MUON FOR RLVR: WHAT WORKS, WHY, AND WHAT IT LEARNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Muon is widely adopted for pretraining large language models and has been extended to supervised fine-tuning, but it is rarely used for reinforcement learning (RL), where reports of its value are mixed. We ask whether Muon can be used for RL with verifiable rewards (RLVR), and if so, how does Muon make a difference. With the right recipe, Muon extends the useful training horizon: it keeps improving after Adam-series optimizers plateau and reaches higher accuracy on both an AdamW-pretrained base (Qwen3-4B-Base, 29.17% versus 24.47% macro-averaged Pass@1 against AdamW) and a Muon-pretrained one (Moonlight-16B-A3B-Instruct, 32.60% versus 31.29% macro-averaged Pass@1 against SignSGD). We identify two critical ingredients. First, the matrix passed to Newton–Schulz should be a single linear operator: orthogonalizing fused QKV projections per head rather than as one stored tensor is what lifts Muon above AdamW on Qwen3-4B-Base, because splitting keeps the cross-head agreement already present in the gradient, which fused orthogonalization whitens away. Second, momentum should be removed, because RLVR offers it no persistent direction to average, Newton–Schulz's scale invariance turns the resulting correlation into extra displacement that AdamW would damp, and on-policy sampling makes that displacement unrecoverable. Finally, Muon, like AdamW, learns off-the-principal directions of the base weights. Muon's endpoint, however, is less compressible and relies on the W_V attention value projections for accuracy where Adam-series optimizers do not.

1 INTRODUCTION

Muon performs spectral-norm steepest descent on matrix-valued weights: unlike AdamW's coordinate-wise scaling (Loshchilov & Hutter, 2019), it approximately orthogonalizes each matrix update with Newton–Schulz iterations (Jordan et al., 2024; Bernstein, 2025). After Moonlight showed Muon scales to language-model pretraining (Liu et al., 2025), it became a common choice for pretraining large models, including Kimi K2, GLM-5, DeepSeek-V4 and Inkling (Kimi Team, 2025; GLM-5 Team, 2026; DeepSeek-AI, 2026; Thinking Machines Lab, 2026). Adoption in post-training is more limited. Kimi K2 uses Muon for both supervised fine-tuning and reinforcement learning (RL), and recommends it for fine-tuning K2 because a Muon-pretrained checkpoint performs best when fine-tuned with Muon (Kimi Team, 2025; Liu et al., 2025). That recommendation rests on matching the pretraining optimizer, and the report does not compare optimizers in RL.

The evidence on Muon for RL is mixed. Muon has been reported to add little value in RL training, particularly when the base model was pretrained with AdamW (NVIDIA, 2025), and Fan et al. (2026) observe model collapse under Muon in RL with verifiable rewards (RLVR), which they attribute to Newton–Schulz amplifying a noise-dominated spectral tail. We find that the outcome depends on the recipe. A properly configured Muon keeps improving under RLVR for longer than coordinate-wise optimizers (Figure 1) and eventually reaches higher accuracy, both on an AdamW-pretrained base (Qwen3-4B-Base) and on a Muon-pretrained one (Moonlight-16B-A3B-Instruct). This paper asks *which implementation details make that recipe work, why they matter, and what a Muon-trained model learns differently from an AdamW-trained one.*

Muon's update is derived for a single linear map, but implementations (Shoeybi et al., 2019; Jordan et al., 2024) often fuse several maps into one stored tensor for efficiency, most commonly the query, key and value (QKV) projections. As the polar map does not separate under concatenation, how the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tensor is partitioned defines a different optimizer. On Qwen3-4B-Base, orthogonalizing the fused QKV matrix does not outperform AdamW, whereas splitting it per head before Newton–Schulz lifts Muon well above AdamW (Table 2). Splitting preserves the input directions shared across the heads' gradients, which fused orthogonalization whitens away, and rebalances the update across heads and across Q, K and V.

Besides, momentum hurts Muon far more than AdamW. Mukherjee et al. (2026) find that plain SGD matches AdamW for RLVR and that adding momentum to SGD often hurts; with Muon, momentum is often the main driver of early collapse. Without momentum, split Muon trains stably for more than 4000 steps, so in our setting orthogonalization alone does not cause collapse. We trace the failure to three conditions that hold together. With only 32 prompt groups per step, RLVR gives momentum no persistent direction to average; Newton–Schulz is scale-invariant, so momentum cannot shrink the step and instead correlates consecutive directions, moving the model farther without a matching gain in reward; and because the policy generates its own data, that extra displacement shifts the sampling distribution and is never recovered. AdamW escapes the second condition: weakly correlated gradients shrink its first moment, and hence its step.

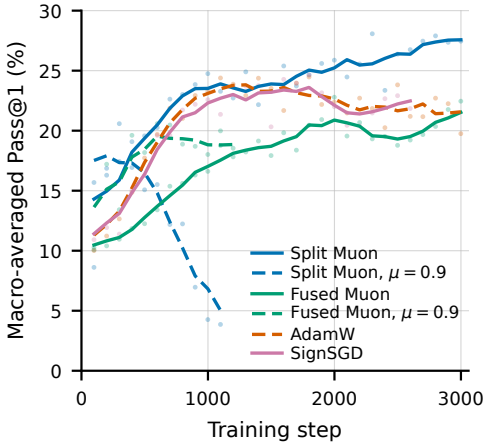

Figure 1: Macro-averaged Pass@1 on Qwen3-4B-Base. Split Muon with $\mu = 0$ continues improving after the coordinate-wise optimizers plateau; per-task curves appear in Figure 2.

Finally, we ask what differs between checkpoints trained with Muon and with coordinate-wise optimizers. Expressing each weight change in the singular frame of its base matrix, we find that Muon, like AdamW, learns off the principal directions: rotation among lower-energy singular directions plus motion outside the base matrix spans, which we call the functional endpoint ΔW_{func}, recovers the trained behavior, and removing it erases the gain. The optimizers differ in how they organize it: AdamW's functional endpoint compresses into a few leading singular directions of each matrix, whereas Muon's spreads over many more, and under either QKV partition, Muon relies on the value projections for accuracy, whereas AdamW and SignSGD do not.

Our contributions are summarized as follows:

1. **A Muon recipe for RLVR.** Per-head QKV splitting without momentum outperforms coordinate-wise baselines on both AdamW- and Muon-pretrained bases.
2. **An explanation of the recipe.** Semantic partition preserves cross-head structure, while removing momentum avoids excess displacement from short-lived directions.
3. **Optimizer-specific functional organization.** Muon and AdamW share an off-principal functional channel, but Muon's endpoint is less compressible and places accuracy in different tensors.

2 The Muon RL Recipe

Let $G_t = \nabla_W \mathcal{L}_t$ denote the stochastic gradient of matrix parameter $W \in \mathbb{R}^{m \times n}$. Muon forms a matrix-valued momentum buffer and steps along its polar factor:

$$B_t = \mu B_{t-1} + (1-\mu) G_t, \qquad \Delta W_t^{\text{Muon}} = -\eta \, \text{Polar}(B_t), \tag{1}$$

where μ is the momentum coefficient, η is the learning rate, and for $B_t = U\Sigma V^\top$ the polar factor $\text{Polar}(B_t) = UV^\top$ is approximated by Newton–Schulz (NS) iterations. Setting all nonzero singular values to one makes this the steepest-descent direction under the spectral norm (Bernstein, 2025).

