ISO: An RLVR-Native Optimization Stack

Hanqing Zhu[1,♠], **Wenyan Cong**[1,♠], **Zhizhou Sha**[1], **Sagnik Mukherjee**[2], **Xinyuan Song**[3], **David González-Martínez**[6], **Xiaoxia Wu**[4], **Yuandong Tian**[5], **Shiwei Liu**[6], **David Z. Pan**[1], **Zhangyang "Atlas" Wang**[1,†]

[1]The University of Texas at Austin [2]UIUC, [3]Emory University, [4]Together AI,
[5]Recursive Superintelligence Inc, [6]ELLIS Institute Tübingen

Code Website

Reinforcement learning with verifiable rewards (RLVR) is rapidly advancing the reasoning capabilities of language models, yet the optimization layer that converts reward feedback into weight-space updates remains poorly understood. In line of our prior analysis [1], we study this missing layer through the singular structure of model weights and identify *spectral inheritance*: RLVR can reuse the base model's weight spectra while acquiring new behavior through changes in the associated input and output singular frames. We further confirm that close reconstruction of learned endpoints requires both retained frames to adapt: remixing only within the incoming input and output spans, or keeping either incoming subspace fixed, leaves substantially more of the checkpoint change unexplained.

We operationalize spectral inheritance into **Isospectral Optimization (ISO)**, an RLVR-native, fixed-spectrum optimization framework with complementary offline and online instantiations. *Offline*, **ISO-Merger** combines the frame changes of shared-base specialists into a single fixed-spectrum model, requiring no post-merge data, rollouts, gradient updates, or on-policy distillation (OPD). It recovers complementary specialist capabilities and achieves the strongest aggregate performance among the compared data-free merging methods. *Online*, **ISO-Optimizer** applies a chosen base optimizer, including AdamW or Muon, to the frame variables while keeping the base spectra fixed. Across reasoning and coding tasks ranging from 1.5B to 8B parameters, ISO-Optimizer consistently improves accuracy in the reported runs and achieves matched scores with substantially fewer actor updates. On Qwen3-8B-Base, ISO-AdamW reaches AdamW's 270-update score by update 100 and improves it from 0.495 to 0.509 by update 210. Together, ISO turns spectral inheritance into a practical design principle for RLVR: *inherit the spectrum, optimize the frames*.

1. Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a major scaling axis for modern reasoning models [2], with rapid progress across data and environments [3], learning objectives [4, 5, 6, 7], and systems [8, 9]. Yet the optimizers and parameterizations that translate reward feedback into weight-space motion remain largely inherited from pre-training, despite the recent proliferation of optimizers designed specifically for that regime [10, 11, 12, 13]. This default is not yet obviously natural: pre-training learns from dense token-level supervision, whereas RLVR adapts an already capable policy using comparatively sparse outcome-level rewards.

We study this missing optimization layer and uncover a separation between *what RLVR reuses and what it changes*. We call this phenomenon **spectral inheritance**: *RLVR can reuse the base model's weight spectra while acquiring new behavior by changing the associated input and output singular frames*.

Work completed when Hanqing Zhu was at UT Austin. This work builds on our prior analysis of RL optimization dynamics (*The Path Not Taken*) and develops a concrete RL-native optimization stack including both offline and online.

♠*Equal contribution.* †*Corresponding author.* hqzhu@utexas.edu, wycong@utexas.edu

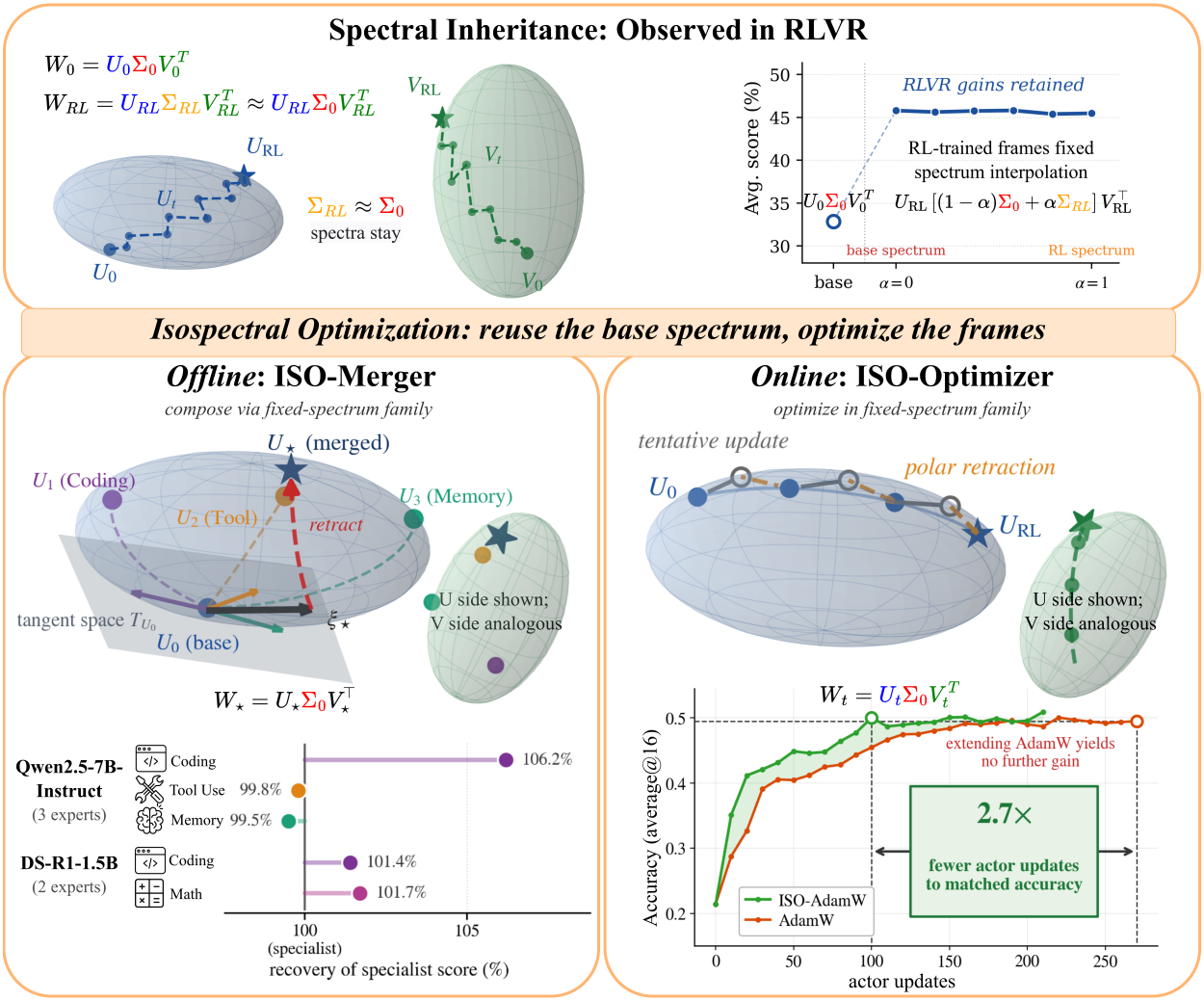


Figure 1 | **From spectral inheritance to Isospectral Optimization. Top.** Unconstrained RLVR changes both singular frames while its spectra remain close to their base values; restoring Σ_0 with the RL-trained frames fixed leaves performance flat at every α. ISO turns this regularity into an inductive bias: reuse Σ_0 and optimize (U, V). **Left.** ISO-Merger (offline) applies project–mask–merge–retract to shared-base specialists' frame changes and reconstructs a fixed-spectrum model that recovers their specialist capabilities. **Right.** ISO-Optimizer (online) applies a conventional optimizer (e.g., AdamW) to (U, V) under fixed Σ_0; on Qwen3-8B-Base, ISO-AdamW reaches AdamW's end-of-run accuracy with 2.7× fewer actor updates.

Existing accounts do not expose this separation, but they do suggest that RLVR differs structurally from pre-training and supervised fine-tuning (SFT). At the policy level, RLVR has been described through reverse-KL and KL-proximal views [1, 14]. At the parameter level, its Euclidean updates have been reported to be sparse and off-principal, with strongly overlapping footprints across independent runs initialized from the same base model [15, 1]. These findings identify distinctive structure in RLVR, but not a coordinate system that separates what is reused from what changes.

Observation: spectra stay. Motivated by our prior observation of limited spectral drift in RLVR [1], we move from Euclidean weight coordinates to a spectral view. For each weight matrix, write $W_t = U_t\Sigma_t V_t^\top$: Σ_t specifies the scales of the singular modes, while (U_t, V_t) specify their output and input directions. Across the unconstrained RLVR runs studied here, the learned checkpoints remain close to the fixed-spectrum families of their base weights, a pattern we call *near-isospectrality*. Because unconstrained RLVR optimization imposes no spectral constraint, we further ask whether this proximity

reflects a genuine optimization preference. A dimension-aware calibration finds no strong additional preference for spectrum-preserving motion beyond what high dimensionality alone predicts, although the contrast with SFT remains pronounced. The calibration therefore refines rather than weakens the observation: RLVR remains near-isospectral, but proximity alone does not establish an intrinsic optimization preference. This leaves the more consequential functional question: *are the small spectral changes that do occur necessary for the acquired behavior?*

Functional regularity: spectral inheritance. We next test a stronger question than spectral proximity: are the small spectral changes produced by unconstrained RLVR functionally necessary? Restoring the base spectra of RLVR checkpoints, while retaining their learned frames, preserves most acquired gains. More stringently, keeping the base spectra fixed throughout training and updating only the associated frames still supports strong RLVR learning. Conversely, a restricted spectrum-only control, which updates the singular values while freezing the base frames, yields only limited improvement.

We refer to this functional reuse as **spectral inheritance**: ***RLVR can reuse the base model's weight spectra rather than having to rewrite them to acquire new capabilities***.

Structure: both frames must remain adaptable. Spectral inheritance identifies what can be reused, but not which variables must remain adaptable. Could the learned endpoint instead be explained by a simpler transformation that either remixes the model only within the incoming input and output spans, or retains one incoming singular subspace while allowing only the other side to change? We find that both alternatives leave a substantially larger portion of the checkpoint change unexplained than retaining the incoming spectrum while allowing both singular frames to adapt. Thus, among the structural choices tested, the spectrum can remain fixed, but both frames must remain adaptable. The same spectral-inheritance and two-frame-adaptability pattern recurs across sequential RL stages with distinct objectives.

ISO: Isospectral Optimization for RLVR. These findings suggest a simple design principle: *inherit the spectrum, optimize the frames*. We operationalize this principle through **Isospectral Optimization (ISO)**, a framework that represents post-training change within the fixed-spectrum families of the base weights while keeping both singular frames adaptable. ISO is not a single numerical optimizer. It has two complementary instantiations spanning the RLVR workflow: ISO-Merger for checkpoint-only composition of shared-base specialists and ISO-Optimizer for online learning with a chosen base optimizer such as AdamW or Muon.

Offline: ISO-Merger. A modular RLVR workflow trains domain specialists from a shared base and later consolidates them, often through an additional on-policy distillation stage [16, 17]. ISO-Merger instead consolidates the specialists directly from their checkpoints. Guided by spectral inheritance, it reuses the shared base spectra and directly combines the experts' singular-frame changes into a single fixed-spectrum model. Without post-merge rollout generation, gradient updates, or distillation, ISO-Merger recovers complementary specialist capabilities and achieves the strongest aggregate performance among the compared data-free methods.

Online: ISO-Optimizer. ISO-Optimizer applies the same principle during RLVR training. Given a conventional base optimizer such as AdamW or Muon, it channels reward feedback into weight-space motion within the corresponding fixed-spectrum families, updating the singular-frame variables while keeping the base spectra fixed. Across reasoning and coding settings and multiple model scales, ISO-Optimizer improves final accuracy and reaches matched accuracy in fewer actor-update iterations than the corresponding weight-space optimizers.

Our contributions are summarized as follows:

- **Spectral inheritance in RLVR.** We formalize near-isospectrality as distance to a fixed-spectrum

family, calibrate it against the geometry of high-dimensional weight spaces, and show through endpoint and training-time interventions that the base model's weight spectra remain functionally reusable. We further show that, among the transformation classes tested, a low-residual reconstruction of learned endpoints requires both singular frames to remain adaptable.
- **RLVR-Native Isospectral Optimization Framework.** We then introduce ISO, a fixed-spectrum framework that reuses the base spectra and represents reward-driven post-training change through the associated singular-frame coordinates.
- **Data-free offline RL expert composition.** We develop ISO-Merger, which directly composes shared-base RLVR specialists in fixed-spectrum coordinates without post-merge data, rollout generation, gradient updates, or distillation.
- **Fixed-spectrum online RLVR.** We develop ISO-Optimizer, which applies a chosen base optimizer, including AdamW or Muon, to the singular-frame variables while preserving the source spectra. Across all runs, its variants improve performance and/or reach matched accuracy in fewer actor updates than their weight-space counterparts. On Qwen3-8B-Base, ISO-AdamW reaches AdamW's 270-update accuracy after 100 actor updates and improves it further by update 210.

2. Spectra Stay: From Near-Isospectrality to Spectral Inheritance

From sparse Euclidean motion to spectral structure. Recent analyses suggest that RLVR differs from pre-training and SFT in its weight-space motion. At the policy level, RLVR has been described as a conservative, KL-proximal improvement of the current policy [18, 14, 1]. At the parameter level, its Euclidean updates have been reported to be sparse and off-principal, with strongly overlapping patterns across runs initialized from the same base model [15, 1]. This raises a puzzle: large behavioral gains arise from apparently sparse and off-principal weight motion.

These regularities suggest that RLVR motion is structured, but raw Euclidean coordinates do not reveal what is reused and what is rewritten. Building on our prior observation of limited spectral drift in RLVR [1], we ask two questions. First, does the observed spectral proximity reflect a genuine preference for spectrum-preserving motion? A dimension-aware calibration finds no strong preference beyond what ambient dimensionality already predicts. Second, are the small spectral changes that remain functionally necessary? We test this through complementary interventions that restore the base spectra after training or keep them fixed throughout training. We call the resulting functional reuse *spectral inheritance*. Section 3 then asks which variables must remain adaptable once the spectrum is reused, and whether the same requirement recurs across sequential RL stages.

Notation. For a weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, let $q = \min\{d_{\text{out}}, d_{\text{in}}\}$ and let $\sigma(W) = (\sigma_1(W), \ldots, \sigma_q(W))$ denote all singular values in nonincreasing order. We write $W = U\Sigma V^\top$ for a thin SVD, with $U \in \mathrm{St}(d_{\text{out}}, q)$, $V \in \mathrm{St}(d_{\text{in}}, q)$, and $\Sigma = \mathrm{Diag}(\sigma(W))$, where

$$\mathrm{St}(d, r) = \{X \in \mathbb{R}^{d \times r} : X^\top X = I_r\}.$$

The spectrum Σ specifies the scales of the singular modes, while (U, V) specify their output and input directions. Throughout this section, the *base model* means the checkpoint that initializes the RL stage under study: it may itself be pretrained, SFT-trained, or already RL-trained. Accordingly, W_0 and Σ_0 denote a base weight matrix and its spectrum for the transition being analyzed. Section 3 reserves $r < q$ for an informative top-r truncation.

The fixed-spectrum family. For a base matrix $W_0 = U_0\Sigma_0 V_0^\top$, define its *fixed-spectrum family*

$$\mathcal{F}(W_0) := \left\{Z \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}} : \sigma(Z) = \sigma(W_0)\right\}. \tag{1}$$

Equivalently,

$$\mathcal{F}(W_0) = \left\{U\Sigma_0 V^\top : U \in \mathrm{St}(d_{\text{out}}, q),\ V \in \mathrm{St}(d_{\text{in}}, q)\right\}. \tag{2}$$

Thus, $\mathcal{F}(W_0)$ is simply the set of matrices of the same shape that share the singular values of W_0 while allowing the associated left and right singular frames to change.

2.1. A Precise Observation: Spectra Stay

We first make "**spectra stay**" precise by measuring the distance from a learned weight matrix W to the fixed-spectrum family $\mathcal{F}(W_0)$ of its base matrix W_0.

Proposition 2.1 (Exact distance to the fixed-spectrum family)**.** *For any* $W, W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$,

$$\operatorname{dist}_F\big(W, \mathcal{F}(W_0)\big) = \|\sigma(W) - \sigma(W_0)\|_2. \tag{3}$$

Moreover, if $W = U\Sigma V^\top$ *and* $\Sigma_0 = \operatorname{Diag}(\sigma(W_0))$*, then*

$$U\Sigma_0 V^\top \in \arg\min_{Z \in \mathcal{F}(W_0)} \|W - Z\|_F. \tag{4}$$

Proposition 2.1 gives singular-value drift a precise checkpoint-level meaning: it is the distance from the learned matrix to $\mathcal{F}(W_0)$, the family of matrices sharing the corresponding base spectrum. The proposition also identifies a closest representative with that spectrum, which motivates the functional intervention in Section 2.2. The proof and the treatment of repeated singular values are given in Appendix B.1.

Measurements. For each matrix ℓ, let $\Delta W^{(\ell)} = W_1^{(\ell)} - W_0^{(\ell)}$. We measure

$$\begin{aligned} \delta_\Sigma^{(\ell)} &:= \frac{\|\sigma(W_1^{(\ell)}) - \sigma(W_0^{(\ell)})\|_2}{\|W_0^{(\ell)}\|_F}, \\ \rho_\Sigma^{(\ell)} &:= \frac{\|\sigma(W_1^{(\ell)}) - \sigma(W_0^{(\ell)})\|_2}{\|\Delta W^{(\ell)}\|_F}. \end{aligned} \tag{5}$$

The first measures spectral drift relative to the base-weight scale. The second compares the exact distance to $\mathcal{F}(W_0)$ with the complete checkpoint displacement. For a collection $\mathcal{L}$ of analyzed matrices, we report the pooled residual $\rho_\Sigma^{\text{pool}} := \left[\frac{\sum_{\ell \in \mathcal{L}} \|\sigma(W_1^{(\ell)}) - \sigma(W_0^{(\ell)})\|_2^2}{\sum_{\ell \in \mathcal{L}} \|\Delta W^{(\ell)}\|_F^2}\right]^{1/2}$.

Long-horizon endpoint evidence. To rule out a short-horizon artifact, we analyze the released endpoint of a reasoning RLVR run on `DeepSeek-R1-Distill-Qwen-1.5B` (DS-1.5B) trained for over 3,000 updates [19, 20]. The checkpoint sequence

$$\texttt{Qwen2.5-Math-1.5B} \to \texttt{DS-1.5B} \to \texttt{Nemotron-Research-Reasoning-Qwen-1.5B}$$

contains an SFT transition followed by an RLVR transition. For the RLVR stage, `DS-1.5B` is the base model. Broader evidence across models, datasets, RL objectives, and training horizons was established in our prior work [1].

Figure 2 shows that the post-RL spectrum nearly overlaps that of its pre-RL base, whereas the illustrative SFT transition produces substantial spectral contraction. Across RLVR layers, $\delta_\Sigma^{(\ell)}$ is approximately $10^{-2}\%$, and the pooled relative residual is $\rho_\Sigma^{\text{pool}} = 0.03$. Thus, the distance from the learned endpoint to a closest checkpoint in $\mathcal{F}(W_0)$ is only 3% of the full base-to-RL displacement.

Evidence across sampled training checkpoints. An endpoint comparison cannot exclude a large intermediate spectral excursion that later cancels. We therefore track an independent unconstrained Qwen3-8B-Base AdamW run using checkpoints saved every 10 actor updates. At each saved checkpoint,

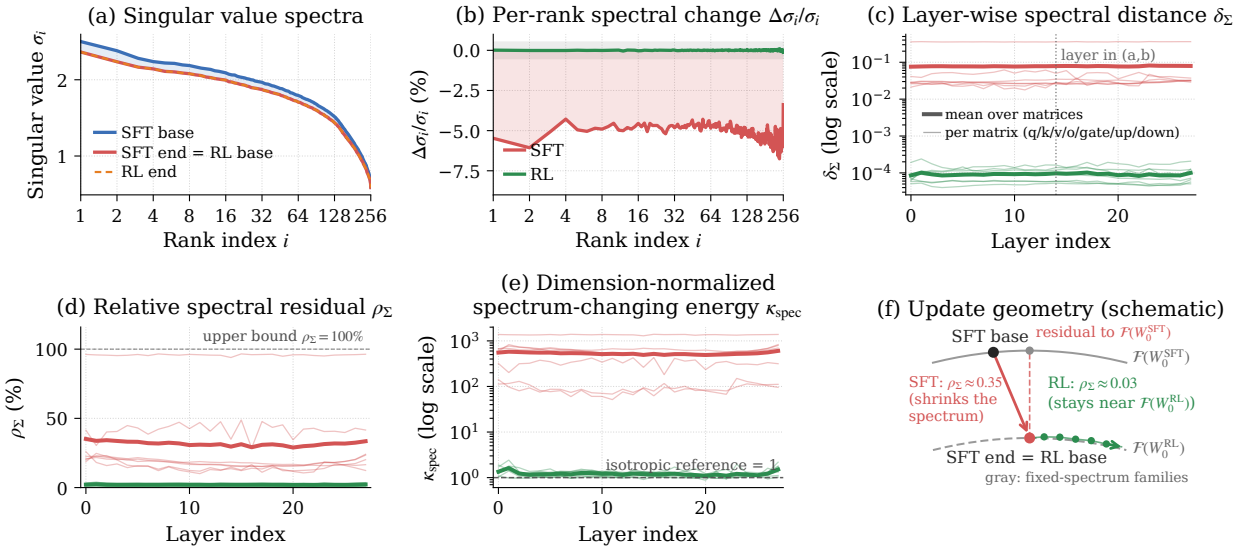


Figure 2 | **Spectral dynamics under SFT and RLVR post-training. (a)** Representative singular-value profiles (`v_proj`, layer 14); the SFT endpoint is also the RLVR base model. **(b)** Rank-wise relative change: SFT reshapes the spectrum, whereas RLVR stays within a fraction of a percent of its pre-RL values. **(c)** Layer-wise spectral distance δ_Σ: the RLVR endpoint is two to three orders of magnitude closer to its fixed-spectrum family than the illustrative SFT endpoint. **(d)** Relative spectral residual ρ_Σ: the mean RLVR residual is approximately 3% of the checkpoint displacement, compared with approximately 35% for SFT. **(e)** Dimension-normalized spectrum-changing energy κ_{spec} (Eq. 6): RLVR remains order-one ($\approx$ 1.0–1.4), whereas SFT lies two to three orders of magnitude above the dimensional reference. **(f)** Schematic endpoint view: SFT substantially rewrites the base spectrum, whereas RLVR checkpoints (dots) remain close to the fixed-spectrum family of the pre-RL base. The straight arrow denotes the SFT endpoint displacement, not a continuous optimization path.

we measure its distance to the same fixed-spectrum family $\mathcal{F}(W_0)$. As shown in Figure 3, both $\delta_\Sigma(t)$ and $\rho_\Sigma(t)$ remain small throughout the observed checkpoint sequence. Thus, at the resolution of the saved checkpoints, spectral stability persists during training rather than appearing only at the final endpoint.

