UNDERSTANDING EVOLUTION STRATEGIES FOR LLM REASONING: BROADER REASONING COVERAGE THAN GRPO

Anonymous authors
Paper under double-blind review

ABSTRACT

Evolution Strategies (ES) have recently emerged as a memory-efficient post-training paradigm for LLM reasoning. However, the optimization behavior of ES remains understudied, making it hard to define its advantage scope compared to mainstream post-training paradigms (e.g., Group Relative Policy Optimization (GRPO)). This paper systematically investigates the behavior and design of ES post-training. **First,** we **identify a performance advantage of ES over GRPO**: broader reasoning coverage that better exploits the reasoning capabilities of pretrained LLMs. Empirically, ES generally improves Pass@1 over the base model and achieves higher Pass@K than GRPO, which can exhibit entropy collapse. Theoretically, we derive sufficient conditions under which an ES update improves the center policy's Pass@K. We further propose a sequential ES–GRPO training method in both orders, which can combine GRPO's Pass@1 strength with ES's Pass@K gains. **Second,** we find that despite substantial parameter drift, removing most small coordinate updates of ES largely preserves target-task Pass@1 performance. Held-out evaluations further show that this large drift need not imply broad forgetting. **Finally,** we study how ES design choices affect its effectiveness, finding that comparable training rewards can be maintained with smaller ES populations in larger LLMs. These findings position ES as a distinct post-training paradigm for LLM reasoning rather than a less effective, memory-efficient alternative to GRPO. Our code is available at https://anonymous.4open.science/r/understanding-es-3D73.

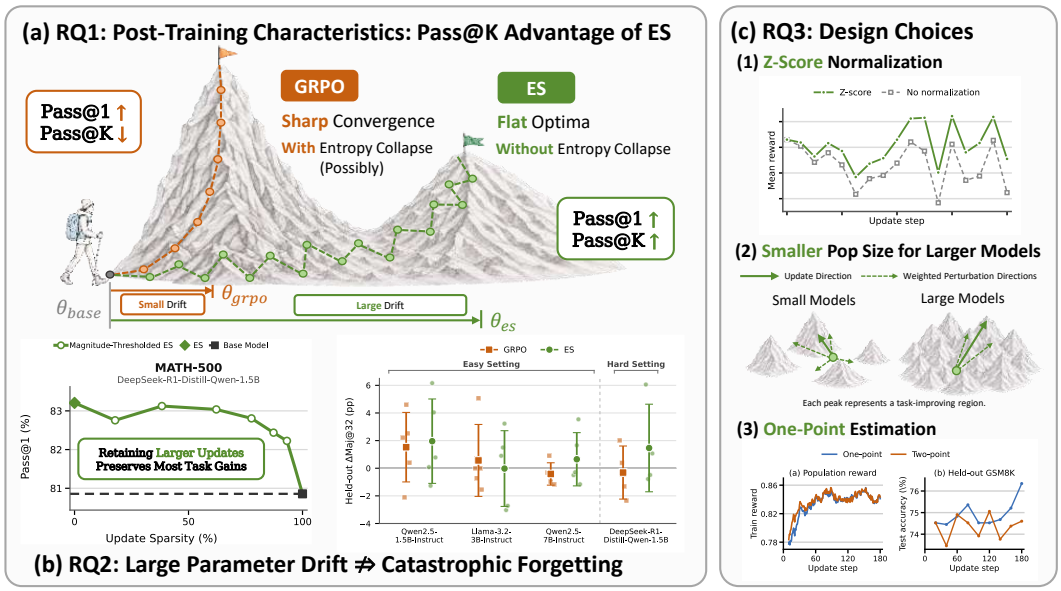


Figure 1: Overview of the three research questions about ES: (a) reasoning coverage, (b) parameter drift and capability preservation, and (c) design choices.

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

1 INTRODUCTION

Recent work has shown that Evolution Strategies (ES) (Salimans et al., 2017) can fine-tune large language models (LLMs) to improve their reasoning capabilities (Qiu et al., 2026; Sarkar et al., 2026; Zheng et al., 2026). ES optimizes a model by perturbing parameters, evaluating the perturbed models through parallelizable forward passes, and aggregating reward-weighted perturbations to estimate an update direction. By avoiding backpropagation, ES enables memory-efficient LLM post-training. Despite this efficiency advantage, the practical viability of ES remains unclear: its post-training behavior, the potential for catastrophic forgetting (Hoy et al., 2026; Abdi et al., 2026), and the conditions for effective and scalable optimization remain poorly understood.

We compare ES with Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a mainstream RL method that likewise optimizes rewards from responses. At each update, ES searches over a population of parameter-perturbed policies, whereas GRPO explores multiple responses sampled from the same policy. While GRPO improves Pass@1, it can exhibit entropy collapse and reduced exploration (Cui et al., 2025; Petrenko et al., 2026; Jin et al., 2026). GRPO may also lower large-K Pass@K relative to the base model (Yue et al., 2025). One possible explanation is that GRPO can concentrate probability mass on solutions that already have high probability under the base model (Zhao et al., 2025; Petrenko et al., 2026), reducing access to other correct but less probable solutions (Wu et al., 2025). We therefore ask whether ES similarly narrows reasoning coverage or avoids this problem through parameter-space exploration.

Beyond reasoning coverage, another important criterion for post-training is whether the model preserves its pretrained capabilities. Compared with supervised fine-tuning (SFT), RL-based post-training methods such as GRPO have been shown to better preserve pretrained capabilities and to be less prone to catastrophic forgetting (Jin et al., 2025; Yuan et al., 2025). Capability preservation therefore provides a key reference for evaluating ES as an alternative to RL post-training.

For ES, however, whether such capability preservation holds remains unclear. Prior work observes substantial parameter drift alongside catastrophic forgetting under ES and suggests that large, dense updates may contribute to forgetting (Abdi et al., 2026). Yet whether parameter drift causes forgetting remains unverified, as existing evidence is limited to specific tasks and small training sets. Beyond capability preservation, deploying ES effectively also requires understanding how ES design choices affect optimization stability and scalability. Together with the distinct exploration behavior discussed above, these open questions motivate our systematic study of ES. We summarize three research questions and main findings in Figure 1 and below:

1. **RQ1: Does ES exhibit the same post-training characteristics as GRPO?** We find that ES exhibits distinct post-training characteristics, with *generally broader reasoning coverage than GRPO*. Across models post-trained on GSM8K and DeepScaleR, ES generally improves Pass@1 over the base model, achieves higher Pass@K than GRPO, and improves the base model's Maj@K. Representative training dynamics show largely stable entropy under ES, in contrast to the substantial decline under GRPO. Our theory analyzes how sampling from different policies and favoring higher-reward policies can help find a correct answer, and gives sufficient conditions under which an ES update improves the center policy's Pass@K. We further propose a new sequential ES–GRPO method in both orders that can combine GRPO's strength in Pass@1 with ES's gains in Pass@K.
2. **RQ2: Does ES necessarily cause catastrophic forgetting?** We find that *ES does not necessarily cause catastrophic forgetting*. ES induces substantially greater parameter drift than GRPO, with a disproportionate share of its displacement concentrated in a minority of larger coordinate updates. Removing most small coordinate updates largely preserves the target-task Pass@1 gains from ES training. Furthermore, despite substantial parameter drift, ES training generally improves average held-out Pass@1 and Pass@K over the base model, while broadly preserving average Maj@K.
3. **RQ3: What design choices make ES effective and scalable?** We identify effective choices for *reward normalization* and *estimator design*, and examine how *population size* affects ES training across model scales. Z-score normalization improves training rewards, while the one-point estimator can achieve higher test performance than the two-point estimator under the same evaluation budget. Our experiments also suggest that, in larger models, comparable training rewards can be maintained with smaller ES populations, although overly small populations still limit progress.

2 PRELIMINARIES

Let θ denote the model parameters and $\mathcal{D}_{\text{train}}$ the distribution of reasoning prompts. Given $x \sim \mathcal{D}_{\text{train}}$, an LLM generates a response $y = (c, a) \sim \pi_\theta(\cdot \mid x)$, consisting of a chain-of-thought (CoT) trajectory c followed by a final answer a. A verifier assigns a scalar reward $r(x, y)$, yielding

$$F(\theta) = \mathbb{E}_{x\sim\mathcal{D}_{\text{train}}}\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(x, y)]. \tag{1}$$

Representative post-training methods for LLM reasoning include SFT, on-policy distillation (OPD), GRPO, and ES. SFT uses demonstrations, including expert CoT trajectories (Chu et al., 2025; Zheng & Lee, 2025), while teacher-guided OPD learns from teacher token distributions (Song & Zheng, 2026). They introduce supervision beyond scalar rewards and are excluded from our main comparison. We therefore focus on GRPO and ES, which optimize these rewards with different updates.

2.1 GRPO FINE-TUNING

GRPO (Shao et al., 2024) samples G responses from $\pi_{\theta_{\text{old}}}$ for each prompt x. With rewards $r_i = r(x, y_i)$, group mean $\bar{r}_G$, and population standard deviation $s_G > 0$, its advantages are

$$\widehat{A}_i = \frac{r_i - \bar{r}_G}{s_G}. \tag{2}$$

The same advantage is applied to all tokens in a response through a PPO-style clipped objective (Schulman et al., 2017), without a learned critic.

Let $T_i = |y_i|$ and define the token-level probability ratio

$$\varrho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}.$$

For a clipping parameter $\varepsilon_{\text{clip}} > 0$, the per-prompt GRPO loss to be minimized is

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\Big(\varrho_{i,t}\widehat{A}_i,\ \text{clip}(\varrho_{i,t}, 1-\varepsilon_{\text{clip}}, 1+\varepsilon_{\text{clip}})\widehat{A}_i\Big) + \beta\mathcal{L}_{\text{KL}}(\theta). \tag{3}$$

Here $\mathcal{L}_{\text{KL}}$ denotes the token-averaged KL regularization term relative to a fixed reference policy, and β is its coefficient.