The design space. Eq. (1) leaves several implementation details open. As is standard, Muon is applied only to hidden-layer weight matrices; biases, normalization gains, the embedding and the language-model head are updated with Adam (Jordan et al., 2024; Liu et al., 2025). Implementations differ most in how they handle fused parameters. Frameworks store the query, key and value projections in a single QKV tensor, and under grouped-query attention each block holds many heads, so the tensor can be orthogonalized as a whole, as three Q/K/V blocks, or head by head. They

Method	Setting				Update geometry	
	QKV	Momentum	β_1	β_2	Update	Normalization unit
$\text{Muon}_{\text{split}}$	split	×	0	–	$UV^\top$	matrix
$\text{Muon}^{\mu}_{\text{split}}$	split	✓	0.9	–	$UV^\top$	matrix
$\text{Muon}_{\text{fused}}$	fused	×	0	–	$UV^\top$	matrix
$\text{Muon}^{\mu}_{\text{fused}}$	fused	✓	0.9	–	$UV^\top$	matrix
AdamW	fused	✓	0.9	0.98	$m/(\sqrt{v}+\epsilon)$	coordinate
SignSGD	fused	×	0	0	$g/(\lvert g\rvert+\epsilon)$	coordinate

Table 1: Optimizer arms; the shaded row is our recipe. *Split* orthogonalizes each head's Q/K/V block separately; *fused* orthogonalizes the stored fused QKV matrix (the partition is irrelevant for coordinate-wise updates). With $\beta_1 = \beta_2 = 0$, Adam reduces to SignSGD (Bernstein et al., 2018) up to ϵ and decoupled weight decay.

also differ in the momentum coefficient μ, the shape-dependent scale of the update, its calibration against the AdamW learning rate, and the NS coefficients. We vary the QKV partition and μ, fix the remaining choices, and compare against two coordinate-wise baselines, AdamW and SignSGD (Table 1).

Fusion changes the optimizer. The partition of a fused tensor matters because the polar map is non-separable. For a vertically stacked QKV gradient of full column rank,

$$\text{Polar}\left(\begin{bmatrix} G_Q \\ G_K \\ G_V \end{bmatrix}\right) = \begin{bmatrix} G_Q \\ G_K \\ G_V \end{bmatrix} \left(G_Q^\top G_Q + G_K^\top G_K + G_V^\top G_V\right)^{-1/2} \neq \begin{bmatrix} \text{Polar}(G_Q) \\ \text{Polar}(G_K) \\ \text{Polar}(G_V) \end{bmatrix}, \tag{2}$$

so the update each block receives depends on every other block through a shared Gram matrix. The shape factor is likewise computed from the fused shape rather than from each operator's own shape. A fused transform therefore changes both the direction and the scale of every block's update.

Our recipe. We keep the stored parameter layout but split each fused QKV gradient into per-head Q/K/V blocks, apply five Newton–Schulz iterations and blockwise scaling by $\sqrt{\max(m_b, n_b)}$, and gather the resulting updates back into the fused layout. Thus, each matrix passed to Newton–Schulz represents one semantic linear operator rather than an arbitrary storage-level tensor.

The full recipe is summarized below, and the following sections evaluate and analyze it.

Recipe

Ingredients of our Muon RLVR recipe: (an asterisk marks performance-critical choices.)

- Split QKV weights per head before the NS iterations.*
- Use no momentum ($\mu = 0$).*
- Calibrate the learning rate against AdamW so that the update norm matches.

3 Experimental Setup and Main Results

We choose `Qwen3-4B-Base` [1] as a representative model for an AdamW-trained base model and `Moonlight-16B-A3B-Instruct` [2] as a representative for a Muon-trained base model.

Training and Evaluation. We train with fully on-policy GRPO (Shao et al., 2024) on DAPO-Math-17k (Yu et al., 2025), using 32 prompts and eight rollouts per prompt (global batch size 256), without PPO clipping. All arms use a constant reference learning rate of 10^{-6} and weight decay 0.1; Muon's learning rate is scaled to match. The optimizer variants are summarized in Table 1 (See more details in Appendix B). And we evaluate Qwen3-4B-Base on AIME24[3], AIME25, and BeyondAIME[4]. Because Moonlight-16B-A3B-Instruct is weaker at math overall, we extend its

[1] https://huggingface.co/Qwen/Qwen3-4B-Base
[2] https://huggingface.co/moonshotai/Moonlight-16B-A3B-Instruct
[3] https://huggingface.co/datasets/math-ai/aime24
[4] https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME

Method	Step	AIME24 P@1	AIME24 P@8	AIME24 Tr.↓	AIME25 P@1	AIME25 P@8	AIME25 Tr.↓	BeyondAIME P@1	BeyondAIME P@8	BeyondAIME Tr.↓	Macro-average P@1	Macro-average P@8	Macro-average Tr.↓
$\text{Muon}_{\text{split}}$	4099	**40.4** ±5.5	**63.3**	10.0	**29.6** ±3.3	**50.0**	10.4	**17.5** ±2.8	**37.0**	15.1	**29.2** ±2.6	**50.1**	11.9
$\text{Muon}_{\text{fused}}$	1999	32.1 ±5.6	53.3	21.3	24.2 ±4.3	43.3	20.0	12.1 ±2.3	29.0	21.6	22.8 ±2.4	41.9	21.0
AdamW	1799	32.5 ±3.5	**63.3**	15.8	26.7 ±4.4	46.7	13.8	14.3 ±0.7	32.0	13.8	24.5 ±1.5	47.3	14.4
SignSGD	1799	32.5 ±3.5	56.7	**1.3**	29.2 ±5.0	46.7	**2.9**	12.3 ±1.4	27.0	**0.9**	24.6 ±2.1	43.5	**1.7**

Table 2: Qwen3-4B-Base with a 64k generation budget. Values in %: P@1 is Pass@1 (mean ± std), P@8 is Pass@8, and Tr. is the truncation rate (lower is better).

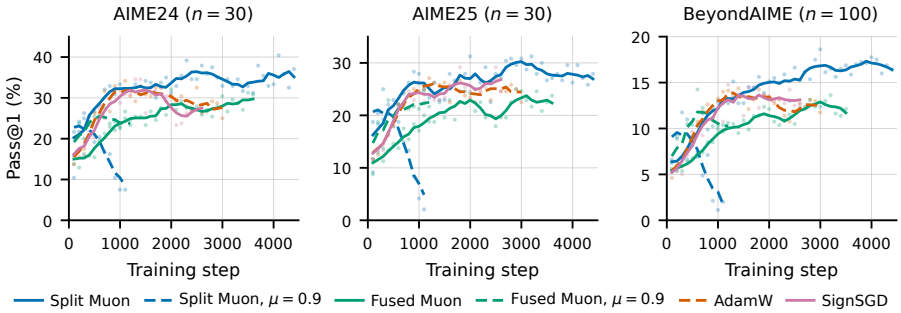

Figure 2: Per-task Pass@1 of Qwen3-4B-Base (64k evaluation). $\text{Muon}_{\text{split}}$ pulls ahead after the coordinate-wise optimizers plateau. Dashed lines use momentum ($\mu = 0.9$ or $\beta_1 = 0.9$); lines are 5-point moving averages of checkpoints (dots).

evaluation to OlympiadBench (He et al., 2024) and MATH500 (Hendrycks et al., 2021; Lightman et al., 2023). We sample eight completions per problem at temperature 0.6 and top-p 0.95, and report Pass@1 and Pass@8 under 16k and 64k generation budgets. Full protocols appear in Appendix B.

Muon on an AdamW-pretrained base: Qwen3-4B-Base. Table 2 and Figure 2 report Qwen3-4B-Base results under the 64k generation budget, with each arm's best checkpoint selected by macro-averaged Pass@1.

- **Split Muon is the strongest arm.** It reaches 29.17% macro-averaged Pass@1, against 24.47% for AdamW, and has the highest Pass@1 on every task and the highest average Pass@8.
- **The advantage appears late in training.** At step 1799, $\text{Muon}_{\text{split}}$, AdamW and SignSGD are nearly tied. $\text{Muon}_{\text{split}}$ keeps improving after the coordinate-wise arms plateau and peaks at step 4099, so its gain is a longer useful training horizon rather than faster early progress.
- **Matrix partition reverses the ordering.** The matched split and fused runs differ only in whether QKV is split per head before Newton–Schulz. $\text{Muon}_{\text{fused}}$ falls below both coordinate-wise baselines and $\text{Muon}_{\text{split}}$ rises above them, under both the 64k and the 16k evaluation (Table 8 and Figure 7 in Appendix B).
- **Momentum is not part of the recipe.** With $\mu = 0.9$, $\text{Muon}_{\text{split}}$ peaks within the first few hundred steps and then collapses, far below the momentum-free arm. $\text{Muon}_{\text{fused}}$ with momentum survives longer, so the collapse is an interaction between partition and momentum rather than a general failure of Muon; Section 5 analyzes why.