Together, these results establish a descriptive fact: **for the analyzed matrices, unconstrained RLVR checkpoints remain close to the fixed-spectrum families of their base weights—a property we call near-isospectrality**.

Does near-isospectrality reflect a preference? A natural hypothesis of the preceding results is that RLVR preferentially suppresses spectrum-changing directions. High dimensionality, however, makes this interpretation nontrivial. In a $d_{\text{out}}d_{\text{in}}$-dimensional matrix space, only $q = \min\{d_{\text{out}}, d_{\text{in}}\}$ independent first-order coordinates change the singular values in the generic full-rank, simple-spectrum case. At $W_0 = U_0\Sigma_0 V_0^\top$, the first-order spectrum-changing coordinates of a displacement ΔW are $\text{diag}(U_0^\top \Delta W V_0)$. Thus, even a generically oriented high-dimensional displacement places only a small fraction of its squared norm in spectrum-changing coordinates.

We calibrate this dimensional effect with

$$\kappa_{\text{spec}}^{(\ell)} := \frac{d_{\text{out}}d_{\text{in}}}{q} \frac{\left\| \text{diag}\left(U_0^{(\ell)\top} \Delta W^{(\ell)} V_0^{(\ell)} \right) \right\|_2^2}{\|\Delta W^{(\ell)}\|_F^2}. \tag{6}$$

An isotropically oriented displacement has expected value one under this normalization. Hence, $\kappa_{\text{spec}} \ll 1$ would indicate additional suppression of spectrum-changing motion; order-one values do not reveal such a strong preference; and $\kappa_{\text{spec}} \gg 1$ indicates concentration in spectrum-changing coordinates.

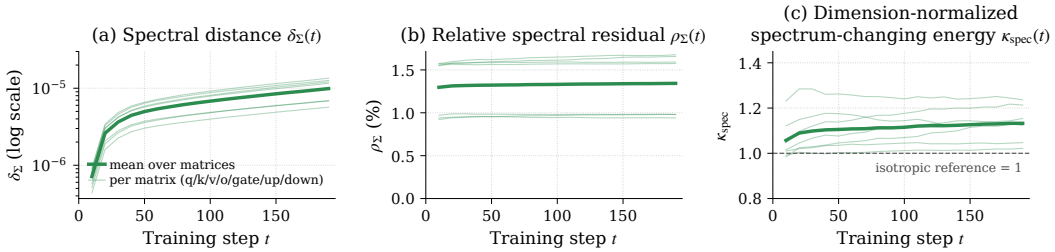


Figure 3 | **Near-isospectrality along an RLVR training trajectory** (Qwen3-8B-Base, checkpoints every 10 steps up to step 190). **(a)** The mean spectral distance $\delta_\Sigma(t)$ stays below 10^{-5}. **(b)** The mean relative spectral residual $\bar{\rho}_\Sigma(t)$ stays at $\approx$1.3% of the total displacement. **(c)** The dimension-normalized spectrum-changing energy $\kappa_{\text{spec}}(t)$ remains order-one relative to the isotropic dimensional reference.

Across matrix types in the public reasoning run, RLVR yields mean κ_{spec} values between 1.02 and 1.35. Thus, after accounting for the small number of spectrum-changing coordinates, RLVR does not exhibit a strong additional suppression of spectral change. The same order-one pattern persists along the sampled Qwen3-8B-Base trajectory in Figure 3(c). Crucially, this calibration *does not erase the contrast with SFT*. The illustrative SFT transition yields κ_{spec} values between 89 and 1364, two to three orders of magnitude above the dimensional reference. Thus, even after correcting for ambient dimensionality, RLVR and SFT remain sharply different in how strongly their updates concentrate in spectrum-changing coordinates.

2.2. Spectral Inheritance: Reusing the Base Model's Spectrum

The dimensional calibration changes the interpretation of the initial observation without overturning it. RLVR remains close to $\mathcal{F}(W_0)$, but this proximity alone does not reveal a strong preference for spectrum-preserving updates. The key functional question is whether the small spectral changes that do occur are needed for the acquired behavior.

We address this question through two complementary interventions. First, we restore the base spectrum after unconstrained RLVR while retaining the learned frames, testing whether the endpoint still requires its acquired spectral change. Second, we keep the base spectrum fixed throughout training, testing whether strong gains can be acquired without allowing that change at all.

Restoring the base spectrum after training. Let $W_0 = U_0\Sigma_0 V_0^\top$, $W_{\text{RL}} = U_{\text{RL}}\Sigma_{\text{RL}}V_{\text{RL}}^\top$. We interpolate only the spectrum:

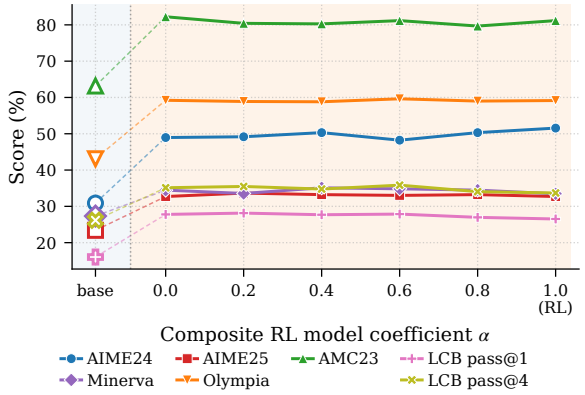
$$\widetilde{W}(\alpha) = U_{\text{RL}}\left[(1-\alpha)\Sigma_0 + \alpha\Sigma_{\text{RL}}\right]V_{\text{RL}}^\top, \qquad \alpha \in [0,1]. \tag{7}$$

At $\alpha = 0$, the base spectrum is restored while the RL-trained frames are retained. At $\alpha = 1$, the original RL checkpoint is recovered. Proposition 2.1 shows that $\widetilde{W}(0)$ is a closest point in $\mathcal{F}(W_0)$.

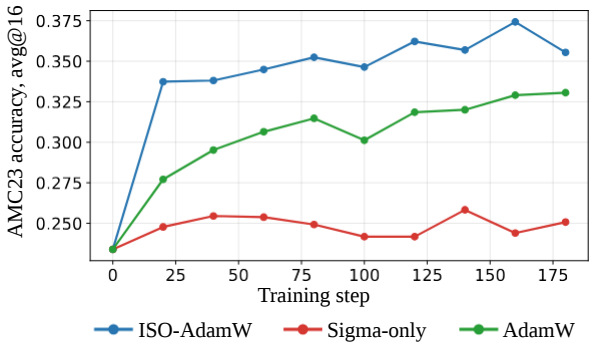
Figure 4(a) shows that restoring the base spectrum preserves most acquired performance. Conversely, replacing the base spectrum with the RL-trained spectrum while retaining the base frames does not improve the base model (Figure 15).

Learning with the base spectrum fixed. A stronger constructive test is to keep Σ_0 fixed from the first RL update onward and optimize only the singular frames. For contrast, we freeze the base frames and optimize only the diagonal spectrum Σ.

Figure 4(b) shows that the fixed-spectrum parameterization acquires strong reasoning gains and, in this run, outperforms the AdamW baseline, whereas spectrum-only training does not achieve



(a) Spectrum restoration after training.



(b) Fixed-spectrum and spectrum-only control.

Figure 4 | **Functional evidence for spectral inheritance. (a)** Restoring the base spectrum while retaining the RL-trained frames preserves most endpoint performance. **(b)** Keeping the base spectrum fixed throughout training still permits strong gains, whereas the restricted spectrum-only parameterization does not recover comparable performance under the studied recipe.

comparable gains. Because the latter control has restricted capacity,[1] its failure shows only that spectral rescaling alone is insufficient in this configuration.

Together, these interventions support *spectral inheritance*: RLVR can functionally reuse the base model's spectral structure rather than having to rewrite it.

Take-away 1: Spectral Inheritance. Unconstrained RLVR checkpoints remain close to the fixed-spectrum families of their base weights. More importantly, the interventions show that RLVR can reuse this spectral structure rather than having to rewrite it, a property we call ***spectral inheritance***.

3. Frames Move: What Changes Under Spectral Inheritance?

Structure: both frames must remain adaptable. Spectral inheritance identifies what can remain fixed, but not which variables must remain adaptable. *Could the endpoint be explained by a simpler transformation that either remixes only within the incoming input and output spans or retains one incoming singular subspace while leaving the other side free?* We find that these simpler alternatives leave substantially more of the checkpoint update unexplained than reusing the incoming spectrum while adapting both frames. Thus, among the transformation classes tested, a low-residual fixed-spectrum description requires both singular frames to remain adaptable.

Q1: Can a more conservative transformation explain the endpoint? For a transition $W_j \to W_i$, we refer to W_j as the *incoming checkpoint* and to W_i as the *learned endpoint*. Let $q = \min\{d_{\text{out}}, d_{\text{in}}\}$ and consider an admissible top-r truncation

$$W_t^{(r)} = U_t^{(r)} \Sigma_t^{(r)} \big(V_t^{(r)}\big)^\top, \qquad P_t^{(r)} = U_t^{(r)} \big(U_t^{(r)}\big)^\top, \qquad Q_t^{(r)} = V_t^{(r)} \big(V_t^{(r)}\big)^\top. \tag{8}$$

where $t \in \{i, j\}$. Admissibility requires a positive rank-r boundary gap at both endpoints, so that the retained projectors are well-defined. We use $r = \lfloor 0.9q \rfloor$ in the main text and report rank sensitivity in Appendix B.4.

[1] This family has only q active degrees of freedom per matrix.

Define the rank-r fixed-spectrum family associated with the incoming checkpoint:

$$\mathcal{F}_j^{(r)} := \left\{ U\Sigma_j^{(r)}V^\top : U \in \mathrm{St}(d_{\mathrm{out}}, r),\ V \in \mathrm{St}(d_{\mathrm{in}}, r) \right\}. \tag{9}$$

This family retains the incoming singular values $\Sigma_j^{(r)}$ while allowing both singular frames to vary.

Proposition 3.1 (Optimal reconstructions and unexplained-update ratios)**.** *For an admissible transition $W_j \to W_i$, the Frobenius-optimal reconstructions under the four structural restrictions are*

$$\begin{aligned}
\widehat{W}_{\mathrm{mix}} &= P_j^{(r)} W_i^{(r)} Q_j^{(r)}, && \text{remix within both incoming subspaces,}\\
\widehat{W}_L &= P_j^{(r)} W_i^{(r)}, && \text{retain the incoming output subspace,}\\
\widehat{W}_R &= W_i^{(r)} Q_j^{(r)}, && \text{retain the incoming input subspace,}\\
\widehat{W}_{\mathrm{iso}} &= U_i^{(r)} \Sigma_j^{(r)} \left(V_i^{(r)}\right)^\top, && \text{retain the incoming spectrum; adapt both frames.}
\end{aligned} \tag{10}$$

For $W_i^{(r)} \neq W_j^{(r)}$, define the unexplained-update ratio

$$u_h := \frac{\|W_i^{(r)} - \widehat{W}_h\|_F}{\|W_i^{(r)} - W_j^{(r)}\|_F}, \qquad h \in \{\mathrm{mix}, L, R, \mathrm{iso}\}. \tag{11}$$

Because $W_j^{(r)}$ is feasible under all four restrictions, optimality gives $0 \le u_h \le 1$.

Thus, $u_h = 0$ denotes exact reconstruction, whereas $u_h = 1$ means that the best reconstruction under restriction h is no closer to the endpoint than the unchanged incoming checkpoint.

Moreover,

$$\widehat{W}_{\mathrm{iso}} \in \arg\min_{Z \in \mathcal{F}_j^{(r)}} \|W_i^{(r)} - Z\|_F, \qquad u_{\mathrm{iso}} = \frac{\|\Sigma_i^{(r)} - \Sigma_j^{(r)}\|_F}{\|W_i^{(r)} - W_j^{(r)}\|_F}. \tag{12}$$

The remix reconstruction can be written as

$$\widehat{W}_{\mathrm{mix}} = U_j^{(r)} B_{ij}^{(r)} \left(V_j^{(r)}\right)^\top, \qquad B_{ij}^{(r)} := \left(U_j^{(r)}\right)^\top W_i^{(r)} V_j^{(r)}. \tag{13}$$

The core $B_{ij}^{(r)}$ is unconstrained and may rotate, mix, rescale, and change the represented spectrum. Thus, this class fixes only the incoming input and output spans. The one-sided classes are similarly permissive: they retain one incoming span while allowing the remaining mapping and spectrum to refit freely. Their u_h values are therefore optimistic lower bounds on the reconstruction error incurred by freezing the corresponding frame structure. We use u_h to test parameterization sufficiency. Formal class definitions and proofs are provided in Appendix B.3.

Q2: Does the same structure recur under a new RL objective? A second question is whether this structure is specific to a single RL stage and objective, or whether it reappears when a new RL stage starts from an already RL-trained checkpoint and targets a distinct capability.

A sequential objective-shift stress test. We evaluate the parameterization test on a two-stage vision-language-action RL pipeline initialized from `Qwen2.5-VL-3B-Instruct` [21]. Stage I optimizes spatial reasoning and produces W_1. Stage II starts from this already RL-trained checkpoint, optimizes an embodied-manipulation objective, and produces W_2. The resulting sequence provides three linked transitions:

$$W_0 \to W_1, \qquad W_1 \to W_2, \qquad W_0 \to W_2. \tag{14}$$

The transition $W_0 \to W_1$ provides the first-stage reference. The decisive objective-shift test is $W_1 \to W_2$: it asks whether the same frame-adaptability requirement reappears after re-anchoring at an already RL-trained checkpoint and changing the RL objective. The cumulative transition $W_0 \to W_2$ tests consistency across the complete two-stage sequence. We analyze both language and vision modules.

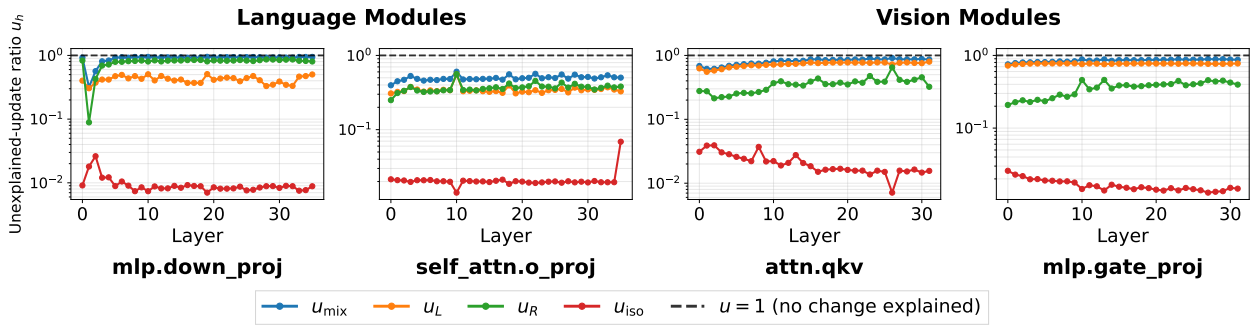


Figure 5 | **Both incoming frames must remain adaptable.** Unexplained-update ratios u_h for $W_0 \to W_2$ at $r = \lfloor 0.9q \rfloor$, for representative language and vision modules. The dashed line at $u = 1$ corresponds to leaving the incoming checkpoint unchanged. Restrictions that freeze one or both incoming singular subspaces (u_{mix}, u_L, u_R) leave a large fraction of the update unexplained at every layer, whereas fixing only the incoming spectrum (u_{iso}) leaves a few percent. All transitions and truncation levels are reported in Appendix B.4.

Results: both frames must remain adaptable. For the cumulative transition $W_0 \to W_2$, Figure 5 shows that an optimal remix within both incoming subspaces leaves a median 87% of the checkpoint update unexplained. Retaining only the incoming output or input subspace still leaves 45% and 42%, respectively. These are optimistic residuals because the corresponding classes may freely refit their remaining core and spectrum.

By contrast, retaining the incoming spectrum while adapting both frames leaves a median residual of only 1.8%. Thus, among the tested structural restrictions, neither incoming singular subspace can be frozen while retaining a low-residual endpoint description: the spectrum can remain fixed, but both frame variables must remain adaptable.

The same qualitative conclusion holds for the key transition $W_1 \to W_2$, where the incoming checkpoint has already undergone RL and Stage II optimizes a distinct embodied-manipulation objective. Thus, the frame-adaptability requirement is not confined to the first RL stage; it reappears after an objective shift and re-anchoring at an RL-trained checkpoint. Results for all transitions and truncation levels are reported in Appendix B.4.

Take-away 2: fix the spectrum, but keep both frames adaptable. Even permissive remix and one-sided reconstruction classes leave much of the checkpoint update unexplained. Thus, among the tested restrictions, the spectrum can remain fixed but both frame variables must remain adaptable, directly motivating ISO's parameterization $W(U, V) = U\Sigma_0 V^\top$.

4. From Spectral Inheritance to Isospectral Optimization (ISO)

Sections 2–3 identify complementary constraints for algorithm design. Section 2 shows that the base spectra can be reused while RLVR gains are retained and acquired. Section 3 then shows that both singular frames must remain adaptable.

We use this separation as an inductive bias rather than as a claim that unconstrained RLVR follows an exact fixed-spectrum update law. Motivated by this evidence, we introduce *Isospectral Optimization* (ISO), an RLVR-native optimization stack that reuses the base spectra and exposes both singular frames as adaptable coordinates for offline expert composition and online RLVR training.

4.1. The ISO Principle: Reuse the Spectrum, Optimize the Frames

Inherit the spectrum, optimize the frames. For a base matrix $W_0 = U_0\Sigma_0 V_0^\top$, ISO represents the weight matrix through the fixed-spectrum parameterization

$$W(U,V) = U\Sigma_0 V^\top, \qquad U \in \mathrm{St}(d_{\text{out}}, q), \quad V \in \mathrm{St}(d_{\text{in}}, q), \quad q = \min\{d_{\text{out}}, d_{\text{in}}\}. \tag{15}$$

Equation 15 parameterizes the fixed-spectrum family $\mathcal{F}(W_0)$ defined in Section 2. Thus, every represented matrix shares the base spectrum Σ_0. In exact arithmetic, $W(U,V) \in \mathcal{F}(W_0)$; in implementation, this property holds up to floating-point error.

This parameterization directly implements the separation established in the preceding sections: it reuses the base spectrum Σ_0 while leaving both frame variables (U,V) adaptable for optimization or composition.

An RLVR-native optimization stack. ISO is a framework rather than a single optimizer. The same fixed-spectrum frame parameterization supports two complementary stages of the RLVR workflow. Offline, ISO-Merger consolidates shared-base RL specialists directly from their checkpoints, without post-merge data, additional rollouts, gradient updates, or an on-policy distillation stage. Online, ISO-Optimizer applies a conventional base optimizer, such as AdamW or Muon, to the frame variables (U,V) during RLVR training. Both instantiations reuse the base spectrum and restore frame feasibility after composition or optimization.

Proposition 4.1 (First-order spectrum preservation)**.** *Let $W = U\Sigma_0 V^\top \in \mathcal{F}(W_0)$, where the singular values in Σ_0 are positive and simple. For any differentiable curve $W(t) \in \mathcal{F}(W_0)$ with $W(0) = W$ and tangent*

$$\dot{W} := \frac{d}{dt} W(t)\Big|_{t=0},$$

we have

$$\mathrm{diag}\big(U^\top \dot{W} V\big) = 0. \tag{16}$$

By contrast, for an arbitrary perturbation H,

$$\frac{d}{d\varepsilon}\sigma_k(W + \varepsilon H)\Big|_{\varepsilon=0} = u_k^\top H v_k, \qquad k = 1, \ldots, q. \tag{17}$$

For Stiefel tangent directions $\xi_U \in T_U\mathrm{St}(d_{\text{out}}, q)$, $\xi_V \in T_V\mathrm{St}(d_{\text{in}}, q)$, differentiating Equation 15 gives the induced weight-space direction

$$\dot{W} = \xi_U \Sigma_0 V^\top + U\Sigma_0 \xi_V^\top. \tag{18}$$

Because $U^\top\xi_U$ and $V^\top\xi_V$ are skew-symmetric, Equation 18 satisfies Equation 16. Thus, feasible frame motion preserves the spectrum to first order, while the finite reconstruction $U\Sigma_0 V^\top$ preserves it exactly. The proof is given in Appendix C.1.

