts independently derived via RL from the same base model. At the 1.5B scale (Base: DeepSeek-Distill-Qwen-1.5B) [31, 32], we use DeepScaleR (Math) and Archer-Code (Code) [33]. At the 14B scale (Base: DeepSeek-Distill-Qwen-14B), we pair OpenReasoner (Math) [21] with DeepCoder (Code) [34]. We use SAR to extract a compact mathematical reasoning rewiring from the math expert and merge these reasoning-effective parameters with the code expert. We benchmark against established weight interpolation methods including Task Arithmetic (TA), TIES, and DARE+TIES.

Results. Table 4 reports zero-shot math-code merging results. Task Arithmetic directly averages the full math and code RL updates, and therefore serves as a simple averaging baseline. Its degradation, especially on the reasoning metric, suggests that naively merging full RL updates can introduce cross-domain interference.

At both 1.5B and 14B scales, SAR produces merged models that **exceed the best single-domain experts** on the primary math and code AVG metrics. At 1.5B, SAR improves AIME AVG@32 to 43.44% and LCB AVG@8 to 32.25%, surpassing the best math expert (40.31%) and code expert (31.80%) simultaneously. At 14B, SAR again surpasses the best math expert on AIME AVG@32 (74.38% vs. 74.27%) and improves beyond the code expert on LCB AVG@8 (63.40% vs. 61.00%), while matching the best AIME Pass@32. Although DARE+TIES obtains a slightly higher LCB score, SAR provides stronger balanced transfer in this setting by improving coding performance while preserving the math expert's advantage.

These results indicate that the reasoning rewiring extracted by SAR is more compatible for merging than the full RL update. In contrast to magnitude-based merging methods, SAR filters the update through the pretrained spectral geometry before merging, reducing interference with the code expert.

Takeaway.

- In Mix-RL, SAR releases suppressed cross-domain capability, improving coding and large-k exploration while preserving math reasoning and instruction following.
- In expert merging, SAR produces merged models that surpass the best single-domain experts on the primary math and code AVG metrics at both 1.5B and 14B scales.
- These gains are consistent with our core view: the transferable reasoning component is concentrated in the pretrained spectral manifold, while residual directions outside this geometry can amplify domain interference.
- Spectral projection therefore makes reasoning updates more compatible across domains by retaining the shared rewiring component and filtering less compatible task-specific directions.
- This suggests a practical path for building stronger multi-domain reasoning models: post-training updates can be purified before deployment or merged across experts without directly combining all dense task deltas.

5 Discussion

Across single-domain extraction, Mix-RL purification, and expert merging, SAR preserves or improves key downstream performance after projecting dense RL updates into the pretrained spectral coordinates. These results support our central insight: a large part of the reasoning-effective change induced by RL is not arbitrary parameter drift, but compact rewiring of capabilities already represented in the reference model's spectral geometry. This insight also defines the scope of the method. SAR is expected to be most effective when the

Table 4 Cross-Domain Generalization via Model Merging. By extracting and integrating compact reasoning-effective parameters, SAR outperforms naive interpolation and improves cross-domain transfer.

Scale	Model / Method	Math (AIME 24)		Code (LCB)
		AVG@32	Pass@32	AVG@8
1.5B	*Single-Domain References*			
	Base: Distill-Qwen-1.5B	31.67%	76.67%	23.00%
	Math Expert: DeepScaleR	**40.31%**	**76.67%**	27.20%
	Code Expert: Archer-Code	39.48%	76.67%	**31.80%**
	Best Single-Domain Expert	*40.31%*	*76.67%*	*31.80%*
	Cross-Domain Merging			
	Task Arithmetic	36.25%	63.33%	31.80%
	TIES Merging	42.08%	73.33%	31.40%
	DARE + TIES	40.05%	73.33%	31.70%
	Projected (Ours)	**43.44%**	**76.67%**	**32.25%**
14B	*Single-Domain References*			
	Base: Distill-Qwen-14B	67.71%	90.00%	54.20%
	Math Expert: OR-Zero	**74.27%**	**93.33%**	56.70%
	Code Expert: DeepCoder	71.04%	86.67%	**61.00%**
	Best Single-Domain Expert	*74.27%*	*93.33%*	*61.00%*
	Cross-Domain Merging			
	Task Arithmetic	71.04%	93.33%	54.60%
	TIES Merging	73.54%	93.33%	63.30%
	DARE + TIES	73.06%	90.00%	**64.30%**
	Projected (Ours)	**74.38%**	**93.33%**	63.40%

reference model already contains the relevant reasoning ability or domain knowledge and RL primarily elicits or reorganizes it.