Muon on a Muon-pretrained base: Moonlight-16B-A3B. Table 3 and Figure 3 report Moonlight-16B-A3B-Instruct results under the 16k generation budget, again selecting each arm's best checkpoint by macro-averaged Pass@1.

- **Muon trains stably where SignSGD does not.** SignSGD improves fastest early, but it peaks around step 340 and then declines on every task. Both Muon arms improve steadily, and $\text{Muon}_{\text{split}}$ is still improving at the end of its run.
- **Split Muon reaches the highest accuracy.** It reaches 32.60% macro-averaged Pass@1, against 31.29% for SignSGD, and leads on three of the four tasks; its Pass@8 is on par with SignSGD's.
- **The split-over-fused margin is small.** Unlike on Qwen, $\text{Muon}_{\text{fused}}$ also exceeds SignSGD, and its curves stay close to $\text{Muon}_{\text{split}}$'s. The arms were trained for different durations, so this margin is descriptive rather than converged (Appendix B).

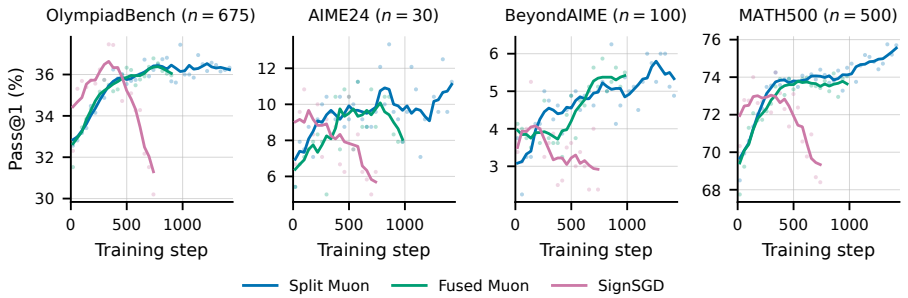

Figure 3: Per-task Pass@1 of Moonlight-16B-A3B-Instruct (16k evaluation). All arms are momentum-free; lines are 5-point moving averages of checkpoints (dots).

	$\text{Muon}_{\text{split}}$ (step 1339)			$\text{Muon}_{\text{fused}}$ (step 699)			SignSGD (step 339)		
Benchmark	P@1	P@8	Tr.↓	P@1	P@8	Tr.↓	P@1	P@8	Tr.↓
OlympiadBench	36.2 ±0.7	54.2	41.6	36.4 ±1.2	56.0	**23.5**	**37.5** ±1.4	**59.0**	33.5
AIME24	**12.5** ±4.3	**26.7**	59.2	10.8 ±3.9	20.0	**35.8**	9.6 ±4.2	23.3	68.3
BeyondAIME	**6.0** ±0.9	12.0	56.3	5.6 ±1.7	**13.0**	**24.9**	3.9 ±1.5	11.0	52.1
MATH500	**75.8** ±0.8	**88.4**	17.8	73.5 ±1.4	86.8	**13.1**	74.3 ±1.6	**88.4**	13.8
Macro-average	**32.6** ±1.1	45.3	43.7	31.6 ±1.2	44.0	**24.3**	31.3 ±1.0	**45.4**	41.9

Table 3: Moonlight-16B-A3B-Instruct with a 16k generation budget, at each optimizer's best checkpoint by macro-averaged Pass@1. Columns as in Table 2.

4 What Per-Head Splitting Changes

Per-head QKV splitting is what lifts Muon above the coordinate-wise baselines on Qwen3-4B-Base (Section 3). To see why, we reconstruct the applied split step and a fused-QKV counterfactual from the same FP32 optimizer input across five layers and 15 checkpoints (Table 4; details in Appendix B.3).

The matrix boundary determines the geometry of the applied step (Table 4): per-head splitting keeps the cross-head agreement already present in the gradient, and fused orthogonalization whitens it away. The optimizer input has stable rank of only about two, so the gradients of all heads are dominated by a few input directions they share. Orthogonalizing each head separately changes its singular values but not its singular directions, so each head stays broad, close to its full rank of 128, while keeping the shared directions it inherits from the gradient. The restacked step $\widetilde{G}$ therefore has low stable rank: the heads sit roughly 80% of the way from mutually orthogonal to fully collinear in their shared input space (Appendix B.3). The fused transform instead equalizes the singular values of the whole stacked matrix, so a direction shared by every head receives unit weight once for the entire stack rather than once per head, and the update spreads over far more directions.

Partition also changes how the update budget is allocated, both across heads and across attention roles. Across heads, fusion leaves update norms unequal, letting heads with larger gradients dominate the update, whereas per-head orthogonalization makes them nearly equal; Kimi Team (2026) give the same reason for adopting per-head orthogonalization in pretraining. Across roles, splitting doubles the Q update relative to the fused step, enlarges K by a fifth, and shrinks V. Thus split Muon is not a fused step computed in smaller pieces: semantic partition jointly changes singular-direction coupling and the scale assigned to each head and attention role.

5 Why Momentum Underperforms

Momentum has been found to be beneficial for Muon pretraining (Li et al., 2026) and inert for AdamW under RLVR settings (Mukherjee et al., 2026), but why would momentum be actively harmful for Muon in RLVR? We identify three conditions for momentum's failure: RLVR at small batch sizes (prompt groups) supplies no persistent direction for momentum to average; Newton–Schulz's scale invariance means that averaging nothing still costs a full-size step; and a policy that generates its own data makes the resulting displacement unrecoverable.

	Fused QKV	Per-head split
Optimizer input		
Stable rank of pre-NS QKV input	2.08	2.08
Applied QKV step		
Stable rank of full QKV step $\widetilde{G}$ (heads restacked)	1195	130.8
Median stable rank per head (of 128)	–	105.2
Coefficient of variation of head norms	0.43	0.008
Update norm, Q (relative to fused)	1	2.02×
Update norm, K (relative to fused)	1	1.20×
Update norm, V (relative to fused)	1	0.87×
Update norm, total QKV (relative to fused)	1	1.29×

Table 4: Semantic partition changes the applied QKV step (Qwen3-4B-Base, $\mu = 0$). Medians over five layers and 15 checkpoints; both steps are reconstructed from the same fp32 optimizer input, and $\widetilde{G}$ is the full QKV update applied to the fused weight: the restacked per-head Q/K/V blocks for the split step, and the single polar transform of the fused gradient for the counterfactual. Momentum-run values appear in Appendix B.3.

Small batch size with noisy gradients → no persistent direction. Let $P_{\text{func}} = P_{\text{trot}} + P_{\text{resid}}$ project a weight change onto its tail-rotation and out-of-span components in the base frame (Appendix B.3); applied to the full displacement $W_T - W_0$, it gives the functional endpoint ΔW_{func}. For checkpoints separated by L saved intervals, define $\Delta_L W_t^{(\mu)} = W_{t+L}^{(\mu)} - W_t^{(\mu)}$ and $D_\mu(L) = \left(\mathbb{E}_t \|P_{\text{func}} \Delta_L W_t^{(\mu)}\|_F^2\right)^{1/2}$. We fit $D_\mu(L) \propto L^{\alpha_\mu}$ over $L \in \{1, 2, 4, 8\}$, where each interval contains 20 optimizer steps. Independent increments give $\alpha = 1/2$ and coherent drift gives $\alpha = 1$. Muon$_{\text{split}}$ has $\alpha_0 = 0.506 \pm 0.002$ across all nine depth-band × Q/K/V cells. Its energy-weighted increment autocorrelation,

$$\rho_\mu(k) = \frac{\mathbb{E}_t \langle \delta_t^{(\mu)}, \delta_{t+k}^{(\mu)} \rangle}{\mathbb{E}_t \|\delta_t^{(\mu)}\|_F^2}, \quad \delta_t^{(\mu)} = P_{\text{func}} \left(W_{t+1}^{(\mu)} - W_t^{(\mu)}\right)$$

stays near zero at every lag (Figure 5b): functional updates accumulate, but no direction persists. With $\mu = 0.9$, α rises to 0.608–0.618, yet this is not a persistent direction either.The tail-rotation autocorrelation is large at lag 1 and gone by lag 4 (Figure 5b), and a correlated random walk built from these measured values reproduces the exponent (squares in Figure 5a).