4.2. ISO-Merger: Merging RL Experts Without Rollouts

Data-free composition of shared-base RL experts. Training a single RLVR policy across heterogeneous domains can be expensive and unstable. A modular alternative is to train several specialists from a shared base and then consolidate their capabilities into a single policy. One common consolidation strategy is an additional on-policy distillation stage driven by online rollouts [16, 17]. ISO-Merger instead addresses the strictly checkpoint-only setting: it requires no post-merge data, rollout generation, gradient updates, or distillation. It pursues the same broad objective as on-policy distillation: recovering complementary specialist capabilities in a single policy.

We focus on specialists trained from the same base checkpoint, with each expert targeting a distinct domain or capability. Merging already-generalist policies trained on broad, substantially overlapping mixtures constitutes a different setting that we do not study here.

Setup. Let $\{W_i\}_{i=1}^K$ be the corresponding matrices from K RL experts fine-tuned from the same base checkpoint W_0. We write $W_0 = U_0\Sigma_0 V_0^\top,\ W_i = U_i\Sigma_i V_i^\top$.

Many data-free merging methods construct $W_\star = \mathrm{Comp}(W_1, \ldots, W_K; W_0)$ by combining Euclidean task vectors. ISO-Merger instead retains each expert's learned frames (U_i, V_i) while reusing the shared base spectrum Σ_0. Equivalently, it represents expert i by the fixed-spectrum checkpoint

$$U_i\Sigma_0 V_i^\top \in \mathcal{F}(W_0)$$

and constructs

$$W_\star = U_\star\Sigma_0 V_\star^\top. \tag{19}$$

The resulting matrix therefore belongs to the fixed-spectrum family $\mathcal{F}(W_0)$ up to numerical precision.

Expert directions in shared frame coordinates. For simple singular values, each singular pair has a joint sign ambiguity,

$$(u_k, v_k) \mapsto (-u_k, -v_k).$$

We first align each expert's singular pairs with the corresponding base pairs using the sign-canonicalization rule described in Appendix E. After alignment, we define the frame displacements

$$\Delta U_i := U_i - U_0, \qquad \Delta V_i := V_i - V_0, \tag{20}$$

and project them onto the Stiefel tangent spaces at the shared base:

$$\xi_{U,i} = \Pi_{U_0}(\Delta U_i), \qquad \xi_{V,i} = \Pi_{V_0}(\Delta V_i), \tag{21}$$

where Π_{U_0} and Π_{V_0} denote the Stiefel tangent projections defined in Appendix E.

We find empirically that trailing modes, the columns associated with small singular values, are less stable across experts and that retaining them degrades the merged model. For a keep ratio $\rho_{\mathrm{keep}} \in (0, 1]$, define

$$\begin{aligned} k_{\mathrm{keep}} &:= \mathrm{round}(\rho_{\mathrm{keep}} q), \\ D_{\mathrm{keep}} &:= \mathrm{Diag}\left(\left(\mathbf{1}\{k \le k_{\mathrm{keep}}\}\right)_{k=1}^{q}\right), \end{aligned} \tag{22}$$

and mask the trailing columns:

$$\widetilde{\xi}_{U,i} = \xi_{U,i} D_{\mathrm{keep}}, \qquad \widetilde{\xi}_{V,i} = \xi_{V,i} D_{\mathrm{keep}}. \tag{23}$$

Unless stated otherwise, we use $\rho_{\mathrm{keep}} = 0.9$.

Masking is a coordinate-selection operation: $\widetilde{\xi}_{U,i}$ and $\widetilde{\xi}_{V,i}$ need not individually satisfy the Stiefel tangent constraints. We use them to represent each expert's change in the dominant shared frame coordinates and impose feasibility only after aggregating the experts. Because all displacements are anchored at the same base factors (U_0, V_0), their linear combinations are well-defined in a common coordinate system.

A unit-retention target. Different expert directions may overlap or interfere. To measure these interactions, we linearize the reconstruction map $W(U, V) = U\Sigma_0 V^\top$ at (U_0, V_0) and associate expert i with the local first-order effect proxy

$$g_i := \widetilde{\xi}_{U,i}\Sigma_0 V_0^\top + U_0\Sigma_0\widetilde{\xi}_{V,i}^\top \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}. \tag{24}$$

We then form the Gram matrix

$$\Gamma_{ij} := \langle g_i, g_j \rangle_F, \qquad \Gamma \in \mathbb{R}^{K \times K}. \tag{25}$$

For coefficients $c \in \mathbb{R}^K$, let

$$g(c) := \sum_{i=1}^{K} c_i g_i. \tag{26}$$

For each nonzero expert proxy g_i, define its self-retention in the merged proxy by

$$\mathrm{ret}_i(c) := \frac{\langle g(c), g_i \rangle_F}{\|g_i\|_F^2} = \frac{(\Gamma c)_i}{\Gamma_{ii}}, \qquad i = 1, \ldots, K. \tag{27}$$

The ideal target $\mathrm{ret}_i(c) = 1$ for every expert yields

$$\Gamma c = b, \qquad b := \mathrm{diag}(\Gamma) \in \mathbb{R}^K.$$

Because Γ may be ill-conditioned and the exact solution may require unstable coefficients, we use the ridge-stabilized system

$$\left(\Gamma + \lambda_{\mathrm{ridge}} I_K\right) \bar{c} = b, \qquad c^\star = \mathrm{clip}\left(\bar{c}; c_{\min}, c_{\max}\right). \tag{28}$$

The clipping bounds limit extreme coefficients and excessive amplification. The complete procedure therefore targets, rather than guarantees, unit self-retention after ridge stabilization, clipping, tangent projection, and retraction.

Project, retract, and reconstruct. After combining the masked expert directions, we project only the aggregate displacement back onto the Stiefel tangent spaces:

$$\begin{aligned} \xi_{U,\star} &= \Pi_{U_0}\left(\sum_{i=1}^{K} c_i^\star \widetilde{\xi}_{U,i}\right), \\ \xi_{V,\star} &= \Pi_{V_0}\left(\sum_{i=1}^{K} c_i^\star \widetilde{\xi}_{V,i}\right). \end{aligned} \tag{29}$$

For a thin SVD

$$X = P_X S_X Q_X^\top,$$

we define the polar factor by

$$\mathrm{polar}(X) := P_X Q_X^\top. \tag{30}$$

When X has full column rank, this is equivalently

$$\mathrm{polar}(X) = X(X^\top X)^{-1/2}.$$

For rank-deficient X, the SVD expression selects one valid nearest Stiefel factor, which need not be unique. For ISO-Merger, the retraction arguments are in fact full column rank. For example, tangent feasibility gives

$$(U_0 + \xi_{U,\star})^\top (U_0 + \xi_{U,\star}) = I + \xi_{U,\star}^\top \xi_{U,\star} \succ 0,$$

and analogously for $V_0 + \xi_{V,\star}$. We then reconstruct the merged factors and weight matrix as

$$\begin{aligned} U_\star &= \mathrm{polar}(U_0 + \xi_{U,\star}), \\ V_\star &= \mathrm{polar}(V_0 + \xi_{V,\star}), \\ W_\star &= U_\star \Sigma_0 V_\star^\top. \end{aligned} \tag{31}$$

Because $U_\star$ and $V_\star$ have orthonormal columns, $W_\star$ shares the base singular values Σ_0 up to floating-point error.

We apply the same construction to two-dimensional embedding and unembedding matrices. One-dimensional parameters, such as normalization scales and biases, are composed using a simple average. ISO-Merger thus performs checkpoint-only composition in shared fixed-spectrum coordinates, requiring no post-merge data, rollouts, gradient updates, or distillation. Full algorithmic details are provided in Appendix E.

4.3. ISO-Optimizer: Online Fixed-Spectrum RLVR Training

A fixed-spectrum transformation of a base optimizer. ISO-Optimizer does not replace the numerical update rule used by standard RLVR. Instead, it applies that rule to the singular-frame variables under a fixed base spectrum. Let

$$\mathsf{Opt}(X, G, s)$$

denote one update of a base optimizer, such as AdamW or Muon, where X is the optimized variable, G is its gradient, and s is the corresponding optimizer state. Standard weight-space training applies

$$(W^+, s_W^+) = \mathsf{Opt}\,(W, G_W, s_W)\,, \qquad G_W = \nabla_W \mathcal{L}(W). \tag{32}$$

ISO-Opt instead uses the fixed-spectrum parameterization

$$W(U, V) = U\Sigma_0 V^\top, \tag{33}$$

where Σ_0 is inherited from the base matrix $W_0 = U_0\Sigma_0 V_0^\top$ and remains fixed throughout training. The base optimizer supplies the update equations and state machinery, while ISO supplies the frame parameterization and the feasibility retraction.

Factor gradients and tentative updates. Given the gradient with respect to the reconstructed weight matrix,

$$G_W = \nabla_W \mathcal{L}\big(U\Sigma_0 V^\top\big),$$

the Euclidean gradients with respect to the frame variables follow from the chain rule:

$$G_U = G_W V\Sigma_0, \qquad G_V = G_W^\top U\Sigma_0. \tag{34}$$

A detailed derivation is provided in Appendix C.2. The factors Σ_0 arise from the Jacobian of the reconstruction map: the same frame displacement produces a larger weight-space change for a mode with a larger singular value.

The base optimizer applies its update rule independently to the two frame variables and their associated states:

$$\begin{aligned}(\bar{U}, s_U^+) &= \mathsf{Opt}\,(U, G_U, s_U)\,,\\ (\bar{V}, s_V^+) &= \mathsf{Opt}\,(V, G_V, s_V)\,.\end{aligned} \tag{35}$$

The tentative factors $\bar{U}$ and $\bar{V}$ need not satisfy the Stiefel constraints. ISO restores feasibility using the polar retraction defined in Equation 30:

$$\begin{aligned}U^+ &= \mathrm{polar}(\bar{U}),\\ V^+ &= \mathrm{polar}(\bar{V}),\\ W^+ &= U^+\Sigma_0 (V^+)^\top.\end{aligned} \tag{36}$$

In exact arithmetic, $W^+ \in \mathcal{F}(W_0)$; in implementation, its singular values match Σ_0 up to floating-point error.

A generic optimizer transformation. This construction defines

$$\mathsf{Opt} \longmapsto \mathrm{ISO}[\mathsf{Opt}]. \tag{37}$$

Using AdamW as the base update rule gives ISO-AdamW, while using Muon gives ISO-Muon. Both variants share the same fixed-spectrum frame parameterization and retraction, while inheriting the update equations and optimizer-state dynamics of their respective base rules. AdamW is our primary instantiation because it is the standard optimizer in the RLVR recipes studied here. ISO-Muon tests whether the same construction transfers across base update rules. Then, the central comparison is

$$\text{AdamW on } W \qquad \text{versus} \qquad \text{ISO[AdamW] on } (U, V) \text{ with fixed } \Sigma_0. \tag{38}$$

Importantly, ISO[Opt] is not an Opt update on W followed by projection onto $\mathcal{F}(W_0)$. The update rule and optimizer states are instantiated directly in the frame coordinates, changing the optimized variables, the effective factor-space geometry, and the feasible training trajectory.

Although ISO introduces two factor variables, it does not enlarge the represented weight-space class. After every retraction, the reconstructed weight remains in the strictly more constrained fixed-spectrum family $\mathcal{F}(W_0)$. Thus, the observed gains cannot be attributed to a larger feasible model class, although the factor-space parameterization and optimizer dynamics are also part of the method. Alternative spectrum-preserving optimizers formulated directly in weight space are also possible; systematically comparing such formulations with factor-space ISO is left to future work.

Weight decay. We set weight decay to zero for both the weight-space and ISO variants. Empirically, it provides no measurable benefit in our RLVR runs. At learning rates of order 10^{-6}, the standard coefficient $\lambda = 10^{-2}$ induces only $\eta\lambda \approx 10^{-8}$ relative shrinkage per step, below the BF16-visible update scale analyzed in prior work [1].

Implementation and overhead. The main additional structured operation in ISO-Optimizer is the polar retraction applied after each factor update. We implement it using an FP64 SVD-based polar computation for numerical stability. Following a DION-style distributed implementation [22], we distribute the matrix retractions across multiple GPUs. In our RLVR profiling, the additional retraction increases end-to-end step time by approximately 7%, as reported in Section 5.2. In modern asynchronous RL systems, this non-dominant optimizer work is amenable to overlap with rollout generation, particularly in long-horizon and agentic settings where environment interaction can be substantially more expensive.

Representing each two-dimensional weight through both U and V increases the nominal storage for trainable tensors and optimizer states relative to a single dense matrix; the fixed spectrum Σ_0 requires no optimizer state. In our implementation, the factor tensors are fully sharded, and optimizer-state offloading can further reduce the per-device memory footprint. Given the observed optimization and actor-update savings, these overheads are manageable in our setting and motivate further infrastructure optimization for both retraction scheduling and factor-state storage.

5. Experiments

We evaluate ISO in two settings corresponding to its two algorithmic instantiations. First, we test whether ISO-Merger can compose RLVR experts without additional rollouts while preserving their specialist capabilities, pursuing an objective similar to that of on-policy distillation. Second, we test whether ISO-Optimizer improves RLVR convergence when applied to base optimizers such as AdamW and Muon.

5.1. ISO-Merger: Composing Heterogeneous RL Experts Without Rollouts

Setup. We evaluate ISO-Merger in the setting it is designed for: composing multiple RLVR specialists from a shared base without post-merge data, rollouts, or distillation. We consider two expert-composition settings. The first uses `Qwen2.5-7B-Instruct` [23] as the shared base and merges three RL experts specialized for coding [24], tool use [25], and long-context memory [26]. The second uses `DeepSeek-R1-Distill-Qwen-1.5B` as the shared base and merges two RL experts specialized for coding [32] and math [33]. We compare against representative training-free merging baselines: Task Arithmetic [27], TIES [28], TSV-Merge [29], RAM [30] and one more recent OrthoMerge-G-TIES [31]. All methods merge the same expert checkpoints and are evaluated directly after merging. Unless otherwise stated, generation results use average@16, and **stochastic evalua-**

Table 1 | **Main results on Qwen2.5-7B-Instruct with 3 RL experts.** Coding: pass accuracy (ACC) and unit-test accuracy (UT) on LiveBench (LB) and LiveCodeBench (LCB). Tool Use: Live and Non-Live averages following the BFCL v4 evaluation protocol. Memory: RULER HotpotQA and SQuAD at 32K–64K context lengths under the Recurrent setting. Best expert/merged result per column in **bold**. Shading indicates Experts and Ours.

	Coding				Tool Use		Memory				
	LB		LCB v2		Live	Non-Live	HotpotQA		SQuAD		
Method	ACC	UT	ACC	UT	ACC	ACC	32K	64K	32K	64K	Avg
Base [23]	36.25 ± 0.86	48.27 ± 1.44	28.31 ± 0.51	43.08 ± 0.82	62.96	69.26	49.87 ± 1.45	45.54 ± 0.58	58.90 ± 1.07	56.77 ± 0.61	49.92
RLVR-Coder [24]	**37.42 ± 1.22**	49.42 ± 1.60	**30.17 ± 0.99**	**44.24 ± 0.99**	63.37	68.07	50.21 ± 0.16	45.61 ± 0.33	60.51 ± 0.70	56.84 ± 0.42	50.59
RLVR-Tool [25]	36.87 ± 1.79	48.08 ± 0.82	28.20 ± 1.95	42.43 ± 1.90	**71.83**	**81.82**	50.39 ± 0.68	45.33 ± 0.26	59.73 ± 1.18	57.86 ± 0.52	52.25
RLVR-Memory [26]	37.32 ± 1.37	**50.05 ± 1.61**	27.41 ± 2.99	41.70 ± 4.63	63.82	72.61	**78.60 ± 0.54**	**77.95 ± 0.29**	**79.98 ± 0.38**	**78.52 ± 0.22**	**60.80**
Task Arithmetic [27]	38.85 ± 0.70	52.18 ± 0.40	31.55 ± 0.18	47.06 ± 0.58	**73.73**	**82.86**	72.22 ± 0.22	70.49 ± 0.10	75.03 ± 0.29	74.95 ± 0.32	61.89
TIES [28]	40.30 ± 2.18	52.04 ± 2.50	31.46 ± 0.35	47.33 ± 0.30	68.24	76.54	76.42 ± 0.36	75.24 ± 0.14	78.27 ± 0.46	77.86 ± 0.59	62.37
TSV [29]	36.96 ± 0.86	49.23 ± 0.59	28.92 ± 1.79	43.37 ± 3.25	73.64	82.69	75.37 ± 0.29	74.10 ± 0.30	78.30 ± 0.17	77.26 ± 0.78	61.98
RAM [30]	38.59 ± 0.79	50.38 ± 1.53	31.31 ± 0.28	46.33 ± 0.30	72.56	81.89	76.17 ± 0.20	75.29 ± 0.49	78.42 ± 0.47	77.83 ± 0.46	62.88
OrthoMerge-G-TIES [31]	40.74 ± 2.00	**53.22 ± 1.90**	**31.98 ± 0.84**	**47.48 ± 0.77**	66.98	78.17	74.79 ± 0.49	74.20 ± 0.49	77.75 ± 0.12	76.32 ± 0.21	62.16
ISO-Merger	**41.81 ± 2.00**	52.73 ± 1.57	31.06 ± 1.88	45.69 ± 3.06	72.32	81.07	**79.46 ± 0.78**	**76.77 ± 0.27**	**79.10 ± 0.14**	**78.01 ± 0.32**	**63.80**

Table 2 | **Main results on DeepSeek-R1-Distill-Qwen-1.5B with 2 RL experts**. Coding: pass accuracy (ACC) and unit-test accuracy (UT) on LiveBench (LB) and LiveCodeBench (LCB). Math: ACC on AIME 2024, AIME 2025, AMC 2023, Minerva, and OlympiadBench. Avg is the average over all columns. Best expert/merged result per column in **bold**. Shading indicates Experts and Ours.

	Coding				Math					
	LB		LCB v5		AIME 2024	AIME 2025	AMC 2023	Minerva	OlympiadBench	Avg
Method	ACC	UT	ACC	UT	Avg@32	Avg@32	Avg@8	Avg@4	Avg@4	
Base [4]	17.99 ± 0.29	26.24 ± 0.53	16.82 ± 0.28	20.85 ± 0.18	30.90 ± 1.15	23.40 ± 0.86	63.15 ± 0.38	27.33 ± 0.43	43.11 ± 0.26	29.98
Archer2.0 [32]	**26.66 ± 0.45**	**37.26 ± 0.55**	**26.90 ± 0.21**	**38.99 ± 0.37**	42.05 ± 0.43	28.02 ± 0.44	73.44 ± 0.50	30.09 ± 0.23	49.99 ± 0.65	39.27
JustRL [33]	22.23 ± 0.23	31.85 ± 0.74	23.05 ± 0.21	30.18 ± 0.31	**53.16 ± 0.44**	**36.84 ± 0.73**	**82.78 ± 0.51**	**34.71 ± 0.50**	**55.67 ± 0.42**	**41.16**
Task Arithmetic [27]	26.22 ± 0.22	35.82 ± 0.11	26.70 ± 0.21	36.58 ± 0.32	50.83 ± 0.39	33.09 ± 1.16	80.82 ± 0.51	33.00 ± 0.34	54.31 ± 0.15	41.93
TIES [28]	26.37 ± 0.73	36.75 ± 0.90	27.62 ± 0.13	39.45 ± 0.15	54.51 ± 0.72	35.76 ± 0.30	82.13 ± 0.14	33.76 ± 0.48	55.36 ± 0.15	43.52
TSV [29]	26.40 ± 0.40	36.83 ± 0.15	27.61 ± 0.17	40.01 ± 0.26	53.16 ± 0.57	35.17 ± 0.72	80.97 ± 0.92	33.88 ± 0.46	54.68 ± 0.27	43.19
RAM [30]	**26.71 ± 0.29**	**36.99 ± 0.47**	27.18 ± 0.63	38.83 ± 0.67	54.20 ± 0.55	35.80 ± 0.71	82.28 ± 0.38	34.10 ± 0.34	55.54 ± 0.25	43.51
OrthoMerge-G-TIES [31]	26.46 ± 0.62	36.73 ± 0.37	27.74 ± 0.30	39.86 ± 0.64	54.55 ± 1.44	35.14 ± 0.87	81.88 ± 0.63	33.55 ± 0.42	55.47 ± 0.24	43.49
ISO-Merger	25.52 ± 0.08	36.42 ± 0.55	**27.84 ± 0.13**	**41.84 ± 0.31**	**55.00 ± 0.52**	**37.81 ± 0.31**	**83.53 ± 1.02**	**34.77 ± 0.23**	**56.64 ± 0.05**	**44.38**

tions are repeated over three independent runs with mean and standard deviation reported. Full benchmark protocols, decoding settings, baseline hyperparameters, and best@4/worst@4 results are provided in Appendix F.