The main boundary of SAR is projection compatibility: how much of an RL update can be represented as a reorganization of knowledge already encoded in the reference model. Appendix D analyzes three representative cases. In heavily optimized reasoning RL, projecting JustRL after more than 4000 training steps still recovers a strong AIME 2024 score of about 43%, comparable to DeepScaleR-level reasoning, but does not match the final RL model. This suggests that the spectral component captures substantial elicitation, while the remaining gap may reflect task-specific rewriting outside the reference manifold. In code RL, direct RL from a base model is less projection-compatible because the update may learn code formatting, tool-interface behavior, and execution-oriented conventions, rather than only recombining existing knowledge and skills. After code SFT establishes this domain knowledge, the subsequent RL update becomes more compatible with spectral projection. PPO-style training with an external critic creates a related boundary: distribution shift between the critic and actor can induce preferences that are not well aligned with the reference model's spectral manifold. Thus, projection compatibility provides a diagnostic for whether RL primarily reorganizes latent capabilities within the reference manifold or acquires behavior beyond it.

6 Conclusion

We presented a first-principles geometric view of RL post-training in large language models. By analyzing updates in the pretrained singular-vector coordinates, we show that much of the reasoning-effective change is captured by compact spectral rewiring, consistent with the view that RL reorganizes latent capabilities already represented in the base model. Subspace-Aligned Rewiring (SAR) turns this insight into a post-hoc editing method that preserves full-RL-level reasoning, improves high-k exploration, reduces multi-domain interference, and strengthens expert model merging. These results position spectral rewiring as both a lens for understanding post-training updates and a practical tool for improving their downstream utility.

References

[1] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

[2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

[3] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

[4] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. arXiv preprint arXiv:2305.20050, 2023.

[5] Hannah Rose Kirk, Bertie Vidgen, Paul Rottger, and Scott A. Hale. Understanding the effects of rlhf on llm generalisation and diversity. arXiv preprint arXiv:2310.06452, 2023.

[6] Ted Moskovitz, Archit Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca Dragan, and Stephen McAleer. Confronting reward model overoptimization with constrained rlhf. arXiv preprint arXiv:2310.04373, 2023.

[7] Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Re, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. arXiv preprint arXiv:2407.21787, 2024.

[8] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. arXiv preprint arXiv:2408.03314, 2024.

[9] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. arXiv preprint arXiv:2212.04089, 2022.

[10] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. arXiv preprint arXiv:2306.01708, 2023.

[11] 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. arXiv preprint arXiv:2311.03099, 2023.

[12] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149, 2015.

[13] Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models. arXiv preprint arXiv:2312.05821, 2023.

[14] Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression. arXiv preprint arXiv:2403.07378, 2024.

[15] Graeme S. Halford, William H. Wilson, and Steven Phillips. Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. Behavioral and Brain Sciences, 21(6): 803–831, 1998.

[16] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. arXiv preprint arXiv:1706.01427, 2017.

[17] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018.

[18] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Li, Li Erran Yang, et al. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. arXiv preprint arXiv:2502.01682, 2025.

[19] POLARIS Project. POLARIS: A post-training recipe for scaling reinforcement learning on advanced reasoning models. https://hkunlp.github.io/blog/2025/Polaris/, 2025. Accessed 2026-05-15.

[20] AI2. OLMo 3: A family of open language models. arXiv preprint arXiv:2512.13961, 2025.

[21] Jian Hu, Xibin Wu, Weixun Fu, Xinyu Chen, Chen Xu, Weizhi Zhu, Jiaxin Pei, Zixuan Zhong, Jiawei Zheng, Zheng Chen, et al. Open-reasoner-zero: An open source approach to scaling reinforcement learning on the base model. arXiv preprint arXiv:2503.24290, 2025.

[22] 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.

[23] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171, 2022.

[24] Xiang Deng, Jeff Da, Edwin Pan, Yan He, Charles Ide, Kanak Garg, Niklas Lauffer, Andrew Park, Nitin Pasari, Chetan Rane, Karmini Sampath, Maya Krishnan, Srivatsa Kundurthy, Sean M. Hendryx, Zifan Wang, Chen Bo Calvin Zhang, Noah Jacobson, Bing Liu, and Brad Kenstler. Swe-bench pro: Can ai agents solve long-horizon software engineering tasks? 2025. URL https://api.semanticscholar.org/CorpusID:281421060.