Nor does momentum reveal a more concentrated spectral signal, which is measured by stable rank (Figure 5c): it slightly broadens the pre-NS input, while the stable rank of the actual split step is almost unchanged, 130.8 versus 130.7. Thus the pretraining mechanism proposed by Li et al. (2026), in which momentum preserves a dominant spectral signal before orthogonalization, is not observed in this regime.

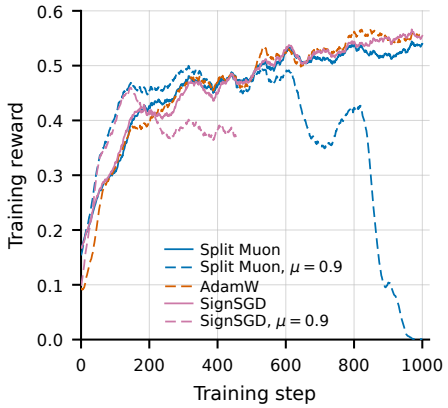

Figure 4: Training reward of Qwen3-4B-Base over the first 1000 steps. Dashed lines use momentum. Both scale-invariant updates gain early and then fall behind their $\mu = 0$ partners, while AdamW tracks SignSGD.

Muon vs. AdamW: scale invariance turns uselessness into cost. Mukherjee et al. (2026) report plain SGD matches AdamW, and the same inertness holds in pretraining: Marek et al. (2025) sweep β_1 from 0 to 0.999 across four orders of magnitude of batch size and find Adam's loss nearly flat in it, most markedly at small batch. The key difference lies in Muon's scale invariance. For independent zero-mean gradients, $\text{Var}(\hat{m}) = \sigma^2 \cdot (1 - \beta_1)/(1 + \beta_1)$, which implies that at $\beta_1 = 0.9$, its first-moment numerator has standard deviation 0.23σ while the second-moment denominator remains finite. AdamW shrinks its update when gradients fail to agree, whereas Muon re-normalizes the buffer. SignSGD with momentum (Signum; Bernstein et al., 2018), whose update $\text{sign}(\hat{m})$ is scale-invariant like Muon's, behaves like Muon$^{\mu}_{\text{split}}$ rather than AdamW: it gains early and then falls behind momentum-free SignSGD (Figure 4).

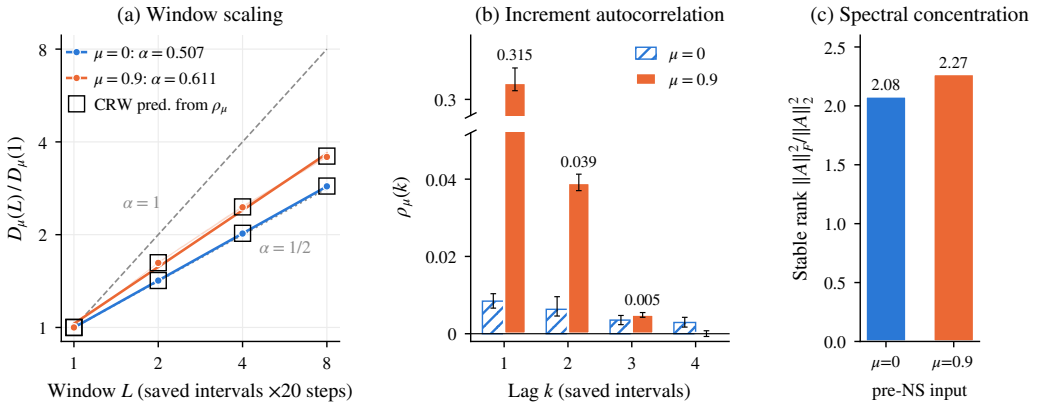

Figure 5: **Functional updates diffuse; momentum adds only short-range correlation.** Split Muon, functional Q/K/V components averaged over nine depth-band×Q/K/V cells; one saved interval is 20 optimizer steps. **(a)** Displacement $D_\mu(L)$ normalized at $L = 1$. Squares show the correlated-random-walk prediction (Eq. (4)) from the measured $\rho_\mu(k)$, which matches both exponents ($\alpha_0 = 0.507$, $\alpha_{0.9} = 0.611$). **(b)** Increment autocorrelation (error bars: s.d. across cells; broken axis). With $\mu = 0.9$ it is large at lag 1 and gone by lag 4. **(c)** Stable rank of the pre-NS QKV input and of the applied split step.

A moving policy makes the displacement unrecoverable. Under a stationary objective, extra displacement along a diffusive path is wasted motion and little else; that is the regime in which Marek et al. (2025) find momentum merely unnecessary rather than harmful. Liu et al. (2025) similarly report that they did not find a consistent improvement when tuning μ. In RLVR, the momentum arms peak early in terms of training reward but overshoot, to the point of no return. On-policy feedback is a plausible amplifier: excess displacement changes the policy that generates subsequent rollouts, so later gradients cannot salvage the model to its original training trajectory.

6 What Muon Learns Differently

Prior RLVR analyses find that learned behavior lies away from the principal singular directions of the base weights (Zhu et al., 2025; 2026) and can often be recovered at low rank (Cai et al., 2025). In our measurements, the optimizers share a coarse off-principal functional channel, but Muon organizes it differently: its endpoint is less compressible and places accuracy in different tensors.

A shared off-principal functional channel. For each base matrix $W_0 = U_0 \Sigma_0 V_0^\top$, we express the endpoint displacement in its base singular frame and decompose it into singular-value change (`diag`), rotations involving leading base directions (`hrot`), tail–tail rotation (`trot`), and motion outside the represented thin-SVD spans (`resid`). We define the functional endpoint as $\Delta W_{\text{func}} = \Delta W_{\text{trot}} + \Delta W_{\text{resid}}$. Adding a component to W_0 tests sufficiency, while removing it from W_T tests necessity. Full definitions and projection details appear in Appendix B.

The functional reconstruction $W_0 + \Delta W_{\text{func}}$ nearly matches the trained endpoint for Muon$_{\text{split}}$ and AdamW. Across the three arms, adding ΔW_{func} to W_0 recovers essentially all of the accuracy and length gains of Muon$_{\text{split}}$ and AdamW, and removing it from W_T erases them (Tables 5 and 6); SignSGD keeps its length gain but recovers less of its accuracy gain on the 30-problem AIME24 read. In contrast, `diag` and `hrot` carry little behavior on their own. Muon and the coordinate-wise optimizers learn through the same off-principal trot + resid channel. The conclusion doesn't depend on the head width: with k raised from 8 to 32, 128 or 512, the scale of the principal subspaces in Zhu et al. (2025), $W_0 + \Delta W_{\text{func}}$ still recovers 83–100% of the length gain and 88–111% of the accuracy gain on every arm.