Results. Tables 1 and 2 show that ISO-Merger achieves the strongest overall data-free composition performance in both settings. On Qwen2.5-7B, ISO-Merger reaches an overall average of 63.80, compared with 62.88 for the strongest training-free baseline. On DeepSeek-R1-Distill-Qwen-1.5B, it reaches 44.38, compared with 43.52 for the strongest baseline. Appendix F shows that ISO-Merger essentially matches the strongest aggregate best@4 baseline under both backbones while improving worst@4 by 1.62 and 1.36 points, respectively. Thus, the mean improvements do not sacrifice upper-tail performance, and the worst@4 results suggest more consistent capability recovery across stochastic generations. Overall, ISO-Merger remains competitive with the original specialists while consolidating multiple capabilities into a single model. These results support the practical utility of spectral inheritance for shared-base expert composition: specialist capabilities can be combined in fixed-spectrum frame coordinates while reusing the shared base spectrum.

5.2. ISO-Optimizer: Optimizing in Fixed-Spectrum Coordinates Improves RLVR

Setup and evaluation. We evaluate ISO-Optimizer by applying conventional base optimizers in fixed-spectrum frame coordinates rather than directly in weight space. All runs use VERL [9]. For mathematical reasoning, we train `Qwen3-1.7B-Base` and `Qwen3-4B-Base` [34] on DEEPMATH-

Table 3 | **Math RLVR results.** We report average@16 accuracy. Because evaluations remain noisy near convergence, for each run we select the checkpoint with the highest aggregate score among the final three evaluations and report all benchmark scores from that checkpoint.

Model	Method	AIME24	AIME25	AMC23	Minerva	Olympiad	Avg.
Qwen3-1.7B-Base	Base	3.75	1.25	23.04	18.49	18.43	12.99
	AdamW	13.96	**12.92**	43.45	32.17	**39.25**	28.35
	Muon	**16.25**	8.33	43.60	32.24	38.06	27.70
	ISO-AdamW	16.04	11.04	**44.50**	**33.11**	39.01	**28.74**
Qwen3-4B-Base	Base	6.88	5.00	25.45	20.97	22.43	16.15
	AdamW	25.21	20.42	63.55	45.40	53.87	41.69
	Muon	25.42	19.58	63.63	**46.51**	**56.02**	42.23
	ISO-AdamW	**27.29**	**23.13**	**65.74**	45.15	55.99	**43.46**

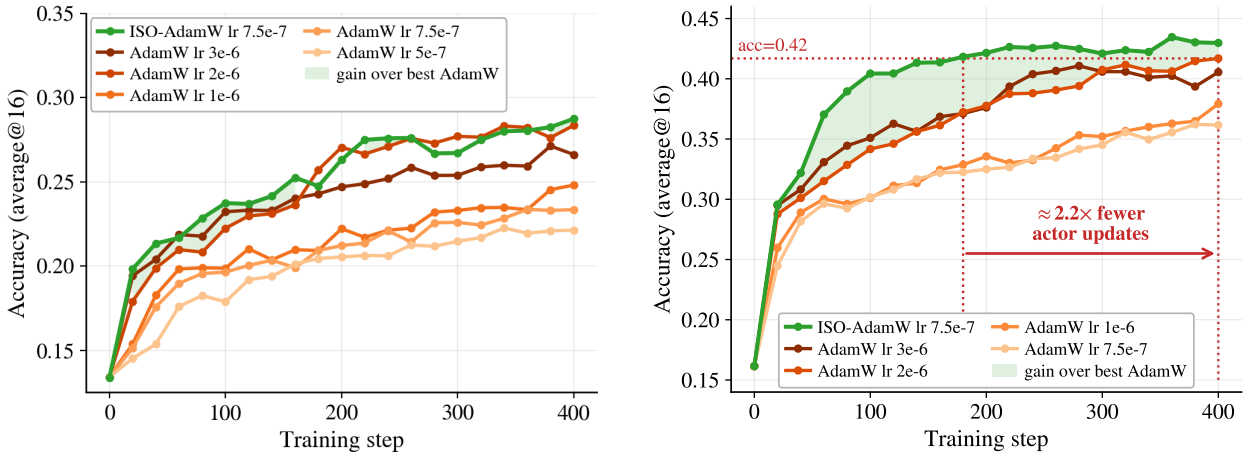


Figure 6 | **Aggregate math accuracy during RLVR training.** ISO-AdamW uses a single learning rate of 7.5×10^{-7}, whereas the weight-space AdamW baselines are tuned over the sweep described in the text. **Left** (`Qwen3-1.7B-Base`): ISO-AdamW attains the highest final accuracy. **Right** (`Qwen3-4B-Base`): ISO-AdamW matches the final aggregate accuracy of the strongest AdamW run while reducing the number of actor updates by a factor of approximately 2.2, and continues to improve thereafter.

103K [35] for 400 actor updates with a global prompt batch size of 256. The 1.7B and 4B runs use 16 and 12 rollouts per prompt, respectively, together with online filtering of all-correct and all-incorrect prompt groups. ISO-AdamW uses a single learning rate of 7.5×10^{-7} for (U, V).[2] For the weight-space baselines, we sweep AdamW learning rates over $\{5 \times 10^{-7}, 7.5 \times 10^{-7}, 1 \times 10^{-6}, 2 \times 10^{-6}, 3 \times 10^{-6}\}$ and Muon learning rates over $\{2.5 \times 10^{-5}, 5 \times 10^{-5}, 7.5 \times 10^{-5}, 1 \times 10^{-4}\}$.

We evaluate AIME 2024, AIME 2025, AMC 2023, Minerva, and OlympiadBench using 16 samples per problem. Evaluations are performed every 20 actor updates (10 actor updates on 8B) under identical decoding settings. Full training and evaluation details are provided in Appendix H.

Table 4 | **Coding results on DS-1.5B.** We report average@8 LiveCodeBench accuracy after 220 actor updates.

Method	LCB v5	LCB v6	Avg.
DS-1.5B	17.43	18.41	17.92
AdamW	24.01	26.24	25.13
ISO-AdamW	**25.49**	**26.91**	**26.20**

Math results. Table 3 shows that ISO-AdamW achieves the strongest aggregate score at both model scales. On `Qwen3-1.7B-Base`, it reaches 28.74, compared with 28.35 for the best AdamW run and 27.70 for Muon. On `Qwen3-4B-Base`, it reaches 43.46, compared with 41.69 for AdamW and 42.23 for Muon.

[2] On `Qwen3-1.7B-Base`, we compare 7.5×10^{-7} with 1×10^{-6} and use the better-performing setting throughout the math experiments.

Figure 6 shows that these gains emerge early rather than only at the selected endpoint. At 1.7B, ISO-AdamW attains the highest final aggregate accuracy. At 4B, it reaches the strongest AdamW run's final accuracy with approximately 2.2× fewer actor updates and continues to improve thereafter. These gains are obtained without enlarging the feasible weight-space model class: every ISO iterate remains in the constrained fixed-spectrum family $\mathcal{F}(W_0)$.

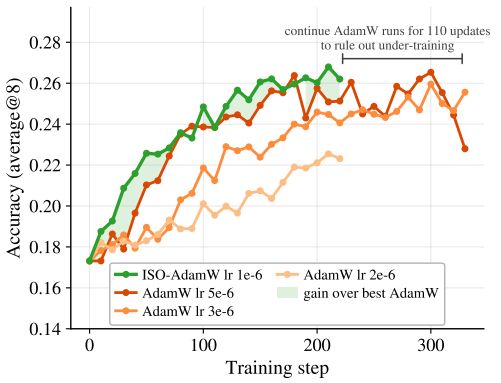


Figure 7 | **Coding accuracy on DS-1.5B.** ISO-AdamW leads the tuned AdamW baselines on LiveCodeBench.

Coding results. Beyond math reasoning, we train `DS-1.5B` on ARCHERCODER using a DAPO-style recipe and evaluate on LiveCodeBench v5 and v6. The ISO-AdamW run is limited to 220 actor updates, by which point the validation curve has plateaued; further training would require repeated passes over the relatively small coding set and may exacerbate overfitting. Table 4 shows that ISO-AdamW improves over AdamW on both LiveCodeBench splits. Figure 7 shows the corresponding training curves: ISO-AdamW outperforms the strongest AdamW setting (5×10^{-6}) at most evaluation checkpoints. To determine whether longer training closes the gap, we extend the two strongest AdamW runs to 330 updates. The 5×10^{-6} run peaks at 0.265 and then declines, while the 3×10^{-6} run ends at 0.256, a level that ISO-AdamW reaches by update 130. Neither extended baseline matches ISO-AdamW's peak accuracy of 0.268 within its 220-update budget. Full details are provided in Appendix H.

ISO-Muon variant. Because ISO is agnostic to the base optimizer, we also instantiate ISO-Muon, which applies the Muon update in fixed-spectrum Stiefel coordinates. Figure 8 compares ISO-Muon, using a learning rate of 5×10^{-5}, with a Muon sweep over $\{5\times10^{-5}, 7.5\times10^{-5}, 10^{-4}\}$ on `Qwen3-4B-Base`. We train all runs for 300 actor updates. Muon with a learning rate of 2.5×10^{-5} performs strictly worse and is omitted from the figure for clarity. The qualitative pattern in Figure 6 persists: ISO-Muon matches the final accuracy of the strongest Muon run (0.422 at update 300) by update 220 and attains a higher final accuracy (0.428 versus 0.422). We omit ISO-Muon from Table 3 because it uses the shorter 300-update budget.

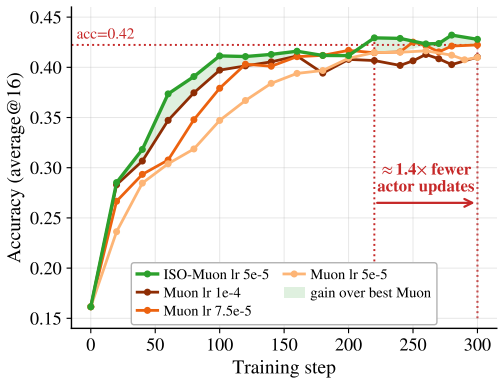


Figure 8 | **ISO-Muon versus Muon on `Qwen3-4B-Base`.** ISO-Muon reaches the strongest Muon baseline's final accuracy with 220 rather than 300 actor updates and finishes with higher accuracy.

Scaling to larger models. Beyond the 1.5B–4B models considered above, we test whether ISO's benefits extend to a larger scale. We train `Qwen3-8B-Base` on DEEPMATH-103K [35] using a global batch size of 256. We compare ISO-AdamW, with a learning rate of 7.5×10^{-7}, against weight-space AdamW using 2×10^{-6}, the strongest AdamW learning rate identified in the smaller-model sweeps. Neither method is further tuned at 8B. Training begins with an 8K maximum response length. At this scale, ISO-AdamW produces noticeably longer responses than AdamW, so we increase the cap to 16K for both runs after update 80 to reduce truncation. Evaluation uses a 16K cap throughout. As shown in Figure 9, ISO-AdamW reaches 0.509 after 210 actor updates, whereas AdamW reaches 0.487 under the same actor-update budget. To test whether AdamW is simply undertrained, we continue it for 60 additional updates. Its accuracy improves only to 0.495 and then plateaus, whereas ISO-AdamW reaches the same level by update 100. This corresponds to a reduction in the actor-update count by a factor of approximately 2.7. The persistent gap across evaluation

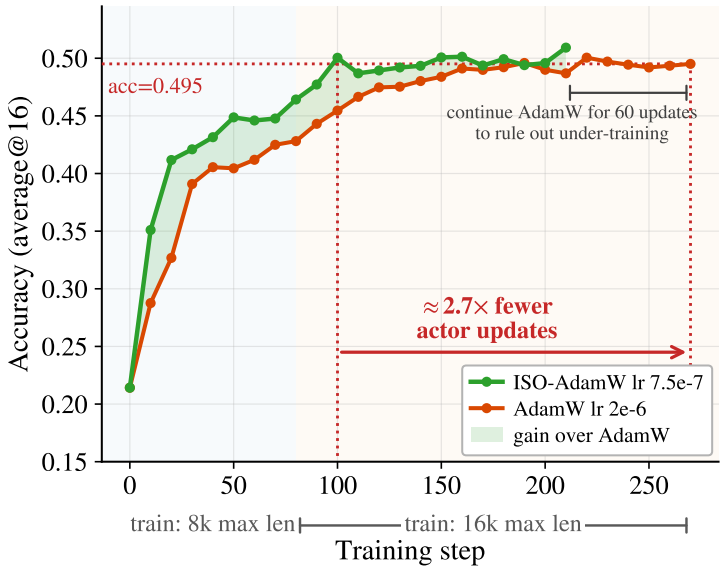


Figure 9 | **Math accuracy on `Qwen3-8B-Base`.** Shading marks the 8K and 16K max response-length phases in training. ISO-AdamW remains above AdamW and reaches its final accuracy earlier.

checkpoints, obtained without additional tuning at 8B, suggests that the benefits of fixed-spectrum coordinates extend to the larger model scale.

Retraction overhead. ISO introduces additional actor-update cost through the SVD-based polar retraction. Figure 10 profiles this cost on a representative `Qwen3-4B-Base` RLVR run with an 8K context window. Relative to AdamW, ISO increases the actor-update time by approximately 86 seconds per step, but this increase accounts for only about 7% of the end-to-end RL step time in this setting. The end-to-end runtime remains dominated by other parts, e.g., rollout generation, rather than by the optimizer update. The difference in total step time also grows because ISO produces longer responses, thereby increasing rollout latency. This cost profile differs from that of pre-training, where optimizer computation lies directly on the critical path of each step. RLVR includes a substantially longer generation phase, providing additional opportunities to potentially amortize the overhead or overlap actor-side computation with rollout generation, especially in modern asynchronous RL systems.

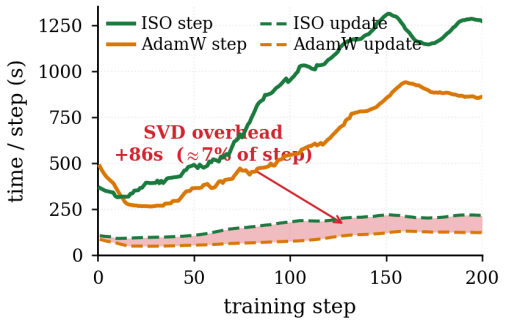


Figure 10 | **Actor-update and end-to-end step time for ISO and AdamW on `Qwen3-4B-Base`.**

6. Related Work

RLVR dynamics and optimization. Most advances in RLVR have focused on data and environments , learning objectives, or systems infrastructure. By contrast, the optimization layer that translates reward feedback into parameter-space motion remains comparatively underexplored and is typically inherited from pre-training, including the choice of optimizer and parameterization. Recent analyses have begun to distinguish RLVR from pre-training and supervised fine-tuning at both the policy and parameter levels. At the policy level, RLVR has been characterized through KL-proximal and conservative policy-improvement views; at the parameter level, its updates have been reported to be sparse, off-principal,

and strongly shared across runs initialized from the same source model [18, 14, 15, 1]. Building on our prior observation of limited spectral drift in RLVR [1], we formalize and test *spectral inheritance*: the base model's singular-value spectra can remain functionally reusable while the associated input and output frames adapt. We then use this principle to guide RLVR optimization algorithm design.

Matrix-geometric optimization and constrained parameterizations. A growing line of work treats matrix-valued parameters and updates as structured objects rather than as flattened Euclidean vectors. These methods act on different objects and impose different forms of structure. Muon orthogonalizes matrix-valued momentum updates, while DION develops scalable distributed approximations to such orthonormalized updates [11, 22]. Hyperball constrains the Frobenius norms of both weight matrices and optimizer updates, making the relative angular step size explicit rather than controlling it indirectly through weight decay [13]. The Modular Manifolds perspective advocates co-designing module-specific constraints and update norms, including Stiefel-constrained weights and a manifold version of Muon [36]. POET and POET-X preserve weight spectra through two-sided orthogonal-equivalence reparameterizations, with an emphasis on stable and efficient LLM training [37, 38]. Related structure has also been used for parameter-efficient adaptation: Spectral Adapter modifies a leading spectral subspace, while StelLA learns orthonormal input and output subspaces for a low-rank adapter [39, 40].

An independent concurrent work Pion [41] also preserves weight spectra through left and right orthogonal-equivalence transformations. Pion is introduced as a spectrum-preserving optimizer for general LLM training, and further adapts its construction to RLVR, where its motivation explicitly draws on the spectral observation from our earlier work [1]. The overlap with ISO is therefore most direct in online RLVR optimization. The two approaches nevertheless differ in both construction and scope. Pion updates each weight matrix directly through a dedicated Lie-algebra update rule, together with its own scale control, momentum design, and approximate exponential map. ISO instead fixes the source spectrum, exposes the associated singular frames as optimization variables, and applies a chosen base optimizer—including AdamW or Muon—to those variables, followed by retraction. More fundamentally, Pion begins from spectrum preservation as an optimizer design principle for general LLM training, with its construction motivated by pre-training stability, scale control, and orthogonal-equivalence geometry. ISO begins from an RLVR-specific empirical and functional characterization: we identify *spectral inheritance*, challenge raw near-isospectrality through dimension-aware calibration, and test source-spectrum reuse through functional interventions and sequential RL stages. We then promote this evidence to an RLVR-native optimization framework rather than a single optimizer. Its online instantiation transforms conventional base optimizers, while its offline instantiation provides a checkpoint-only method for composing shared-base RL experts.

From matrix-geometric priors to an RLVR-native post-training stack. Taken together, these works demonstrate the value of matrix-geometric structure in optimization. Our contribution is not the first use of orthogonality, manifolds, or spectral constraints. Rather, ISO follows an evidence-to-design route tailored to RLVR. We begin by studying unconstrained RLVR and show, through dimension-aware calibration, that raw near-isospectrality alone is not discriminative. We then test whether the learned spectral changes are functionally necessary, whether the source spectrum can remain fixed throughout learning, and how the remaining checkpoint change is organized in the associated singular frames. The same learning pattern recurs across sequential RL stages with distinct objectives. Together, these analyses establish *spectral inheritance*: RLVR can reuse the source model's weight spectra while acquiring new behavior through changes in the associated input and output singular frames. ISO promotes this empirically and functionally supported regularity from a descriptive observation to an explicit RLVR inductive bias.

ISO uses this inductive bias to redesign the RLVR post-training stack around a common fixed-

spectrum principle. Rather than tying the idea to a particular optimizer or to one singular-frame reparameterization, ISO treats spectral inheritance as a reusable interface across different stages of post-training. During online RLVR, it governs how reward-driven updates are represented and executed under the inherited spectra. After training, it provides a shared coordinate system for consolidating independently trained RL experts without additional rollouts or distillation. ISO-Optimizer and ISO-Merger are therefore not two unrelated algorithms, but two complementary realizations of the same stack-level design, spanning online policy learning and offline expert consolidation. The singular-frame construction studied here is one numerical realization of this principle rather than the definition of ISO itself. Alternative enforcement mechanisms and further applications within the same framework—including geometry-aware adapter initialization, adapter training, and adapter composition—remain open directions.

In summary, ISO contributes an evidence-to-stack design: it identifies spectral inheritance from RLVR's own dynamics, validates its functional relevance, and turns it into a common optimization principle spanning online policy learning and offline expert consolidation.

Geometry-aware model merging. Most training-free merging methods combine Euclidean task vectors, with techniques such as Task Arithmetic, TIES, and TSV-Merge reducing interference through sign resolution or low-rank structure [27, 28, 29]. OrthoMerge instead merges orthogonal fine-tuning transformations in a Lie algebra and extends to generally fine-tuned models by separating orthogonal and residual components [31]. ISO-Merger is specialized to shared-base RL experts: it projects each expert to the source-spectrum geometry, represents its frame displacement in a common base-anchored Stiefel tangent space, and performs checkpoint-only composition without post-merge prompts, rollouts, teacher queries, or optimization.

7. Conclusion

We identify *spectral inheritance* as a recurring structure in RLVR: the base model's weight spectra remain functionally reusable while new behavior is acquired through changes in the associated singular frames. Restoring the base spectra after training preserves most acquired performance, and keeping them fixed throughout training still supports strong reasoning and coding gains. Among the structural restrictions tested, simpler subspace-remixing and one-sided transformations leave substantially more of the checkpoint change unexplained, indicating that both frames must remain adaptable; this pattern also recurs across sequential RL stages with distinct objectives.

Building on this separation, we introduce Isospectral Optimization (ISO). ISO-Merger composes shared-base RL experts without post-merge data, rollouts, gradient updates, or distillation, achieving the strongest aggregate performance among the compared data-free methods. ISO-Optimizer applies conventional base optimizers to the frame variables under fixed base spectra, improving aggregate accuracy and reaching matched accuracy in fewer actor-update iterations, including 2.7× fewer updates on Qwen3-8B-Base. Together, these results identify the spectral structure learned before RL as a reusable substrate for post-training: *RLVR can inherit the spectrum and learn how it acts*.

References

[1] 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.

[2] xAI. Grok: Ai assistant, 2025. Accessed: 2025-09-24, continuously updated.

[3] Scale AI. The next frontier of data training: Rl environments, February 2026.

[4] 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.

[5] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*, 1:2, 2025.

[6] Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*, 2025.

[7] 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.

[8] Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An llm post-training framework for rl scaling. https://github.com/THUDM/slime, 2025. GitHub repository. Corresponding author: Xin Lv.

