When Does Muon Help Agentic Reinforcement Learning?

Kai Ruan[1], **Jinghao Lin**[2], **Zihe Huang**[3]
Ziqi Zhou[4], **Qianshan Wei**[5], **Xuan Wang**[6], **Hao Sun**[1,*]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Independent Researcher
[3]Institute of Computing Technology, Chinese Academy of Sciences
[4]Duke University
[5]Institute of Automation, Chinese Academy of Sciences
[6]College of Computer Science and Technology, Zhejiang University
[*]Corresponding author

Abstract

Muon is competitive with AdamW in large-scale pre-training, but its value for reinforcement-learning (RL) post-training remains unclear. We study vanilla Muon in sparse-reward agentic RL through matched comparisons with AdamW on ALFWorld using Qwen2.5-0.5B-Instruct. Under Group-in-Group Policy Optimization (GiGPO), applying Muon only to hidden weight matrices raises final-window validation success from 0.290 to 0.546 (**+88%**); high-rate AdamW controls retain no post-update success. The effect depends on the advantage estimator and learning rate. At 3×10^{-5}, Muon improves GRPO from 0.161 to 0.268, whereas at 10^{-5} it reaches 0.185. Meanwhile, GraphGPO's late-window gap narrows near saturation. At 10^{-5}, GraphGPO Muon reaches 0.901, raises normalized validation AUC from 0.399 to 0.556, and reaches $0.5/0.75$ success $30/60$ updates earlier. A second matched GraphGPO seed preserves Muon's late-window and AUC advantage. These exploratory results motivate studying the policy optimizer, advantage estimator, and learning rate jointly; broader multi-seed and cross-task validation remain open.

Introduction

Muon (Jordan et al. 2024) replaces the element-wise adaptive scaling of Adam-family optimizers with an approximate spectral normalization of the momentum matrix, computed via Newton–Schulz (NS) iterations. In pre-training, this simple change is remarkably effective: Muon matches AdamW's final loss with roughly 52% of the training FLOPs at billion-parameter scale (Liu et al. 2025), and has since been used in trillion-parameter pre-training (Kimi Team 2025). Yet its authors explicitly identified whether Muon transfers to *post-training*, particularly reinforcement learning (RL), as an open question (Jordan et al. 2024).

Existing evidence is mixed. NeMo RL has added Muon support and reports "minor improvements" when Muon is introduced only during post-training (NVIDIA 2026). Studies of *optimizer mismatch* report a worse learning–forgetting tradeoff when Adam-pretrained models are fully fine-tuned with Muon, with greater disruption or forgetting at larger learning rates (Qu, Huang, and Horvath 2026; Liu, Wang, and Zhang 2026). These results are confined to supervised fine-tuning and show substantial learning-rate sensitivity. Most recently, Fan et al. (2026) report that vanilla Muon *fails* under RLVR with GRPO-style objectives, attributing the failure to spectral whitening amplifying the noise-dominated tail of low signal-to-noise policy gradients. Public engineering reports also describe persistent entropy floors and gradient explosions in mathematical-reasoning RL; Hopper combines variance normalization with a single NS step to improve stability (Wei 2026a,b). The negative RL evidence, however, is concentrated in *single-turn* RLVR with *episode-level* (outcome-only) advantage estimation.

We test whether Muon's viability in RL post-training depends on the structure of the advantage estimator. We study long-horizon, sparse-reward agentic RL on ALFWorld (Shridhar et al. 2021), training a Qwen2.5-0.5B-Instruct (Yang et al. 2024) agent, and compare a family of group-based RL algorithms spanning distinct credit-assignment structures: GRPO (Shao et al. 2024) with episode-level advantages; GiGPO (Feng et al. 2025), which augments the identical episode-level advantage with a step-level term computed by grouping actions taken from repeated *anchor states* across trajectories; and GraphGPO (Cheng et al. 2026), which aggregates rollouts into a unified state-transition graph and assigns transition-level credit from successor-state distance to the task goal. GiGPO provides a controlled comparison because its advantage is $A = A^E + \omega A^S$, and it reduces to GRPO when the step-level term vanishes. This exposes the step-level contribution within a single GiGPO code path without conflating it with a separate GRPO implementation.

Figure 1 previews the central empirical comparison across the three advantage estimators. The experiments yield three observations. First, applying Muon only to the policy's hidden weight matrices (embeddings, norms, and all other parameters keep the baseline AdamW configuration) lifts GiGPO's validation success rate on ALFWorld from 0.29 to 0.55 averaged over the final evaluation window, an 88% relative improvement. It reaches 0.63 versus 0.32 at the final checkpoint, with the largest gains on task categories where AdamW attains its lowest success. Second, learning-rate controls test whether a larger nominal AdamW rate alone reproduces the Muon gain. Intermediate AdamW rates of 3×10^{-6} and 5×10^{-6} underperform the 10^{-6} baseline, while controls at 10^{-5} and 3×10^{-5} record zero post-update validation suc-

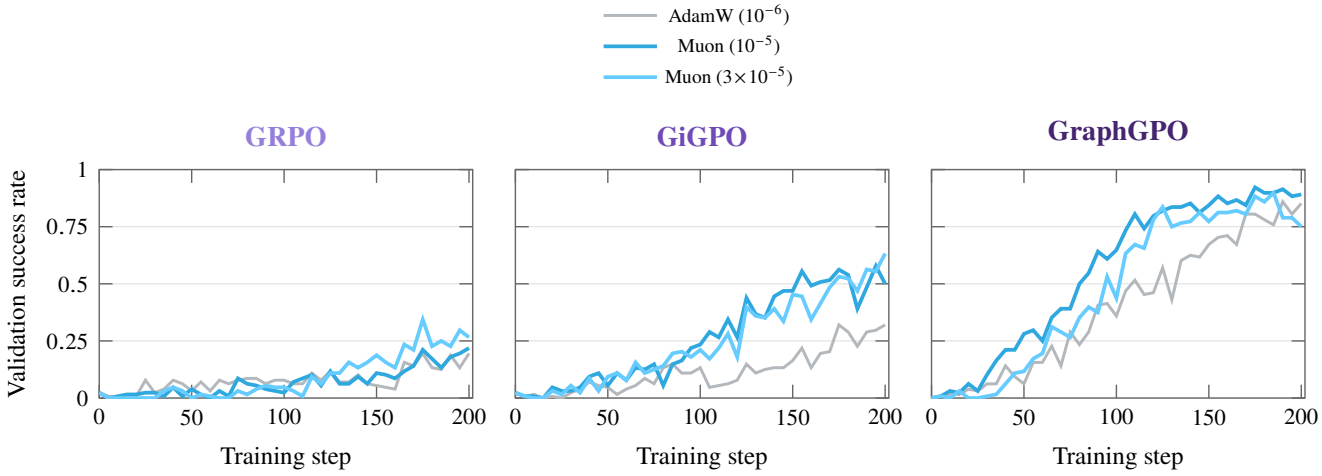

Figure 1: Muon improves late-window quality and learning efficiency, with learning-rate sensitivity that varies by advantage estimator. Every run reaches step 200; curves show raw validation checkpoints without smoothing. AdamW is gray throughout. Blue shades distinguish Muon learning rates, while subplot titles identify the advantage estimator. All three estimators include both 10^{-5} and 3×10^{-5}.

cess. Muon retains nonzero validation success through step 200 at 3×10^{-5}. Third, the effect varies across advantage estimators. Under GRPO, Muon reaches 0.185 at 10^{-5} and 0.268 at 3×10^{-5}, compared with 0.161 for AdamW. Under GraphGPO, the late-window gap narrows as the curves approach saturation: Muon at 3×10^{-5} reaches 0.83 versus 0.81 for AdamW, but its normalized validation AUC is 0.47 versus 0.40. At 10^{-5}, Muon reaches 0.90, with normalized AUC 0.56 and earlier threshold crossings. A matched second seed preserves both GraphGPO Muon advantages.