Muon's functional endpoint is less compressible. We truncate each matrix's functional endpoint to a common leading rank and normalize recovery between W_0 and the untruncated reconstruction (Figure 6). AdamW retains its full accuracy gain from rank 64 on, whereas split Muon retains only

	Acc. (AIME24 avg@16)			# Tokens		
Method	W_T	$W_0 + \Delta W_{\text{func}}$	Rec.	W_T	$W_0 + \Delta W_{\text{func}}$	Rec.
$\text{Muon}_{\text{split}}$	.2917	.2875	98%	9514	9626	101%
AdamW	.2375	.2500	108%	8979	9042	102%
SignSGD	.2958	.2542	**77%**	10266	10217	100%

Table 5: Reconstruction $W_0 + \Delta W_{\text{func}}$ from the functional endpoint on Qwen3-4B-Base at step 2599. # Tokens is the average number of generated tokens. Recovery (Rec.) is the fraction of each arm's $W_0 \to W_T$ gain in that metric retained by the reconstruction.

	$\text{Muon}_{\text{split}}$				AdamW				SignSGD			
	Add		Remove		Add		Remove		Add		Remove	
Channel	acc.	len.	acc.	len.	acc.	len.	acc.	len.	acc.	len.	acc.	len.
`diag`	3%	−1%	−1%	−3%	5%	−9%	−13%	5%	−1%	−1%	9%	0%
`hrot`	−1%	0%	−12%	5%	8%	−6%	−6%	5%	4%	2%	1%	0%
`trot`	24%	7%	−9%	13%	56%	25%	−22%	31%	28%	30%	19%	15%
`resid`	87%	87%	73%	84%	102%	66%	51%	79%	75%	82%	68%	66%
ΔW_{func}	109%	96%	100%	100%	116%	90%	90%	98%	85%	101%	94%	97%

Note. Evaluated separately from Table 5, with recoveries normalized to this evaluation's own W_0 and W_T.

Table 6: Whole-model channel interventions (Qwen3-4B-Base, step 2599) as a share of each arm's $W_0 \to W_T$ gain. Add: gain recovered by $W_0 + \Delta W_C$; Remove: gain lost by $W_T - \Delta W_C$. Accuracy is AIME24 avg@16; length is the mean number of generated tokens.

about two thirds even at rank 128, and its length curve shows no knee in the plotted range. The gap is relative: even AdamW loses about half of its length gain at rank 16, and no optimizer approaches the rank-1 recovery reported by Cai et al. (2025). SignSGD compresses about as well as AdamW, though its accuracy recoveries are less reliable because its reconstruction misses part of its accuracy gain. Muon's functional endpoint thus spreads the learned behavior over many more singular directions of each matrix, consistent with Newton–Schulz giving every step equal singular values.

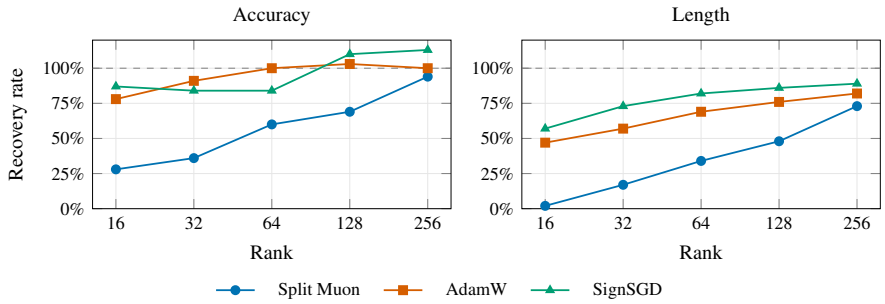

Figure 6: Accuracy and length recovery after truncating each matrix's ΔW_{func} to its leading singular directions (step 2599), normalized so that W_0 is 0% and each optimizer's untruncated $W_0 + \Delta W_{\text{func}}$ is 100%. This is a diagnostic reconstruction, not a low-rank training method.

Optimizers place function in different tensors. We next ask where each optimizer places the learned function across tensor families. We test tensor-family sufficiency by adding one family's functional endpoint to the base checkpoint weights W_0, and necessity by deleting one family from the endpoint weights W_T. Table 7 gives both for split Muon, fused Muon, AdamW and SignSGD.

- **Muon places accuracy in W_V; the coordinate-wise optimizers do not.** Under both split and fused Muon, W_V is the largest single accuracy carrier. Under AdamW and SignSGD, W_{down} carries the most accuracy, and W_V carries no more than the other attention projections.
- **W_{down} carries response length under every optimizer.** It leads length recovery in all four arms, and removing it loses the most length in each, 14–35% of the length gain.
- **The partition does not change the tensor profile.** $\text{Muon}_{\text{split}}$ and $\text{Muon}_{\text{fused}}$ give nearly the same per-tensor recoveries, so the tensor-level differences mainly come from the optimizer.

	$\text{Muon}_{\text{split}}$		$\text{Muon}_{\text{fused}}$		AdamW		SignSGD	
Tensor	acc.	length	acc.	length	acc.	length	acc.	length
Added alone to W_0: share of the gain recovered								
W_{down}	12%	**17%**	13%	**19%**	**47%**	**20%**	**27%**	**23%**
W_V	**24%**	6%	**22%**	11%	19%	8%	4%	4%
W_O	−1%	2%	−3%	7%	22%	9%	5%	11%
W_Q	3%	−5%	6%	2%	22%	0%	13%	3%
W_K	−2%	1%	6%	2%	21%	−2%	10%	−1%
Removed from W_T: share of the gain lost								
W_{down}	10%	**14%**	22%	**35%**	−10%	**27%**	−9%	**26%**
W_V	3%	11%	26%	21%	−19%	11%	−3%	6%
W_O	8%	−4%	13%	20%	−18%	16%	0%	10%
W_Q	11%	1%	9%	6%	6%	−1%	−5%	−3%
W_K	1%	0%	4%	5%	−1%	−5%	−1%	3%

Table 7: Per-tensor interventions (Qwen3-4B-Base, AIME24) as a share of each arm's $W_0 \to W_T$ gain: one tensor family's ΔW_{func} is added to W_0 (top) or removed from W_T (bottom). Length is the mean number of generated tokens. $\text{Muon}_{\text{fused}}$ is evaluated at step 2739, the others at step 2599. Bold marks each column's largest entry; removal accuracy is left unmarked because it is within the noise of 30 problems.

- **Function is distributed.** Under $\text{Muon}_{\text{split}}$, the five single-tensor recoveries sum to only about a fifth of the length gain and a third of the accuracy gain, and removing any one family keeps at least 85% of the length gain. Adding W_V and W_{down} together at the same step, in a separate evaluation, recovers 50% of the accuracy gain and 63% of the length gain, more than the two single-tensor recoveries combined, suggesting a possible pathway: value projections change what attention carries, and MLP output projections change what is written back to the residual stream.

7 Related Work

Muon and matrix-level optimization. Muon orthogonalizes each matrix update with Newton–Schulz iterations and is widely used for pretraining (Jordan et al., 2024; Liu et al., 2025; Wang et al., 2025). Recent pretraining reports orthogonalize attention at semantic boundaries rather than on fused tensors, separating QKV (Qwen Team, 2026) or each head (GLM-5 Team, 2026; Kimi Team, 2026); we test per-head partitioning in RLVR and trace its effect to the optimizer step.

Geometry of RLVR updates. RLVR updates move away from the principal singular directions of the base weights (Zhu et al., 2025; 2026), are often recoverable at rank one (Cai et al., 2025; Wei et al., 2026), and are learned well by plain SGD without momentum (Mukherjee et al., 2026). Under Muon, the first property holds but the second does not, and the lack of a persistent direction that makes momentum weak for SGD becomes excess displacement (See detailed discussion in Appendix A).

8 Conclusion

With per-head QKV splitting and no momentum, Muon extends the useful RLVR training horizon and outperforms coordinate-wise optimizers on both AdamW- and Muon-pretrained bases. Splitting preserves cross-head agreement, and removing momentum avoids turning short-lived correlation into excess displacement. Muon learns through the same off-principal channel as AdamW, but less compressibly and through different tensors. Matrix partition and momentum are thus algorithmic components of Muon, not implementation details; broader validation across base models remains needed.