[9] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[11] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

[12] Hanqing Zhu, Zhenyu Zhang, Wenyan Cong, Xi Liu, Sem Park, Vikas Chandra, Bo Long, David Z Pan, Zhangyang Wang, and Jinwon Lee. Apollo: Sgd-like memory, adamw-level performance. *Proceedings of Machine Learning and Systems*, 7, 2025.

[13] Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. *arXiv preprint arXiv:2509.02046*, 2025.

[14] Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

[15] Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *arXiv preprint arXiv:2505.11711*, 2025.

[16] Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. https://thinkingmachines.ai/blog/on-policy-distillation.

[17] Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

[18] Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

[19] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

[20] Jian Hu, Mingjie Liu, Ximing Lu, Fang Wu, Zaid Harchaoui, Shizhe Diao, Yejin Choi, Pavlo Molchanov, Jun Yang, Jan Kautz, et al. Brorl: Scaling reinforcement learning via broadened exploration. *arXiv preprint arXiv:2510.01180*, 2025.

[21] Yifu Yuan, Haiqin Cui, Yaoting Huang, Yibin Chen, Fei Ni, Zibin Dong, Pengyi Li, Yan Zheng, and Jianye Hao. Embodied-r1: Reinforced embodied reasoning for general robotic manipulation. *arXiv preprint arXiv:2508.13998*, 2025.

[22] Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. Dion: Distributed orthonormalized updates. *arXiv preprint arXiv:2504.05295*, 2025.

[23] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[24] Yinjie Wang, Ling Yang, Ye Tian, Ke Shen, and Mengdi Wang. Co-evolving llm coder and unit tester via reinforcement learning. *arXiv preprint arXiv:2506.03136*, 2025.

[25] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

[26] Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025.

[27] 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.

[28] 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.

[29] Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18695–18705, 2025.

[30] Xiangchi Yuan, Dachuan Shi, Chunhui Zhang, Zheyuan Liu, Shenglong Yao, Soroush Vosoughi, and Wenke Lee. Behavior knowledge merge in reinforced agentic models, 2026.

[31] Sihan Yang, Kexuan Shi, and Weiyang Liu. Orthogonal model merging. *arXiv preprint arXiv:2602.05943*, 2026.

[32] Jiakang Wang, Runze Liu, Lei Lin, Wenping Hu, Xiu Li, Fuzheng Zhang, Guorui Zhou, and Kun Gai. Aspo: Asymmetric importance sampling policy optimization. *arXiv preprint arXiv:2510.06062*, 2025.

[33] Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025.

[34] Qwen Team. Qwen3 technical report, 2025.

[35] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. 2025.

[36] Jeremy Bernstein. Modular manifolds. *Thinking Machines Lab: Connectionism*, 2025. https://thinkingmachines.ai/blog/modular-manifolds/.

[37] Zeju Qiu, Simon Buchholz, Tim Xiao, Maximilian Dax, Bernhard Schölkopf, and Weiyang Liu. Reparameterized llm training via orthogonal equivalence transformation. *Advances in Neural Information Processing Systems*, 38:140775–140821, 2026.

[38] Zeju Qiu, Lixin Liu, Adrian Weller, Han Shi, and Weiyang Liu. Poet-x: Memory-efficient llm training by scaling orthogonal transformation. *arXiv preprint arXiv:2603.05500*, 2026.

[39] Fangzhao Zhang and Mert Pilanci. Spectral adapter: Fine-tuning in spectral space. *arXiv preprint arXiv:2405.13952*, 2024.

[40] Zhizhong Li, Sina Sajadmanesh, Jingtao Li, and Lingjuan Lyu. Stella: Subspace learning in low-rank adaptation using stiefel manifold. *Advances in Neural Information Processing Systems*, 38:75066–75092, 2026.

[41] Kexuan Shi, Hanxuan Li, Zeju Qiu, Yandong Wen, Simon Buchholz, and Weiyang Liu. Pion: A spectrum-preserving optimizer via orthogonal equivalence transformation. *arXiv preprint arXiv:2605.12492*, 2026.

[42] John Schulman and Thinking Machines Lab. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. https://thinkingmachines.ai/blog/lora/.

[43] Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr. *arXiv preprint arXiv:2507.15778*, 2025.

[44] 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.

Appendix Outline

The appendices are organized as follows.

- **Appendix A: Qualitative Index-Alignment Check.** Provides rank-index singular-direction sanity checks for the spectrum-restoration intervention.
- **Appendix B: Fixed-Spectrum Theory and Frame-Adaptability Diagnostics.** Establishes the fixed-spectrum distance and nearest representatives, derives the dimension-aware calibration, formalizes the reconstruction classes and unexplained-update ratios, and reports rank sensitivity.
- **Appendix C: First-Order Properties of the ISO Parameterization.** Proves first-order spectrum preservation for feasible frame motion and derives the ISO factor gradients.
- **Appendix D: A Same-Base SFT–RLVR Case Study.** Compares SFT and RLVR checkpoints derived from the same 14B backbone and reports an additional spectral-interpolation intervention.
- **Appendix E: ISO-Merger Details.** Specifies sign canonicalization, tangent projection, mode masking, retention coefficients, retraction, parameter scope, and the complete merging algorithm.
- **Appendix F: Data-Free Merging Experimental Details.** Describes the experts, baselines, hyperparameters, evaluation protocols, and additional distributional results for ISO-Merger.
- **Appendix G: Numerical Precision of the SVD-Based Retraction.** Documents the SVD-based retraction, evaluates the accuracy–runtime trade-off of PyTorch SVD configurations, and motivates the FP64 GPU implementation.
- **Appendix H: Online RLVR Training Details.** Gives the shared setup and the mathematical-reasoning and competitive-coding RLVR configurations, optimizer settings, training budgets, and evaluation protocols.

A. Qualitative Index-Alignment Check

Why this check is needed. The main text avoids using individual singular-vector identities as formal evidence, since SVD bases are gauge-dependent under sign flips and rotations inside near-degenerate singular-value clusters. Nevertheless, because our spectrum-rebasing intervention pairs singular values by rank index, it is useful to verify whether rank-index tracking is qualitatively stable in the RLVR runs we analyze.

Rank-index singular-vector rotations. For each layer, we compute the principal angle between rank-index paired singular directions from the base model and the post-training checkpoint, after standard sign alignment. These plots should be interpreted only as qualitative sanity checks. The formal frame-adaptability conclusions in Section 3 use projector-based reconstructions and unexplained-update ratios.

Figures 11 and 12 provide a qualitative contrast between RLVR and SFT. Under RLVR, rank-index paired singular directions rotate mildly and smoothly, suggesting that the spectrum-restoration intervention is not dominated by large index-wise mismatch. Under SFT, many matched directions become nearly orthogonal, indicating that rank-index tracking is much less stable. We emphasize that these plots are not used as formal evidence for singular-subspace motion. The main diagnostic evidence is the gauge-invariant projector and reconstruction analysis in Section 3.

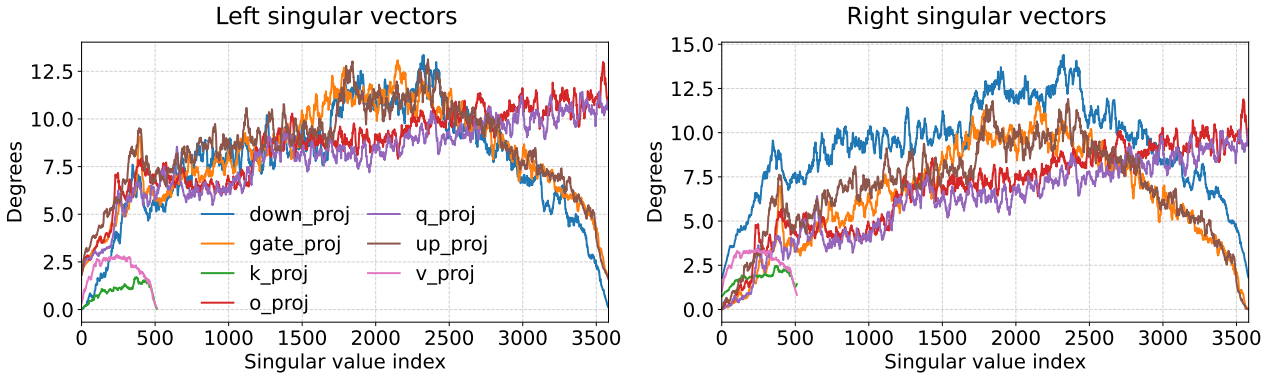


Figure 11 | **Qualitative singular-direction rotations under RL post-training.** Mean principal angles between rank-index paired base and RL-trained singular directions remain small and smooth across attention and MLP layers. This suggests that rank-index tracking is qualitatively stable under RLVR in these runs. These plots are used only as a sanity check. Formal claims use projector-based diagnostics.

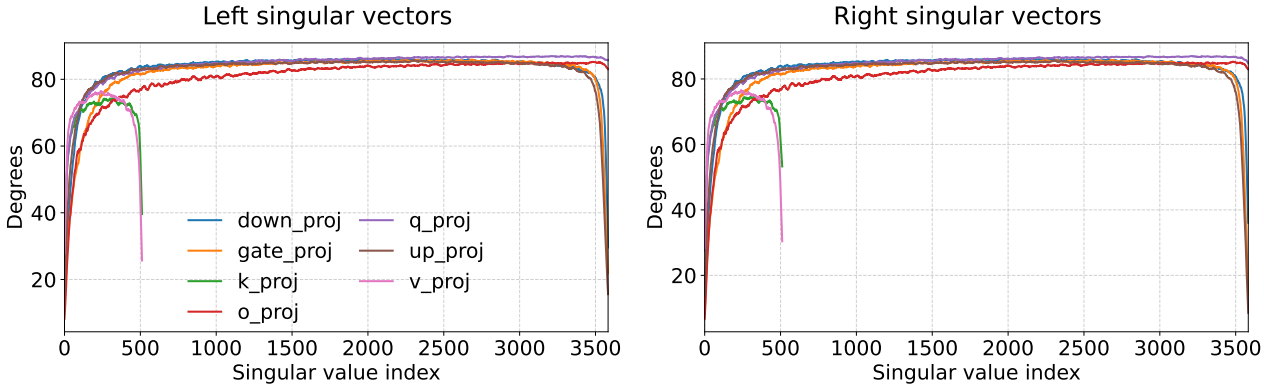


Figure 12 | **Qualitative singular-direction rotations under SFT.** Mean angles between rank-index paired base and SFT singular directions increase toward near-orthogonality for many ranks. In high-dimensional spaces, unrelated directions are typically nearly orthogonal, so this indicates that simple rank-index tracking becomes much less informative under SFT.

B. Fixed-Spectrum Theory and Frame-Adaptability Diagnostics

Notation. Let $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and $q = \min\{d_{\text{out}}, d_{\text{in}}\}$. We write

$$\sigma(W) = \big(\sigma_1(W), \dots, \sigma_q(W)\big)$$

for all singular values in nonincreasing order, including zeros with multiplicity. A thin SVD is

$$W = U\Sigma V^\top, \qquad U \in \mathrm{St}(d_{\text{out}}, q), \quad V \in \mathrm{St}(d_{\text{in}}, q), \quad \Sigma = \mathrm{Diag}(\sigma(W)).$$

Here $\mathrm{Diag}(a)$ constructs a diagonal matrix from a vector, whereas $\mathrm{diag}(A)$ extracts the diagonal of a matrix.

B.1. The Fixed-Spectrum Family and Nearest Representatives

Proposition B.1 (Two-sided characterization of the fixed-spectrum family)**.** *For any source matrix W_0,*

$$\begin{aligned} \mathcal{F}(W_0) &:= \left\{ Z \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}} : \sigma(Z) = \sigma(W_0) \right\} \\ &= \left\{ Q_L W_0 Q_R^\top : Q_L \in \mathrm{O}(d_{\text{out}}),\ Q_R \in \mathrm{O}(d_{\text{in}}) \right\}. \end{aligned} \tag{39}$$

Proof. Orthogonal multiplication preserves singular values, proving one inclusion. For the reverse inclusion, write

$$W_0 = U_0 \Sigma_0 V_0^\top, \qquad Z = U_Z \Sigma_0 V_Z^\top.$$

Extend U_0, U_Z and V_0, V_Z to complete orthogonal bases and choose Q_L, Q_R mapping the source bases to the target bases. Then $Q_L W_0 Q_R^\top = Z$. □

Proposition B.2 (Exact distance to the fixed-spectrum family)**.** *For $W, W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$,*

$$\mathrm{dist}_F\big(W, \mathcal{F}(W_0)\big) = \|\sigma(W) - \sigma(W_0)\|_2. \tag{40}$$

If $W = U\Sigma V^\top$ and $\Sigma_0 = \mathrm{Diag}(\sigma(W_0))$, then

$$U\Sigma_0 V^\top \in \operatorname*{arg\,min}_{Z \in \mathcal{F}(W_0)} \|W - Z\|_F. \tag{41}$$

Proof. Every $Z \in \mathcal{F}(W_0)$ has Frobenius norm $\|W_0\|_F$, so

$$\|W - Z\|_F^2 = \|W\|_F^2 + \|W_0\|_F^2 - 2\langle W, Z\rangle_F.$$

By von Neumann's trace inequality,

$$\langle W, Z\rangle_F \le \sum_{k=1}^{q} \sigma_k(W)\sigma_k(W_0).$$

The bound is attained by $Z = U\Sigma_0 V^\top$. Substitution gives

$$\min_{Z \in \mathcal{F}(W_0)} \|W - Z\|_F^2 = \sum_{k=1}^{q} \big(\sigma_k(W) - \sigma_k(W_0)\big)^2.$$

□

Remark B.3 (Gauge and nonuniqueness of a rebased representative). The distance to the fixed-spectrum family and the set of nearest points are basis-independent, but a particular representative $U\Sigma_0 V^\top$ need not be unique.

Suppose a block C of the target spectrum is exactly repeated. Replacing

$$U_C \leftarrow U_C R, \qquad V_C \leftarrow V_C R, \qquad R \in \mathrm{O}(|C|)$$

leaves W unchanged but produces the rebased block

$$U_C R \Sigma_{0,C} R^\top V_C^\top.$$

Every such representative is a nearest point in $\mathcal{F}(W_0)$. It is identical for all R when $\Sigma_{0,C}$ is proportional to the identity. More generally, with $\bar{\sigma}_C$ denoting the block mean,

$$\left\| R\Sigma_{0,C}R^\top - \Sigma_{0,C} \right\|_F \le 2 \left\| \Sigma_{0,C} - \bar{\sigma}_C I \right\|_F . \tag{42}$$

Thus the ambiguity is small when the source spectrum is nearly flat inside the cluster. Near-degenerate but distinct singular values do not create exact gauge freedom, but can make the numerical singular vectors ill-conditioned. Our rebasing experiment evaluates the representative selected by the numerical SVD; Appendix A reports a qualitative stability check.

Corollary B.4 (Finite-horizon spectral drift)**.** *Let $W_0, \dots, W_T$ be a matrix trajectory and $E_t = W_{t+1} - W_t$. Then*

$$\|\sigma(W_t) - \sigma(W_0)\|_2 = \mathrm{dist}_F\big(W_t, \mathcal{F}(W_0)\big) \le \|W_t - W_0\|_F \le \sum_{\tau=0}^{t-1} \|E_\tau\|_F. \tag{43}$$

B.2. Spectral-Distance Metrics and Dimension-Aware Calibration

For matrix ℓ, let $\Delta W^{(\ell)} = W_1^{(\ell)} - W_0^{(\ell)}$. We use

$$\begin{aligned} \delta_\Sigma^{(\ell)} &= \frac{\mathrm{dist}_F\left(W_1^{(\ell)}, \mathcal{F}(W_0^{(\ell)})\right)}{\|W_0^{(\ell)}\|_F}, \\ \rho_\Sigma^{(\ell)} &= \frac{\mathrm{dist}_F\left(W_1^{(\ell)}, \mathcal{F}(W_0^{(\ell)})\right)}{\|\Delta W^{(\ell)}\|_F}. \end{aligned} \tag{44}$$

The pooled residual reported in the main text is

$$\rho_\Sigma^{\mathrm{pool}} = \left[\frac{\sum_\ell \mathrm{dist}_F^2\left(W_1^{(\ell)}, \mathcal{F}(W_0^{(\ell)})\right)}{\sum_\ell \|\Delta W^{(\ell)}\|_F^2} \right]^{1/2}. \tag{45}$$

Proposition B.5 (First-order spectrum-changing subspace and isotropic calibration)**.** *Assume W_0 is full rank with simple singular values and write $W_0 = U_0\Sigma_0 V_0^\top$. The first-order spectrum-changing subspace at W_0, equivalently the normal space of its fixed-spectrum family, is*

$$N_{W_0}\mathcal{F}(W_0) = \left\{ U_0 \,\mathrm{Diag}(a) V_0^\top : a \in \mathbb{R}^q \right\}. \tag{46}$$

The orthogonal projection of a perturbation H onto this spectrum-changing subspace is

$$\Pi_{\mathrm{spec}}(H) = U_0 \,\mathrm{Diag}\left(\mathrm{diag}(U_0^\top H V_0)\right) V_0^\top. \tag{47}$$

If $\mathrm{vec}(H)/\|H\|_F$ *is isotropic, then*

$$\mathbb{E}\left[\frac{\|\Pi_{\mathrm{spec}}(H)\|_F^2}{\|H\|_F^2}\right] = \frac{q}{d_{\mathrm{out}}d_{\mathrm{in}}}. \tag{48}$$

Consequently,

$$\kappa_{\mathrm{spec}} = \frac{d_{\mathrm{out}}d_{\mathrm{in}}}{q}\frac{\|\operatorname{diag}(U_0^\top H V_0)\|_2^2}{\|H\|_F^2} \tag{49}$$

satisfies $\mathbb{E}[\kappa_{\mathrm{spec}}] = 1$. *Thus, one is the isotropic dimensional reference under this normalization.*

Proof. For a simple full-rank spectrum, fixing the q singular values imposes q independent first-order constraints. The matrices $\{u_{0,k}v_{0,k}^\top\}_{k=1}^q$ form an orthonormal basis for this spectrum-changing subspace, which gives Equation (47). An isotropic direction places expected squared energy in a fixed subspace in proportion to its dimension. The spectrum-changing and ambient dimensions are q and $d_{\mathrm{out}}d_{\mathrm{in}}$, respectively. □

B.3. Reconstruction Classes and Unexplained-Update Ratios

For $r < q$, write

$$W_t^{(r)} = U_t^{(r)}\Sigma_t^{(r)}\big(V_t^{(r)}\big)^\top, \qquad P_t^{(r)} = U_t^{(r)}\big(U_t^{(r)}\big)^\top, \qquad Q_t^{(r)} = V_t^{(r)}\big(V_t^{(r)}\big)^\top.$$

We call a transition $W_j \to W_i$ admissible at rank r when

$$\sigma_r(W_t) > \sigma_{r+1}(W_t), \qquad t \in \{i, j\},$$

so that the retained projectors are uniquely defined.

Let $A = W_i^{(r)}$. Define

$$\begin{aligned} \mathcal{C}_{\mathrm{mix}} &= \left\{U_j^{(r)} C \big(V_j^{(r)}\big)^\top : C \in \mathbb{R}^{r\times r}\right\}, \\ \mathcal{C}_L &= \left\{U_j^{(r)} B : B \in \mathbb{R}^{r\times d_{\mathrm{in}}}\right\}, \\ \mathcal{C}_R &= \left\{B\big(V_j^{(r)}\big)^\top : B \in \mathbb{R}^{d_{\mathrm{out}}\times r}\right\}, \\ \mathcal{F}_j^{(r)} &= \left\{U\Sigma_j^{(r)}V^\top : U \in \mathrm{St}(d_{\mathrm{out}}, r), V \in \mathrm{St}(d_{\mathrm{in}}, r)\right\}. \end{aligned} \tag{50}$$

Proposition B.6 (Optimal checkpoint reconstructions)**.** *The Frobenius-optimal reconstructions are*

$$\begin{aligned} \widehat{W}_{\mathrm{mix}} &= P_j^{(r)} A Q_j^{(r)}, \qquad \widehat{W}_L = P_j^{(r)} A, \\ \widehat{W}_R &= A Q_j^{(r)}, \qquad \widehat{W}_{\mathrm{iso}} = U_i^{(r)}\Sigma_j^{(r)}\big(V_i^{(r)}\big)^\top. \end{aligned} \tag{51}$$

The remix reconstruction can equivalently be written as

$$\widehat{W}_{\mathrm{mix}} = U_j^{(r)} B_{ij}^{(r)} \big(V_j^{(r)}\big)^\top, \qquad B_{ij}^{(r)} = \big(U_j^{(r)}\big)^\top W_i^{(r)} V_j^{(r)}. \tag{52}$$

The core $B_{ij}^{(r)}$ *is unconstrained and may rotate, mix, rescale, and change the represented spectrum. Thus,* $\mathcal{C}_{\mathrm{mix}}$ *fixes only the incoming input and output spans, not the individual frames or spectrum. Moreover,*

$$\widehat{W}_{\mathrm{iso}} \in \operatorname*{arg\,min}_{Z\in\mathcal{F}_j^{(r)}} \|A - Z\|_F, \tag{53}$$

and the corresponding normalized residuals are

$$e_h = \frac{\|A - \widehat{W}_h\|_F}{\|A\|_F}, \qquad h \in \{\text{mix}, L, R, \text{iso}\}, \tag{54}$$

while the displacement from the unchanged incoming checkpoint, used as a scale reference, is

$$e_{\text{drift}} := \frac{\|W_i^{(r)} - W_j^{(r)}\|_F}{\|W_i^{(r)}\|_F}. \tag{55}$$

This reference is not an additional reconstruction hypothesis. The isospectral residual has the closed form

$$e_{\text{iso}} = \frac{\|\Sigma_i^{(r)} - \Sigma_j^{(r)}\|_F}{\|\Sigma_i^{(r)}\|_F}. \tag{56}$$

The corresponding unexplained-update ratios are

$$u_h := \frac{e_h}{e_{\text{drift}}} = \frac{\|W_i^{(r)} - \widehat{W}_h\|_F}{\|W_i^{(r)} - W_j^{(r)}\|_F}, \qquad h \in \{\text{mix}, L, R, \text{iso}\}. \tag{57}$$

If $e_{\text{drift}} > 0$*, then*

$$0 \le u_h \le 1, \qquad h \in \{\text{mix}, L, R, \text{iso}\}.$$

Proof. The first three expressions are orthogonal projections onto C_{mix}, C_L, and C_R. In particular,

$$P_j^{(r)} W_i^{(r)} Q_j^{(r)} = U_j^{(r)} \left[\left(U_j^{(r)}\right)^\top W_i^{(r)} V_j^{(r)} \right] \left(V_j^{(r)}\right)^\top,$$

which gives Equation (52). The isospectral expression follows by applying Proposition B.2 to the rank-r matrices $W_i^{(r)}$ and $W_j^{(r)}$.