2.2 ES OPTIMIZATION

ES optimizes a Gaussian-smoothed reward objective (Salimans et al., 2017):

$$F_\sigma(\theta) = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}[F(\theta + \sigma\epsilon)].$$

For scale $\sigma > 0$, ES samples N independent perturbations $\epsilon_i \sim \mathcal{N}(0, I)$ and obtains rollout rewards R_i. Assuming $\mathbb{E}[R_i \mid \epsilon_i] = F(\theta + \sigma\epsilon_i)$, the raw one-point estimator satisfies

$$\widehat{g}^{\text{raw}}_{\text{ES}} = \frac{1}{N\sigma}\sum_{i=1}^{N} R_i\epsilon_i, \qquad \mathbb{E}[\widehat{g}^{\text{raw}}_{\text{ES}}] = \nabla F_\sigma(\theta). \tag{4}$$

In practice, following Qiu et al. (2026), we standardize rewards within each population using z-scores to make the update less sensitive to reward scaling. Let $\bar{R}$ and s_R denote the mean and population standard deviation of $\{R_i\}_{i=1}^N$, and set $z_i = (R_i - \bar{R})/(s_R + \varepsilon_{\text{num}})$, where $\varepsilon_{\text{num}} > 0$ ensures numerical stability. The resulting search direction and center-model update are

$$\widehat{d}_{\text{ES}} = \frac{1}{N}\sum_{i=1}^{N} z_i\epsilon_i, \qquad \theta^+ = \theta + \alpha\widehat{d}_{\text{ES}}, \tag{5}$$

where $\alpha > 0$ is the update scale and absorbs the fixed $1/\sigma$ factor.

Table 1 contrasts GRPO's single-policy token gradients with ES's parameter perturbations.

Table 1: Comparison of ES and GRPO optimization procedures.

Aspect	ES	GRPO
Optimization objective	Gaussian-smoothed expected reward	PPO-style clipped surrogate objective
Rollout policies	Multiple parameter-perturbed policies	A shared rollout policy
Update construction	Reward-weighted perturbation aggregation	Advantage-weighted token-level gradients
Backpropagation	Not required	Required

3 RQ1: Does ES Exhibit the Same Post-Training Characteristics as GRPO?

3.1 Sufficient Conditions for ES to Improve Pass@K

Prior work shows that RL post-training can improve Pass@1 while reducing Pass@K at large K relative to the base model (Yue et al., 2025; Wu et al., 2025). We investigate conditions under which an ES update improves the center policy's Pass@K. Our analysis examines how sampling from perturbed policies and reward-based policy selection affect the probability of obtaining a correct response, then establishes sufficient conditions for Pass@K improvement in the updated policy.

For prompts $x \sim \mathcal{D}$ and a correctness verifier $v(x, y) \in \{0, 1\}$, let $p_\pi(x) = \Pr_{Y\sim\pi(\cdot|x)}[v(x, Y) = 1]$. Population members $\pi_i = \pi_{\theta+\sigma\epsilon_i}$, with $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, I)$, have success probabilities $p_i(x) = p_{\pi_i}(x)$.

Step 1. Sampling responses from different policies can help find a correct answer. Lemma 1 compares one independent response per perturbed policy with N independent responses from a policy with success probability $\bar{p}(x) = N^{-1}\sum_i p_i(x)$. Define success-probability Jensen–Shannon (JS) divergence as $\mathrm{JS}_N^{\mathrm{succ}}(x) = h(\bar{p}(x)) - N^{-1}\sum_i h(p_i(x))$, where h is binary entropy.

Lemma 1 (Different policies can help find a correct answer)**.** *For at least one correct response:*

$$P_N^{\mathrm{pop}}(x) := 1 - \prod_{i=1}^{N}(1 - p_i(x)) \geq 1 - (1 - \bar{p}(x))^N =: P_N^{\mathrm{same}}(x). \tag{6}$$

Equality holds if and only if all $p_i(x)$ are equal. When the policies' success probabilities are close and $\bar{p}(x)$ is bounded away from 0 and 1, Appendix B.2 gives

$$P_N^{\mathrm{pop}}(x) - P_N^{\mathrm{same}}(x) \approx N\bar{p}(x)(1 - \bar{p}(x))^{N-1}\,\mathrm{JS}_N^{\mathrm{succ}}(x). \tag{7}$$

Step 2. Favoring higher-reward policies can improve the probability of answering correctly. For analysis, let $w_i \geq 0$, $\sum_i w_i = 1$, be reward-based policy-selection probabilities. Selecting policy i with probability w_i and then sampling a response defines $\pi_w = \sum_i w_i\pi_i$ with success probability $p_w(x) = \sum_i w_i p_i(x)$. Positive correlation between w_i and $p_i(x)$ improves success over uniform selection (Lemma 3, Appendix B.3).

Step 3. ES updates can improve the center policy's Pass@K. ES uses reward-weighted perturbations to update its center policy from π_θ to π_{θ^+}. The following proposition gives sufficient conditions for this ES update to improve the center policy's Pass@K: π_w must outperform π_θ by more than the transfer-error bound between π_w and π_{θ^+}.

Proposition 1 (Sufficient conditions for center-policy Pass@K improvement)**.** *Let $B_w(x) = \mathrm{Bern}(p_w(x))$ and $B_{\theta^+}(x) = \mathrm{Bern}(p_{\theta^+}(x))$. Define expected Pass@$K$, $J_K(\pi) = \mathbb{E}_{x\sim\mathcal{D}}[1 - (1 - p_\pi(x))^K]$, and the Pass@$K$ gain of π_w over the center policy before the ES update, $m_K = J_K(\pi_w) - J_K(\pi_\theta)$. For $\varepsilon_{\mathrm{succ}} \geq 0$, assume*

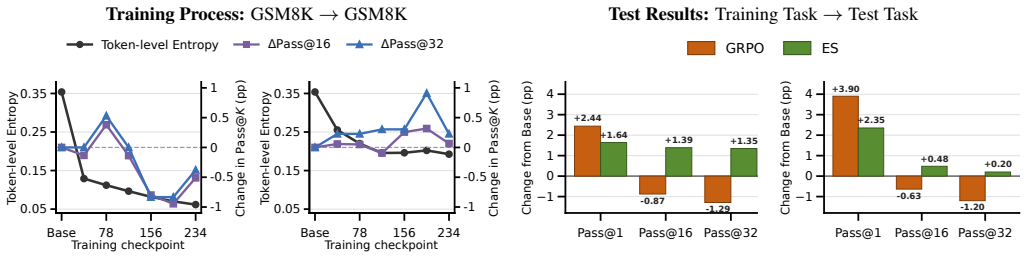
$$\mathbb{E}_{x\sim\mathcal{D}} D_{\mathrm{KL}}(B_w(x) \,\|\, B_{\theta^+}(x)) \leq \varepsilon_{\mathrm{succ}}, \tag{8}$$

$$m_K > K\sqrt{\varepsilon_{\mathrm{succ}}/2}. \tag{9}$$

Then

$$J_K(\pi_{\theta^+}) \geq J_K(\pi_w) - K\sqrt{\varepsilon_{\mathrm{succ}}/2} > J_K(\pi_\theta). \tag{10}$$

The proof is given in Appendix B.4.



(a) **GRPO** Qwen2.5-1.5B-Instruct (b) **ES** Qwen2.5-1.5B-Instruct (c) **GSM8K → Average** Llama-3.2-3B-Instruct (d) **DeepScaleR → MATH-500** DeepSeek-R1-Distill-Qwen-1.5B

Figure 2: Training dynamics (a–b) and relative test performance changes (c–d) of ES and GRPO.

Table 2: Pass@K results ($\times 100$) in the Easy Setting after two epochs of GSM8K post-training. P@K and M@K denote Pass@K and Maj@K. Underlining marks the best task scores. Bold and shading mark the best and second-best Average scores.

	GSM8K			HotpotQA			Countdown			MBPP			Average		
Method	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32
Qwen2.5-1.5B-Instruct															
Base	70.3	94.8	96.4	27.6	46.6	50.3	7.5	46.8	56.9	58.6	83.8	87.2	41.0	68.0	72.7
GRPO	77.3	94.3	96.0	25.0	44.3	48.1	8.5	49.5	58.8	60.0	82.4	86.0	**42.7**	67.7	72.2
ES	73.2	94.9	96.6	24.9	45.8	49.9	8.8	49.4	59.5	58.7	83.0	86.4	41.4	**68.3**	**73.1**
ES→GRPO	77.1	94.4	96.0	24.3	44.0	47.9	8.3	47.9	57.1	59.8	83.4	86.8	42.3	67.4	71.9
GRPO→ES	76.1	94.6	96.1	24.6	44.7	48.8	7.8	48.4	58.0	58.5	81.6	84.8	41.8	67.3	71.9
Llama-3.2-3B-Instruct															
Base	77.7	96.8	98.1	26.6	44.2	47.6	7.7	47.4	56.6	62.1	80.6	83.7	43.5	67.2	71.5
GRPO	84.7	95.6	96.4	27.5	40.4	43.0	9.6	52.9	61.6	62.1	76.7	79.8	46.0	66.4	70.2
ES	81.6	96.4	97.9	29.0	45.7	48.9	9.0	51.3	60.5	61.1	81.2	84.0	45.2	**68.6**	**72.8**
ES→GRPO	84.1	95.0	96.2	26.1	41.0	44.1	10.3	53.5	63.4	63.5	80.4	82.1	**46.0**	67.5	71.4
GRPO→ES	84.1	95.9	96.9	27.5	43.6	47.0	8.5	49.9	59.5	61.5	77.2	80.5	45.4	66.7	71.0
Qwen2.5-7B-Instruct															
Base	91.8	97.6	98.0	45.4	52.4	53.6	34.1	70.3	74.8	76.1	86.4	87.9	61.8	76.7	78.6
GRPO	92.3	97.0	97.3	46.2	52.3	53.3	36.7	68.4	72.9	74.8	85.4	87.5	**62.5**	75.8	77.8
ES	91.0	97.6	98.1	45.1	52.1	53.4	33.0	71.9	76.7	75.1	85.7	87.5	61.1	**76.8**	**78.9**
ES→GRPO	92.1	97.6	98.0	46.0	52.1	53.1	36.0	70.7	75.5	75.1	85.1	86.0	62.3	76.4	78.1
GRPO→ES	92.3	97.4	98.0	45.9	52.4	53.6	35.0	69.4	73.8	75.1	86.0	87.9	62.1	76.3	78.3

3.2 ES Can Achieve Higher Pass@16 and Pass@32 Than GRPO

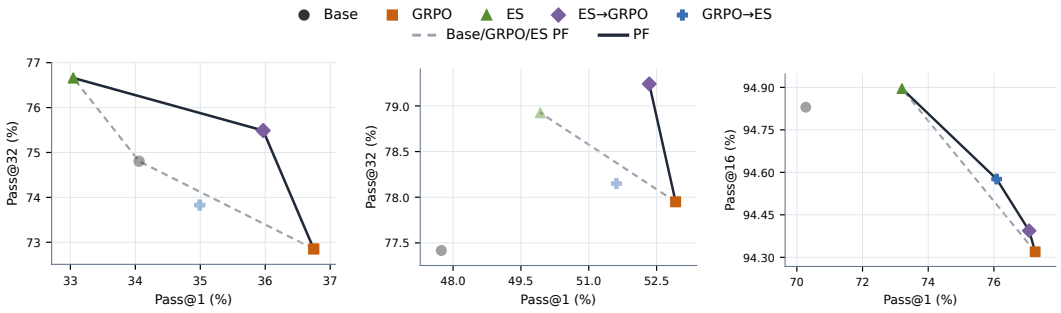
Experimental Setup. In the Easy Setting, Qwen2.5-1.5B-Instruct, Qwen2.5-7B-Instruct, and Llama-3.2-3B-Instruct are post-trained on GSM8K for two epochs (Cobbe et al., 2021). In the Hard Setting, DeepSeek-R1-Distill-Qwen-1.5B is post-trained on DeepScaleR for one epoch (Luo et al., 2025). Further experimental details are provided in Appendix A.