AI USE STATEMENT

We used AI tools to assist with language polishing, code implementation, and literature research. The authors are responsible for the final manuscript, implementation, and references.

REPRODUCIBILITY STATEMENT

Section 2 and Table 1 specify every optimizer arm, including the QKV partition, momentum, Newton–Schulz iterations, per-block scaling and learning-rate calibration. Section 3 and Appendix B give the training and evaluation protocols, and Appendix B.3 describes how the update-geometry and functional-endpoint measurements are computed. All base models (Qwen3-4B-Base, Moonlight-16B-A3B-Instruct), the training data (DAPO-Math-17k) and the evaluation benchmarks are publicly available. We will release our code after the review period.

REFERENCES

Jeremy Bernstein. Deriving muon, 2025. URL https://jeremybernste.in/writing/deriving-muon.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signSGD: Compressed optimisation for non-convex problems. *International Conference on Machine Learning*, 2018.

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

Franz Louis Cesista. Squeezing 1-2% efficiency gains out of Muon by optimizing the Newton–Schulz coefficients. https://leloykun.github.io/ponder/muon-opt-coeffs/, 2025. Published 21 February 2025.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Chongyu Fan, Gaowen Liu, Mingyi Hong, Ramana Rao Kompella, and Sijia Liu. Rethinking Muon beyond pretraining: Spectral failures and high-pass remedies for VLA and RLVR. *arXiv preprint arXiv:2605.19282*, May 2026. URL https://arxiv.org/abs/2605.19282.

GLM-5 Team. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*, 2021.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. Technical blog post, 2024. URL https://kellerjordan.github.io/posts/muon/.

Keller Jordan, Jeremy Bernstein, Franz Louis Cesista, and Jiacheng You. modded-nanogpt: Speedrunning NanoGPT. https://github.com/KellerJordan/modded-nanogpt, 2025. Quintic Newton–Schulz coefficient schedule, `train_gpt_medium.py`; strategy credited to @jxbz, @leloykun, and @YouJiacheng.

Kimi Team. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Xianliang Li, Zihan Zhang, Weiyang Liu, and Han Bao. Denoise first, orthogonalize later: Understanding momentum in Muon via spectral filtering. *arXiv preprint arXiv:2606.03899*, 2026.

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

Jingyuan Liu, Jianlin Su, Xingcheng Yao, et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025.

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

Martin Marek, Sanae Lotfi, Aditya Somasundaram, Andrew Gordon Wilson, and Micah Goldblum. Small batch size training for language models: When vanilla SGD works, and why gradient accumulation is wasteful. *arXiv preprint arXiv:2507.07101*, 2025. NeurIPS 2025.

Sagnik Mukherjee, Lifan Yuan, Pavan Jayasinha, Dilek Hakkani-Tür, and Hao Peng. Do we need Adam? surprisingly strong and sparse reinforcement learning with SGD in LLMs. *arXiv preprint arXiv:2602.07729*, 2026.

NVIDIA. Nemo rl: A scalable and efficient post-training library. https://github.com/NVIDIA-NeMo/RL, 2025. GitHub repository.

Qwen Team. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. *Technical report*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Thinking Machines Lab. Inkling: Our open-weights model. https://thinkingmachines.ai/news/introducing-inkling/, 2026. Accessed 2026-09-23.

Shuche Wang, Fengzhuo Zhang, Jiaxiang Li, Cunxiao Du, Chao Du, Tianyu Pang, Zhuoran Yang, Mingyi Hong, and Vincent Y. F. Tan. Muon outperforms Adam in tail-end associative memory learning. *arXiv preprint arXiv:2509.26030*, 2025.

Zhepei Wei, Xinyu Zhu, Wei-Lin Chen, Chengsong Huang, Jiaxin Huang, and Yu Meng. You only need minimal RLVR training: Extrapolating LLMs via rank-1 trajectories. *arXiv preprint arXiv:2605.21468*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Huan Zhu, Jiwei Xu, J. Li, Kunlun Chen, and Ting He. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, David Z. Pan, Zhangyang Wang, Yuandong Tian, and Kai Sheng Tai. The path not taken: RLVR provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

Hanqing Zhu, Wenyan Cong, Zhizhou Sha, Sagnik Mukherjee, Xinyuan Song, David González-Martínez, Xiaoxia Wu, Yuandong Tian, Shiwei Liu, David Z. Pan, and Zhangyang Wang. ISO: An RLVR-native optimization stack. *arXiv preprint arXiv:2607.19331*, 2026.

A Extended Related Work

Muon and matrix-level optimization. Muon orthogonalizes each matrix update with Newton–Schulz iterations (Jordan et al., 2024), which makes it steepest descent under the spectral norm (Bernstein, 2025); later work tunes the Newton–Schulz coefficients for faster convergence (Jordan et al., 2025; Cesista, 2025). After Moonlight showed that Muon scales to language-model pretraining (Liu et al., 2025), it was adopted for pretraining Kimi K2, GLM-5, DeepSeek-V4 and Inkling (Kimi Team, 2025; GLM-5 Team, 2026; DeepSeek-AI, 2026; Thinking Machines Lab, 2026). Its benefit can depend on the parameter family (Wang et al., 2025).

Muon for reinforcement learning. Kimi K2 uses Muon for both supervised fine-tuning and RL, motivated by matching the pretraining optimizer, but does not compare optimizers in RL (Kimi Team, 2025). Muon has been reported to add little value in RL training, particularly on AdamW-pretrained bases (NVIDIA, 2025), and Fan et al. (2026) observe collapse under Muon in RLVR, which they attribute to Newton–Schulz amplifying a noise-dominated spectral tail. We find that the outcome depends on the recipe, and that in our setting momentum rather than orthogonalization is the main driver of early collapse (Sections 3 and 5).

Geometry of RLVR updates. Prior work studies how RLVR modifies pretrained singular directions and argues that useful motion can occur away from the principal subspace (Zhu et al., 2025; 2026), and Mukherjee et al. (2026) find that plain SGD produces sparse updates in RLVR. RLVR updates are also functionally low-rank: Cai et al. (2025) and Wei et al. (2026) find that a rank-1 approximation of the parameter update recovers most of the performance gain, and use that direction, estimated from an early training window, to extrapolate later checkpoints. We revisit these observations under Muon. Its functional endpoint also lies off the principal directions, but it is far less compressible than AdamW's, so the low-rank observation does not carry over (Section 6).

B Additional Experimental Details

B.1 Training and evaluation protocols

Qwen3-4B-Base training. All Qwen arms start from the same initialization and train on DAPO-Math-17k using chat-templated prompts and a math verifier that does not require an instruction-model thinking tag. GRPO uses 32 prompts per rollout, eight samples per prompt, global batch size 256, a 16k rollout cap, zero KL coefficient, zero entropy coefficient, and no PPO clipping, since updates are fully on-policy. All arms use learning rate 10^{-6} and weight decay 0.1. The learning-rate calibration of Muon (Section 2) is its block scaling: multiplying each orthogonalized block by $\sqrt{\max(m,n)}$ gives the update unit root-mean-square entries, matching the per-coordinate scale of the coordinate-wise updates, so no additional multiplier is applied. All arms are compared through step 3000; arms trained longer are evaluated through their last saved checkpoint (split Muon through step 4399). Fp32 optimizer masters are saved every 20 steps. Split Muon uses five quintic Newton–Schulz iterations, spectral scaling $\sqrt{\max(m,n)}$, 48 per-head $[128, 2560]$ QKV blocks, and distributed full-matrix polar transforms for unsplit matrices.

Qwen evaluation. AIME24, AIME25, and BeyondAIME are evaluated with eight stochastic samples per problem and math verification. We report the macro-average of the three task metrics. Raw maxima are descriptive; the matched-step trajectory from steps 1499–2599 (Table 9) provides selection-robust context.