Finally, $W_j^{(r)}$ belongs to all four reconstruction classes. Therefore, the optimality of $\widehat{W}_h$ gives

$$\|W_i^{(r)} - \widehat{W}_h\|_F \le \|W_i^{(r)} - W_j^{(r)}\|_F,$$

which proves $0 \le u_h \le 1$.

□

Remark B.7 (Interpretation of the reconstruction test). The reconstruction classes have different dimensions, so we do not interpret their residual ordering as a complexity-normalized model-selection result. The purpose of the test is parameterization sufficiency: whether freezing the corresponding incoming span still permits a low-residual endpoint description. Because C_{mix}, C_L, and C_R are more permissive than the associated fixed-spectrum restrictions, their residuals are optimistic lower bounds on the error caused by freezing that frame structure.

Remark B.8 (Gauge and interpretation). The projectors $P_t^{(r)}, Q_t^{(r)}$ and all four residual values are invariant to basis changes inside the retained subspaces. If $\Sigma_i^{(r)}$ has repeated blocks, $\widehat{W}_{\text{iso}}$ may not be the unique nearest point, but e_{iso} remains unique.

The remix class contains the internal fixed-spectrum slice

$$\left\{ U_j^{(r)} R_U \Sigma_j^{(r)} R_V^\top \left(V_j^{(r)}\right)^\top : R_U, R_V \in \mathrm{O}(r) \right\}.$$

Hence, failure of the more permissive remix class also rules out an explanation based solely on internal fixed-spectrum rotations.

Remark B.9 (Why $r < q$). At full thin rank, at least one of $P_t^{(r)}, Q_t^{(r)}$ is the identity; for square matrices both are. The remix and one-sided tests would therefore degenerate. We use $r < q$ so that both retained subspaces are proper and informative.

B.4. Rank Sensitivity of the Frame-Adaptability Test

Rank sensitivity. For $\alpha \in \{0.2, 0.4, 0.6, 0.8, 0.9\}$, we use the integer truncation rank

$$r_\alpha = \max\{1, \lfloor \alpha q \rfloor\}. \tag{58}$$

At each α, a matrix–transition pair is included only when r_α is admissible at both endpoints; inadmissible pairs are omitted at that value of α. For each included pair, we compute

$$u_h = \frac{e_h}{e_{\text{drift}}}, \qquad h \in \{\text{mix}, L, R, \text{iso}\}. \tag{59}$$

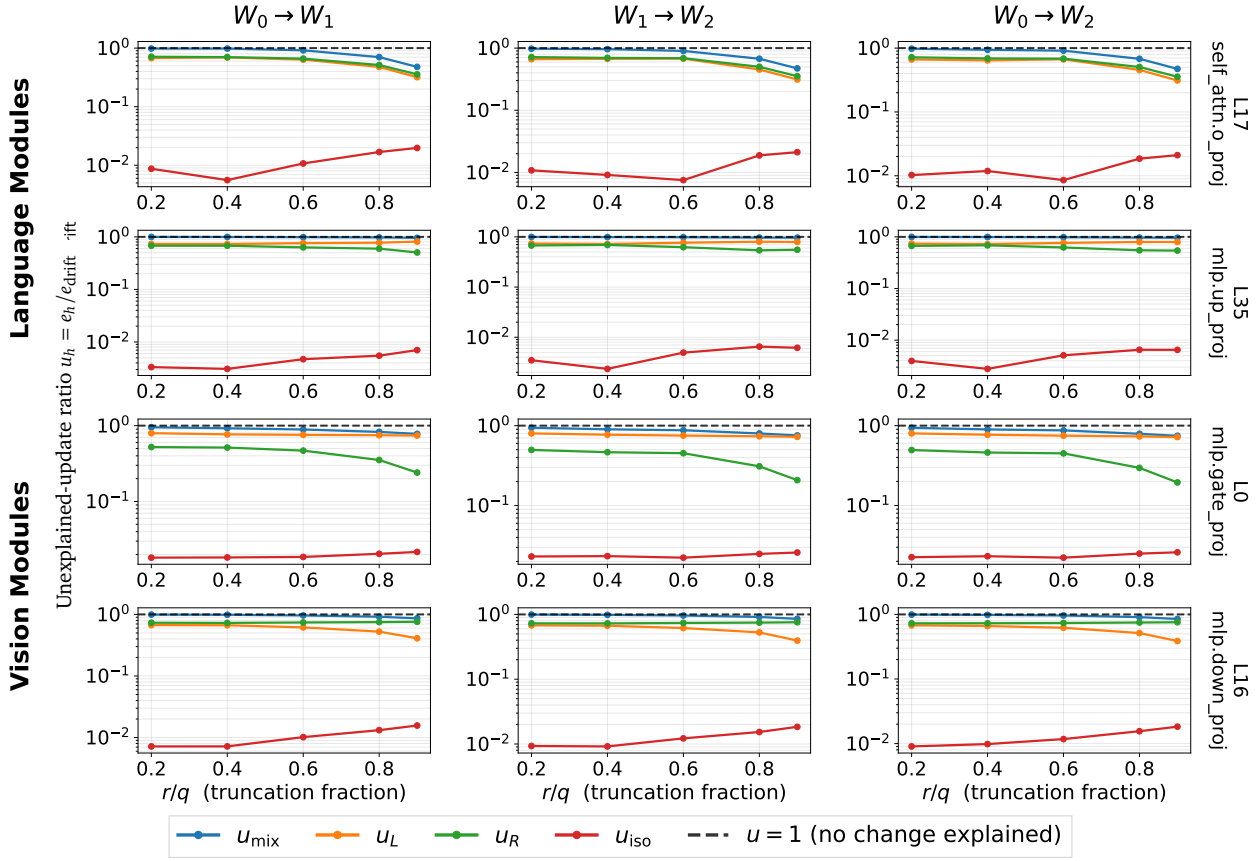


Figure 13 | **Rank sensitivity of the unexplained-update ratio.** Unexplained-update ratio u_h versus the truncation fraction r/q for representative language and vision modules and all three transitions; the dashed line $u = 1$ corresponds to retaining the unmodified incoming checkpoint. The ordering in Figure 5 is stable across ranks: u_{iso} stays below 2% throughout the sweep, while the subspace-retaining alternatives leave tens of percent unexplained at every rank. At $r/q = 0.2$, the median across the twelve displayed matrix–transition panels is $u_{\text{mix}} = 0.99$. The gap is already pronounced at small r: restricting the reconstruction to the dominant incoming input and output spans explains little of the observed rank-r checkpoint change, whereas retaining the incoming spectrum and adapting both frames remains low-residual.

C. First-Order Properties of the ISO Parameterization

C.1. Proof of Proposition 4.1

Proof. We prove the two claims in Proposition 4.1.

First, consider a differentiable curve $W(\varepsilon) \in \mathcal{F}(W_0)$ with $W(0) = W = U\Sigma_0 V^\top$. Since all q singular values are positive and simple, we may locally choose differentiable Stiefel representatives $U(\varepsilon)$ and $V(\varepsilon)$ such that

$$W(\varepsilon) = U(\varepsilon)\Sigma_0 V(\varepsilon)^\top, \qquad U(0) = U, \quad V(0) = V. \tag{60}$$

Let

$$\dot{U} = \frac{d}{d\varepsilon} U(\varepsilon)\Big|_{\varepsilon=0}, \qquad \dot{V} = \frac{d}{d\varepsilon} V(\varepsilon)\Big|_{\varepsilon=0}, \qquad \dot{W} = \frac{d}{d\varepsilon} W(\varepsilon)\Big|_{\varepsilon=0}. \tag{61}$$

Differentiating the fixed-spectrum representation gives

$$\dot{W} = \dot{U}\Sigma_0 V^\top + U\Sigma_0 \dot{V}^\top. \tag{62}$$

Multiplying on the left by $U^\top$ and on the right by V yields

$$U^\top \dot{W} V = U^\top \dot{U}\,\Sigma_0 + \Sigma_0\,\dot{V}^\top V. \tag{63}$$

Because $U(\varepsilon)$ and $V(\varepsilon)$ remain on the Stiefel manifold, differentiating $U(\varepsilon)^\top U(\varepsilon) = I$ and $V(\varepsilon)^\top V(\varepsilon) = I$ at $\varepsilon = 0$ gives

$$U^\top \dot{U} + \dot{U}^\top U = 0, \qquad V^\top \dot{V} + \dot{V}^\top V = 0. \tag{64}$$

Thus $U^\top \dot{U}$ and $\dot{V}^\top V$ are skew-symmetric. Since Σ_0 is diagonal, both $U^\top \dot{U}\,\Sigma_0$ and $\Sigma_0 \dot{V}^\top V$ have zero diagonal. Therefore

$$\operatorname{diag}(U^\top \dot{W} V) = 0. \tag{65}$$

Second, let H be an arbitrary perturbation and define

$$W(\varepsilon) = W + \varepsilon H. \tag{66}$$

For each singular value, choose differentiable singular vectors $u_k(\varepsilon)$ and $v_k(\varepsilon)$ with singular value $\sigma_k(\varepsilon)$. Then

$$\sigma_k(\varepsilon) = u_k(\varepsilon)^\top W(\varepsilon) v_k(\varepsilon). \tag{67}$$

Differentiating at $\varepsilon = 0$ gives

$$\sigma_k'(0) = \dot{u}_k^\top W v_k + u_k^\top H v_k + u_k^\top W \dot{v}_k. \tag{68}$$

Using $W v_k = \sigma_k u_k$ and $u_k^\top W = \sigma_k v_k^\top$, this becomes

$$\sigma_k'(0) = \sigma_k \dot{u}_k^\top u_k + u_k^\top H v_k + \sigma_k v_k^\top \dot{v}_k. \tag{69}$$

Since $u_k(\varepsilon)$ and $v_k(\varepsilon)$ remain unit vectors, differentiating $u_k(\varepsilon)^\top u_k(\varepsilon) = 1$ and $v_k(\varepsilon)^\top v_k(\varepsilon) = 1$ gives

$$\dot{u}_k^\top u_k = 0, \qquad v_k^\top \dot{v}_k = 0. \tag{70}$$

Hence

$$\frac{d}{d\varepsilon} \sigma_k(W + \varepsilon H)\Big|_{\varepsilon=0} = u_k^\top H v_k. \tag{71}$$

Collecting this identity for $k = 1, \ldots, q$ shows that $\operatorname{diag}(U^\top H V)$ is exactly the vector of first-order singular-value changes. This proves the proposition. □

C.2. Factor-Gradient Derivation

Let $\mathcal{L}(W)$ be the RLVR loss, let $G_W = \nabla_W \mathcal{L}(W)$, and use the fixed-spectrum parameterization

$$W(U, V) = U\Sigma_0 V^\top \in \mathcal{F}(W_0). \tag{72}$$

For infinitesimal Stiefel-factor changes $\xi_U \in T_U \mathrm{St}(d_{\text{out}}, q)$ and $\xi_V \in T_V \mathrm{St}(d_{\text{in}}, q)$, the induced weight motion is

$$\dot{W} = \xi_U \Sigma_0 V^\top + U\Sigma_0 \xi_V^\top. \tag{73}$$

The first-order change in loss is

$$\begin{aligned} d\mathcal{L}[\dot{W}] &= \langle G_W, \dot{W} \rangle_F \\ &= \langle G_W V \Sigma_0, \xi_U \rangle_F + \langle G_W^\top U \Sigma_0, \xi_V \rangle_F. \end{aligned} \tag{74}$$

Therefore the raw factor gradients used by ISO are

$$G_U = G_W V \Sigma_0, \qquad G_V = G_W^\top U \Sigma_0. \tag{75}$$

These are the Euclidean gradients in the ambient factor coordinates. ISO does not explicitly construct a weight-space projected gradient. Instead, the fixed-spectrum parameterization and polar retraction restrict every represented iterate to $\mathcal{F}(W_0)$ up to numerical precision.

D. A Same-Base SFT–RLVR Case Study

The main-text case study in Section 2.1 compares an SFT transition and an RLVR transition initialized from different base checkpoints. To control for this difference, we repeat the analysis using two public 14B checkpoints derived from the same Qwen2.5-14B base:

$$\texttt{Qwen2.5-14B} \longrightarrow \begin{cases} \texttt{DeepSeek-R1-Distill-Qwen-14B} & \text{(distilled SFT)}, \\ \texttt{Qwen-2.5-14B-SimpleRL-Zoo} & \text{(RLVR)}. \end{cases}$$

Both endpoints are measured relative to the shared base checkpoint. We restrict the comparison to the seven transformer projection matrices whose parameter names and shapes match the shared Qwen2.5-14B backbone, avoiding differences introduced by checkpoint-specific tokenizer or configuration changes. This comparison therefore controls for the starting backbone weights, while the training data, objectives, and optimization procedures remain intentionally different. Figure 14 summarizes the results.

Raw spectral proximity. The two endpoints differ sharply in their spectral displacement from the shared base. At the scale of Figure 14b, the per-rank changes of the RLVR endpoint are visually near zero. Its base-normalized spectral distance is approximately

$$\delta_\Sigma \approx 6 \times 10^{-6}$$

across layers and projection types. By contrast, the distilled-SFT endpoint substantially changes the spectrum: δ_Σ is approximately 10^{-2} for most projection types and reaches approximately 0.4 for the attention output projection.

Dimension-aware calibration. Raw proximity alone does not establish a preferentially spectrum-preserving update direction, so we additionally report the dimension-normalized spectrum-changing energy κ_{spec}. Across projection types, RLVR yields mean values between 1.4 and 3.2, with an overall mean of 1.9. These values remain order-one relative to the isotropic dimensional reference, with the largest deviations concentrated in the earliest layers.

The distilled-SFT endpoint yields mean κ_{spec} values between 117 and 4810, with an overall mean of 873 and the largest concentration in the attention output projection. The same-base comparison therefore reproduces the main-text contrast: SFT concentrates its displacement in spectrum-changing coordinates by roughly two to three orders of magnitude, whereas RLVR exhibits no comparable concentration. This difference cannot be explained by the two transitions starting from different backbone checkpoints.

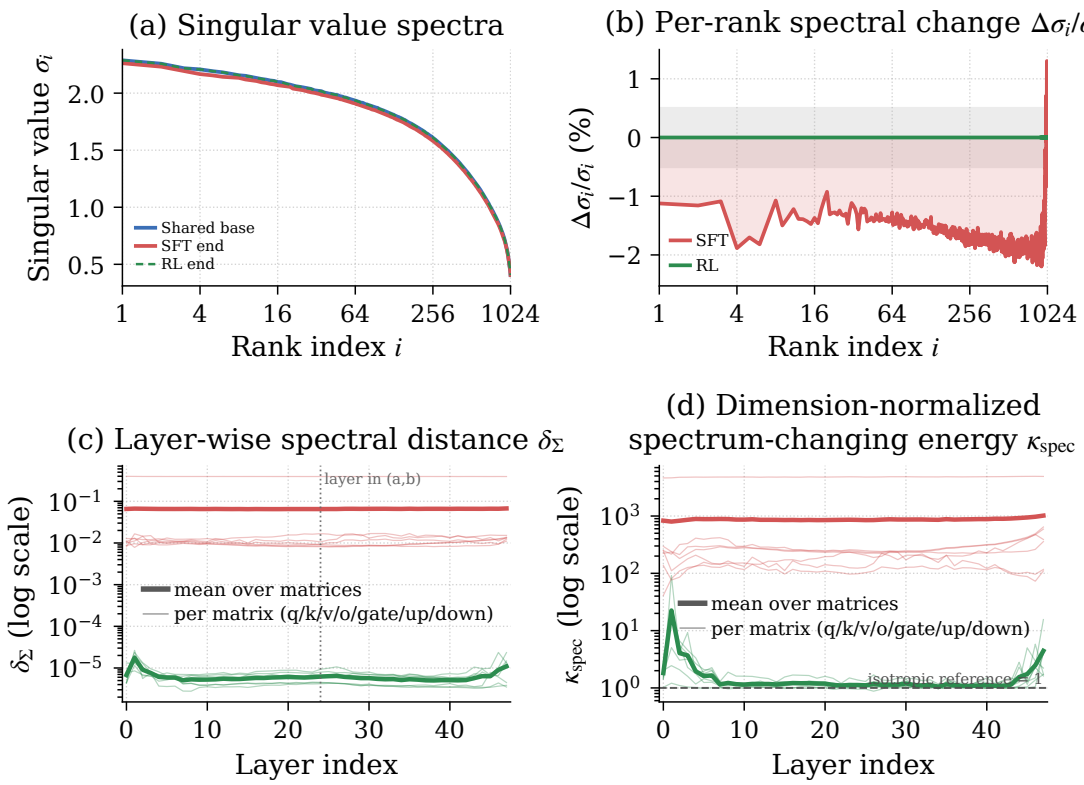


Figure 14 | **Same-base spectral dynamics.** The distilled-SFT and RLVR endpoints are both compared with their shared `Qwen2.5-14B` base. **(a)** Singular-value profiles for `v_proj` in layer 24. **(b)** Rank-wise relative spectral change: the RLVR curve remains near zero at the plotted scale, whereas distilled SFT systematically reshapes the spectrum. **(c)** Layer-wise base-normalized spectral distance δ_Σ, shown per projection matrix (thin) and averaged across projection types (thick); the RLVR endpoint remains several orders of magnitude closer to the fixed-spectrum family of the shared base. **(d)** Dimension-normalized spectrum-changing energy κ_{spec}: RLVR remains order-one relative to the isotropic dimensional reference, whereas distilled SFT lies roughly two to three orders of magnitude above it.

E. ISO-Merger Details

E.1. Setup

For each 2D weight matrix, let

$$W_0 = U_0 \Sigma_0 V_0^\top \tag{76}$$

be the SVD of the shared base model, and let

$$W_i = U_i \Sigma_i V_i^\top, \qquad i = 1, \ldots, K \tag{77}$$

be the corresponding SVDs of K RL experts fine-tuned from the same base. ISO-Merger uses the base spectrum Σ_0 as the shared spectrum and merges expert-specific motion through the Stiefel factors (U_i, V_i).

E.2. Sign canonicalization

The SVD determines each singular-vector pair only up to a joint sign: (u_k, v_k) and $(-u_k, -v_k)$ represent the same matrix. Numerical routines choose these signs arbitrarily, so an unaligned displacement

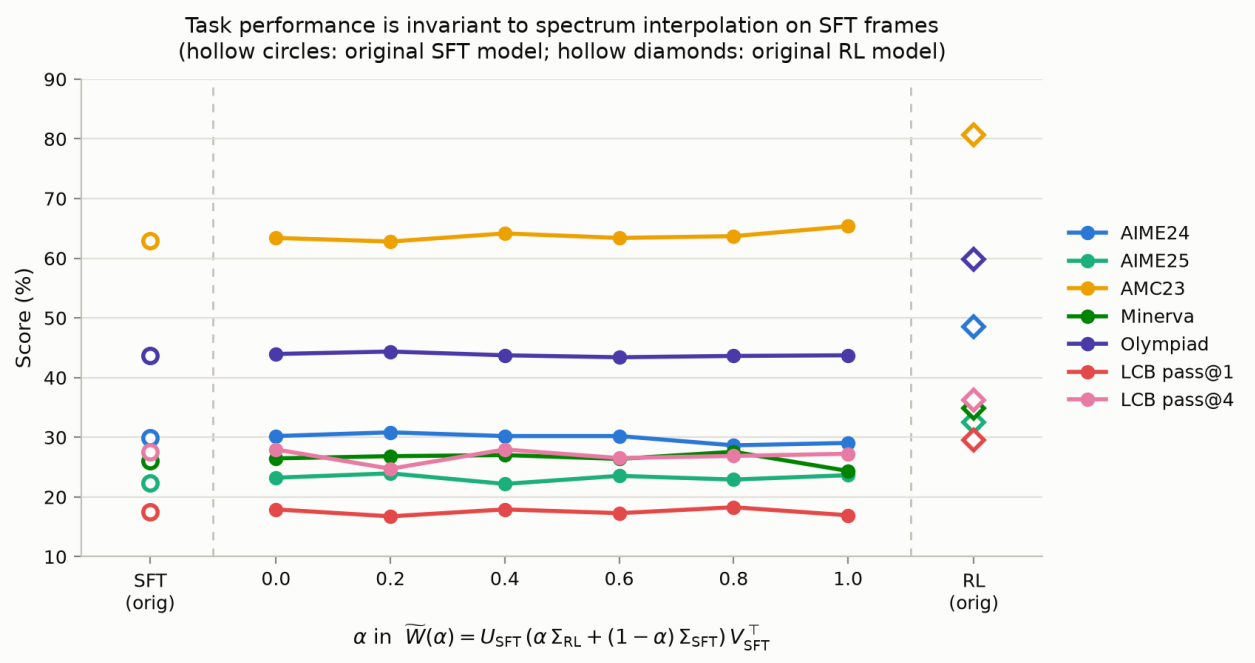


Figure 15 | **Task performance is insensitive to spectral interpolation in the SFT singular frames.** We hold the SFT left and right singular frames fixed and interpolate only the spectrum, $\widetilde{W}(\alpha) = U_{\mathrm{SFT}}[(1-\alpha)\Sigma_{\mathrm{SFT}} + \alpha\Sigma_{\mathrm{RL}}]V_{\mathrm{SFT}}^{\top}$, for $\alpha \in [0,1]$. Performance remains nearly unchanged as an arbitrary proportion of the RL spectrum is substituted into the SFT frames. Hollow circles and diamonds show the original SFT and RL checkpoints, respectively. This complements Figure 4(a), which performs the converse intervention by restoring the SFT/base spectrum in the RL-trained frames.