[25] Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Daniel Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. Openhands: An open platform for ai software developers as generalist agents. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, International Conference on Learning Representations, volume 2025, pages 65882–65919, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a4b6ad6b48850c0c331d1259fc66a69c-Paper-Conference.pdf.

[26] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. Swe-bench: Can language models resolve real-world github issues? In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

[27] Daoguang Zan, Zhirong Huang, Wei Liu, Hanwu Chen, Linhao Zhang, Shulin Xin, Lu Chen, Qi Liu, Xiaojian Zhong, Aoyan Li, Siyao Liu, Yongsheng Xiao, Liangqiang Chen, Yuyu Zhang, Jing Su, Tianyu Liu, Rui Long, Kai Shen, and Liang Xiang. Multi-swe-bench: A multilingual benchmark for issue resolving, 2025. URL https://arxiv.org/abs/2504.02605.

[28] Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Jenia Jitsev, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Jesse Hu, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces, 2026. URL https://arxiv.org/abs/2601.11868.

[29] Naman Jain, King Han, 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. arXiv preprint arXiv:2403.07974, 2024.

[30] 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.

[31] Qwen Team. Qwen2.5 technical report. arXiv preprint arXiv:2412.15115, 2024.

[32] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowei 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.

[33] Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr, 2025. URL https://arxiv.org/abs/2507.15778.

[34] Together AI. Deepcoder: A fully open-source 14b coder at o3-mini level. https://www.together.ai/blog/deepcoder, 2025. Accessed 2026-05-15.

[35] Xiang Yue, Zhuo Chen, and Wenhu Chen. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? arXiv preprint arXiv:2504.13837, 2025.

[36] Christian Walder and Deep Karkhanis. Pass@k policy optimization: Solving harder reinforcement learning problems. arXiv preprint arXiv:2505.15201, 2025.

[37] Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models. arXiv preprint arXiv:2508.10751, 2025.

[38] Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. arXiv preprint arXiv:2602.02710, 2026.

[39] Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. Simko: Simple pass@k policy optimization. arXiv preprint arXiv:2510.14807, 2025.

[40] 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. arXiv preprint arXiv:2106.09685, 2021.

[41] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. arXiv preprint arXiv:2402.09353, 2024.

[42] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. arXiv preprint arXiv:2404.02948, 2024.

[43] Klaudia Balazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. Lora-xs: Low-rank adaptation with extremely small number of parameters. arXiv preprint arXiv:2405.17604, 2024.

[44] Yuchen Cai, Ding Cao, Xin Xu, Zijun Yao, Yuqing Huang, Zhenyu Tan, Benyi Zhang, Guiquan Liu, and Junfeng Fang. On predictability of reinforcement learning dynamics for large language models. arXiv preprint arXiv:2510.00553, 2025.

[45] John X. Morris, Niloofar Mireshghallah, Mark Ibrahim, and Saeed Mahloujifar. Learning to reason in 13 parameters. arXiv preprint arXiv:2602.04118, 2026.

[46] Zixuan Ren, Jinliang Lu, Junhong Wu, Yang Zhao, Dai Dai, Hua Wu, Haifeng Wang, and Chengqing Zong. Enough is as good as a feast: A comprehensive analysis of how reinforcement learning mitigates task conflicts in llms. In International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=N4l4Jp50R4.

[47] JustRL Team. JustRL: Scaling a 1.5b llm with a simple rl recipe. https://iclr-blogposts.github.io/2026/blog/2026/justrl/, 2026. ICLR Blogposts 2026.

[48] Jie Wu, Haoling Li, Xin Zhang, Jiani Guo, Jane Luo, Steven Liu, Yangyu Huang, Ruihang Chu, Scarlett Li, and Yujiu Yang. X-Coder: Advancing competitive programming with fully synthetic tasks, solutions, and tests. arXiv preprint arXiv:2601.06953, 2026.

Appendix

A Related Work

Exploration and exploitation in outcome-reward RL. Recent work studies reasoning RL not only as a way to improve Pass@1, but also as a process that reshapes the exploration–exploitation trade-off of reasoning models [35]. A related line of methods redesigns the RL objective to better optimize Pass@k-style behavior or test-time scaling efficiency [36–39]. These approaches treat exploration primarily as a training-time objective design problem. SAR takes a **different view**: a completed RL update can already contain a compact high-k-effective component, but this component is entangled with residual off-manifold drift. By separating them in pretrained spectral coordinates, SAR improves Pass@k through post-hoc model editing, without changing the RL objective, collecting new rollouts, or retraining the model.