We use gradient signal-to-noise ratio (SNR) as an organizing hypothesis for the positive and negative evidence in the literature. Muon's orthogonalization flattens the singular spectrum of the update, boosting weak directions relative to dominant ones. In single-turn, low-success RLVR, those directions may be dominated by sampling noise, consistent with the failures reported by Fan et al. (2026). Our positive GRPO result on long-horizon agentic tasks suggests that episode-level credit alone does not determine whether Muon helps. GiGPO's anchor-state grouping adds per-state contrasts (over 65% of states recur across trajectories in ALFWorld (Feng et al. 2025)). These contrasts change the scalar weights applied to token-level score gradients and may alter the aggregate gradient spectrum, a connection not established by GiGPO theory. Its larger observed gain motivates testing this hypothesis directly and suggests that *advantage-estimator design* complements optimizer-side remedies such as Hopper's variance-normalized, single-step NS update (Wei 2026b) and Pion's high-pass spectral filtering (Fan et al. 2026).

Our contributions are:

- A controlled study of vanilla Muon for long-horizon, sparse-reward agentic RL, including a matched second GraphGPO seed. Under GiGPO, Muon improves final-window success by 88% relative to matched AdamW, with gains spanning several ALFWorld task categories.
- GiGPO learning-rate controls showing that the result is not reproduced by increasing AdamW's nominal rate: intermediate rates underperform the baseline, and AdamW at 10^{-5} and 3×10^{-5} retains no post-update success, whereas Muon at 3×10^{-5} retains success through all 200 updates.
- A matched 2×2 GiGPO ablation showing Muon gains with and without step-level credit, together with a trajectory-level comparison across GRPO, GiGPO, and GraphGPO. A gradient-SNR conjecture connects the estimator-dependent pattern to recent negative Muon results under RLVR.

Related Work

Muon and matrix-aware optimizers. Muon (Jordan et al. 2024) orthogonalizes the momentum matrix of each 2D hidden-layer parameter via Newton–Schulz iterations, and is typically deployed with an auxiliary Adam-family optimizer for embeddings, norms, and output heads. Liu et al. (2025) demonstrated scalability to LLM pre-training with roughly half of AdamW's FLOP budget, and MuonClip (Kimi Team 2025) stabilized trillion-parameter pre-training via QK-clip; we note that QK-clip addresses attention-logit explosion in pre-training and is orthogonal to the RL-stability questions studied here. With decoupled weight decay, Muon is a nuclear-norm Lion-$\mathcal{K}$ method that implicitly enforces a spectral-norm constraint, although this characterization does not predict RL behavior (Chen, Li, and Liu 2025). Pion replaces uniform spectral whitening with a high-pass NS transformation that suppresses tail components (Fan et al. 2026); our work is complementary in that we hold the optimizer fixed (vanilla Muon) and vary the RL objective.

Muon in post-training and optimizer mismatch. Qu, Huang, and Horvath (2026) and Liu, Wang, and Zhang (2026) study optimizer mismatch in supervised fine-tuning. The former finds that full Muon fine-tuning underperforms matched Adam on Adam-pretrained models and becomes more disruptive as its learning rate grows; the latter finds a better learning–forgetting tradeoff when pre-training and fine-tuning use the same optimizer, with more forgetting at larger learning rates. In the evaluated Llama-2 settings, Qu, Huang, and Horvath (2026) select 2×10^{-5}–5×10^{-5} for full-Muon instruction tuning, while Liu, Wang, and Zhang (2026) sweep 1×10^{-5}–1×10^{-4} when analyzing forgetting; these are experiment-specific settings, not universal thresholds. Our RL experiments use lower nominal rates and KL regularization toward a reference policy; Qwen2.5's pretraining optimizer is undisclosed. Fan et al. (2026) report that vanilla Muon fails under RLVR, attributing failure to spectral whitening under low gradient SNR, and propose a high-pass spectral remedy; community experiments similarly report entropy floors and gradient explosions with GRPO-family objectives on mathematical reasoning. Hopper reports improved stability from a combined recipe of variance normalization and one NS step (Wei 2026a,b). In supervised post-training, Gupta et al. (2025) instruction-tune a Muon-pretrained GPT checkpoint with Muon, establishing feasibility under optimizer continuity in supervised post-training. Kimi K2 and INTELLECT-3 describe RL stages that continue from Muon-pretrained bases (Kimi Team 2025; Prime Intellect Team et al. 2025), making optimizer continuity one plausible factor. Our experiments compare Muon's behavior across three advantage-estimator implementations.

Group-based RL for LLM agents. Group-based methods, including RLOO (Kool, van Hoof, and Welling 2019; Ahmadian et al. 2024), GRPO (Shao et al. 2024), and DAPO (Yu et al. 2025), estimate advantages from within-group statistics of rollouts. For multi-turn agents, credit assignment across long horizons is the central difficulty. GiGPO (Feng et al. 2025) adds a step-level relative advantage by retroactively grouping actions taken from repeated anchor states across trajectories, at negligible cost and without extra rollouts. GraphGPO (Cheng et al. 2026) aggregates rollouts into a unified state-transition graph, estimates each state's distance to the task goal, and assigns edge-level advantages from successor-state distance; under deterministic- environment and fixed-policy assumptions, it also gives a conditional variance result for scalar feedback. This result establishes neither lower policy-gradient variance nor higher gradient-matrix SNR. The three methods therefore span different credit-signal granularities and conditioning.

Preliminaries

Setup. An LLM agent interacts with an environment over discrete steps $t = 1, \ldots, T$: it observes state $\boldsymbol{s}_t$, emits a textual action $\boldsymbol{a}_t \sim \pi_\theta(\cdot \mid \boldsymbol{s}_t, x)$ for task prompt x, and receives reward r_t. Rewards are sparse: in ALFWorld a positive terminal reward is given only on task success, plus a small penalty for invalid actions.

GiGPO. GiGPO (Feng et al. 2025) samples N trajectories from identical initial conditions and combines two group-relative advantages. The *episode-level* term normalizes total returns within the trajectory group, exactly as GRPO:

$$A^E(\boldsymbol{\tau}_i) = \frac{R(\boldsymbol{\tau}_i) - \mathrm{mean}(\{R(\boldsymbol{\tau}_j)\}_{j=1}^N)}{F_{\mathrm{norm}}(\{R(\boldsymbol{\tau}_j)\}_{j=1}^N)}. \tag{1}$$

Here F_{norm} denotes the within-group standard deviation used by the mean-std normalization in our experiments. The *step-level* term exploits the fact that trajectories from the same initial state repeatedly visit identical environment states (*anchor states* $\tilde{\boldsymbol{s}}$). Actions taken from the same anchor state are grouped, and their discounted returns $R_t^{(i)} = \sum_{k \geq t} \gamma^{k-t} r_k^{(i)}$ are normalized within the group to give $A^S(\boldsymbol{a}_t^{(i)})$. The combined advantage is

$$A(\boldsymbol{a}_t^{(i)}) = A^E(\boldsymbol{\tau}_i) + \omega\, A^S(\boldsymbol{a}_t^{(i)}), \tag{2}$$

optimized with a PPO-style clipped objective and a KL penalty to the reference policy. Under matched mean-std normalization, setting $\omega = 0$ yields the same episode-level advantage as GRPO within the GiGPO code path; γ then affects only the discarded step term. The same reduction occurs when no anchor states recur and $A^S = 0$ (Feng et al. 2025). Thus step-level credit is a toggleable component of a single implementation.

Muon. For each 2D hidden weight matrix with momentum-accumulated gradient M, Muon (Jordan et al. 2024) applies the update $\Delta W \propto \mathrm{NS}_5(M) \approx UV^\top$ where $M = U\Sigma V^\top$, i.e., it preserves the singular directions of the momentum while discarding singular-value magnitudes (5 Newton–Schulz iterations, momentum 0.95, Nesterov). Non-matrix parameters (embeddings, norms; the LM head is tied to the embedding in Qwen2.5-0.5B) are delegated to AdamW.

Learning-rate comparability. Muon and AdamW learning rates are not directly comparable quantities because Muon implementations use different matrix-shape scaling conventions. For the non-RMS-matched variants, a practical rule of thumb is to multiply an Adam learning rate by roughly $10\times$ for common hidden dimensions; the RMS-matched Moonlight variant instead reuses the Adam rate (Su 2025; Liu et al. 2025). We therefore impose no universal conversion: each optimizer is evaluated over its own learning-rate range, and all points are reported in Table 2.