$U_i - U_0$ can contain spurious columns of norm ~2 that reflect bookkeeping rather than learned motion. We canonicalize the gauge column-wise against the base:

$$s_{i,k} := \mathrm{sign}\big(\langle u_{0,k}, u_{i,k}\rangle\big), \qquad \mathrm{sign}(0) := 1, \tag{78}$$

$$u_{i,k} \leftarrow s_{i,k}u_{i,k}, \quad v_{i,k} \leftarrow s_{i,k}v_{i,k}. \tag{79}$$

The same sign is applied to both frames, so the alignment is a valid gauge transformation that leaves W_i unchanged. Exactly repeated singular values additionally admit a joint in-block rotation $U_c \to U_c Q$, $V_c \to V_c Q$; generic weight matrices have simple spectra, and we do not canonicalize this freedom explicitly.

E.3. Stiefel tangent projection

For an anchor $U_0 \in \mathrm{St}(m,q)$, the tangent space is

$$T_{U_0}\mathrm{St}(m,q) = \{\xi \in \mathbb{R}^{m\times q} : U_0^{\top}\xi + \xi^{\top}U_0 = 0\}. \tag{80}$$

For expert i, define the frame displacements

$$\Delta U_i := U_i - U_0, \qquad \Delta V_i := V_i - V_0. \tag{81}$$

ISO-Merger uses their orthogonal projections onto the corresponding tangent spaces:

$$\xi_{U,i} = \Pi_{U_0}(\Delta U_i) = \Delta U_i - U_0\ \mathrm{sym}(U_0^{\top}\Delta U_i), \qquad \mathrm{sym}(A) = \frac{1}{2}(A + A^{\top}). \tag{82}$$

The right-frame tangent $\xi_{V,i} = \Pi_{V_0}(\Delta V_i)$ is computed analogously.

E.4. Top-k_{keep} singular-mode masking

Let $\rho_{\text{keep}} \in (0, 1]$ and define

$$k_{\text{keep}} := \lfloor \rho_{\text{keep}} q \rceil, \qquad D_{\text{keep}} := \text{Diag}\left(\left(\mathbf{1}\{k \le k_{\text{keep}}\}\right)_{k=1}^{q}\right). \tag{83}$$

Thus, D_{keep} keeps the leading k_{keep} columns. Empirically, the trailing modes (columns associated with small singular values) are estimated noisily and disagree across experts, and retaining them degrades the merged model. We therefore mask the trailing columns of each displacement,

$$\widetilde{\xi}_{U,i} = \xi_{U,i} D_{\text{keep}}, \qquad \widetilde{\xi}_{V,i} = \xi_{V,i} D_{\text{keep}}. \tag{84}$$

Masking is a coordinate-selection operation and need not preserve the Stiefel tangent constraints. We use the masked displacements only to define local first-order effect proxies and impose feasibility after aggregation. We use $\rho_{\text{keep}} = 0.9$ unless otherwise stated.

E.5. Retention Coefficients with a Unit-Retention Target

For expert i, define the local first-order effect proxy

$$g_i = \widetilde{\xi}_{U,i} \Sigma_0 V_0^\top + U_0 \Sigma_0 \widetilde{\xi}_{V,i}^\top. \tag{85}$$

We then form the Gram matrix

$$\Gamma_{ij} := \langle g_i, g_j \rangle_F, \qquad \Gamma \in \mathbb{R}^{K \times K}. \tag{86}$$

For $c \in \mathbb{R}^K$, define $g(c) := \sum_{i=1}^K c_i g_i$. For each nonzero proxy g_i, define its self-retention in the merged proxy by

$$\text{ret}_i(c) := \frac{\langle g(c), g_i \rangle_F}{\|g_i\|_F^2} = \frac{(\Gamma c)_i}{\Gamma_{ii}}. \tag{87}$$

For any zero-norm proxy g_i, we omit the corresponding row and column from the Gram system and set its coefficient to zero. The ideal target $\text{ret}_i(c) = 1$ gives

$$\Gamma c = b, \qquad b := \text{diag}(\Gamma) \in \mathbb{R}^K. \tag{88}$$

We solve the ridge-stabilized system

$$(\Gamma + \lambda_{\text{ridge}} I_K)\bar{c} = b, \tag{89}$$

and clip the coefficients to prevent sign reversal or over-amplification of any expert:

$$c^\star = \text{clip}(\bar{c},\, c_{\min},\, c_{\max}), \qquad [c_{\min}, c_{\max}] = [0, 1.5]. \tag{90}$$

We use $\lambda_{\text{ridge}} = 10^{-12}$ as a small numerical stabilizer. The procedure targets, rather than guarantees, unit self-retention after ridge stabilization, clipping, aggregate tangent projection, and retraction.

E.6. Retraction and reconstruction

We combine the masked displacements and project the result onto the tangent space at the anchor:

$$\xi_{U,\star} = \Pi_{U_0}\Big(\sum_{i=1}^K c_i^\star \widetilde{\xi}_{U,i}\Big), \qquad \xi_{V,\star} = \Pi_{V_0}\Big(\sum_{i=1}^K c_i^\star \widetilde{\xi}_{V,i}\Big). \tag{91}$$

For a thin SVD

$$X = P_X S_X Q_X^\top,$$

we define the polar factor by

$$\text{polar}(X) := P_X Q_X^\top. \tag{92}$$

When X has full column rank, this is equivalently

$$\text{polar}(X) = X(X^\top X)^{-1/2}.$$

For rank-deficient X, the SVD expression selects one valid nearest Stiefel factor, which need not be unique. Here the aggregate displacements have been projected into the tangent spaces, so

$$(U_0 + \xi_{U,\star})^\top (U_0 + \xi_{U,\star}) = I + \xi_{U,\star}^\top \xi_{U,\star} \succ 0,$$

with the analogous identity for $V_0 + \xi_{V,\star}$. Thus the ISO-Merger retraction arguments are full column rank. The final merged factors and matrix are

$$U_\star = \text{polar}(U_0 + \xi_{U,\star}), \qquad V_\star = \text{polar}(V_0 + \xi_{V,\star}), \qquad W_\star = U_\star \Sigma_0 V_\star^\top, \tag{93}$$

which carries the base spectrum Σ_0 up to numerical precision.

E.7. Merged parameter scope

All per-layer 2D projection matrices and the embedding/unembedding matrices are merged with the construction above. One-dimensional parameters (normalization scales, attention biases) are merged with a standard task-vector average,

$$w_\star = w_0 + \frac{1}{K} \sum_{i=1}^{K} (w_i - w_0). \tag{94}$$

E.8. Full ISO-Merger algorithm

Full algorithm is in Algorithm 1.

F. Data-Free Merging Experimental Details

Backbones and experts. For the Qwen2.5 setting, we use `Qwen2.5-7B-Instruct` as the shared base and merge three RLVR experts: CURE for coding [24], ToolRL for tool use [25], and MemAgent for long-context memory [26]. For the DeepSeek-R1-Distill setting, we use `DeepSeek-R1-Distill-Qwen-1.5B` as the shared base and merge Archer2.0 for coding [32] and JustRL for math [33].

Merging baselines. We compare against five data-free baselines: Task Arithmetic [27], TIES [28], TSV-Merge [29], RAM [30], and OrthoMerge-G-TIES [31]. Task Arithmetic linearly combines task vectors. The first four methods operate on Euclidean task-vector representations, while OrthoMerge-G-TIES additionally introduces a geometry-aware orthogonal component and is described separately below. Let θ_0 denote the shared base model and let $\{\theta_i\}_{i=1}^K$ be K expert models fine-tuned from this same initialization. The task vector of expert i is

$$\tau_i := \theta_i - \theta_0. \tag{95}$$

A data-free merging method constructs a merged update τ_{merge} without using training data and outputs

$$\theta_{\text{merge}} = \theta_0 + \tau_{\text{merge}}. \tag{96}$$

For a matrix-shaped parameter in layer ℓ, we use the layer-wise notation

$$\Delta W_i^{(\ell)} := W_i^{(\ell)} - W_0^{(\ell)}, \qquad W^{(\ell)} \in \mathbb{R}^{m_\ell \times n_\ell}. \tag{97}$$

Algorithm 1 ISO-Merger

Require: Base model θ_0, expert models $\{\theta_i\}_{i=1}^K$, keep ratio ρ_{keep} (0.9), ridge λ_{ridge} (10^{-12}), clip range $[c_{\min}, c_{\max}]$ ($[0, 1.5]$)
Ensure: Merged model $\theta_\star$

for each 2D weight matrix W_0 (per-layer projections and embedding/unembedding) **do**
  Compute $W_0 = U_0 \Sigma_0 V_0^\top$ with q singular values
  $k_{\text{keep}} \leftarrow \text{round}(\rho_{\text{keep}} q)$
  $D_{\text{keep}} \leftarrow \text{Diag}((\mathbf{1}\{k \le k_{\text{keep}}\})_{k=1}^q)$
  for each expert i **do**
    Compute $W_i = U_i \Sigma_i V_i^\top$
    for each singular mode k **do** ▷ joint sign canonicalization (App. E.2)
      $s_{i,k} \leftarrow \text{sign}(\langle u_{0,k}, u_{i,k} \rangle)$
      $u_{i,k} \leftarrow s_{i,k} u_{i,k}, \quad v_{i,k} \leftarrow s_{i,k} v_{i,k}$
    end for
    $\Delta U_i \leftarrow U_i - U_0, \quad \Delta V_i \leftarrow V_i - V_0$
    $\xi_{U,i} \leftarrow \Pi_{U_0}(\Delta U_i), \quad \xi_{V,i} \leftarrow \Pi_{V_0}(\Delta V_i)$
    $\widetilde{\xi}_{U,i} \leftarrow \xi_{U,i} D_{\text{keep}}, \quad \widetilde{\xi}_{V,i} \leftarrow \xi_{V,i} D_{\text{keep}}$ ▷ mask trailing modes (App. E.4)
    $g_i \leftarrow \widetilde{\xi}_{U,i} \Sigma_0 V_0^\top + U_0 \Sigma_0 \widetilde{\xi}_{V,i}^\top$
  end for
  $\mathcal{I} \leftarrow \{i : \|g_i\|_F > 0\}$; set $c_i^\star \leftarrow 0$ for $i \notin \mathcal{I}$
  $\Gamma_{ij} \leftarrow \langle g_i, g_j \rangle_F$ for $i, j \in \mathcal{I}$
  $b \leftarrow \text{diag}(\Gamma)$
  Solve $(\Gamma + \lambda_{\text{ridge}} I_{|\mathcal{I}|}) \bar{c}_{\mathcal{I}} = b$
  $c_{\mathcal{I}}^\star \leftarrow \text{clip}(\bar{c}_{\mathcal{I}}, c_{\min}, c_{\max})$ ▷ entrywise
  $\xi_{U,\star} \leftarrow \Pi_{U_0}(\sum_{i=1}^K c_i^\star \widetilde{\xi}_{U,i}), \quad \xi_{V,\star} \leftarrow \Pi_{V_0}(\sum_{i=1}^K c_i^\star \widetilde{\xi}_{V,i})$ ▷ project and retract
  $U_\star \leftarrow \text{polar}(U_0 + \xi_{U,\star}), \quad V_\star \leftarrow \text{polar}(V_0 + \xi_{V,\star})$
  $W_\star \leftarrow U_\star \Sigma_0 V_\star^\top$
end for
Merge 1D parameters by task-vector average: $w_\star = w_0 + \frac{1}{K} \sum_{i=1}^K (w_i - w_0)$
return $\theta_\star$

Task Arithmetic merges experts by linearly combining their task vectors and adding the resulting update back to the base model:

$$\tau_{\text{TA}} = \lambda \sum_{i=1}^K \tau_i, \qquad \theta_{\text{TA}} = \theta_0 + \tau_{\text{TA}}. \tag{98}$$

Task Arithmetic is computationally cheap, requiring only vector additions and a global scaling coefficient. However, it does not explicitly address sign conflicts, redundant updates, layer-wise geometry, or sparse task-specific signals. As a result, interference and signal dilution can occur when expert updates overlap or when important updates are distributed over disjoint parameter regions.

TIES-Merging follows three steps: *Trim*, *Elect Sign*, and *Merge*. It first removes small-magnitude entries from each task vector, then elects a consensus sign for every coordinate, and finally averages only the updates whose signs agree with the elected sign.

Let $\rho_{\text{TIES}} \in (0, 1]$ be the retained density. The trimmed update for expert i is

$$m_i = \text{TopKMask}(|\tau_i|, \rho_{\text{TIES}}), \qquad \widetilde{\tau}_i = m_i \odot \tau_i, \tag{99}$$

where $m_i^p = 1$ indicates that coordinate p is among the largest ρ_{TIES} fraction of entries of $|\tau_i|$. The elected sign at coordinate p is

$$\gamma^p = \operatorname{sign}\left(\sum_{i=1}^{K} \widetilde{\tau}_i^p\right). \tag{100}$$

TIES then keeps only the experts aligned with this sign:

$$A_p = \left\{ i : \widetilde{\tau}_i^p \neq 0,\ \operatorname{sign}(\widetilde{\tau}_i^p) = \gamma^p \right\}. \tag{101}$$

The merged coordinate is

$$\tau_{\text{TIES}}^p = \begin{cases} \dfrac{1}{|A_p|} \sum_{i \in A_p} \widetilde{\tau}_i^p, & |A_p| > 0, \\ 0, & |A_p| = 0, \end{cases} \tag{102}$$

and the final model is

$$\theta_{\text{TIES}} = \theta_0 + \lambda \tau_{\text{TIES}}. \tag{103}$$

TIES remains an element-wise method, but it is more robust than naive averaging when many small updates are irrelevant or when experts update the same coordinate in opposite directions. Its main limitation is that it does not explicitly model matrix-level subspace geometry.

TSV-Merge, short for Task Singular Vectors Merge, operates on layer-wise task matrices rather than fully flattened vectors. For each 2D weight matrix, TSV decomposes each expert update with SVD, keeps the leading singular components, orthogonalizes the concatenated singular-vector bases, and reconstructs the merged matrix.

For layer ℓ, compute

$$\Delta W_i^{(\ell)} = U_i \Sigma_i V_i^\top. \tag{104}$$

After truncating to rank k_ℓ,

$$\Delta W_i^{(\ell)} \approx U_i^{(k)} \Sigma_i^{(k)} {V_i^{(k)}}^\top. \tag{105}$$

The truncated bases and spectra are concatenated as

$$U = [U_1^{(k)} | \cdots | U_K^{(k)}], \qquad V = [V_1^{(k)} | \cdots | V_K^{(k)}], \qquad \Sigma = \operatorname{blockdiag}(\Sigma_1^{(k)}, \ldots, \Sigma_K^{(k)}). \tag{106}$$

TSV reduces singular-vector interference by orthogonalizing U and V. In the whitening form,

$$U_\perp = U(U^\top U)^{-1/2}, \qquad V_\perp = V(V^\top V)^{-1/2}. \tag{107}$$

A numerically stable implementation can use the orthogonal Procrustes form. If $U = P_U D_U Q_U^\top$ and $V = P_V D_V Q_V^\top$, then

$$U_\perp = P_U Q_U^\top, \qquad V_\perp = P_V Q_V^\top. \tag{108}$$

The merged task matrix is reconstructed as

$$\widehat{\Delta}^{(\ell)} = U_\perp \Sigma V_\perp^\top, \qquad W_{\text{TSV}}^{(\ell)} = W_0^{(\ell)} + \alpha \widehat{\Delta}^{(\ell)}. \tag{109}$$

For non-matrix parameters, such as biases or normalization vectors, TSV typically falls back to Task Arithmetic.

TSV is a spectral, subspace-level method. By decorrelating singular-vector bases, it can reduce interference that is invisible to element-wise rules. This comes at a higher computational cost because SVDs are required for matrix-shaped layers. The rank k_ℓ is either treated as a hyperparameter or chosen through an automatic rank-reduction rule.

RAM is designed for sparse RL updates, where naive averaging can dilute coordinates that are important for only one expert. For each coordinate p, RAM identifies the experts with non-negligible updates:

$$A_p = \left\{ i \in \{1, \dots, K\} : |\tau_i^p| > \epsilon_{\text{act}} \right\}, \qquad c^p = |A_p|, \tag{110}$$

where ϵ_{act} is an active-update threshold. The merged coordinate is

$$\tau_{\text{RAM}}^p = \begin{cases} 0, & c^p = 0, \\ \tau_j^p, & c^p = 1 \text{ and } A_p = \{j\}, \\ \dfrac{1}{c^p} \displaystyle\sum_{i \in A_p} \tau_i^p, & c^p \geq 2. \end{cases} \tag{111}$$

Thus, inactive coordinates are discarded, task-unique coordinates are preserved without being divided by K, and shared coordinates are averaged only over the experts that actively update them. The final model is

$$\theta_{\text{RAM}} = \theta_0 + \tau_{\text{RAM}}. \tag{112}$$

RAM is element-wise like TIES, but its selection criterion is different. Instead of trimming by global magnitude density and resolving signs, RAM separates inactive, unique, and shared update regions. This makes it suitable when RL-trained experts store task-specific behavior in sparse and partially non-overlapping parameter subsets.

OrthoMerge [31] is a geometry-preserving merging framework that factors each expert update into an *implicit orthogonal transformation* of the base weights and a Euclidean residual, merges the orthogonal components on the rotation manifold, and merges the residuals with a standard element-wise rule. The released implementation provides six variants: two Procrustes-target constructions (C, which fits the rotation only to rows whose update conflicts with the mean task vector, and G, which fits it to the full expert weights) combined with three residual backends (Task Arithmetic, TIES, TSV). We report the G-TIES variant for two reasons. First, TIES is a strong residual backend in both of our merging settings: among the element-wise baselines it attains the best or near-best overall average on both backbones (Tables 1 and 2), so we pair OrthoMerge with the classical method that is already competitive in our comparison. Second, the G construction fits the Procrustes rotation to the *full* expert weights, so the extracted transformation represents the entire expert update. The C construction instead fits it only to the rows that conflict with the mean task vector. Since ISO-Merger likewise composes each expert's full update—represented as motion of its singular frames under the fixed base spectrum—G-TIES is the directly comparable variant.

For each 2D weight matrix (excluding layer norms), OrthoMerge first solves a one-sided orthogonal Procrustes problem per expert,

$$R_i^{(\ell)} = \underset{R \in \mathrm{O}(n_\ell)}{\arg\min} \left\| W_0^{(\ell)} R - W_i^{(\ell)} \right\|_F = \bar{U}\bar{V}^\top, \qquad {W_0^{(\ell)}}^\top W_i^{(\ell)} = \bar{U}\bar{S}\bar{V}^\top, \tag{113}$$

so that the rotation acts only on the input (column) space of the layer. Each rotation is mapped to a skew-symmetric matrix by the inverse Cayley transform,

$$A_i^{(\ell)} = \left(R_i^{(\ell)} + I\right)^{-1} \left(R_i^{(\ell)} - I\right), \qquad A_i^{(\ell)} = -{A_i^{(\ell)}}^\top, \tag{114}$$

and the K skew matrices are aggregated with a magnitude-corrected mean that averages direction and intensity separately,

$$A_{\text{mrg}}^{(\ell)} = \left(\tfrac{1}{K} \textstyle\sum_{i=1}^K \left\| A_i^{(\ell)} \right\|_F \right) \cdot \dfrac{\sum_{i=1}^K A_i^{(\ell)}}{\left\| \sum_{i=1}^K A_i^{(\ell)} \right\|_F}, \tag{115}$$

which prevents the norm shrinkage that a plain average of nearly orthogonal directions would induce. The merged rotation is recovered with the forward Cayley map and applied to the base weights,

$$R_{\text{mrg}}^{(\ell)} = \left(I - A_{\text{mrg}}^{(\ell)}\right)^{-1}\left(I + A_{\text{mrg}}^{(\ell)}\right), \qquad W_{\text{rot}}^{(\ell)} = W_0^{(\ell)} R_{\text{mrg}}^{(\ell)}. \tag{116}$$

The part of each expert update not captured by its own rotation is treated as a Euclidean residual,

$$\delta_i^{(\ell)} = W_i^{(\ell)} - W_0^{(\ell)} R_i^{(\ell)}, \tag{117}$$

(for non-matrix parameters $\delta_i = \tau_i$), and the residuals are merged with the TIES rule of trimming, sign election, and sign-consistent averaging described above. The final model is

$$W_{\text{OM}}^{(\ell)} = W_{\text{rot}}^{(\ell)} + \lambda\, \tau_{\text{TIES}}\left(\{\delta_i\}\right). \tag{118}$$

OrthoMerge is the geometry-aware baseline closest to ISO-Merger, but the two methods act in different coordinates. OrthoMerge applies a one-sided orthogonal transformation on the input side of each matrix: $W_0^{(\ell)} R$ preserves the left singular directions and singular values of $W_0^{(\ell)}$, so any output-side frame change must be represented through its Euclidean residual path.