LoRA and low-dimensional RL updates. Parameter-efficient adaptation and recent analyses of RL dynamics both suggest that LLM adaptation can be highly low-dimensional [40–45]. These works use low dimensionality as a training parameterization, compression principle, or predictability property. SAR instead identifies a more specific mechanism: the reasoning-effective component of a full RL update is not merely low-rank, but aligned with the pretrained model's spectral coordinates and structured as a compact rewiring matrix among pretrained components. This distinction enables post-hoc editing of a completed full-RL model and can surpass the RL baseline in high-k exploration and multi-domain transfer, including at the 32B scale.

Model merging and interference. Model merging methods combine separately trained models by operating on weight-space task deltas, typically using arithmetic, magnitude, sign, or sparsity-based rules to reduce interference [9–11]. Recent analysis further shows that RL-trained models are more mergeable than SFT-trained models, but merged models can still degrade relative to their corresponding experts [46]. These works establish task conflict as a central obstacle in model composition. SAR addresses this conflict at a different level: before composition, it filters each RL update through the pretrained spectral manifold, extracting reasoning-effective rewiring while removing off-manifold residual drift. This enables positive-sum cross-domain composition where the merged model can surpass the best single-domain experts.

B Illustrative Example of Spectral Rewiring

Triangle area calculation. Consider a geometric reasoning task that requires computing the area of a triangle from two side lengths and the included angle. In the pretrained base model, the relevant capabilities may be represented in separated spectral components: a right singular direction v_1 detects side-length information, another direction v_2 detects angle information, and an output direction u_3 corresponds to applying the sine-area formula. Under the original diagonal mapping Σ, neither v_1 nor v_2 alone is sufficient to activate u_3. SAR isolates the RL-induced off-diagonal entries $M_{3,1}$ and $M_{3,2}$, which route both premise directions into the same output direction. Thus, when the model detects both required premises, Equation 6 combines their signals to activate the appropriate multi-condition deduction.

C Implementation Details

Algorithmic Scope. For each Transformer block, SAR is applied only to the attention linear weights (W_q, W_k, W_v, W_o) and MLP linear weights $(W_{\text{gate}}, W_{\text{up}}, W_{\text{down}})$. We do not modify biases, normalization parameters, token embeddings, or the language-modeling head; these parameters are kept from the base model. Empirically, projecting both attention and MLP weights is necessary to recover the full effect: projecting only attention or only MLP degrades relative to the full-RL model, while using the base or RL versions of the embeddings and lm_head has little effect on the final results.

Evaluation Configuration. Unless otherwise specified, we use the same decoding configuration for all reasoning and coding evaluations. We sample with temperature $T = 0.6$, top-$p = 0.95$, and a maximum generation