Method: Muon for Group-Based Agentic RL

Starting from the AdamW GiGPO baseline, we apply Muon to the policy's hidden weight matrices while retaining AdamW for all remaining parameters:

- **Matrix parameters** (all attention and MLP 2D weights): Muon. The primary GiGPO setting uses $\eta_{\mathrm{muon}} = 3 \times 10^{-5}$; the learning-rate study also evaluates 1×10^{-5} under every estimator.
- **Fallback parameters** (embeddings, norms, and all other non-matrix parameters): AdamW at the baseline learning rate 1×10^{-6}, identical to the baseline configuration.

For each AdamW–Muon pair, the archived configurations match in model, data, rollout, advantage estimation, and training schedule; Appendix A provides the full configuration. Our implementation exposes full, unsharded 2D matrices to the Newton–Schulz iteration (FSDP `NO_SHARD`); Appendix B discusses the resulting memory implications.

Conjecture: credit quality and Muon's weak directions. Before clipping and KL terms, write a sampled layer gradient as $\widehat{G} = \sum_t \widehat{A}_t \nabla_W \log \pi_\theta(a_t \mid s_t) = G^\star + E$, separating a latent signal $G^\star$ from credit and sampling error E. In a simplified trajectory with T decisions but only $K \ll T$ credit-relevant decisions, independent equal-energy errors give

$$\frac{\mathrm{SNR}_{\mathrm{local}}}{\mathrm{SNR}_{\mathrm{episode}}} \approx \frac{T}{K}, \tag{3}$$

if an ideal local-credit estimator suppresses the other $T - K$ error terms. This toy scaling describes the corresponding local-credit limit.

Muon approximately applies the polar factor $\mathcal{P}(U\Sigma V^\top) = UV^\top$. In a fixed-singular-direction slice with signal coefficients $s_i > 0$ and Gaussian errors $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, its expected alignment is

$$\mathbb{E}\langle \mathcal{P}(\widehat{G}), G^\star \rangle_F = \sum_i s_i [2\Phi(s_i/\sigma_i) - 1], \tag{4}$$

where Φ is the standard normal CDF. Thus spectral flattening is useful only when weak-direction signs are reliable. We conjecture that GiGPO's anchor-state contrasts and GraphGPO's transition-level credit can reduce relevant confounders and improve some s_i/σ_i, while GraphGPO's stronger baseline can still leave less late-window headroom. Appendix D states the assumptions, derivation, and testable predictions; direct measurements of gradient SNR and update spectra provide the next test.

Experiments

Experimental Setup

Environment and model. ALFWorld (Shridhar et al. 2021) comprises embodied household tasks (six categories: Pick, Look, Clean, Heat, Cool, Pick Two), with a horizon up to 50 steps, sparse terminal success reward with an invalid-action penalty. Base model: Qwen2.5-0.5B-Instruct (Yang et al. 2024). Training uses verl-agent (Feng et al. 2025), an agentic extension of veRL/HybridFlow (Sheng et al. 2024), with vLLM rollouts (Kwon et al. 2023) on eight NVIDIA H20 GPUs for 200 updates. Each update uses groups of eight over a 16-task batch, and validation evaluates 128 episodes every five updates. Each matched AdamW–Muon pair uses the same model, data, rollout, estimator, and training schedule; estimator-specific and execution hyperparameters appear in Appendix A. Because single-evaluation success rates fluctuate under sampled validation, we report both final-checkpoint values and means over a fixed 25-update late window comprising six evaluations (steps 175–200), computed identically for all runs. For the learning-efficiency view, we also report normalized validation AUC: the trapezoidal area under the raw checkpoint curve from steps 0–200, divided by 200. The main comparison uses the designated seed-0 trajectories; a complete matched GraphGPO replication at seed 2 is reported in Appendix C.

Estimator	Muon lr	Late AdamW	Late Muon	Δ_{late}	Δ_{AUC}
GRPO	1×10^{-5}	0.161	0.185	+0.023	−0.007
GRPO	3×10^{-5}	0.161	**0.268**	+0.107	+0.013
GiGPO	1×10^{-5}	0.290	0.509	+0.219	+0.147
GiGPO	3×10^{-5}	0.290	**0.546**	+0.255	+0.127
GraphGPO	1×10^{-5}	0.810	**0.901**	+0.091	+0.157
GraphGPO	3×10^{-5}	0.810	0.828	+0.018	+0.076

Table 1: Late-window quality and learning efficiency. Late values are mean validation success over steps 175–200; Δ_{AUC} is Muon minus AdamW normalized validation AUC over steps 0–200. AdamW uses 10^{-6} in every row. Bold marks the best tested Muon rate for each estimator under the corresponding metric. Deltas are computed from unrounded values.

Compared configurations. (i) **AdamW baseline**: policy AdamW, lr 1×10^{-6} (the baseline configuration of Feng et al. 2025). (ii) **Muon**: matrix parameters Muon at 3×10^{-5}, fallback AdamW 1×10^{-6}. (iii) **AdamW lr controls**: policy AdamW at $\{3 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}\}$. (iv) **Algorithm family**: matched AdamW–Muon comparisons under GRPO and GraphGPO (adapted to ALFWorld). GRPO, GiGPO, and GraphGPO each include Muon 10^{-5} and 3×10^{-5} as learning-rate sensitivity checks. We also evaluate the matched GiGPO AdamW–Muon pair with the step weight disabled ($\omega = 0$).

Main Results Across Advantage Estimators

Table 1 separates late-window quality from full-trajectory learning efficiency. GiGPO at 3×10^{-5} has the largest late-window gain (about 0.26), whereas GraphGPO at 10^{-5} has the largest normalized AUC gain (about 0.16). GRPO improves late by about 0.11 and normalized AUC by about 0.01 at 3×10^{-5}; its 10^{-5} run improves the late window slightly but has lower full-trajectory AUC than AdamW.

GiGPO shows the largest observed effect. Averaged over the late window, Muon (3×10^{-5}) reaches 0.55 versus 0.29 for AdamW, an 88% relative improvement. It leads consistently from step ∼50 onward and reaches 0.63 versus 0.32 at the final checkpoint, approximately a twofold difference. Its normalized AUC is 0.24 versus 0.11 for AdamW; Muon at 10^{-5} has a slightly higher normalized AUC of 0.26. Detailed values appear in Table 1 and Appendix C.

Observed task-level gains span all three estimators. Figure 2 compares final-window success across the six ALFWorld task categories. Under GRPO at 3×10^{-5}, the largest gains are on Clean, Pick Two, and Pick, while Look slightly favors AdamW. GiGPO gains most on Cool, Pick Two, and Heat; GraphGPO gains most on Look, Pick Two, and Pick. Thus no aggregate gain is driven by one category; Figure 2 reports the task-level magnitudes.

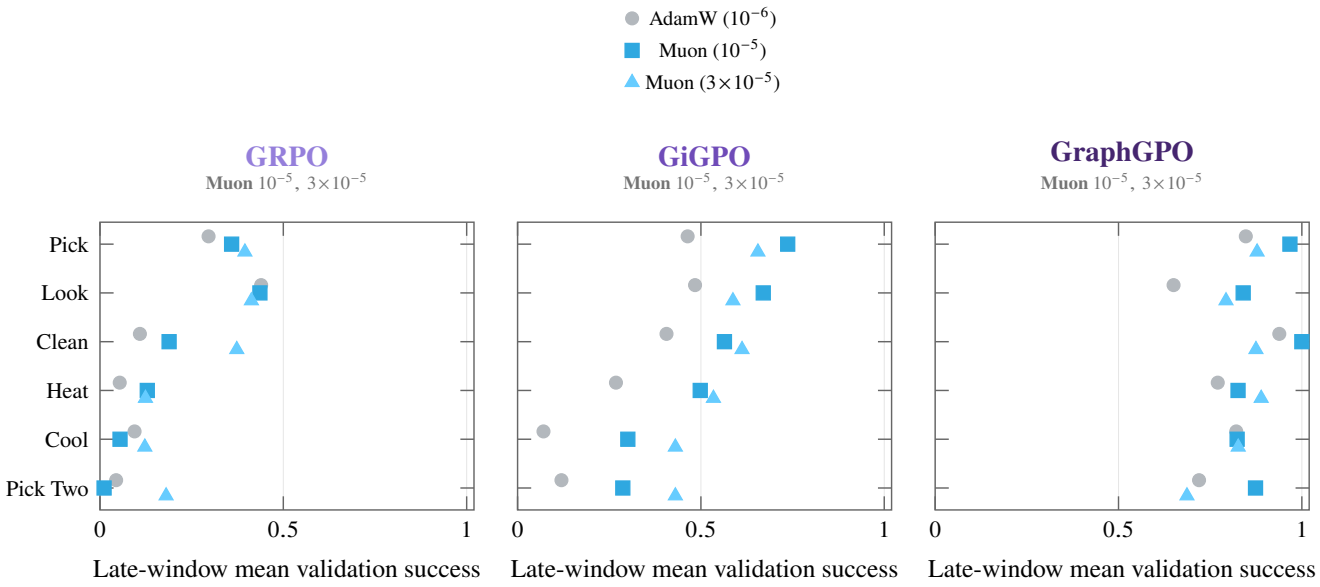

Figure 2: Per-task late-window mean validation success across all three estimators. Gray circles denote AdamW; blue squares/triangles denote Muon rates. The late window contains the six evaluations at steps 175–200. Every estimator includes both tested Muon rates. Offsets are visual only.