Generation budgets. The 16k and 64k labels in the main text are nominal. For Qwen3-4B-Base, the 64k budget allows 63,488 completion tokens after a 2,048-token prompt reserve, and the 16k budget allows 16,384 completion tokens. For Moonlight-16B-A3B-Instruct, the 16k budget allows 14,336 completion tokens after a 2,048-token prompt reserve. Truncation rates (Tr.) in all tables are measured against these limits.

Moonlight-16B-A3B-Instruct. The 16k evaluation uses $K = 8$, math verification, and a 14,336-token completion allowance after a 2,048-token prompt reserve. The four-task aggregate contains

OlympiadBench ($n = 675$), AIME24 ($n = 30$), BeyondAIME ($n = 100$), and MATH500 ($n = 500$); AIME25 remains excluded from the reported Moonlight aggregate. Each arm is selected independently by its raw macro-averaged Pass@1: split Muon reaches 0.3260 at step 1339, fused Muon reaches 0.3159 at step 699, and SignSGD reaches 0.3129 at step 339. Their corresponding macro-averaged truncation rates are 43.71%, 24.33%, and 41.92%. Because the arms have unequal training durations and the split arm's selected checkpoint has substantial truncation, the small split–fused margin is descriptive rather than a final converged estimate; a duration-matched, longer-context evaluation and additional seeds remain necessary.

B.2 Supplementary performance results

Qwen 16k evaluation. See Table 8 and Figure 7.

	AIME24			AIME25			BeyondAIME			Macro-average		
Method (selected step)	P@1	P@8	Tr.↓	P@1	P@8	Tr.↓	P@1	P@8	Tr.↓	P@1	P@8	Tr.↓
$\text{Muon}_{\text{split}}$ (2999)	**35.0**	**56.7**	**20.8**	**29.2**	50.0	**16.7**	12.8	26.0	15.0	**25.6**	**44.2**	**17.5**
$\text{Muon}_{\text{fused}}$ (3719)	30.4	53.3	32.1	25.8	46.7	32.9	10.9	24.0	35.1	22.4	41.3	33.4
AdamW (1719)	31.3	50.0	27.5	28.3	**53.3**	29.2	10.5	24.0	21.4	23.4	42.4	26.0
SignSGD (1759)	29.6	40.0	22.5	28.3	36.7	18.3	**13.4**	**30.0**	**12.0**	23.8	35.6	17.6

Table 8: Qwen3-4B-Base 16k evaluation at each optimizer's macro-averaged Pass@1-best checkpoint. All values are percentages.

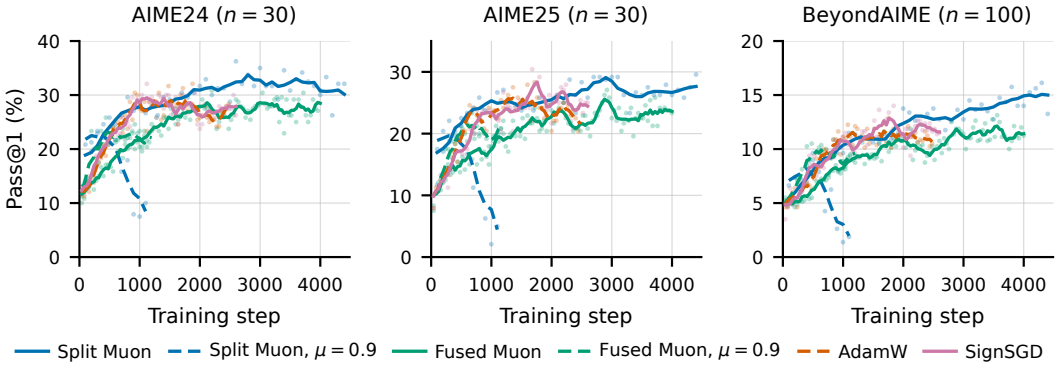

Figure 7: Per-task Pass@1 of Qwen3-4B-Base on the 16k evaluation over all evaluated checkpoints, for the matched recipe comparison. Split and fused Muon differ only in whether QKV is partitioned into heads before Newton–Schulz. Dashed lines use momentum ($\mu = 0.9$ for Muon, $\beta_1 = 0.9$ for AdamW); lines are 5-point moving averages of the checkpoints (dots).

Matched-step Qwen 64k evaluation. See Table 9.

step	$\text{Muon}_{\text{split}}$	$\text{Muon}_{\text{fused}}$	AdamW	SignSGD
1499	23.3	17.9	23.5	20.3
1599	25.4	19.8	21.7	24.3
1799	24.7	20.5	24.5	24.6
1999	25.9	22.8	22.4	22.0
2199	25.6	20.7	22.7	20.4
2299	28.1	19.3	21.9	21.5
2499	26.4	18.8	23.8	22.9
2599	26.8	19.2	21.2	21.9

Table 9: Macro-averaged Pass@1 (%) at selected common checkpoints from the Qwen3-4B 64k evaluation. These matched-step values provide trajectory context beyond the per-arm best-checkpoint summary in Table 2.

Momentum at larger batch sizes. With $\mu = 0.9$, batches of 512 and 1024 both fall behind momentum-free $\text{Muon}_{\text{split}}$ at equal rollout samples, and the batch-1024 run still collapses (Figure 8).

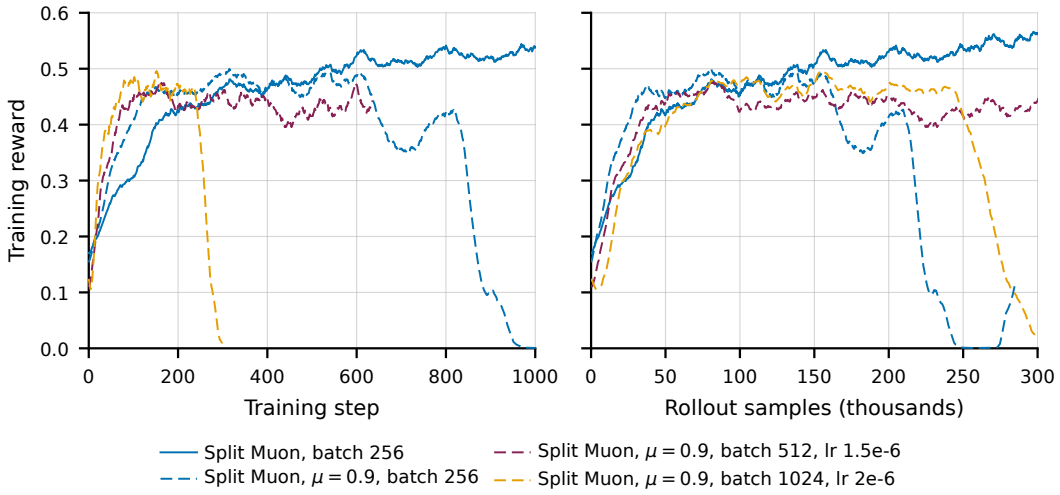

Figure 8: Training reward of Qwen3-4B-Base for Muon$^{\mu}_{\text{split}}$ at global batch sizes 256, 512 and 1024 (32, 64 and 128 prompts per rollout with eight samples each; learning rates 10^{-6}, 1.5×10^{-6} and 2×10^{-6}), with momentum-free Muon$_{\text{split}}$ at batch 256 for reference, against training steps (left) and rollout samples (right). Each line is a centered moving average over about 13k samples (51, 25 and 13 steps at batch 256, 512 and 1024).