Training Dynamics. Figure 2(a–b) tracks GSM8K test performance and mean token-level entropy during post-training of Qwen2.5-1.5B-Instruct. Under GRPO, token-level entropy declines substantially, and Pass@16/32 finish below those of the base model. This pattern is consistent with prior reports of entropy collapse and weaker Pass@K scaling under GRPO-style optimization (Jang et al., 2026). In contrast, under ES, entropy declines less and largely stabilizes later in training. ES finishes above the base model on both Pass@16 and Pass@32.

Test Results. ES generally improves Pass@16/32 over the base model and outperforms GRPO on these metrics, while GRPO typically achieves higher Pass@1, as shown in Tables 2 and 3. Notably, GRPO's Pass@16/32 declines relative to the base model occur in 10 of 12 model–benchmark pairs in the Easy Setting and in the examples in Figure 2(c–d), where ES improves Pass@1, Pass@16, and Pass@32 over the base model. Consistent with this contrast, Appendix G shows that ES overtakes GRPO as K increases on representative tasks, despite lower Pass@1. Beyond Pass@K, Appendix F reports gains in Maj@16 and Maj@32 from ES over the base model when averaged across benchmarks in both settings, with ES also exceeding GRPO in the Hard Setting. Additional results on

Table 3: Pass@K results ($\times 100$) on mathematical reasoning benchmarks in the Hard Setting.

	AIME24			AIME25			AMC23			MATH500			Average		
Method	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32
DeepSeek-R1-Distill-Qwen-1.5B															
Base	23.4	59.9	66.7	22.6	43.8	50.0	64.1	93.6	95.0	80.9	96.7	98.0	47.7	73.5	77.4
GRPO	27.6	62.2	66.7	25.4	45.8	53.3	73.9	94.7	95.0	84.8	96.1	96.8	**52.9**	74.7	78.0
ES	22.8	60.3	70.0	23.2	46.7	50.0	70.5	96.0	97.5	83.2	97.2	98.2	49.9	75.0	78.9
ES→GRPO	29.0	70.6	76.7	23.8	43.0	50.0	72.3	92.4	92.5	84.4	97.1	97.8	52.3	75.8	**79.2**
GRPO→ES	25.8	61.8	63.3	24.5	51.0	56.7	71.6	94.9	95.0	84.5	97.0	97.6	51.6	**76.2**	78.2



(a) **Countdown** Qwen2.5-7B-Instruct

(b) **Math Average** DeepSeek-R1-Distill-Qwen-1.5B

(c) **GSM8K** Qwen2.5-1.5B-Instruct

Figure 3: Pass@1–Pass@K Pareto fronts without (dashed) and with (solid) our sequential ES–GRPO training method in both orders. Faded points are dominated.

CommonsenseQA and GPQA Diamond are reported in Appendix E. While concurrent work (Hayes et al., 2026) also reports ES's Pass@K advantage over RL on math benchmarks, we extend the evaluation to held-out code generation, multi-hop question answering, and commonsense reasoning.

3.3 Our Method: Sequential ES–GRPO Training.

To combine GRPO's Pass@1 gains with ES's large-K coverage, we propose a sequential ES–GRPO training method in both orders, with the same total training budget as ES or GRPO. Figure 3 shows that our method can add non-dominated Pass@1–Pass@K points beyond Base, ES, and GRPO. In the Hard Setting, our method achieves the highest average Pass@16/32 with different training orders, as reported in Table 3. Appendix G shows that the preferred training order varies by task.

Summary.

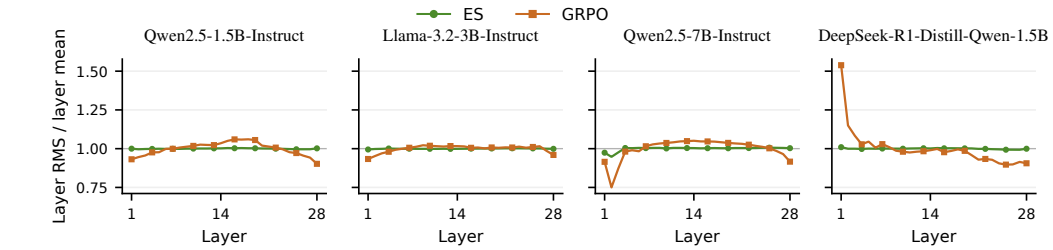
- **ES updates can improve the center policy's Pass@K.** Center-policy Pass@K improves if the reward-weighted policy's gain over the pre-update center exceeds the transfer-error bound.
- **ES can offer broader reasoning coverage than GRPO.** ES generally improves Pass@1 over Base and exceeds GRPO in Pass@16/32. ES also improves mean Maj@16/32 over Base. Representative trajectories show less entropy decline under ES than under GRPO.
- **Sequential ES–GRPO training can combine GRPO's Pass@1 gains with ES's large-K coverage.** It adds non-dominated Pass@1–Pass@K points relative to Base, ES, and GRPO.

4 RQ2: Does ES Necessarily Cause Catastrophic Forgetting?

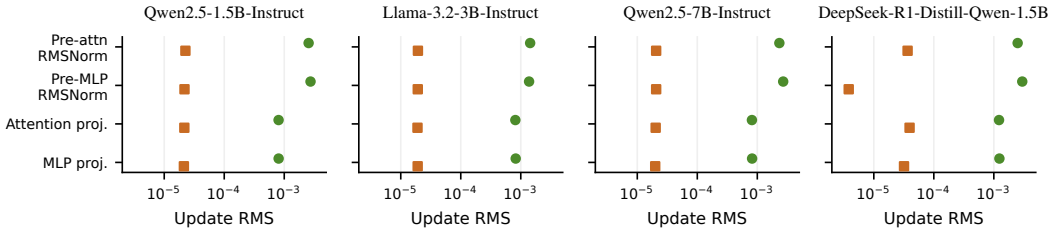
4.1 ES Induces Larger Parameter Drift with Uneven Module-Level Changes

Whole-model drift. We quantify drift by relative L_2 distance $D_{\text{rel}}(\theta, \theta_0) = \|\theta - \theta_0\|_2 / \|\theta_0\|_2$, where θ_0 and θ are pre- and post-training parameters. Table 4 shows that ES induces substantially greater parameter drift than GRPO, consistent with prior work (Hoy et al., 2026; Abdi et al., 2026).

Across layers. Figure 4a shows less variation in the root-mean-square (RMS) magnitude of parameter changes under ES than GRPO, with no common decrease with depth across the four models.



(a) Layer-level changes, normalized by each method's mean across layers.



(b) Module-level changes, measured by update RMS across layers.

Figure 4: Parameter changes across layers and modules after ES and GRPO post-training.

Across modules. Figure 4b shows larger ES update RMS in Pre-attention and Pre-MLP RMSNorm than in attention and MLP projection weights, with RMS computed over all parameters of each module type across layers. Appendix H.1 provides further details.

4.2 Removing Small Updates Largely Preserves Target-Task Pass@1

We next examine how parameter changes are distributed across individual coordinates. Let $\Delta\theta_i = \theta_i - \theta_{0,i}$ denote coordinate i's change from the base model, and define the fraction of nonzero changes with $|\Delta\theta_i| \le \tau$ as

$$s_\tau = \frac{|\{i : 0 < |\Delta\theta_i| \le \tau\}|}{|\{i : |\Delta\theta_i| > 0\}|}. \tag{11}$$

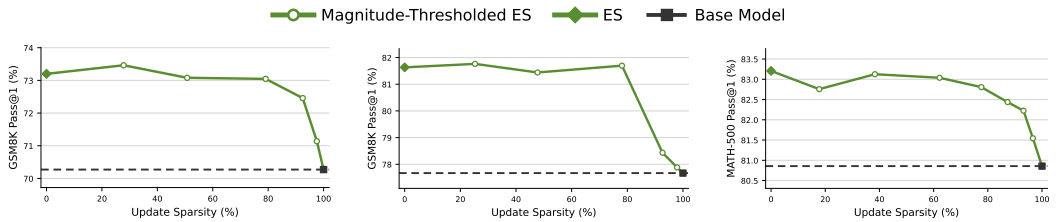
Table 4 shows that removing 77.6–93.0% of nonzero updates at $\tau = \sigma$ retains 60.5–86.3% of the original relative L_2 distance, indicating an outsized contribution from larger updates.

Turning to task performance, removing the smallest 77.6–79.1% of nonzero updates keeps target-task Pass@1 within 0.5 percentage points of that of the ES-trained model for all three models in Figure 5, as detailed in Appendix H.2. Comparisons with random coordinate truncation in Appendix H.3 further support the concentration of task-relevant information in large-magnitude updates. Additional experiments show that both the overall displacement norm and the distribution of updates across parameter coordinates are associated with target-task performance.

Finally, coordinate sparsity does not imply low rank. Specifically, appendix H.4 shows that the update matrices remain full-rank or nearly so after entries whose magnitudes do not exceed τ are set to zero, with little change in spectral concentration at the analyzed thresholds.

Table 4: Relative parameter distances ($\times 10^{-2}$). Parentheses give ES/GRPO distance ratios for Full update and percentages of nonzero updates removed for thresholded ES.

Model	GRPO	Full update	Magnitude-Thresholded ES		
			$\tau = 1.0 \times 10^{-3}$	$\tau = 1.5 \times 10^{-3}$	$\tau = 2.0 \times 10^{-3}$
Qwen2.5-1.5B-Instruct	0.0475	1.933 (40.7×)	1.615 (79.11%)	1.242 (92.47%)	0.871 (97.57%)
Llama-3.2-3B-Instruct	0.0954	4.185 (43.9×)	3.482 (78.08%)	2.589 (92.64%)	1.707 (97.89%)
Qwen2.5-7B-Instruct	0.0831	3.664 (44.1×)	3.032 (78.41%)	2.218 (93.02%)	1.427 (98.11%)
DeepSeek-R1-Distill-Qwen-1.5B	0.0548	2.338 (42.7×)	2.201 (62.18%)	2.017 (77.61%)	1.772 (87.29%)



(a) **GSM8K** Qwen2.5-1.5B-Instruct (b) **GSM8K** Llama-3.2-3B-Instruct (c) **MATH-500** DeepSeek-R1-Distill-Qwen-1.5B

Figure 5: Pass@1 performance after small-update removal. The ES-trained model corresponds to 0% update sparsity. The Base Model corresponds to 100% update sparsity.