Learning-Rate Controls

The Muon recipe changes both the update rule and the effective step size of matrix parameters ($1 \times 10^{-6} \rightarrow 3 \times 10^{-5}$). To test whether a larger nominal AdamW rate alone reproduces the gain, we sweep AdamW from 10^{-6} through 3×10^{-5} (Table 2). AdamW at 3×10^{-6} and 5×10^{-6} retains nonzero success but finishes below the baseline, while the 10^{-5} and 3×10^{-5} controls lose all post-update validation success, accompanied by KL spikes and action-format degradation. Meanwhile Muon completes the full 200 steps at 3×10^{-5} with policy entropy broadly comparable to the matched AdamW baseline (Figure 3). In these GiGPO controls, increasing AdamW's nominal learning rate does not reproduce the observed Muon gain. Muon retains task success through 200 updates at the tested nominal rate, whereas the AdamW controls lose task success. Supervised fine-tuning experiments likewise find Muon sensitive to learning rate on Adam-pretrained models (Qu, Huang, and Horvath 2026); here, Muon's spectrally-normalized updates yield useful policy improvement at its tested rate, whereas the corresponding high-rate AdamW controls lose validation success.

Training Dynamics

Figure 3 places GiGPO and GraphGPO under the same diagnostic view. Muon induces greater policy movement under GiGPO, while the matched GraphGPO runs retain similar KL and gradient-norm scales at the stronger 10^{-5} setting. Raising the GraphGPO Muon rate increases policy movement but weakens the late-window result. These diagnostics accompany the estimator-dependent variation in useful Muon learning rate.

Optimizer (lr)	Success retained?	Final	Failure signature
AdamW (1×10^{-6})	✓	0.320	–
AdamW (3×10^{-6})	✓	0.234	Lower final success
AdamW (5×10^{-6})	✓	0.070	Lower final success
AdamW (1×10^{-5})	×	0.000	Zero from steps 5–200
AdamW (3×10^{-5})	×	0.000	Zero from steps 5–200
Muon (3×10^{-5})	✓	**0.633**	–

Table 2: Learning-rate controls under the reported seed-0 GiGPO configuration. Final-checkpoint success is reported for the designated standard runs and the additional AdamW controls. Intermediate AdamW rates underperform the 10^{-6} baseline; the two high-rate controls lose all post-update success, while Muon retains task success.

The Benefit Varies Across Advantage Estimators

We repeat the comparison under GRPO and GraphGPO on the same environment, model, and training setup (Figure 1). Under GRPO, Muon at 10^{-5} reaches 0.185 late versus 0.161 for AdamW, while normalized AUC changes from 0.078 to 0.070. At 3×10^{-5}, Muon reaches 0.268 late with normalized AUC 0.091, improving both metrics. Under GiGPO, the 10^{-5} and 3×10^{-5} Muon runs improve normalized AUC by about 0.15 and 0.13; the lower rate reaches 0.5 success at step 155, 20 steps earlier, while the higher rate has the better late-window mean (0.55 versus 0.51).

GraphGPO exposes the clearest saturation effect. Muon runs at 10^{-5} and 3×10^{-5} raise normalized AUC from

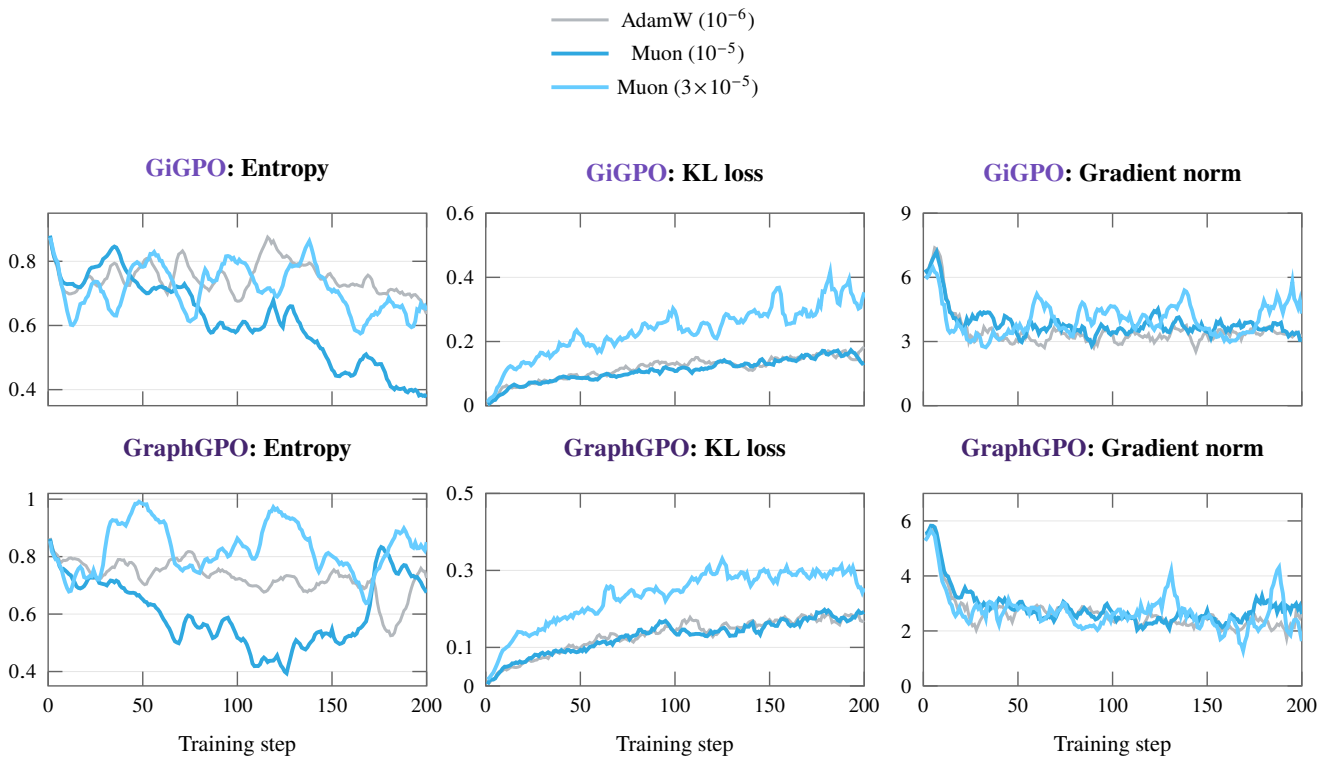

Figure 3: Matched optimizer diagnostics under GiGPO (top) and GraphGPO (bottom). Curves are 5-step trailing means over raw step-level logs; entropy is mean policy entropy over response tokens in nats. AdamW is gray, and blue shades distinguish the two Muon learning rates. Gradient norms remain on the same scale across configurations, while KL movement depends on both estimator and Muon learning rate.

AdamW's 0.40 to 0.56 and 0.47, although the corresponding late-window gains narrow to about 0.09 and 0.02. The 10^{-5} run crosses 0.5/0.75 success at steps 80/110, compared with 110/170 for AdamW. Together, the normalized AUC and threshold crossings indicate earlier learning under the stronger Muon setting. The matched seed-2 replication retains positive late-window and AUC gaps for both Muon rates (Appendix C).