B.3 Mechanistic measurement protocols and diagnostics

Base-frame intervention and compression. For $W_0 = U_0 \Sigma_0 V_0^\top$ and displacement X, define $M_X = U_0^\top X V_0$ and decompose

$$\begin{aligned} X_{\text{diag}} &= U_0 \,\text{diag}(M_X) V_0^\top, \\ X_{\text{hrot}} &= \text{off-diagonal terms involving the top-}k\text{ base directions}, \\ X_{\text{trot}} &= \text{tail–tail off-diagonal terms}, \\ X_{\text{resid}} &= X - U_0 M_X V_0^\top. \end{aligned}$$

We use head width $k = 8$ throughout, except in the head-width check of Section 6, which repeats $W_0 + \Delta W_{\text{func}}$ at $k \in \{32, 128, 512\}$. Add-alone checkpoints $W_0 + \Delta W_C$ test sufficiency; leave-one-out checkpoints $W_T - \Delta W_C$ test necessity. Geometry is computed after fp32 subtraction and projection; inference checkpoints are reconstructed in bf16. Leading-SVD functional compression is performed independently per projection matrix and is an endpoint diagnostic rather than a low-rank training method.

Semantic-partition reconstruction and controlled construction. We reconstruct the optimizer map from fp32 masters for five layers and 15 checkpoints between steps 39 and 319. The stored QKV optimizer state remains fused even in the split run. From the same pre-transform gradient or momentum tensor, we compute the update actually applied by the split recipe and a fused-QKV counterfactual. This holds the rollout batch, optimizer input, Newton–Schulz polynomial, number of iterations, learning rate, and weight decay fixed; only the matrix boundary and its induced shape scaling change.

Local breadth and global alignment. Across the measured layers and checkpoints, the zero-momentum pre-NS QKV input has median stable rank 2.08. The full QKV step $\widetilde{G}$ has stable rank 1195 under the fused transform and 130.8 when the split transform's head blocks are restacked. The momentum run gives the same ordering: input stable rank 2.27, fused counterfactual 1441, and split step 130.7. Individual heads remain broad, with median stable rank 105.2 out of 128, while their Frobenius-norm coefficient of variation falls from 0.43 to 0.008 at $\mu = 0$ and from 0.41 to 0.007 at $\mu = 0.9$.

Let F denote a head's Frobenius norm and $r_h = 105.2$ its stable rank. For the stacked matrix A,

$$\|A\|_F^2 = 48F^2, \qquad \frac{\sigma_1(A)^2}{\sigma_1(A_h)^2} = \frac{48 r_h}{\text{srank}(A)} \approx 38.6. \tag{3}$$

The maximum is 48 when all heads share the same leading right-singular direction. The observed ratio places them roughly 80% of the way from mutually orthogonal to fully collinear, explaining how locally broad heads can form a much more concentrated $\widetilde{G}$.

Shape-dependent scaling and endpoint persistence. Muon evaluates $\sqrt{\max(m,n)}$ on each semantic block after splitting rather than once on the fused storage shape. A shape-only calculation predicts a uniform $\sqrt{2560}/\sqrt{6144} = 0.6455$ multiplier, but the reconstructed split/fused norm ratios are 2.02/1.20/0.87 for Q/K/V at $\mu = 0$ and 1.86/1.08/0.82 at $\mu = 0.9$, with total QKV ratios 1.29 and 1.17. Independent polar normalization therefore changes both singular-direction coupling and relative block scale. The current experiment does not identify which component causes the performance gain; a norm-matched control would separate them.

The step-level signature survives at the endpoint. Functional W_V stable rank falls from 244.4 under fused QKV to 82.6 under per-head splitting, whereas W_{down}, whose Newton–Schulz boundary is unchanged, rises from 673.5 to 758.0. This opposite-moving internal control rules out generic training progress as the explanation for the W_V shift, although rotating directions partially average the instantaneous difference over training.

Endpoint-rank terminology. For a matrix A, stable rank is $\|A\|_F^2/\|A\|_2^2$. It measures spectral concentration and can change substantially when only the leading singular value changes. Entropy effective rank and rank at 90% energy are not interchangeable with stable rank. "Causal compression" in this paper denotes behavioral recovery after leading-SVD endpoint reconstruction; it is neither a claim about the minimum intrinsic dimension nor a low-rank training prescription.

Channel accounting. The four base-frame components sum exactly to the measured displacement. Nevertheless, add-alone behavioral effects need not sum because model function is nonlinear. A successful $\mathrm{trot} + \mathrm{resid}$ reconstruction establishes joint sufficiency at the endpoint, not independent necessity of every singular direction. Likewise, an inert single channel can participate in a nonlinear interaction that is absent when it is added alone. We therefore state only sufficiency when a matched leave-one-out result is unavailable, and make necessity claims only for tensor/channel interventions that have both directions measured.

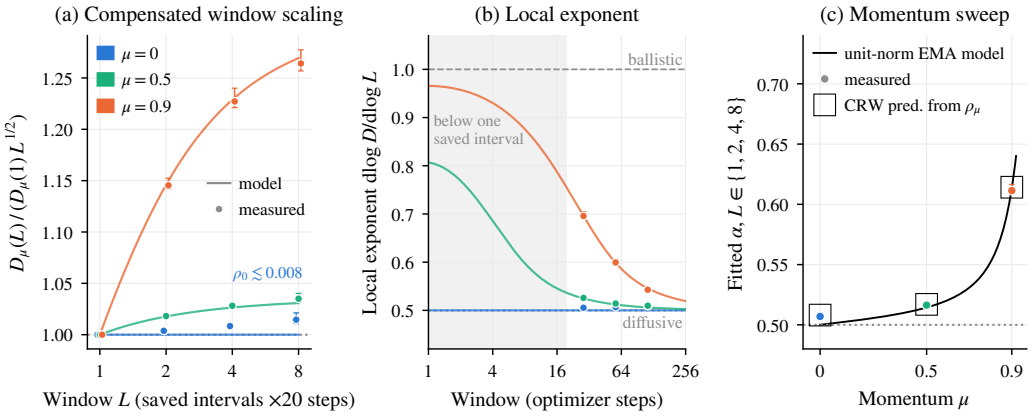

Figure 9: **Short-range correlation fully accounts for window scaling under momentum.** Functional projection P_{func}; windows in saved intervals of 20 optimizer steps. Lines: the parameter-free unit-norm EMA model described above. Points: measurements averaged over nine depth-band×Q/K/V cells (error bars: s.d. across cells). **(a)** Diffusion-compensated displacement $D_\mu(L)/\big(D_\mu(1)\,L^{1/2}\big)$; a flat line means $\alpha = 1/2$. The slight rise at $\mu = 0$ reflects weak intrinsic correlation that the i.i.d. model omits. **(b)** Local exponent versus window length in optimizer steps. The ballistic-to-diffusive transition falls mostly inside the first saved interval (shaded), so the measured windows see only its tail. **(c)** Exponent fitted over $L \in \{1, 2, 4, 8\}$ versus μ: the model, the measurements and the correlated-random-walk prediction of Eq. (4) (squares) agree for all arms.

Window scaling under momentum. Section 5 fits $D_\mu(L) \propto L^{\alpha_\mu}$, where $\alpha = 1/2$ indicates independent increments and $\alpha = 1$ coherent drift. For increments with autocorrelation $\rho(k)$, the

normalized cumulative variance over a window of L intervals is

$$V(L) = L + 2\sum_{k=1}^{L-1}(L-k)\,\rho(k), \tag{4}$$

so a correlated random walk predicts $D_\mu(L) \propto V(L)^{1/2}$, which reduces to $L^{1/2}$ when $\rho \equiv 0$. Figure 9 compares the measurements with a unit-norm EMA model that has no fitted parameters: i.i.d. gradient directions pass through an EMA with coefficient μ, each step is normalized to unit norm as after Newton–Schulz, and the trajectory is sampled every 20 steps. Momentum correlates steps over roughly $(1-\mu)^{-1}$ optimizer steps, so most of the transition from ballistic to diffusive motion happens within the first saved interval, and a fit over $L \in \{1, 2, 4, 8\}$ sees only its tail. This is why $\mu = 0.9$ yields $\alpha \approx 0.61$ rather than a value near 1. The model, the measurements, and the correlated-random-walk prediction of Eq. (4) agree for every measured arm, so no persistent drift direction is needed to explain the exponents.