Table 5: Pass@1 and Maj@K ($\times 100$) on four held-out benchmarks in the Hard Setting.

	GPQA			MBPP			CSQA			Countdown			Average		
Method	P@1	M@16	M@32	P@1	M@16	M@32	P@1	M@16	M@32	P@1	M@16	M@32	P@1	M@16	M@32
DeepSeek-R1-Distill-Qwen-1.5B															
Base	24.2	33.3	33.3	62.9	69.6	70.4	46.8	54.1	54.1	37.5	71.1	75.1	42.9	57.0	58.2
GRPO	31.8	35.9	35.4	66.6	69.6	70.8	47.5	53.7	52.7	36.6	68.5	72.8	**45.6**	56.9	57.9
ES	28.0	39.4	39.4	63.5	70.8	69.6	46.0	52.7	53.6	38.5	71.7	76.2	44.0	**58.6**	**59.7**
ES→GRPO	30.6	36.4	37.4	64.6	71.6	72.0	46.8	52.4	53.0	38.8	70.9	73.6	45.2	57.8	59.0
GRPO→ES	30.6	33.8	34.8	64.2	66.5	69.6	47.3	53.3	54.1	38.1	72.2	76.9	45.1	56.5	58.9

4.3 Large Parameter Drift Does Not Necessarily Indicate Broad Forgetting

To assess capability retention, we evaluate symbolic (e.g., Countdown), commonsense (e.g., CommonsenseQA), and specialized reasoning (e.g., MBPP and GPQA Diamond), following Besta et al. (2025). Table 39 in Appendix I and Table 5 show that ES largely preserves or improves Maj@16/32 averaged over held-out tasks and improves average held-out Pass@1 for most models.

Held-out Pass@16/32 results provide a complementary view of ES's capability retention. For non-multiple-choice tasks, Table 40 in Appendix I shows that ES improves average Pass@16/32 over Base across all three Easy Setting models, with small decreases in the Hard Setting. Additional multiple-choice results are reported in Appendix E. Together, these results show that ES can broadly preserve held-out capabilities despite substantial parameter drift.

Summary.

- **ES induces larger parameter drift with uneven changes across modules.** ES produces substantially greater relative L_2 drift than GRPO, with larger update RMS in normalization parameters than in attention and MLP projection weights.
- **Removing most small updates largely preserves target-task Pass@1.** A minority of larger updates accounts for much of the parameter displacement and retains most task gains.
- **Large parameter drift need not imply broad forgetting.** Despite its larger drift, ES broadly preserves held-out reasoning performance.

5 RQ3: What Design Choices Make ES Effective and Scalable?

5.1 Reward Normalization, Perturbation Scale, and Population-Size Scaling

Reward Normalization. We z-score population rewards before weighting perturbation directions, so updates depend on relative reward scales. In the matched one-point ES ablation, normalization raises mean rewards at every post-initialization update, as shown in item (1) of Figure 1(c).

Perturbation Scale. The perturbation scale σ sets both the radius of parameter-space exploration and the smoothing level of the effective objective F_σ. It should be chosen to allow sufficient exploration while avoiding perturbations that destabilize training.

Table 6: Smoothed GSM8K rewards across model and population sizes. Shading marks rewards for smaller populations within 0.01 of the corresponding $N = 64$ result.

	Qwen2.5-0.5B-Instruct				Qwen2.5-1.5B-Instruct				Qwen2.5-3B-Instruct			
Step	$N = 8$	$N = 16$	$N = 32$	$N = 64$ (ref.)	$N = 8$	$N = 16$	$N = 32$	$N = 64$ (ref.)	$N = 8$	$N = 16$	$N = 32$	$N = 64$ (ref.)
100	0.4782	0.5021	0.5100	0.5085	0.8211	0.8250	0.8259	0.8244	0.8912	0.8945	0.8960	0.8957
200	0.4614	0.4995	0.5180	0.5201	0.8242	0.8314	0.8366	0.8352	0.8903	0.8970	0.8966	0.8992
300	0.4421	0.4935	0.5253	0.5287	0.8276	0.8387	0.8449	0.8438	0.8901	0.9001	0.9004	0.9031

Population-Size Scaling. Motivated by the finding that task-improving perturbations are more prevalent around larger pretrained models (Gan & Isola, 2026), we compare Qwen2.5-{0.5B, 1.5B, 3B}-Instruct models in matched GSM8K runs with $N \in \{8, 16, 32, 64\}$. Table 6 shows rewards comparable to those at $N = 64$ with $N = 32$ for all three models and $N = 16$ for the two larger models. Appendix J shows later reward declines for the smallest model at $N = 8, 16$, while the larger models fall behind their $N = 64$ counterparts at $N = 8$. These results suggest that larger models tolerate smaller populations, but excessive reductions still limit progress.

5.2 One-Point ES Generally Achieves Higher GSM8K Test Accuracy

At equal evaluation budgets, one-point ES generally achieves higher GSM8K test accuracy, while two-point ES offers no training-reward advantage, as shown in item (3) of Figure 1(c). Two-point estimation pairs opposite perturbations to potentially cancel shared evaluation fluctuations, as in supervised zeroth-order fine-tuning (Malladi et al., 2023). Appendix K shows that paired subtraction reduces raw scalar evaluation variance for fixed-batch supervised and fixed-rollout objectives. With regenerated reasoning responses, this reduction occurs only at the smallest tested perturbation scale with shared generation seeds. Thus, the potential variance-reduction benefit of two-point estimation does not guarantee an advantage over one-point ES in reasoning training.

Summary.

- **ES training benefits from reward normalization.** In the matched one-point ablation, z-score normalization yields higher mean rewards throughout training after initialization.
- **Larger models may tolerate smaller ES populations.** In our GSM8K experiments, larger models can maintain comparable training rewards with smaller ES populations, although excessively small populations can hinder training progress.
- **One-point ES generally achieves higher test accuracy.** At equal evaluation budgets, it outperforms two-point ES on GSM8K, with no training-reward advantage from two-point estimation.

6 Conclusion and Future Work

Conclusion. This work characterizes ES post-training for reasoning in three aspects: **(1) Broader reasoning coverage than GRPO.** Empirically, ES generally improves Pass@1 over Base and exceeds GRPO in Pass@K. Theoretically, we give sufficient conditions for an ES update to improve the center policy's Pass@K. Our sequential ES–GRPO training method can combine both methods' strengths and yield additional non-dominated trade-offs. **(2) Large drift need not imply broad forgetting.** A minority of larger coordinate updates accounts for much of the parameter displacement and retains most target-task Pass@1 gains. Despite greater drift than GRPO, ES training improves average held-out Pass@1 for most models while broadly preserving Maj@K. **(3) Design choices for effective ES.** Z-score normalization improves training rewards. Besides, one-point estimation can achieve higher test accuracy than two-point estimation at equal evaluation budgets. Finally, larger models can maintain comparable training rewards with smaller populations. These findings position ES as a distinct post-training paradigm for LLM reasoning rather than a less effective, memory-efficient alternative to GRPO.

Future Work. Future work could better exploit ES's incentivized-reasoning advantage by expanding access to correct paths assigned low probability by the base policy. Continual learning could be further explored to clarify how parameter drift affects previously acquired capabilities over longer training horizons spanning multiple tasks, and to investigate whether approaches such as parameter-efficient updates can suppress harmful drift without constraining beneficial exploration.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing proofs and implementing research methods. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide key ingredients for their proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, translate text, clean and reformat datasets, or interpret experimental results. Qualitative and thematic data analysis is not applicable to this work. Additionally, we used generative AI tools to search for relevant literature, verify references, improve the readability of the manuscript, and format references.

The authors reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, code, and results produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide an anonymous source-code release at https://anonymous.4open.science/r/understanding-es-3D73. Appendix A details the implementation, compute-budget matching, training and evaluation configurations, and evaluation metrics. Appendix A.6 provides the prompt templates, while Appendix A.2 documents dataset sources and licensing information.

REFERENCES

Immanuel Abdi, Akshat Gupta, Micah Mok, Alex Lu, Nicholas Lee, and Gopala Anumanchipalli. Evolutionary strategies at scale lead to catastrophic forgetting. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 194–204, 2026. doi: 10.18653/v1/2026.acl-short.18.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. doi: 10.48550/arXiv.2108.07732.

Maciej Besta, Florim Memedi, Zhenyu Zhang, Robert Gerstenberger, Guangyuan Piao, Nils Blach, Piotr Nyczyk, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Lukas Gianinazzi, Ales Kubicek, Hubert Niewiadomski, Aidan O'Mahony, Onur Mutlu, and Torsten Hoefler. Demystifying chains, trees, and graphs of thoughts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):10967–10989, 2025. doi: 10.1109/TPAMI.2025.3598182.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, second edition, 2006. doi: 10.1002/047174882X.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025. doi: 10.48550/arXiv.2505.22617.

Yulu Gan and Phillip Isola. Neural thickets: Diverse task experts are dense around pretrained weights. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Tanmay Gautam, Youngsuk Park, Hao Zhou, Parameswaran Raman, and Wooseok Ha. Variance-reduced zeroth-order methods for fine-tuning language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15180–15208. PMLR, 2024.

Conor F. Hayes, Elliot Meyerson, Kajetan Schweighofer, Roberto Dailey, Babak Hodjat, Risto Miikkulainen, and Xin Qiu. Beyond the best guess: Improving LLM solution coverage with evolution strategies, 2026.

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, Ning Ding, and Zhiyuan Liu. JustRL: Scaling a 1.5B LLM with a simple RL recipe. *arXiv preprint arXiv:2512.16649*, 2025. doi: 10.48550/arXiv.2512.16649.

William Hoy, Binxu Wang, and Xu Pan. Matching accuracy, different geometry: Evolution strategies vs GRPO in LLM post-training. *arXiv preprint arXiv:2604.01499*, 2026. doi: 10.48550/arXiv.2604.01499.

Jaeeun Jang, Hansle Lee, and Sangmin Kim. A few bad apples spoil the bunch: Preventing global entropy collapse driven by a small set of tokens in LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13134–13154. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.641.

Hangzhan Jin, Sitao Luan, Tianwei Ni, Sicheng Lyu, Guillaume Rabusseau, Reihaneh Rabbany, Doina Precup, and Mohammad Hamdaqa. RL fine-tuning heals OOD forgetting in SFT. *arXiv preprint arXiv:2509.12235*, 2025.