Step-Level Credit Ablation

	Late-window success		Norm. AUC	
ω	AdamW	Muon	AdamW	Muon
0	0.141	**0.361**	0.063	**0.132**
1	0.290	**0.546**	0.114	**0.241**

Table 3: Matched 2×2 GiGPO step-credit ablation. AdamW uses 10^{-6} and Muon uses 3×10^{-5}.

Table 3 completes the matched optimizer-by-ω comparison. Muon improves final-window success at both $\omega = 0$ ($+0.220$) and $\omega = 1$ ($+0.255$). Enabling the step term also improves both optimizers, from 0.141 to 0.290 for AdamW and from 0.361 to 0.546 for Muon. The AUC gap grows from $+0.069$ to $+0.127$, suggesting that step-level credit and Muon may be complementary; Appendix C shows the complete trajectories.

Discussion: Reconciling Positive and Negative Evidence

Our two GRPO rates and the negative RLVR results (Fan et al. 2026; Wei 2026a) indicate that neither episode-level advantages nor optimizer identity alone determines Muon's behavior. Muon has a small late-window gain but lower AUC at 10^{-5}, while 3×10^{-5} improves both metrics in our long-horizon setting. The cited studies use single-turn tasks and different learning-rate regimes. The matched GiGPO factorial shows that Muon improves both episode-level and step-level settings, while the step term improves both optimizers. The complete GraphGPO runs show a complementary sensitivity: reducing Muon's rate from 3×10^{-5} to 10^{-5} raises normalized AUC from 0.47 to 0.56, against AdamW's 0.40, while late-window means rise from 0.83 to 0.90, against AdamW's 0.81. These results motivate two complementary routes: optimizer-side spectral modifications, such as Hopper's combined variance-normalized, single-step NS recipe (Wei 2026b) or Pion's high-pass filtering (Fan et al. 2026), and advantage estimators with additional step-level structure. Testing the SNR hypothesis requires measuring the effective

rank of policy-gradient or update matrices across estimators.

The Qwen2.5 report does not disclose the pretraining optimizer. Our controlled comparisons vary the policy optimizer within each advantage-estimator implementation.

Limitations

This is an exploratory study on one 0.5B model and ALFWorld. Most comparisons, including GRPO, GiGPO, and the matched ω ablation, use one configured seed; GraphGPO adds only one matched second seed. The values therefore do not characterize population-level uncertainty. Learning-rate coverage is also uneven: GiGPO has an AdamW sweep, whereas GRPO and GraphGPO include only two Muon rates and incomplete AdamW sweeps. The mechanism evidence is correlational and does not directly measure update spectra or gradient SNR. Finally, the full-matrix `NO_SHARD` implementation increases per-device memory and will require distributed Muon variants for larger models.

Conclusion

Across the matched runs, Muon's observed RL behavior varies with the optimizer–estimator pair. With GiGPO's anchor-state step advantages, hidden-matrix Muon raises final-window success by 88% over AdamW and retains task success through 200 updates at a nominal rate for which the tested GiGPO AdamW controls lose task success. Under GRPO, Muon at 10^{-5} has a small late-window gain but lower AUC, while 3×10^{-5} improves both metrics. The matched GiGPO ablation shows that Muon's gain persists at $\omega = 0$, while step-level credit improves both optimizers. Under GraphGPO, late-window differences narrow near saturation, while Muon at 10^{-5} raises normalized AUC by about 0.16 and reaches fixed thresholds earlier. GiGPO at 10^{-5} also learns earlier, though with a slightly lower late-window mean. Together, these exploratory results motivate studying advantage estimation jointly with the policy optimizer. A matched second GraphGPO seed preserves the Muon advantage, but broader seeds, models, and environments are needed to determine how consistently the pattern recurs.

Acknowledgments

We thank Chenyu Zheng for helpful discussions.

References

Ahmadian, A.; Cremer, C.; Gallé, M.; Fadaee, M.; Kreutzer, J.; Pietquin, O.; Üstün, A.; and Hooker, S. 2024. Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Chen, L.; Li, J.; and Liu, Q. 2025. Muon Optimizes Under Spectral Norm Constraints. *arXiv preprint arXiv:2506.15054*.

Cheng, X.; He, S.; Feng, L.; Xu, H.; Yan, M.; Feng, L.; and An, B. 2026. Beyond Trajectory-Level Attribution: Graph-Based Credit Assignment for Agentic Reinforcement Learning. *arXiv preprint arXiv:2605.26684*. Accepted at ICML 2026.

Fan, C.; Liu, G.; Hong, M.; Kompella, R. R.; and Liu, S. 2026. Rethinking Muon Beyond Pretraining: Spectral Failures and High-Pass Remedies for VLA and RLVR. *arXiv preprint arXiv:2605.19282*.

Feng, L.; Xue, Z.; Liu, T.; and An, B. 2025. Group-in-Group Policy Optimization for LLM Agent Training. In *Advances in Neural Information Processing Systems*. ArXiv:2505.10978.

Gupta, A.; Celente, R.; Shivanna, A.; Braithwaite, D. T.; Dexter, G.; Tang, S.; Udagawa, H.; Silva, D.; Ramanath, R.; and Keerthi, S. S. 2025. Effective Quantization of Muon Optimizer States. *arXiv preprint arXiv:2509.23106*.

Jordan, K.; Jin, Y.; Boza, V.; You, J.; Cesista, F.; Newhouse, L.; and Bernstein, J. 2024. Muon: An Optimizer for Hidden Layers in Neural Networks. https://kellerjordan.github.io/posts/muon/.

Kimi Team. 2025. Kimi K2: Open Agentic Intelligence. *arXiv preprint arXiv:2507.20534*.

Kool, W.; van Hoof, H.; and Welling, M. 2019. Buy 4 REINFORCE Samples, Get a Baseline for Free! In *ICLR Workshop on Deep RL Meets Structured Prediction*.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J. E.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Liu, J.; Su, J.; Yao, X.; Jiang, Z.; Lai, G.; Du, Y.; et al. 2025. Muon is Scalable for LLM Training. *arXiv preprint arXiv:2502.16982*.

Liu, Y.; Wang, J.; and Zhang, T. 2026. Optimizer-Model Consistency: Full Finetuning with the Same Optimizer as Pretraining Forgets Less. *arXiv preprint arXiv:2605.06654*.

NVIDIA. 2026. Muon Optimizer. https://docs.nvidia.com/nemo/rl/latest/guides/muon-optimizer.html.

Prime Intellect Team; Senghaas, M.; Obeid, F.; et al. 2025. INTELLECT-3: Technical Report. *arXiv preprint arXiv:2512.16144*.

Qu, X.; Huang, P.; and Horvath, S. 2026. Can Muon Fine-tune Adam-Pretrained Models? *arXiv preprint arXiv:2605.10468*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*.

Sheng, G.; Zhang, C.; Ye, Z.; Wu, X.; Zhang, W.; Zhang, R.; Peng, Y.; Lin, H.; and Wu, C. 2024. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv:2409.19256*.

Shridhar, M.; Yuan, X.; Côté, M.-A.; Bisk, Y.; Trischler, A.; and Hausknecht, M. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*.

Su, J. 2025. A Guide to the Muon Optimizer: Quick Start and Key Details. Scientific Spaces, https://www.kexue.fm/archives/11416.

Wei, J. 2026a. Field Notes: Why Muon "Hollows Out" in RL (and What We Plan To DO Next). https://huggingface.co/blog/bird-of-paradise/training-rl-with-muon-2.

Wei, J. 2026b. Hopper: The Optimizer That Learns Parallelism 2x Faster Than Adam. https://huggingface.co/blog/bird-of-paradise/training-rl-with-muon-3.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; et al. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Fan, T.; Liu, G.; Liu, L.; et al. 2025. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint arXiv:2503.14476*.