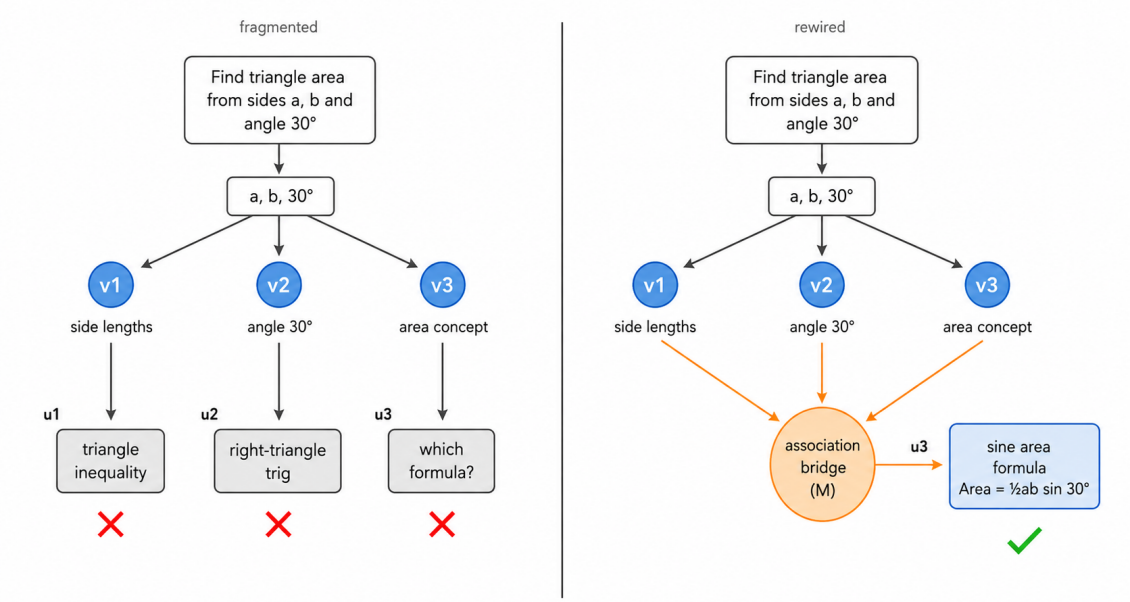


Figure 4 Illustration of spectral rewiring. The base model primarily performs point-to-point associations between isolated spectral components. RL-induced rewiring introduces many-to-one routing, allowing multiple premise directions to jointly activate an output direction associated with a compositional reasoning step.

Table 5 Coverage analysis on AIME 2024 with 256 rollouts. A problem is counted as covered if any rollout produces the correct final answer.

Model	Coverage@256	Solved / Total	Gain
DeepScaleR full RL	83.33%	25 / 30	–
SAR, 1% rank	86.66%	26 / 30	+1 problem

length of 32,768 tokens. For POLARIS, we follow its longer-generation setting and use a maximum generation length of 96,000 tokens. All compared models within the same experiment use identical sampling parameters to ensure a fair comparison between the base model, full RL model, SAR-projected model, and no-projection control.

Numerical Precision. All SVD decompositions and SAR projections are computed in FP32. Since the reconstructed SAR update is compact and sensitive to rounding, directly storing the projected weights in low precision can perturb the intended spectral geometry, especially for smaller models. We therefore store and evaluate the 1.5B, 4B, and 7B models in FP16, which better preserves fine-grained updates than BF16. For the 32B OLMo-3 models, we store the projected weights in FP16 and evaluate with BF16 computation, as they are empirically less sensitive to floating-point perturbations. Unless otherwise stated, all compared models within the same experiment use the same runtime precision.

Coverage analysis. We report a coverage-based view of the large-k improvement in Table 5. A problem is counted as covered if any of 256 rollouts produces the correct final answer. Under this metric, SAR solves one additional AIME 2024 problem beyond the full RL model, indicating that the projected model expands the set of reachable correct solutions rather than merely changing single-sample accuracy.

D Boundary Cases of Spectral Rewiring

SAR is most effective when RL remains in an elicitation regime, where sparse verifiable rewards and token-averaged updates produce relatively small per-step changes that reorganize capabilities already encoded in the reference model. We observe several boundary cases where this assumption does not fully hold.

Heavily trained reasoning RL. For reasoning models trained with many RL steps, the accumulated update can move beyond compact spectral elicitation. In our JustRL case study, which is trained for over 4000 steps [47], projecting the RL update into the base spectral space recovers a non-trivial AIME 2024 score of roughly 43%. This is comparable to the earlier DeepScaleR-level result, but remains below the final RL model ($\sim 52\%$). The result suggests that the base spectral manifold still contains meaningful recoverable reasoning capacity, while the remaining gain may come from stronger task-specific parameter rewriting induced by extended optimization. We therefore interpret this case as a boundary between elicitation and rewriting, rather than as evidence that spectral structure is absent.

Direct code RL from a base model. Code RL introduces domain requirements beyond abstract reasoning. The model must satisfy an execution protocol: syntactically valid code, function-completion or stdin/stdout conventions, exact input-output formatting, unit-test behavior, and edge-case handling. These behaviors can require adapting programming knowledge and interface behavior that are not sufficiently established in the base model's spectral geometry. Consistent with this interpretation, direct code RL from the base model is less projection-compatible in our experiments, as in the DeepCoder case [34]. However, once code SFT establishes the relevant programming interface, the subsequent RL update becomes more compatible with spectral projection onto the SFT model's spectral space. For example, projecting X-coder-RL onto X-coder-SFT recovers the full RL performance [48]. Our OLMo-3 experiments further support this view: after coding-oriented training establishes the relevant domain knowledge, RL primarily elicits reasoning capability already present in the reference model.