Renren Jin, Pengzhi Gao, Yuqi Ren, Zhuowen Han, Tongxuan Zhang, Wuwei Huang, Wei Liu, Jian Luan, and Deyi Xiong. Revisiting entropy in reinforcement learning for large reasoning models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25300–25322, 2026. doi: 10.18653/v1/2026.findings-acl.1266.

Daria Korotyshova, Boris Shaposhnikov, Alexey Malakhov, Alexey Khokhulin, Nikita Surnachev, Kirill Ovcharenko, George Bredis, Alexey Gorbatovski, Viacheslav Sinii, and Daniil Gavrilov. ESSA: Evolutionary strategies for scalable alignment. *arXiv preprint arXiv:2507.04453*, 2025.

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. doi: 10.48550/arXiv.2305.20050.

Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-Preview with a 1.5B model by scaling RL. Notion Blog, 2025.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53038–53075, 2023. doi: 10.52202/075280-2308.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017. doi: 10.1007/s10208-015-9296-2.

Sangjun Park, Elliot Meyerson, Xin Qiu, and Risto Miikkulainen. Fine-tuning language models to know what they know, 2026.

Aleksei Petrenko, Ben Lipkin, Kevin Chen, Erik Wijmans, Marco Cusumano-Towner, Raja Giryes, and Philipp Krähenbühl. Entropy-preserving reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xin Qiu, Yulu Gan, Conor F. Hayes, Qiyao Liang, Yinggan Xu, Roberto Dailey, Elliot Meyerson, Babak Hodjat, and Risto Miikkulainen. Evolution strategies at scale: LLM fine-tuning beyond reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023. doi: 10.48550/arXiv.2311.12022.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017. doi: 10.48550/arXiv.1703.03864.

Bidipta Sarkar, Mattie Fellows, Juan Agustin Duque, Alistair Letcher, Antonio León Villares, Anya Sims, Clarisse Wibault, Dmitry Samsonov, Dylan Cope, Jarek Liesen, Kang Li, Lukas Seier, Theo Wolf, Uljad Berdica, Valentin Mohl, Alexander David Goldie, Aaron Courville, Karin Sevegnani, Shimon Whiteson, and Jakob Nicolaus Foerster. Evolution strategies at the hyperscale. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/arXiv.1707.06347.

Kajetan Schweighofer, Conor F. Hayes, Roberto Dailey, Risto Miikkulainen, and Xin Qiu. Overcoming forgetting in LLM fine-tuning with evolution strategies. *arXiv preprint arXiv:2605.30148*, 2026. doi: 10.48550/arXiv.2605.30148.

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. doi: 10.48550/arXiv.2402.03300.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Zhishen Sun, Sizhe Dang, Guang Dai, and Haishan Ye. ESSAM: A novel competitive evolution strategies approach to reinforcement learning for memory efficient LLMs fine-tuning. *arXiv preprint arXiv:2602.01003*, 2026.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1421.

Fang Wu, Weihao Xuan, Ximing Lu, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why RLVR may not escape its origin. In *The 2nd AI for MATH Workshop at the 42nd International Conference on Machine Learning*, 2025.

Huimin Xu, Shuai Zhao, Xiaobao Wu, and Anh Tuan Luu. Understanding and preventing entropy collapse in RLVR with on-policy entropy flow optimization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17759–17771. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.879.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259.

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

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, volume 38, pp. 57654–57689, 2025.

Yifan Zhang and Math-AI Team. American invitational mathematics examination (AIME) 2024. Hugging Face dataset, 2024.

Yifan Zhang and Math-AI Team. American invitational mathematics examination (AIME) 2025. Hugging Face dataset, 2025.

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D. Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 59173–59190, 2024.

Rosie Zhao, Alexandru Meterez, Sham M. Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: RL post-training amplifies behaviors learned in pretraining. In *Proceedings of the 2nd Conference on Language Modeling*, 2025.

Zhi Zheng and Wee Sun Lee. Reasoning-CV: Fine-tuning powerful reasoning LLMs for knowledge-assisted claim verification. *arXiv preprint arXiv:2505.12348*, 2025.

Zhi Zheng, Rongsheng Chen, Yunpeng Ba, Zhenkun Wang, Yee Whye Teh, and Wee Sun Lee. Agentic ESOpt: Fine-tuning long-horizon LLM agents with minimal GPU requirements. *arXiv preprint arXiv:2608.17310*, 2026.

APPENDIX CONTENTS

A Experimental Details **15**
- A.1 Implementation-Level Training Procedure . . . 15
- A.2 Datasets . . . 15
- A.3 ES–GRPO FLOP Matching . . . 15
- A.4 Training and Evaluation Configurations . . . 16
- A.5 Evaluation Metrics . . . 17
- A.6 Prompt Templates . . . 17

B Proofs for the RQ1 Diversity–Coverage Analysis **19**
- B.1 Proof of Lemma 2 . . . 19
- B.2 Proof of Lemma 1 . . . 20
- B.3 Proof and Illustration of Lemma 3 . . . 20
- B.4 Proof of Proposition 1 . . . 21

C Results Across Training Seeds **22**

D Sensitivity to Test-Time Temperature **23**

E Complete Multiple-Choice Results **24**

F Complete Majority-Vote Results **25**

G Pass@K Profiles of Individual Post-Training Methods and Sequential ES–GRPO Training **26**

H Additional RQ2 Results and Parameter Analyses **27**
- H.1 Parameter Changes Across Layers and Modules . . . 27
- H.2 Task Performance After Magnitude Thresholding . . . 39
- H.3 Random Coordinate Truncation and Displacement Norm Matching . . . 40
- H.4 Spectral Structure After Magnitude Thresholding . . . 42

I Detailed Held-Out Performance Changes **46**

J Population-Size Scaling Trajectories **47**

K ES–ZO Estimator Equivalence and Its Finite-Sample Boundary **48**
- K.1 A Unified Smoothed-Objective View . . . 48
- K.2 Covariance Required by Paired Subtraction . . . 48
- K.3 Local Perturbation Diagnostic . . . 49
- K.4 Diagnostic Results . . . 49

L Related Work **51**
- L.1 Evolution Strategies for LLM Reasoning Post-Training . . . 51
- L.2 Reinforcement Learning with Verifiable Rewards . . . 51
- L.3 Zeroth-Order Optimization for LLMs . . . 52

A Experimental Details

A.1 Implementation-Level Training Procedure

Full-parameter ES. Algorithm 1 details the full-parameter ES implementation of the population-standardized update in Eq. (5).

Algorithm 1 Full-parameter ES training

Require: Initial parameters θ_0, mini-batches $\{\mathcal{B}_t\}_{t=0}^{T-1}$, population size N, perturbation scale σ, update scale α, numerical constant $\varepsilon_{\text{num}} > 0$

for $t = 0, \ldots, T-1$ **do**
  Sample independent full-parameter directions $\epsilon_i \sim \mathcal{N}(0, I)$ for $i = 1, \ldots, N$.
  for $i = 1, \ldots, N$ **do**
    Generate one response per prompt in $\mathcal{B}_t$ using $\theta_t + \sigma\epsilon_i$.
    Set R_i to the mean verifier reward over $\mathcal{B}_t$.
  end for
  $\bar{R} \leftarrow N^{-1} \sum_{i=1}^{N} R_i$
  $z_i \leftarrow (R_i - \bar{R}) / \left(\sqrt{N^{-1} \sum_{j=1}^{N} (R_j - \bar{R})^2} + \varepsilon_{\text{num}}\right)$ for $i = 1, \ldots, N$
  $\theta_{t+1} \leftarrow \theta_t + \frac{\alpha}{N} \sum_{i=1}^{N} z_i \epsilon_i$
end for
return θ_T

A.2 Datasets

The Easy Setting uses GSM8K to compare ES and GRPO across three instruction-tuned model scales using readily verifiable mathematical answers. The Hard Setting pairs DeepSeek-R1-Distill-Qwen-1.5B with DeepScaleR training data to evaluate both methods under a reasoning-oriented post-training protocol aligned with prior mathematical RL work. Its backbone and mathematical evaluation benchmarks match those used by JustRL (He et al., 2025). Held-out tasks assess whether training-domain improvements accompany capability changes across other task families.

A.3 ES–GRPO FLOP Matching

Following Gan & Isola (2026), we approximate the forward and backward costs for a model with P parameters and response length L as $2PL$ and $4PL$ FLOPs. GRPO uses a policy forward, a reference forward, and a policy backward per response, whereas ES uses only a forward pass:

$$\text{FLOPs}_{\text{GRPO}} = 8T_{\text{GRPO}}BGPL, \tag{12}$$

$$\text{FLOPs}_{\text{ES}} = 2T_{\text{ES}}NDPL, \tag{13}$$

where T_{GRPO} and T_{ES} denote the respective update counts, B the GRPO prompt batch size, G responses per prompt, N ES population size, and D prompts per direction. With equal update counts, $D = B$, and $N = 32 = 4G$ for $G = 8$, both methods have equal estimated FLOPs.

Table 7: Dataset sources.

Dataset	Source	License
GSM8K (Cobbe et al., 2021)	`openai/gsm8k`	MIT
CommonsenseQA (Talmor et al., 2019)	`tau/commonsense_qa`	MIT
HotpotQA (Yang et al., 2018)	Official HotpotQA dataset release	CC BY-SA 4.0
Countdown	`HuggingFaceTB/Countdown-Task-GOLD`	Not specified
GPQA Diamond (Rein et al., 2023)	`Idavidrein/gpqa`	CC BY 4.0
MBPP (Austin et al., 2021)	`google-research-datasets/mbpp`	CC BY 4.0
DeepScaleR Preview (Luo et al., 2025)	`agentica-org/DeepScaleR-Preview-Dataset`	MIT
AIME 2024 (Zhang & Math-AI Team, 2024)	`math-ai/aime24`	Apache-2.0
AIME 2025 (Zhang & Math-AI Team, 2025)	`math-ai/aime25`	Apache-2.0
AMC 2023	`math-ai/amc23`	Not specified
MATH-500 (Lightman et al., 2023)	`HuggingFaceH4/MATH-500`	Not specified

Table 8: Dataset roles and task types in the Easy and Hard settings.

Setting	Role	Dataset	Task
Easy	Post-training / test	GSM8K	Grade-school mathematics
	Held-out test	CommonsenseQA	Commonsense reasoning
		HotpotQA distractor	Multi-hop question answering
		Countdown	Arithmetic-expression construction
		GPQA Diamond	Scientific reasoning
		MBPP sanitized	Python code generation
Hard	Post-training	DeepScaleR	Mathematical reasoning
	Mathematical test	AIME 2024	Competition mathematics
		AIME 2025	Competition mathematics
		AMC 2023	Competition mathematics
		MATH-500	Advanced mathematical reasoning
	Held-out test	GPQA Diamond	Scientific reasoning
		MBPP sanitized	Python code generation
		CommonsenseQA	Commonsense reasoning
		Countdown	Arithmetic-expression construction

A.4 Training and Evaluation Configurations

Training and evaluation use NVIDIA A100-SXM4-80GB GPUs, with evaluation running on a single GPU.