A. Experimental Hyperparameters

Parameter	Value
Shared setup	
Model / environment	Qwen2.5-0.5B-Instruct / ALFWorld
Training horizon / evaluation cadence	200 updates / every 5 updates
Max environment steps / history length	50 / 2
Group size / train batch / validation batch	8 / 16 / 128
Max prompt / response length	2048 / 512
PPO mini-batch / epochs	128 / 1
Rollout / validation temperature	1.0 / 0.4 (sampling enabled)
Hardware	8× NVIDIA H20
Optimizer and loss	
Policy optimizer	Muon on hidden 2D matrices; AdamW fallback otherwise
AdamW baseline / fallback lr	1×10^{-6}
Muon momentum / Nesterov / NS steps	0.95 / yes / 5
Weight decay	0.01
KL loss coefficient / type / KL in reward	0.01 / low-var KL / no
Invalid-action penalty coefficient	0.01
Estimator-specific settings	
GRPO	Muon lr $1 \times 10^{-5},\ 3 \times 10^{-5}$
GiGPO	return discount $\gamma = 0.95$; mean-std norm; step weight $\omega = 1$ (main), 0 (ablation); Muon lr $1 \times 10^{-5},\ 3 \times 10^{-5}$
GraphGPO	distance decay $\gamma = 0.10$; mean-std norm; step / episode weights $1/1$; Muon lr $1 \times 10^{-5},\ 3 \times 10^{-5}$
GraphGPO distance normalization / similarity	disabled / disabled

Table 4: Core training configuration. Low-level execution settings are reported separately in Appendix B.

B. Implementation Notes for Muon

Execution. Experiments use PyTorch 2.11 with CUDA 13, veRL 0.3, and vLLM 0.22.

Sharding. Newton–Schulz orthogonalization operates on full 2D matrices. Our implementation forces FSDP `NO_SHARD` on the policy so that Muon sees unsharded parameters; this is mathematically the reference Muon update but increases per-device memory. Scaling to larger models therefore requires a distributed Muon implementation.

C. Additional Per-Algorithm Diagnostics

C.1 GiGPO Same-Step Validation Values

Table 5 preserves the same-step GiGPO values underlying the aggregate comparison in the main paper.

Step	AdamW (10^{-6})	Muon (3×10^{-5})	Step	AdamW (10^{-6})	Muon (3×10^{-5})
50	0.047	0.094	175	0.320	0.531
75	0.062	0.125	180	0.289	0.523
100	0.133	0.211	185	0.227	0.469
125	0.148	0.398	190	0.289	0.562
150	0.164	0.453	195	0.297	0.555
165	0.195	0.414	200	0.320	**0.633**
Late-window mean (steps 175–200)				0.290	**0.546**

Table 5: GiGPO same-step validation success on ALFWorld. The final checkpoint is the within-window maximum for both runs; the late-window mean remains the headline statistic.

C.2 Matched Step-Credit Factorial

Figure 4 shows the complete matched trajectories. Muon retains an advantage when the GiGPO step term is disabled, while enabling the term raises both the AdamW and Muon trajectories. The larger AUC separation at $\omega = 1$ motivates testing the interaction across additional seeds.

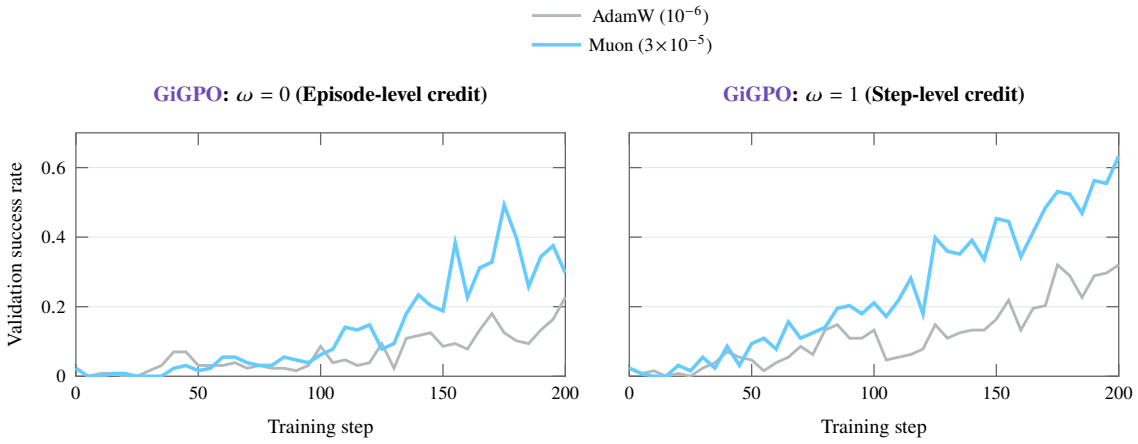

Figure 4: Matched GiGPO optimizer-by-step-credit ablation. The two panels share axes and differ only in the step weight. Curves show raw validation checkpoints every five training steps.

C.3 GRPO Diagnostics

The stronger GRPO comparison reported in the main paper is supported by the 3×10^{-5} task-level trajectories in Figure 5. Muon has the largest final-window gains on Clean, Pick Two, and Pick. Heat and Cool improve more modestly, while Look slightly favors AdamW. Thus, the aggregate improvement is not uniform across task categories; the figure reports the corresponding magnitudes.

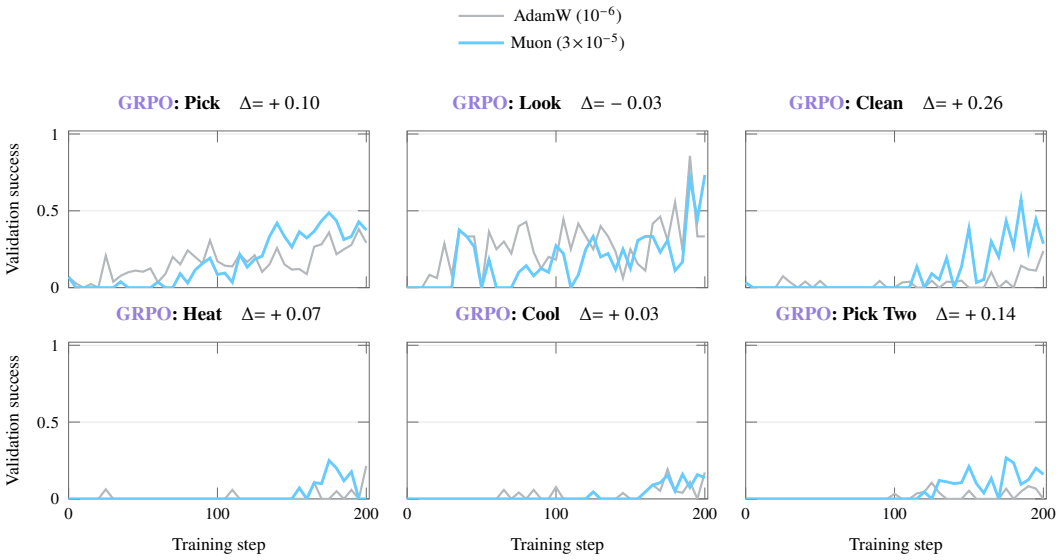

Figure 5: Per-task validation success for the complete GRPO AdamW and Muon (3×10^{-5}) runs. Curves are evaluated every five training steps. Each title reports Muon minus AdamW late-window mean success (steps 175–200).

Figure 6 shows that both GRPO runs retain nonzero validation success through step 200. Muon moves farther from the reference policy while remaining on the same gradient-norm scale.

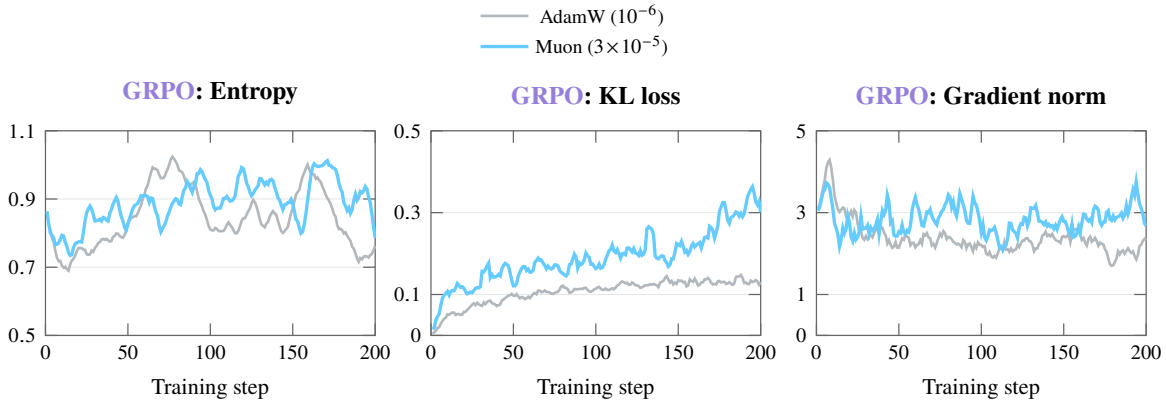

Figure 6: Training dynamics for the complete matched GRPO AdamW and Muon (3×10^{-5}) runs. Curves are 5-step trailing means over raw step-level logs. Entropy is mean policy entropy over response tokens in nats. Muon exhibits larger policy movement while both runs remain within the plotted gradient-norm range.