Dense critic rewards and PPO-style training. PPO-style training with an external critic or dense reward can also reduce projection compatibility. The key factor is not the PPO algorithm itself, but the effective optimization strength: dense token-level rewards and critic-shaped objectives can produce larger and more continuous parameter movement than sparse outcome-reward RL. This can push the update away from compact rewiring in the reference spectral coordinates and toward broader behavioral rewriting. Thus, SAR should be understood as most directly characterizing sparse outcome-reward elicitation, while dense-reward RL may require either a different reference point, intermediate projection during training, or a richer spectral basis.

Together, these cases delineate the scope of SAR. The method captures the component of RL that is aligned with the reference model's spectral manifold. When the reference model already contains the relevant reasoning ability or domain knowledge, this component can account for the dominant effect of RL. When training intensity or missing domain knowledge forces the model to acquire behavior outside that manifold, spectral projection remains informative but becomes an incomplete reconstruction of the full RL update.

E Comparison with the No-Projection Control

To isolate the effect of spectral alignment, we compare SAR with a no-projection control. This control extracts the same top-ranked low-rank component from the RL update ΔW, but directly adds it back to the base model without projecting it onto the pretrained SVD subspace. Thus, both methods use the same rank budget, and the only difference is whether the compact update is aligned with the base model's spectral geometry.

Reasoning and Exploration. We first evaluate reasoning and exploration on AIME 2024 using the DeepScaleR models. For each problem, we generate 256 rollouts and report the Pass@k scaling behavior in Figure 5. SAR and the no-projection control achieve similar Pass@1 performance, showing that low-rank extraction alone can preserve the single-sample reasoning gains of RL. However, their large-k behavior differs clearly:

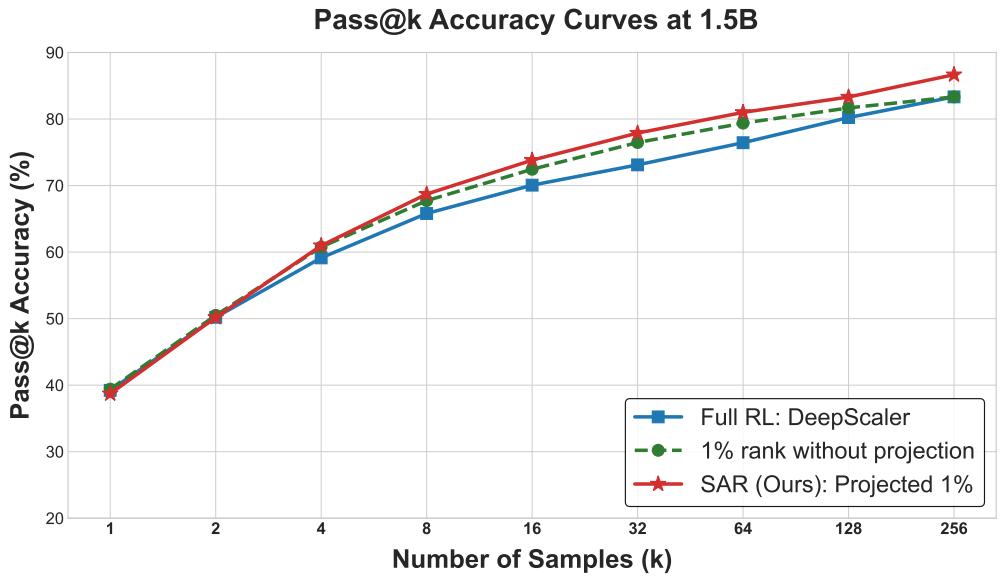


Figure 5 Pass@k scaling on AIME 24 among full RL, SAR, and the low-rank component without projection.

the no-projection control saturates earlier, while SAR continues to improve. At Pass@256, SAR solves one additional problem that neither the full RL model nor the no-projection control solves within 256 attempts. This indicates that the reasoning-effective update is not merely low-rank; aligning it with the base model's SVD space is **important for recovering reasoning and exploration capability**.

Multi-Domain RL. We then evaluate the same control in the multi-domain setting using the OLMo-3-32B models on AIME 2025 and LiveCodeBench v5. Results are shown in Table 6. Consistent with the single-domain results, the no-projection control retains part of the AIME performance, but fails to recover SAR's exploration benefit. On AIME 2025, the control reaches only 86.67% Pass@32, whereas SAR reaches 93.33%. On LiveCodeBench, the control recovers part of the performance, suggesting that low-rank extraction can remove some redundant update directions. Nevertheless, SAR achieves the strongest coding performance, improving AVG@10 to 69.09%, compared with 68.24% for the control and 67.61% for full RL.