ISO-Merger instead composes two-sided frame changes under the shared base spectrum, adapting both U and V in fixed-spectrum Stiefel coordinates. OrthoMerge's rotation class is contained in the more permissive output-subspace-retaining class C_L analyzed in Section 3. Even this larger class leaves a substantial unexplained-update ratio, whereas retaining the incoming spectrum while adapting both frames gives a low-residual description. This motivates ISO-Merger's two-sided frame parameterization.

Baseline hyperparameters. We follow the official implementations for all baselines. For TIES, we use $\lambda = 1.0$ and $\rho_{\text{TIES}} = 0.2$. For TSV-Merge, we use $\alpha = 1.0$ with automatic rank reduction. For RAM, we use $\epsilon = 10^{-5}$. For ISO-Merger, all SVD operations are performed in FP64. For OrthoMerge-G-TIES, we use the official implementation with its default hyperparameters: FP32 SVD-based Procrustes, 20% retained density and $\lambda = 1.0$ for the TIES residual stage.

Evaluation protocol. For the Qwen2.5 setting, we evaluate coding on LiveBench and LiveCodeBench, tool use on BFCL, and long-context memory on RULER HotpotQA and SQuAD at 32K–64K context lengths. For the DeepSeek-R1-Distill setting, we evaluate coding on LiveBench and LiveCodeBench, and math on AIME24, AIME25, AMC23, Minerva, and OlympiadBench. Unless otherwise specified, we generate 16 rollouts per prompt and report mean and standard deviation over three independent runs.

Decoding settings. For DeepSeek-R1-Distill math and coding evaluations, we use temperature 0.6 and top-$p = 0.95$. For Qwen2.5 LiveBench and LiveCodeBench, we use temperature 1.0. For BFCL, we use single-sample near-greedy decoding with temperature 0.001, following the official evaluation protocol. For RULER long-context evaluation, we use temperature 0.7. Inference is conducted with vLLM on NVIDIA H100 GPUs.

Additional distributional metrics. Among n sampled responses, let c denote the number with binary correctness equal to one. For $n \geq k$, we report

$$\widehat{\text{best@}k} = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}, \qquad \widehat{\text{worst@}k} = \frac{\binom{c}{k}}{\binom{n}{k}}.$$

In addition to average@16, we report best@4 and worst@4 to characterize the upper and lower tails of stochastic decoding performance.

- **best@k**=pass@k, the probability that at least one of k randomly drawn samples solves the problem (best-case behaviour over k draws).
- **worst@k**= $\binom{c}{k} / \binom{n}{k}$, the probability that all k randomly drawn samples solve the problem (consistency / worst-case reading).

For the Qwen2.5 setting, unit-test accuracy of LB/LCB coding benchmark is the fraction of unit tests passed, a real value in $[0, 1]$ rather than a binary outcome. BFCL is evaluated with single-sample, near-greedy decoding with temperature 0.001, following the official leaderboard protocol. So we only report best@4 and worst@4 for rest benchmark and metrics.

Table 5 | **Main results on Qwen2.5-7B-Instruct with 3 RL experts under Best@4.** Coding: pass accuracy (ACC) on LiveBench (LB) and LiveCodeBench (LCB). Memory: RULER HotpotQA and SQuAD at 32K–64K context lengths under the Recurrent setting. Best expert/merged result per column in **bold**. Shading indicates Experts and Ours.

	Coding		Memory				
	LB	LCB v2	HotpotQA		SQuAD		
Method	ACC	ACC	32K	64K	32K	64K	Avg
Base	48.35 ± 1.46	39.93 ± 0.64	68.98 ± 1.44	66.54 ± 1.51	79.62 ± 1.23	77.20 ± 0.30	63.44
RLVR-Coder	48.97 ± 1.20	**41.13 ± 1.50**	70.03 ± 0.33	66.33 ± 0.79	81.90 ± 0.59	77.60 ± 0.42	64.33
RLVR-Tool	**51.54 ± 2.17**	39.91 ± 1.66	69.31 ± 0.61	67.71 ± 1.53	81.12 ± 0.80	78.74 ± 0.65	64.72
RLVR-Memory	49.74 ± 2.61	37.49 ± 3.49	**86.21 ± 0.26**	**86.38 ± 0.90**	**87.26 ± 0.32**	**88.54 ± 0.39**	72.60
Task Arithmetic	51.99 ± 1.78	42.46 ± 0.28	81.65 ± 0.60	81.37 ± 0.37	84.13 ± 0.44	87.58 ± 0.31	71.53
TIES	50.04 ± 2.18	42.34 ± 1.42	85.80 ± 0.18	84.32 ± 0.04	85.84 ± 0.36	88.05 ± 0.37	72.73
TSV	48.82 ± 0.29	39.04 ± 2.44	84.75 ± 0.37	83.54 ± 0.37	86.39 ± 0.35	87.72 ± 0.44	71.71
RAM	48.73 ± 0.54	41.06 ± 0.25	85.91 ± 0.52	84.17 ± 0.61	86.62 ± 0.43	**88.39 ± 0.57**	72.48
OrthoMerge-G-TIES [31]	**52.20 ± 2.00**	**42.52± 1.77**	84.38 ± 0.37	83.47 ± 0.40	86.06 ± 0.26	87.66 ± 0.41	72.72
Ours	50.39 ± 1.98	40.68 ± 1.99	**86.05 ± 0.78**	**84.92 ± 0.37**	**86.98 ± 0.21**	88.23 ± 0.38	**72.88**

Tables 5, 6, 7, and 8 show that ISO-Merger achieves the highest upper and lower tails under both backbones. In particular, for worst@4, which is substantially more challenging, ISO-Merger reaches an overall average of 54.85, compared with 53.23 for the strongest training-free baseline. On `DeepSeek-R1-Distill-Qwen-1.5B`, ISO-Merger reaches a total average of 34.98, improving over the best baseline at 33.62. For best@4, ISO-Merger remains competitive with the original specialists while composing multiple skills into a single model. These results further demonstrate that different RL experts initialized from the same base model can be combined in fixed-spectrum Stiefel coordinates.

G. Numerical Precision of the SVD-Based Retraction

In exact arithmetic, the ISO reconstruction

$$W^+ = U^+ \Sigma_0 (V^+)^\top$$

has singular values $\mathrm{diag}(\Sigma_0)$ whenever U^+ and V^+ have orthonormal columns. In practice, however, the polar retraction is computed using a finite-precision SVD, so numerical errors in the decomposition can weaken this invariant.

We therefore compare four PyTorch SVD configurations: FP32 and FP64 on CPU and GPU. As a representative sanity check, we extract the `q_proj` matrix from layer 10 of `Qwen3-1.7B-Base`,

$$W_0 \in \mathbb{R}^{2048\times 2048},$$

Table 6 | **Main results on Qwen2.5-7B-Instruct with 3 RL experts under Worst@4.** Coding: pass accuracy (ACC) on LiveBench (LB) and LiveCodeBench (LCB). Memory: RULER HotpotQA and SQuAD at 32K–64K context lengths under the Recurrent setting. Best expert/merged result per column in **bold**. Shading indicates Experts and Ours.

	Coding		Memory				
	LB	LCB v2	HotpotQA		SQuAD		
Method	ACC	ACC	32K	64K	32K	64K	Avg
Base	24.44 ± 0.38	16.99 ± 0.11	30.76 ± 0.95	24.34 ± 0.24	37.71 ± 1.31	35.55 ± 0.92	28.30
RLVR-Coder	**25.84 ± 1.57**	**19.96 ± 0.41**	30.37 ± 0.84	24.72 ± 0.80	38.49 ± 1.38	35.89 ± 0.65	29.21
RLVR-Tool	21.87 ± 0.84	16.78 ± 1.66	30.61 ± 1.23	23.48 ± 1.03	37.83 ± 1.69	35.89 ± 1.27	27.74
RLVR-Memory	24.26 ± 0.76	17.28 ± 2.64	**70.22 ± 0.73**	**68.40 ± 0.14**	**69.66 ± 0.80**	**67.09 ± 0.51**	**52.82**
Task Arithmetic	26.24 ± 2.29	21.46 ± 0.52	61.86 ± 0.16	58.54 ± 0.90	64.89 ± 0.89	61.44 ± 0.56	49.07
TIES	30.48 ± 2.58	21.41 ± 0.66	65.91 ± 0.74	65.12 ± 0.20	**69.79 ± 0.59**	**66.66 ± 0.46**	53.23
TSV	26.12 ± 1.78	19.39 ± 1.36	65.33 ± 0.99	63.43 ± 0.55	69.36 ± 0.90	65.32 ± 0.98	51.49
RAM	27.41 ± 1.87	21.72 ± 0.57	65.30 ± 0.34	65.82 ± 0.90	69.57 ± 0.43	65.75 ± 0.61	52.59
OrthoMerge-G-TIES [31]	28.71 ± 2.11	21.65 ± 0.66	64.88 ± 1.15	64.80 ± 0.59	68.31 ± 0.38	63.62 ± 0.54	52.00
Ours	**33.50 ± 2.28**	**21.80 ± 1.61**	**71.78 ± 0.96**	**66.12 ± 0.31**	69.75 ± 0.29	66.13 ± 0.52	**54.85**

compute

$$W_0 = U\Sigma V^\top,$$

and reconstruct

$$W_{\text{rec}} = U\Sigma V^\top.$$

We report the mean squared spectral reconstruction error

$$\epsilon_\sigma := \frac{1}{q}\,\|\sigma(W_{\text{rec}}) - \sigma(W_0)\|_2^2, \tag{119}$$

together with the wall-clock decomposition time.

Table 9 shows that FP32 GPU SVD introduces substantially larger numerical error than FP64 GPU SVD: 7.7899×10^{-4} versus 2.8924×10^{-8} in this representative matrix. FP64 reduces the error by approximately 2.7×10^4 while increasing the measured runtime only from 0.894 to 0.923 seconds. We therefore use FP64 GPU SVD for the polar retraction throughout ISO.

H. Online RLVR Training Details

The ISO parameterization, factor gradients, and retraction are described in Section 4.3. This appendix reports the training and evaluation configurations used in the online RLVR experiments.

Common setup. All experiments are implemented using VERL [9] and run on NVIDIA A100 80 GB GPUs. We use the DAPO algorithm [5] and hold all non-optimizer settings fixed between the weight-space and ISO variants.

We use asymmetric policy-ratio clipping with

$$(\epsilon_{\text{low}}, \epsilon_{\text{high}}) = (0.2, 0.28). \tag{120}$$

The KL coefficient is set to

$$\beta_{\text{KL}} = 10^{-3},$$

Table 7 | **Main results on DeepSeek-R1-Distill-Qwen-1.5B with 2 RL experts under Best@4**. Coding ACC on LB and LCB v5. Math: ACC on AIME24, AIME25, AMC23, Minerva, and OlympiadBench. Best expert/merged result per column in **bold**. Shading indicates Experts and Ours.

	Coding		Math					
	LB	LCB v5	AIME24	AIME25	AMC23	Minerva	Olympiad	Total
Method	ACC	ACC	ACC	ACC	ACC	ACC	ACC	Avg
Base [4]	29.68 ± 0.17	25.96 ± 0.42	54.49 ± 1.34	33.34 ± 0.99	80.49 ± 0.80	41.67 ± 2.00	56.05 ± 0.56	45.95
Archer2.0 [32]	**36.11 ± 0.45**	**36.75 ± 0.27**	64.74 ± 0.72	39.88 ± 1.12	86.49 ± 0.12	43.38 ± 0.52	61.19 ± 0.32	52.65
JustRL [33]	31.67 ± 0.42	32.63 ± 0.62	**71.62 ± 0.16**	**49.25 ± 1.32**	**90.83 ± 0.65**	**45.22 ± 1.08**	**64.49 ± 0.91**	55.10
Task Arithmetic [27]	**36.10 ± 0.45**	37.18 ± 0.35	72.34 ± 0.54	47.12 ± 2.32	90.51 ± 0.17	44.61 ± 0.46	64.20 ± 0.56	56.01
TIES [28]	34.31 ± 0.69	37.66 ± 0.31	**73.40 ± 0.42**	47.37 ± 0.77	89.62 ± 0.38	45.10 ± 0.87	64.35 ± 0.35	55.97
TSV [29]	35.84 ± 0.34	**37.67 ± 0.39**	72.41 ± 1.07	46.94 ± 1.18	89.93 ± 1.10	**45.10 ± 0.92**	64.94 ± 0.42	56.12
RAM [30]	35.34 ± 0.87	36.96 ± 1.04	73.06 ± 0.62	49.50 ± 1.81	90.36 ± 0.63	44.98 ± 0.76	64.59 ± 0.55	56.40
OrthoMerge-G-TIES [31]	35.18 ± 0.72	37.47 ± 0.50	73.95 ± 1.16	47.13 ± 0.96	89.82 ± 0.51	44.36 ± 0.56	64.74 ± 0.68	56.09
Ours	34.04 ± 0.56	36.87 ± 0.09	72.69 ± 0.34	**50.43 ± 0.71**	**91.31 ± 0.61**	44.73 ± 0.87	**64.99 ± 0.56**	**56.44**

Table 8 | **Main results on DeepSeek-R1-Distill-Qwen-1.5B with 2 RL experts under Worst@4**. Coding ACC on LB and LCB v5. Math: ACC on AIME24, AIME25, AMC23, Minerva, and OlympiadBench. Best expert/merged result per column in **bold**. Shading indicates Experts and Ours.

	Coding		Math					
	LB	LCB v5	AIME24	AIME25	AMC23	Minerva	Olympiad	Total
Method	ACC	ACC	ACC	ACC	ACC	ACC	ACC	Avg
Base	8.38 ± 0.38	8.95 ± 0.47	12.42 ± 0.92	12.96 ± 0.56	46.02 ± 0.64	14.95 ± 0.76	30.17 ± 0.73	19.12
Archer2.0	**15.95 ± 0.80**	**17.39 ± 0.24**	19.80 ± 0.63	17.36 ± 0.36	57.83 ± 1.03	18.75 ± 1.08	38.22 ± 0.60	26.47
JustRL	14.02 ± 0.22	14.42 ± 0.14	**33.61 ± 0.93**	**27.25 ± 1.15**	**72.89 ± 0.65**	**23.90 ± 0.60**	**46.32 ± 0.14**	**33.20**
Task Arithmetic	15.82 ± 0.26	16.67 ± 0.50	28.49 ± 0.56	20.99 ± 0.47	68.35 ± 0.85	21.94 ± 0.92	43.41 ± 0.64	30.81
TIES	**17.76 ± 0.87**	18.22 ± 0.27	33.40 ± 1.45	25.21 ± 0.22	71.85 ± 0.97	23.28 ± 1.25	45.23 ± 0.07	33.57
TSV	15.91 ± 0.52	17.46 ± 0.16	32.33 ± 0.38	24.69 ± 1.48	68.36 ± 1.44	24.02 ± 0.87	42.86 ± 0.81	32.23
RAM	17.41 ± 1.11	17.84 ± 0.25	33.90 ± 0.86	24.10 ± 0.14	71.46 ± 0.29	25.12 ± 0.35	45.48 ± 0.12	33.62
OrthoMerge-G-TIES [31]	16.45 ± 0.67	18.27 ± 0.19	33.21 ± 1.99	24.86 ± 0.60	70.85 ± 0.80	23.77 ± 0.21	45.28 ± 0.60	33.24
Ours	16.18 ± 0.18	**18.95 ± 0.45**	**36.10 ± 1.01**	**27.41 ± 0.83**	**73.23 ± 1.81**	**25.12 ± 1.21**	**47.85 ± 0.12**	**34.98**

Table 9 | Runtime and mean squared spectral reconstruction error under different PyTorch SVD precision and device configurations.

Precision	Device	Mean squared spectral error	Time (s)
FP32	CPU	3.6633e-7	2.115
FP64	CPU	1.1126e-8	3.699
FP32	GPU	**7.7899e-4**	0.894
FP64	GPU	2.8924e-8	0.923

with the KL term applied as a loss-shaping term rather than as part of the rollout reward. We enable online dynamic filtering by removing prompt groups whose sampled rollouts are either all correct or all incorrect. Unless stated otherwise, training rollouts use temperature 1.0 and top-p = 1.0.

We set weight decay to zero for both the weight-space and ISO variants. At the learning rates used here, the standard coefficient $\lambda = 10^{-2}$ falls below the BF16-visible update scale and provides no measurable benefit in our RLVR runs [1].

Mathematical reasoning. We train `Qwen3-1.7B-Base`, `Qwen3-4B-Base`, and `Qwen3-8B-Base` [34] on DeepMath-103K [35], following prior RLVR post-training recipes [42].

For `Qwen3-1.7B-Base` and `Qwen3-4B-Base`, we use a global prompt batch size of 256 and train for 400 actor updates, corresponding to approximately one nominal pass over the prompt set before online filtering. The 1.7B runs sample 16 rollouts per prompt, whereas the 4B runs sample 12. Both use a maximum prompt length of 1,024 tokens and a maximum response length of 8,192 tokens.

ISO-AdamW uses a learning rate of 7.5×10^{-7} for the frame variables (U, V), followed by polar retraction after every tentative factor update.[3] For weight-space AdamW, we sweep

$$\left\{5 \times 10^{-7},\ 7.5 \times 10^{-7},\ 1 \times 10^{-6},\ 2 \times 10^{-6},\ 3 \times 10^{-6}\right\}.$$

For the weight-space Muon baselines, we sweep

$$\left\{2.5 \times 10^{-5},\ 5 \times 10^{-5},\ 7.5 \times 10^{-5},\ 1 \times 10^{-4}\right\}.$$

The ISO-Muon experiment is conducted on `Qwen3-4B-Base` for 300 actor updates using a learning rate of 5×10^{-5}.

For `Qwen3-8B-Base`, we retain the global prompt batch size of 256 with a rollout size of 12. We compare ISO-AdamW with learning rate 7.5×10^{-7} against weight-space AdamW with learning rate 2×10^{-6}, the strongest AdamW setting identified in the smaller-model sweeps; neither method is further tuned at 8B. Both runs are trained for 210 actor updates. To test whether the AdamW baseline is undertrained, we continue it for an additional 60 updates.

The 8B runs begin with a maximum response length of 8,192 tokens. Because ISO-AdamW produces longer responses at this scale, we increase the maximum response length to 16,384 tokens for both methods after update 80. Evaluation uses a 16,384-token cap throughout.

We evaluate mathematical reasoning on AIME 2024, AIME 2025, AMC 2023, Minerva, and OlympiadBench. For each problem, we sample 16 responses using temperature 1.0 and top-$p = 0.8$. The maximum evaluation length is 8,192 tokens for the 1.7B and 4B models and 16,384 tokens for the 8B model. For Table 3, we select the checkpoint with the highest aggregate score among the final three evaluations and report all benchmark scores from that checkpoint.

Competitive coding. For competitive coding, we train `DeepSeek-R1-Distill-Qwen-1.5B` [4] on the ArcherCodeR training split [43], which contains 6,753 problems. We use a global prompt batch size of 128 and sample 16 rollouts per problem. The maximum prompt length is 2,048 tokens, and the maximum response length is 32,768 tokens.

ISO-AdamW uses a learning rate of 1×10^{-6} for (U, V), followed by polar retraction after each factor update. For the weight-space AdamW baseline, we sweep

$$\left\{1 \times 10^{-6},\ 2 \times 10^{-6},\ 3 \times 10^{-6},\ 5 \times 10^{-6}\right\},$$

where 1×10^{-6} is the learning rate used by the original Archer-style recipe.

We train the primary runs for 220 actor updates. At a prompt batch size of 128, this budget corresponds to approximately

$$\frac{220 \times 128}{6753} \approx 4.17$$

nominal passes over the prompt set before accounting for online filtering. Because the coding dataset is relatively small, longer training can enter a multi-pass overfitting regime. To test whether the

[3] On `Qwen3-1.7B-Base`, we compare learning rates 7.5×10^{-7} and 1×10^{-6}. The former performs better, so we use it throughout the math experiments.

primary budget truncates the weight-space baseline prematurely, we continue the two strongest AdamW runs, with learning rates 3×10^{-6} and 5×10^{-6}, to 330 actor updates.

For evaluation, we report results on LiveCodeBench [44] v5 and v6. We sample 8 responses per problem using temperature 0.8 and top-p = 1.0, and report average accuracy under the same evaluation protocol for all methods.