C.4 GiGPO and GraphGPO Task Trajectories

Figure 7 provides the complete GiGPO and GraphGPO per-task validation trajectories summarized in the main paper. Both Muon learning rates are shown for each estimator.

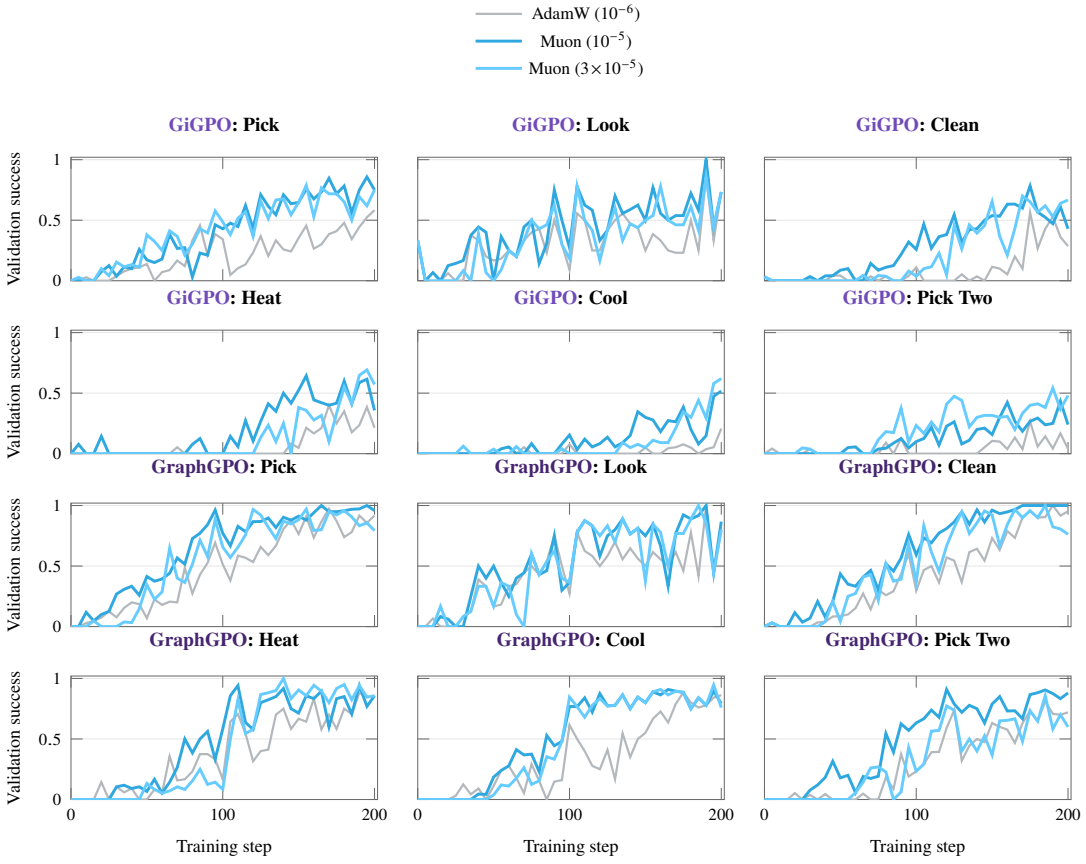

Figure 7: Complete per-task validation trajectories for GiGPO (top) and GraphGPO (bottom), using the same visual encoding and axis scales. Curves show raw checkpoints every five training steps for AdamW and both tested Muon learning rates. For GraphGPO, 10^{-5} gives the strongest aggregate final-window result and exceeds 3×10^{-5} on Pick, Look, Clean, and Pick Two.

C.5 GraphGPO Replication

Table 6 reports the complete matched seed-0 and seed-2 runs. Both Muon rates exceed AdamW in the late window and normalized AUC at each seed. Averaging the two trajectories gives a late-window gain of 0.138 for Muon 10^{-5} and 0.104 for Muon 3×10^{-5}. A separate same-seed rerun of Muon 10^{-5} reaches 0.848 in the late window, compared with 0.901 for the designated run, which indicates nontrivial run-to-run variation without changing the optimizer ordering.

Metric	Configuration	Seed 0	Seed 2	Mean
Late-window	AdamW (10^{-6})	0.810	0.620	0.715
	Muon (10^{-5})	**0.901**	0.805	**0.853**
	Muon (3×10^{-5})	0.828	**0.810**	0.819
Norm. AUC	AdamW (10^{-6})	0.399	0.250	0.324
	Muon (10^{-5})	**0.556**	**0.488**	**0.522**
	Muon (3×10^{-5})	0.474	0.473	0.474

Table 6: Matched GraphGPO replication. Late-window success is the mean over steps 175–200; normalized AUC covers steps 0–200. The final column is the descriptive mean across two configured seeds.

D. A Credit-Quality Conjecture for Muon

This appendix develops the simplified mechanism proposed in the main paper. The analysis focuses on one layer and one update, omitting clipping, KL regularization, momentum history, and singular-vector rotation to isolate the role of credit noise.

D.1 Long-Horizon Credit Noise

Let

$$Z_t = \nabla_W \log \pi_\theta(a_t \mid s_t), \qquad \widehat{G} = \sum_{t=1}^{T} \widehat{A}_t Z_t = G^\star + E.$$

Suppose an idealized trajectory has a set $\mathcal{I}$ of $K \ll T$ decisions carrying task-relevant signal. Write e_t for the matrix error contributed at step t. If these errors are zero mean, uncorrelated across time, and have equal energy $\mathbb{E}\|e_t\|_F^2 = \sigma^2$, then a trajectory-level estimator that retains all steps has

$$\mathbb{E}\|E_{\text{episode}}\|_F^2 = \mathbb{E}\Big\|\sum_{t=1}^{T} e_t\Big\|_F^2 = T\sigma^2. \tag{5}$$

An oracle local estimator that suppresses errors outside $\mathcal{I}$ instead gives

$$\mathbb{E}\|E_{\text{local}}\|_F^2 = \mathbb{E}\Big\|\sum_{t\in\mathcal{I}} e_t\Big\|_F^2 = K\sigma^2. \tag{6}$$

Holding $\|G^\star\|_F^2$ fixed yields

$$\frac{\text{SNR}_{\text{local}}}{\text{SNR}_{\text{episode}}} \approx \frac{T}{K}. \tag{7}$$

Temporal covariance adds $2\sum_{t<u}\mathbb{E}\langle e_t, e_u\rangle_F$, so the scaling can be smaller or larger. Actual GiGPO also retains its episode-level term and does not literally gate all irrelevant decisions; T/K is therefore an interpretive limit, not an empirical estimate.

D.2 Two Ways to Remove Credit Confounders

Anchor-state contrasts. For actions sampled from a repeated anchor state s, consider

$$R_j = b(s) + q(s, a_j) + \epsilon_j,$$

where $b(s)$ is shared state difficulty, $q(s, a_j)$ distinguishes the action, and ϵ_j is rollout noise. Centering within the anchor group gives the exact identity

$$R_j - \bar{R}_s = q(s, a_j) - \bar{q}_s + \epsilon_j - \bar{\epsilon}_s. \tag{8}$$

The shared difficulty $b(s)$ cancels. With n iid errors of variance σ^2, $\text{Var}(\epsilon_j - \bar{\epsilon}_s) = \sigma^2(1 - 1/n)$. This slice explains how GiGPO's normalized within-state comparison can replace a trajectory-level confounder with a more local action contrast; it does not make the estimate noise free.