Cross-Domain Merging. We also evaluate a no-projection control for math-to-code expert merging. This control extracts the same 1% low-rank component from the math RL update and merges it into the code expert, but does not align the update with the pretrained spectral basis. As shown in Table 7, the no-projection merge preserves the code-side gain, but substantially weakens math reasoning compared with SAR. This indicates that low-rank extraction alone is insufficient for positive-sum composition; the spectral projection step is what makes the math reasoning update compatible with the code expert.

Overall, this comparison supports our central insight: the reasoning-effective parameters are not only compact, but are specifically **expressed in the base model's pretrained SVD space**. Low-rank extraction explains part of the parameter efficiency, but SAR's spectral alignment is what yields stronger exploration and cross-domain transfer. In several settings, SAR even outperforms the full RL model, indicating that extracting the compact spectral reasoning core is not merely a compression technique, but a useful model-editing operation in its own right.

F Method Ablations

We further ablate the structure of SAR on DeepScaleR-1.5B using the same 1% spectral budget on AIME 2024. Table 8 compares SAR with three controls: replacing the pretrained spectral basis with a random

Table 6 Head-to-head comparison on AIME 2025 and LiveCodeBench v5. The no-projection control keeps the same top-1% low-rank component but removes the projection onto the pretrained SVD subspace. SAR achieves the strongest large-k reasoning coverage and coding performance.

Model / Method	AIME 2025		LiveCodeBench v5	
	AVG@32	Pass@32	AVG@10	Pass@10
Base: OLMo3-32B-Think-DPO	70.83%	90.00%	64.81%	78.07%
Full RL: OLMo-3.1-32B-Think	**75.76%**	90.00%	67.61%	80.91%
No projection, 1%	75.50%	86.67%	68.24%	81.13%
SAR projection, 1%	75.16%	**93.33%**	**69.09%**	**81.59%**

Table 7 No-projection control for 1.5B math-code merging. The no-projection merge uses the same 1% low-rank math update budget as SAR but skips alignment to the pretrained spectral basis. It preserves code performance but loses much of the math gain, whereas SAR achieves positive-sum transfer.

Scale	Model / Method	Math (AIME 24)		Code (LCB)
		AVG@32	Pass@32	AVG@8
1.5B	*Single-Domain References*			
	Base: Distill-Qwen-1.5B	31.67%	76.67%	23.00%
	Math Expert: DeepScaleR	**40.31%**	**76.67%**	27.20%
	Code Expert: Archer-Code	39.48%	76.67%	31.80%
	Best Single-Domain Expert	*40.31%*	*76.67%*	*31.80%*
	Cross-Domain Merging			
	No Projection: 1% Math-to-Code	36.25%	63.33%	32.25%
	Spectral Projection Merge (SAR)	**43.44%**	**76.67%**	**32.25%**

projection, retaining only the diagonal entries of the rewiring matrix M, and retaining only its off-diagonal entries.

These ablations support three conclusions. First, the random projection baseline improves over the base model in Pass@1 but remains clearly below SAR, showing that the pretrained singular-vector coordinates are important rather than interchangeable with an arbitrary low-dimensional projection. Second, the diagonal-only variant fails to recover the RL gain, suggesting that simply rescaling isolated pretrained components is insufficient for reasoning elicitation. Third, the off-diagonal-only variant recovers most of the Pass@1 gain but loses Pass@32 coverage, indicating that cross-component rewiring carries a large part of the reasoning signal but works best together with the diagonal component. Overall, SAR's effect comes from the combination of pretrained spectral alignment and the full rewiring matrix, rather than from low-rank structure, random projection, or either matrix component alone.

Table 8 Method ablations on DeepScaleR-1.5B with a 1% update budget on AIME 2024. SAR preserves full-RL-level Pass@1 while maintaining Pass@32, whereas random projection and partial rewiring controls lose reliability or coverage.

Method	Pass@1	Pass@32
Base	31.67%	80.00%
Full RL	**40.31%**	76.67%
SAR, 1%	40.21%	76.67%
Random projection, 1%	36.67%	80.00%
Diagonal-only rewiring, 1%	30.73%	80.00%
Off-diagonal-only rewiring, 1%	38.54%	73.33%