Training. Table 9 summarizes the training configurations. GRPO uses a higher sampling temperature to promote response diversity and reduce equal-reward groups with no relative reward signal (Equation (2)). ES can use a lower temperature because parameter perturbations provide exploration even under deterministic decoding.

Table 9: Training configurations under the Easy and Hard settings.

(a) Configurations that differ between settings

Parameter	Easy Setting	Hard Setting
Model	Qwen2.5-1.5B-Instruct; Qwen2.5-7B-Instruct; Llama-3.2-3B-Instruct	DeepSeek-R1-Distill-Qwen-1.5B
Training data	GSM8K	DeepScaleR
Epochs	2	1
Response limit (tokens)	2,048	8,192
ES sampling temperature	0	0.6

(b) Parameters shared across settings

Method	Parameter	Value
ES and GRPO	Prompt batch size	64
ES	Population size N	32
	Perturbation scale σ	1.5×10^{-3}
	Update scale α	2.5×10^{-4}
	Reward normalization	Population z-score
GRPO	Responses per prompt G	8
	Learning rate	1.0×10^{-6}
	Sampling temperature	1.0
	Clipping parameter	0.2
	KL coefficient β	0.001

Evaluation. Table 10 summarizes the evaluation configurations. In both settings, Pass@1 is estimated from 32 responses sampled at $\tau = 0.6$.

Table 10: Offline evaluation configurations under the Easy and Hard settings.

(a) Configurations that differ between settings

Parameter	Easy Setting	Hard Setting
Maximum new tokens	2,048	8,192

(b) Shared evaluation settings

Parameter	Value
Pass@1 decoding	Sampled
Pass@1 aggregation	Mean correctness over 32 responses
Responses per problem	32
Sampling temperature	0.6
Reported K	16, 32

Entropy and test-performance trajectories. Figure 2(a–b) tracks Qwen2.5-1.5B-Instruct's mean token-level full-vocabulary Shannon entropy (nats) on 512 fixed GSM8K training prompts and test Pass@16/32 changes from Base (percentage points).

A.5 Evaluation Metrics

Single-response accuracy, correct-response coverage, and voting accuracy are evaluated using Pass@1, Pass@K, and Maj@K, respectively. For a prompt x, let $p_\theta(x)$ denote the probability of generating a correct response under a fixed decoding configuration. Pass@1 equals $p_\theta(x)$, while Pass@K, the probability of obtaining at least one correct response from K independent samples, equals $1-(1-p_\theta(x))^K$. We report these metrics averaged over evaluation prompts. Although Pass@K increases monotonically with $p_\theta(x)$ for each prompt, improvements in dataset-average Pass@1 need not translate into improvements in dataset-average Pass@K, because success probabilities can change unevenly across prompts.

- **Pass@1.** Single-response accuracy is measured by Pass@1. In both settings, we sample $M = 32$ responses per problem and estimate Pass@1 as c/M, where c is the number of correct responses.
- **Pass@K.** The probability of obtaining at least one correct response within K samples ($2 \le K \le M$) is estimated by

$$\widehat{\text{Pass@}}K = 1 - \frac{\binom{M-c}{K}}{\binom{M}{K}}. \tag{14}$$

  Pass@32 is thus computed as the fraction of problems with at least one correct response.
- **Maj@K.** An answer is selected by deterministic plurality voting over the first K normalized answers and scored against the gold answer.

All methods share the evaluation protocol within each setting (Table 10).

A.6 Prompt Templates

The following messages are rendered with each model's official chat template.

Prompt for Mathematical Reasoning

system
Please reason step by step, and put your final answer within `\boxed{}`.
user
{Question}

Prompt for Multiple-Choice Reasoning

system
Please reason step by step to solve the following multiple-choice question. Put your final choice letter inside `\boxed{}`, e.g., `\boxed{C}`.

user
{Question and labeled choices}

Prompt for Open Question Answering

system
Answer the question using the provided context. Keep the final answer short, and put it inside `<answer></answer>` tags.

user
{Context and question}

Prompt for MBPP Code Generation

system
You are a Python programming assistant. Write clean, correct Python code that passes the given tests. The function name and signature must match exactly as specified in the test cases. The final code block must not include the tests; do not include the tests in your answer. Do not include explanations or text outside the code block. Output exactly one Python markdown code block containing only the final solution code.

user
Task: {Programming task}
Test Cases: {Public tests}

Countdown uses the role/content messages distributed with the dataset and renders them without rewriting the task statement. Answer extraction follows the same task-family contract as scoring: boxed answers for mathematics and multiple choice, `<answer>` tags for open QA and Countdown, and one Python code block for MBPP.

B PROOFS FOR THE RQ1 DIVERSITY–COVERAGE ANALYSIS

This appendix proves Lemmas 1, 2, and 3 and Proposition 1. In Section 3.1 and this appendix, the realized population, fitness values, and resulting update are held fixed unless an expectation explicitly averages over these quantities. The Jensen–Shannon (JS) definitions and entropy representation follow the standard formulation of Lin (1991). Data processing and Pinsker bounds follow Cover & Thomas (2006).

B.1 PROOF OF LEMMA 2

To quantify perturbation-induced policy diversity, define the score $s_\theta(y \mid x) = \nabla_\theta \log \pi_\theta(y \mid x)$ and the prompt-conditioned Fisher information

$$\mathcal{I}_x(\theta) = \mathbb{E}_{Y \sim \pi_\theta(\cdot|x)} \left[s_\theta(Y \mid x) s_\theta(Y \mid x)^\top \right]. \tag{15}$$

Lemma 2 (Perturbations induce policy diversity)**.** *For a fixed prompt x and population size N, draw $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, I)$, define $\pi_i(y \mid x) = \pi_{\theta+\sigma\epsilon_i}(y \mid x)$ and $\bar{\pi}_N = N^{-1} \sum_i \pi_i$, and let $\mathrm{JS}_N^{\mathrm{pol}}(x) = N^{-1} \sum_i D_{\mathrm{KL}}(\pi_i \,\|\, \bar{\pi}_N)$. The prompt-conditioned Fisher information $\mathcal{I}_x(\theta)$ is defined in Equation (15). Assuming finite entropy and Fisher information, common local support, and sufficient smoothness under the regularity conditions stated below, as $\sigma \to 0$,*

$$\mathbb{E}_{\epsilon_{1:N}} \left[\mathrm{JS}_N^{\mathrm{pol}}(x) \right] = \frac{\sigma^2}{2} \left(1 - \frac{1}{N} \right) \operatorname{tr} \mathcal{I}_x(\theta) + O(\sigma^4). \tag{16}$$

Proof. Finite response-distribution entropies imply this identity:

$$H(\bar{\pi}_N) = \frac{1}{N} \sum_{i=1}^{N} H(\pi_i) + \mathrm{JS}_N^{\mathrm{pol}}(x). \tag{17}$$

Thus, $\mathrm{JS}_N^{\mathrm{pol}}$ measures diversity across population members. Sampling entropy describes variability within an individual policy. For a GRPO rollout group drawn from one behavior policy, all member policies are identical and the corresponding cross-parameter JS is zero, although within-policy sampling entropy can remain nonzero.

Take N and the parameter dimension to be fixed and finite, and let $\mathcal{Y}_x$ be the countable response space for prompt x. Write $q_\Delta(y) = \pi_{\theta+\Delta}(y \mid x)$, $q_0 = q_{\Delta=0}$, $s_\theta = s_\theta(y \mid x)$, $\bar{\Delta} = N^{-1} \sum_i \Delta_i$, $\bar{q} = N^{-1} \sum_i q_{\Delta_i}$, and $\|\Delta_{1:N}\| = \max_i \|\Delta_i\|$. For some $r > 0$, assume that q_Δ has common positive support and finite entropy on $\{\|\Delta\| \le r\}$, is four times continuously differentiable in Δ, and has finite Fisher information at $\Delta = 0$. Define the per-response JS integrand $\ell_y(\Delta_{1:N}) = N^{-1} \sum_i q_{\Delta_i}(y) \log[q_{\Delta_i}(y)/\bar{q}(y)]$. For every multi-index $|\nu| \le 4$, assume $\sup_{\|\Delta_{1:N}\| \le r} |\partial^\nu \ell_y(\Delta_{1:N})| \le M_\nu(y)$ for an envelope satisfying $\sum_{y \in \mathcal{Y}_x} M_\nu(y) < \infty$. These conditions justify termwise differentiation of the response sum and a uniform fourth-order Taylor remainder. For local offsets $\Delta_1, \ldots, \Delta_N$, Taylor expansion gives $q_\Delta = q_0[1 + s_\theta^\top \Delta] + O(\|\Delta\|^2)$ and $\bar{q} = q_0[1 + s_\theta^\top \bar{\Delta}] + O(\|\Delta_{1:N}\|^2)$. Substituting into the KL divergence and using $\mathbb{E}_{q_0}[s_\theta] = 0$ and $\mathbb{E}_{q_0}[s_\theta s_\theta^\top] = \mathcal{I}_x(\theta)$ yields the quadratic term below. Retaining the third- and fourth-order terms gives

$$\begin{aligned} \mathrm{JS}_N^{\mathrm{pol}}(x) = \frac{1}{2N} \sum_{i=1}^{N} (\Delta_i - \bar{\Delta})^\top \mathcal{I}_x(\theta) (\Delta_i - \bar{\Delta}) \\ + \mathcal{T}_3(\Delta_{1:N}) + \mathcal{R}_4(\Delta_{1:N}), \end{aligned} \tag{18}$$

where $\mathcal{T}_3$ is homogeneous of degree three and $|\mathcal{R}_4(\Delta_{1:N})| \le C \|\Delta_{1:N}\|^4$ whenever $\|\Delta_{1:N}\| \le r$. For $\Delta_i = \sigma\epsilon_i$, the covariance becomes

$$\mathbb{E}\left[(\Delta_i - \bar{\Delta})(\Delta_i - \bar{\Delta})^\top \right] = \sigma^2 \left(1 - \frac{1}{N} \right) I. \tag{19}$$

The expected quadratic term is therefore $\frac{\sigma^2}{2}(1 - 1/N) \operatorname{tr} \mathcal{I}_x(\theta)$. Let $E_\sigma = \{\max_i \|\sigma\epsilon_i\| \le r\}$. It is sign invariant, so $\mathbb{E}[\mathcal{T}_3(\sigma\epsilon_{1:N}) \mathbf{1}_{E_\sigma}] = 0$. The remainder on E_σ has expectation $O(\sigma^4)$. Moreover, $\mathrm{JS}_N^{\mathrm{pol}} \le \log N$ and $\Pr(E_\sigma^c) = O(e^{-c/\sigma^2})$, so the JS tail and quadratic truncation are $o(\sigma^4)$, establishing the expansion in Equation (16).