Potential-like transition credit. If a state graph induces a progress potential $\Psi(s) \approx -d(s, \mathrm{goal})$, an idealized transition signal is $\delta_t^{\Psi} = \gamma\Psi(s_{t+1}) - \Psi(s_t)$. Its discounted sum telescopes:

$$\sum_{t=0}^{T-1} \gamma^t \delta_t^{\Psi} = -\Psi(s_0) + \gamma^T \Psi(s_T). \tag{9}$$

This identity shows how a delayed terminal quantity can be distributed across local transitions without accumulating arbitrary intermediate potential terms. GraphGPO is not identical to potential-based reward shaping; Equation 9 is an analogy for why distance-based transition credit can reduce long-horizon delay.

D.3 Why Directional Reliability Matters to Muon

Consider a fixed-singular-direction model

$$G^\star = U\,\mathrm{diag}(s_i)V^\top, \qquad \widehat{G} = U\,\mathrm{diag}(s_i + \epsilon_i)V^\top,$$

with $s_i > 0$ and independent $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$. The ideal polar update uses $\mathrm{sign}(s_i + \epsilon_i)$ in direction i, so

$$\Pr[s_i + \epsilon_i > 0] = \Phi(s_i/\sigma_i).$$

Taking the Frobenius inner product with $G^\star$ gives

$$\mathbb{E}\langle \mathcal{P}(\widehat{G}), G^\star \rangle_F = \sum_i s_i[2\Phi(s_i/\sigma_i) - 1]\,. \tag{10}$$

A direction with $s_i/\sigma_i \gg 1$ contributes almost s_i in expectation, whereas a direction with $s_i/\sigma_i \approx 0$ contributes almost zero. Muon equalizes singular magnitudes, so a weak direction is useful only if its sign and singular vectors are reliable.

Let Π_{tail} project onto weak signal directions. The paper's mechanism conjecture is that, in some long-horizon agentic settings, finer-grained credit approximately satisfies

$$\mathbb{E}\|\Pi_{\mathrm{tail}} E_{\mathrm{local}}\|_F^2 \le c\,\mathbb{E}\|\Pi_{\mathrm{tail}} E_{\mathrm{episode}}\|_F^2, \qquad c < 1, \tag{11}$$

without removing the corresponding signal. In the fixed-basis Gaussian model, Equation 10 is monotone in every s_i/σ_i, implying better expected alignment of the local-credit polar update with $G^\star$. Extending that implication to rotating singular vectors and optimizer momentum is an open problem.

D.4 Testable Predictions and Scope

The conjecture predicts that finer-grained credit should improve weak-tail directional SNR or sign agreement before it changes saturated final success. This motivates the normalized-AUC and time-to-threshold analysis in the main paper. In GiGPO, it predicts a benefit from activating ω under both optimizers and, if credit quality is especially important for Muon, a larger effect under Muon. A paired optimizer-by-ω comparison is needed to distinguish those two possibilities. GraphGPO may instead show an early learning gain despite limited late-window headroom.

The conjecture can be tested by logging per-layer update spectra, effective rank, tail-projected signal-to-noise estimates, and cross-minibatch sign agreement under matched AdamW and Muon runs. Its scope follows directly from the assumptions: correlated errors can alter the T/K scaling, scalar credit quality need not transfer to matrix-tail directions, and momentum or Newton–Schulz approximation can depart from the ideal polar model.

E. GiGPO Learning-Rate Control Diagnostics

E.1 Learning-Rate Controls and Run Sensitivity

Figure 8 keeps the designated standard runs from the main-paper comparison and adds AdamW controls at 3×10^{-6} and 5×10^{-6} under the same reported GiGPO hyperparameters. Their lower final success shows that increasing AdamW's rate does not reproduce the designated Muon result. Table 7 records one additional complete matched run group. The groups share the same configured seed and are therefore shown separately as a run-sensitivity diagnostic. Muon 3×10^{-5} improves over AdamW in both groups, while the 10^{-5} comparison changes direction.

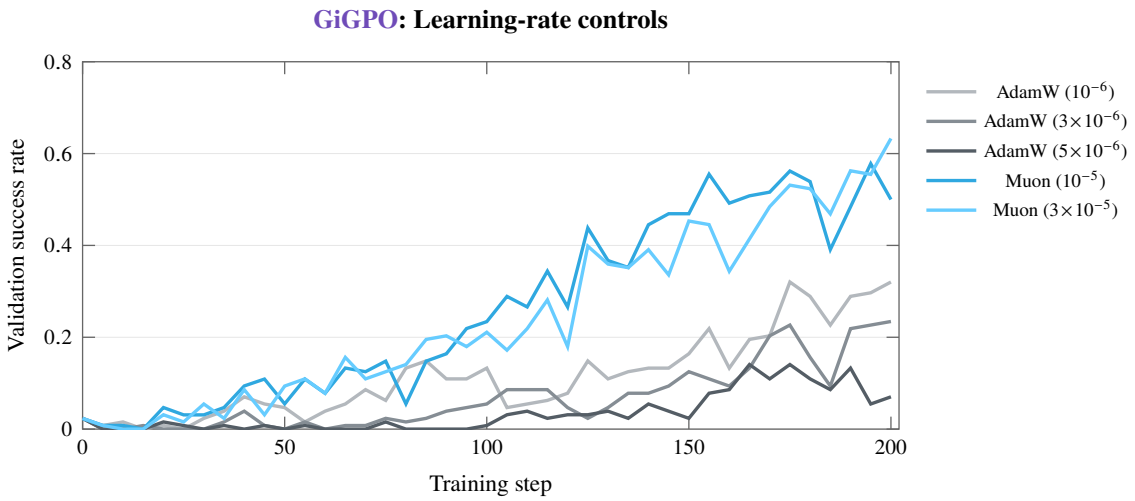

Figure 8: GiGPO learning-rate controls using the designated standard runs and intermediate-rate AdamW controls under the same reported hyperparameters. Curves show raw validation checkpoints every five updates; AdamW is gray and Muon is blue.

Run group	AdamW (10^{-6})	Muon (10^{-5})	Muon (3×10^{-5})
Designated	0.290	0.509	**0.546**
Additional matched group	0.448	0.372	**0.603**

Table 7: GiGPO late-window success for two complete run groups. Each entry is one run's mean over steps 175–200. Both groups use the same configured seed and are reported separately.

E.2 High-Rate Failure Diagnostics

Figure 9 distinguishes unsuccessful policy behavior from process termination. Both high-rate GiGPO AdamW controls complete all 200 updates but record zero validation success after step 0. At 3×10^{-5}, responses become fully clipped and valid actions vanish; at 10^{-5}, validity declines with length saturation and late KL excursions. Muon at 3×10^{-5} retains valid behavior and improves success.

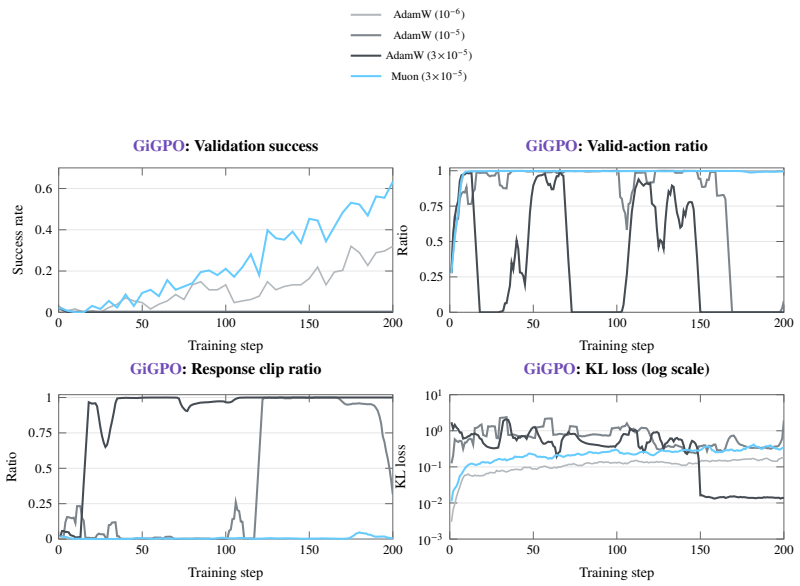

Figure 9: GiGPO learning-rate stress diagnostics. Validation curves show raw five-step checkpoints; the other panels show 5-step trailing means. AdamW validation remains zero after step 0; KL is log-scaled.