B.2 Proof of Lemma 1

Conditional on the fixed population, one response is sampled independently from each member policy. The comparator uses N independent responses from a policy with success probability $\bar{p}(x)$.

Let $\mathcal{C}(x) = \{y : v(x, y) = 1\}$, so that $p_i(x) = \Pr_{Y \sim \pi_i(\cdot|x)}[Y \in \mathcal{C}(x)]$. Using binary entropy h, define the verifier-projected diversity $\mathrm{JS}_N^{\mathrm{succ}}(x) = h(\bar{p}(x)) - N^{-1} \sum_i h(p_i(x))$. The verifier maps $\pi_i(\cdot \mid x)$ to $B_i(x) = \mathrm{Bernoulli}(p_i(x))$, so data processing gives

$$0 \le \mathrm{JS}_N^{\mathrm{succ}}(x) \le \mathrm{JS}_N^{\mathrm{pol}}(x). \tag{20}$$

This projection is necessary because full policy JS may arise solely from variation among incorrect responses.

For a fixed prompt x, suppress the dependence on x and write $p_i = p_i(x)$ and $\bar{p} = N^{-1} \sum_i p_i$. By the arithmetic–geometric mean inequality,

$$\prod_{i=1}^{N}(1 - p_i) \le \left(\frac{1}{N} \sum_{i=1}^{N}(1 - p_i) \right)^N = (1 - \bar{p})^N. \tag{21}$$

Subtracting both sides from one proves $P_N^{\mathrm{pop}} \ge P_N^{\mathrm{same}}$. Equality in the arithmetic–geometric mean inequality holds exactly when $1 - p_1 = \cdots = 1 - p_N$, equivalently $p_1 = \cdots = p_N$.

For the local relation, let $p_i = \bar{p} + \xi_i$, $\sum_i \xi_i = 0$, and $\beta = 1 - \bar{p}$, where ξ_i is the centered member-success deviation and β the average failure probability. Assume that $\bar{p}$ remains in a compact subset of $(0, 1)$. Expanding at $\xi = 0$ and subtracting from one gives

$$\prod_{i=1}^{N}(\beta - \xi_i) = \beta^N - \frac{\beta^{N-2}}{2} \sum_{i=1}^{N} \xi_i^2 + O(\|\xi\|^3), \tag{22}$$

$$P_N^{\mathrm{pop}} - P_N^{\mathrm{same}} = \frac{\beta^{N-2}}{2} \sum_{i=1}^{N} \xi_i^2 + O(\|\xi\|^3). \tag{23}$$

For $h(p) = -p \log p - (1 - p) \log(1 - p)$, the binary entropy satisfies $h''(p) = -1/[p(1 - p)]$. A Taylor expansion of the success-JS definition above, with the linear term canceled by $\sum_i \xi_i = 0$, yields

$$\mathrm{JS}_N^{\mathrm{succ}} = \frac{1}{2N\bar{p}(1 - \bar{p})} \sum_{i=1}^{N} \xi_i^2 + O(\|\xi\|^3). \tag{24}$$

Eliminating $\sum_i \xi_i^2$ between Equations (23) and (24) gives

$$P_N^{\mathrm{pop}}(x) - P_N^{\mathrm{same}}(x) = N\bar{p}(x)(1 - \bar{p}(x))^{N-1}\, \mathrm{JS}_N^{\mathrm{succ}}(x) + O(\|\xi(x)\|^3). \tag{25}$$

B.3 Proof and Illustration of Lemma 3

Lemma 3 (Reward weighting improves success under positive correlation)**.** *Let $w_i \ge 0$, $\sum_i w_i = 1$, be normalized weights obtained by a monotone transformation of realized fitness R_i. Selecting policy i with probability w_i and then sampling a response from it defines the reward-weighted policy $\pi_w = \sum_i w_i \pi_i$, with success probability $p_w(x) = \sum_i w_i p_i(x)$. With Cov_i taken under a uniform member draw, $p_w(x) - \bar{p}(x) = N\,\mathrm{Cov}_i(w_i, p_i(x))$. Thus, weighting improves success on x if and only if the weights positively correlate with $p_i(x)$.*

Proof. Under a uniform draw of member index i, $\mathbb{E}_i[w_i] = 1/N$ and $\mathbb{E}_i[p_i] = \bar{p}$. Hence

$$N\,\mathrm{Cov}_i(w_i, p_i) = \sum_{i=1}^{N} w_i p_i - \bar{p} = p_w - \bar{p}, \tag{26}$$

which proves the covariance identity in Lemma 3. Monotonicity in observed fitness R_i alone does not guarantee positive correlation with the true prompt-wise success probabilities $p_i(x)$. The policy

π_w serves as an analytical comparator. Relating it to the policy produced by the ES parameter update requires the transfer conditions below. Its nonnegative weights w_i differ from the signed standardized rewards z_i used to aggregate parameter perturbations in Appendix A.1.

For intuition only, define the prompt-adaptive oracle gate

$$\widetilde{w}_i(x) = \frac{\exp(\lambda p_i(x))}{\sum_{j=1}^{N} \exp(\lambda p_j(x))}, \qquad \widetilde{p}_w(x) = \sum_{i=1}^{N} \widetilde{w}_i(x) p_i(x), \tag{27}$$

where $\lambda > 0$. For fixed $N \geq 2$, write $p_i = \bar{p} + \xi_i$ with $\sum_i \xi_i = 0$ and assume that $\bar{p}$ is bounded away from zero and one. Expanding the weights gives

$$\widetilde{w}_i = \frac{\exp(\lambda \xi_i)}{\sum_j \exp(\lambda \xi_j)} = \frac{1}{N} + \frac{\lambda}{N} \xi_i + O(\|\xi\|^2). \tag{28}$$

Using $\sum_i \xi_i = 0$ and Equation (24),

$$\widetilde{p}_w - \bar{p} = \sum_i \widetilde{w}_i \xi_i = \frac{\lambda}{N} \sum_i \xi_i^2 + O(\|\xi\|^3) \tag{29}$$

$$= 2\lambda \bar{p}(1-\bar{p})\, \mathrm{JS}_N^{\mathrm{succ}} + O(\|\xi\|^3). \tag{30}$$

This oracle calculation illustrates the benefit of weights aligned with true prompt-wise success probabilities. It does not establish such alignment for observed fitness.

B.4 Proof of Proposition 1

The preceding lemmas compare against population-average success. Improvement over the center policy before the ES update requires an additional assumption: a positive margin m_K. We additionally assume a bound on the transfer divergence. The improvement guarantee for ES is conditional on both assumptions.

Data processing yields a verifier-projected KL bound:

$$D_{\mathrm{KL}}(B_w \,\|\, B_{\theta^+}) \leq D_{\mathrm{KL}}(\pi_w \,\|\, \pi_{\theta^+}). \tag{31}$$

Bounding the verifier-projected KL controls the difference in success probabilities without requiring the full response distributions to be close.

Let $D_x = D_{\mathrm{KL}}(B_w(x) \,\|\, B_{\theta^+}(x))$. For Bernoulli distributions, total variation equals the absolute success-probability difference. Pinsker's inequality gives

$$|p_w(x) - p_{\theta^+}(x)| \leq \sqrt{\frac{D_x}{2}}. \tag{32}$$

The map $f_K(p) = 1 - (1-p)^K$ is K-Lipschitz since $0 \leq f_K'(p) = K(1-p)^{K-1} \leq K$ on $[0,1]$. Hence,

$$f_K(p_{\theta^+}(x)) \geq f_K(p_w(x)) - K\sqrt{\frac{D_x}{2}}. \tag{33}$$

Taking expectations and applying Jensen's inequality and Equation (8) yields

$$\begin{aligned} J_K(\pi_{\theta^+}) &\geq J_K(\pi_w) - K\mathbb{E}_{x\sim\mathcal{D}}\sqrt{D_x/2} \\ &\geq J_K(\pi_w) - K\sqrt{\mathbb{E}_{x\sim\mathcal{D}}[D_x]/2} \\ &\geq J_K(\pi_w) - K\sqrt{\varepsilon_{\mathrm{succ}}/2}. \end{aligned} \tag{34}$$

Combining this bound with the margin condition in Equation (9) proves the chained conclusion in Equation (10).

C Results Across Training Seeds

We evaluate ES and GRPO using three training seeds on Qwen2.5-1.5B-Instruct. Tables 11 and 12 summarize task-level and four-task average results, respectively.

Table 11: Task-level results across three training seeds on Qwen2.5-1.5B-Instruct ($\times 100$).

	GSM8K			HotpotQA		
Method	P@1	P@16	P@32	P@1	P@16	P@32
Base	$70.48_{\pm 0.34}$	$94.90_{\pm 0.14}$	$96.48_{\pm 0.16}$	$27.84_{\pm 0.39}$	$46.59_{\pm 0.01}$	$50.30_{\pm 0.03}$
GRPO	$77.66_{\pm 0.64}$	$94.10_{\pm 0.29}$	$95.74_{\pm 0.24}$	$24.82_{\pm 0.36}$	$44.44_{\pm 0.42}$	$48.32_{\pm 0.50}$
ES	$73.23_{\pm 0.11}$	$95.12_{\pm 0.21}$	$96.82_{\pm 0.19}$	$26.50_{\pm 2.23}$	$45.29_{\pm 0.55}$	$49.04_{\pm 1.12}$
	Countdown			MBPP		
Method	P@1	P@16	P@32	P@1	P@16	P@32
Base	$7.50_{\pm 0.01}$	$47.16_{\pm 0.59}$	$57.18_{\pm 0.46}$	$58.66_{\pm 0.08}$	$83.13_{\pm 1.31}$	$86.52_{\pm 1.48}$
GRPO	$8.48_{\pm 0.35}$	$48.74_{\pm 0.66}$	$58.24_{\pm 0.54}$	$60.18_{\pm 0.22}$	$82.76_{\pm 0.72}$	$86.12_{\pm 0.59}$
ES	$8.31_{\pm 0.43}$	$48.67_{\pm 0.77}$	$58.70_{\pm 1.36}$	$58.76_{\pm 0.05}$	$83.24_{\pm 0.24}$	$86.65_{\pm 0.44}$

Table 12: Four-task average results across three training seeds on Qwen2.5-1.5B-Instruct ($\times 100$).

Method	P@1	P@16	P@32
Base	$41.12_{\pm 0.20}$	$67.95_{\pm 0.15}$	$72.62_{\pm 0.22}$
GRPO	$42.79_{\pm 0.13}$	$67.54_{\pm 0.23}$	$72.10_{\pm 0.32}$
ES	$41.70_{\pm 0.50}$	$68.09_{\pm 0.19}$	$72.80_{\pm 0.32}$

To quantify question-sampling uncertainty, we compute paired percentile bootstrap confidence intervals using 10,000 replicates. Questions are resampled within each task, with shared indices across methods and runs. Bootstrap intervals quantify question-sampling uncertainty conditional on the evaluated runs. Table 13 reports the resulting intervals.

Table 13: Paired question-bootstrap confidence intervals for the four-task average ES–GRPO differences, averaged across three runs. Intervals are marginal 95% confidence intervals.

Metric	ES − GRPO (pp)	95% CI (pp)
P@1	-1.088	$[-1.400, -0.783]$
P@16	$+0.559$	$[+0.082, +1.023]$
P@32	$+0.698$	$[+0.093, +1.275]$

These results support the repeatability of ES's average coverage advantage over GRPO.

D SENSITIVITY TO TEST-TIME TEMPERATURE

We examine test-time temperatures $T \in \{0.4, 0.6, 0.8\}$ for Llama-3.2-3B-Instruct in the Easy Setting and $T \in \{0.6, 0.8, 1.0\}$ for DeepSeek-R1-Distill-Qwen-1.5B in the Hard Setting. The $T = 0.6$ results appear in Tables 2 and 3, respectively.

Llama-3.2-3B-Instruct. Tables 14 and 15, together with Table 2, support the main-text conclusion at temperatures near the default setting: ES improves average Pass@1 over Base and achieves higher average Pass@16 and Pass@32 than both Base and GRPO. We focus on this nearby temperature range because higher test-time temperatures create a large training–inference temperature mismatch that may cause average Pass@1 to collapse.

Table 14: Llama-3.2-3B-Instruct: Pass@K ($\times 100$) at $T = 0.4$.

	GSM8K			HotpotQA			Countdown			MBPP			Average		
Method	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32
Llama-3.2-3B-Instruct															
Base	78.5	95.9	97.2	27.8	40.2	42.8	6.8	43.2	52.4	63.2	77.7	79.8	44.1	64.3	68.0
GRPO	84.7	95.2	96.3	27.8	36.7	38.5	8.6	50.7	59.9	62.4	73.3	75.1	**45.9**	64.0	67.4
ES	82.0	95.6	97.0	29.9	41.6	44.0	8.2	48.5	58.6	62.0	77.9	81.3	45.5	**65.9**	**70.2**

Table 15: Llama-3.2-3B-Instruct: Pass@K ($\times 100$) at $T = 0.8$.

	GSM8K			HotpotQA			Countdown			MBPP			Average		
Method	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32
Llama-3.2-3B-Instruct															
Base	75.6	97.1	98.6	24.9	47.5	51.9	6.2	44.8	55.9	60.6	81.4	83.3	41.8	67.7	72.4
GRPO	84.4	95.9	97.0	26.8	43.5	47.2	10.4	53.7	62.2	61.6	79.3	81.7	**45.8**	68.1	72.0
ES	80.7	96.4	97.4	27.4	49.0	53.3	6.8	47.4	58.1	59.9	82.6	84.8	43.7	**68.9**	**73.4**

DeepSeek-R1-Distill-Qwen-1.5B. Tables 16 and 17, together with Table 3, support the main-text conclusion across the reported temperatures: ES improves average Pass@1 over Base and achieves higher average Pass@16 and Pass@32 than both Base and GRPO.

Table 16: DeepSeek-R1-Distill-Qwen-1.5B: Pass@K ($\times 100$) at $T = 0.8$.

	AIME24			AIME25			AMC23			MATH500			Average		
Method	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32
DeepSeek-R1-Distill-Qwen-1.5B															
Base	25.0	53.2	56.7	22.2	44.3	46.7	66.5	93.1	95.0	81.1	96.3	97.2	48.7	71.7	73.9
GRPO	27.5	54.9	60.0	23.6	44.2	50.0	75.8	94.4	95.0	85.2	96.7	97.4	**53.0**	72.6	75.6
ES	22.6	63.0	70.0	22.9	48.0	53.3	69.5	94.1	95.0	82.9	96.7	97.8	49.5	**75.5**	**79.0**

Table 17: DeepSeek-R1-Distill-Qwen-1.5B: Pass@K ($\times 100$) at $T = 1.0$.

	AIME24			AIME25			AMC23			MATH500			Average		
Method	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32
DeepSeek-R1-Distill-Qwen-1.5B															
Base	20.6	55.0	63.3	21.4	43.3	46.7	62.5	92.1	92.5	80.1	95.7	96.6	46.1	71.5	74.8
GRPO	26.2	57.4	63.3	23.5	44.0	50.0	74.3	94.4	95.0	84.9	96.2	96.8	**52.2**	73.0	76.3
ES	22.4	58.5	63.3	22.1	46.3	53.3	67.1	96.8	97.5	82.4	96.3	97.4	48.5	**74.5**	**77.9**

E Complete Multiple-Choice Results

Tables 18 and 19 report Pass@K results on CommonsenseQA and GPQA Diamond. Large-K results describe correct-option sampling coverage. Multiple-choice tasks have a small answer space, so repeated sampling can include the correct option even when the model frequently selects incorrect answers. Consequently, Pass@K can approach saturation as K increases, limiting its ability to distinguish reliable answer selection from occasional correct guesses. We therefore assess capability retention primarily through Pass@1 and Maj@K, which measure single-response accuracy and the accuracy of the answer selected by voting, respectively. P@K denotes Pass@K. Task-cell underlining marks the best result across all five methods.

Table 18: Pass@K results ($\times 100$) on CommonsenseQA and GPQA Diamond in the Easy Setting.

	CSQA			GPQA		
Method	P@1	P@16	P@32	P@1	P@16	P@32
Qwen2.5-1.5B-Instruct						
Base	60.6	93.4	96.1	21.4	87.1	94.4
GRPO	63.7	93.2	95.5	22.2	87.3	96.0
ES	60.7	94.1	96.8	22.9	88.5	96.5
ES→GRPO	62.1	93.9	96.5	22.2	88.1	96.0
GRPO→ES	62.5	93.7	96.0	22.8	86.6	95.5
Llama-3.2-3B-Instruct						
Base	67.3	92.8	95.3	23.0	82.0	90.4
GRPO	72.0	90.5	92.8	26.9	80.3	88.4
ES	69.9	93.0	95.5	25.0	86.9	95.5
ES→GRPO	72.0	91.3	93.7	24.4	71.4	79.3
GRPO→ES	71.3	91.8	94.3	26.6	84.5	92.9
Qwen2.5-7B-Instruct						
Base	80.0	92.9	94.3	33.3	82.2	89.4
GRPO	80.7	91.7	92.7	34.8	79.2	85.4
ES	79.0	92.6	94.3	34.4	83.6	88.9
ES→GRPO	80.2	92.4	93.7	35.5	79.9	85.9
GRPO→ES	80.2	91.9	93.2	34.2	80.8	87.4

Table 19: Pass@K results ($\times 100$) on CommonsenseQA and GPQA Diamond in the one-epoch Hard Setting.

	CSQA			GPQA		
Method	P@1	P@16	P@32	P@1	P@16	P@32
DeepSeek-R1-Distill-Qwen-1.5B						
Base	46.8	89.2	93.7	24.2	80.8	91.4
GRPO	47.5	88.9	93.6	31.8	86.4	91.9
ES	46.0	88.9	93.4	28.0	84.6	92.4
ES→GRPO	46.8	88.8	93.3	30.6	84.5	91.4
GRPO→ES	47.3	88.7	93.2	30.6	86.7	93.4

F Complete Majority-Vote Results

This appendix reports complete Maj@16 and Maj@32 results under the Easy Setting and on both the mathematical and held-out benchmarks under the Hard Setting. Each Average column includes all benchmarks displayed in its table. Maj@K applies deterministic plurality voting after task-specific answer normalization, as defined in Appendix A.5.

	GSM8K		CSQA		HotpotQA		Countdown		GPQA		MBPP		Average	
Method	M@16	M@32	M@16	M@32	M@16	M@32	M@16	M@32	M@16	M@32	M@16	M@32	M@16	M@32
Qwen2.5-1.5B-Instruct														
Base	82.6	84.3	67.8	68.8	31.1	31.1	21.7	27.0	20.7	20.2	61.9	61.5	47.6	48.8
GRPO	84.2	84.8	70.0	71.0	28.8	29.0	27.8	31.5	22.7	22.7	60.3	61.9	**49.0**	50.2
ES	82.3	83.9	67.2	68.9	29.8	29.8	27.6	33.1	25.3	24.2	58.4	62.3	48.4	**50.4**
ES→GRPO	83.9	84.2	69.2	69.6	28.8	29.1	26.2	31.6	20.7	18.7	61.5	62.6	48.4	49.3
GRPO→ES	83.1	83.6	68.7	70.0	29.2	29.4	20.7	24.6	23.7	23.7	58.8	61.9	47.4	48.9
Llama-3.2-3B-Instruct														
Base	86.9	87.8	74.1	74.6	28.1	28.5	39.2	45.7	33.8	33.8	62.3	64.6	54.1	55.8
GRPO	88.9	89.5	74.3	74.6	27.4	27.7	44.6	50.8	35.4	33.8	62.3	63.0	**55.5**	**56.6**
ES	88.6	88.9	74.9	75.5	29.5	30.0	40.4	48.9	30.3	30.8	62.3	61.9	54.3	56.0
ES→GRPO	88.4	88.2	75.9	75.8	26.6	26.8	41.9	48.7	29.8	29.3	63.8	62.3	54.4	55.2
GRPO→ES	89.1	89.8	74.9	75.6	28.5	28.5	40.0	47.5	30.8	31.3	63.0	62.6	54.4	55.9
Qwen2.5-7B-Instruct														
Base	94.5	94.5	81.4	81.7	45.9	46.0	56.6	58.4	37.4	38.4	77.0	77.0	65.5	66.0
GRPO	93.8	93.6	81.9	81.4	46.8	46.9	56.6	57.4	34.8	37.9	75.9	75.9	65.0	65.5
ES	94.2	94.2	80.8	81.2	45.7	45.7	57.9	60.1	40.4	41.9	75.1	75.9	**65.7**	**66.5**
ES→GRPO	94.3	94.6	81.4	81.4	46.5	46.5	57.3	59.2	37.4	37.9	75.1	74.7	65.3	65.7
GRPO→ES	93.7	94.0	82.2	81.7	46.2	46.3	56.1	57.2	40.4	39.4	75.1	75.9	65.6	65.8

Table 20: Maj@K results ($\times 100$) in the Easy Setting.

	AIME24		AIME25		AMC23		MATH500		Average	
Method	M@16	M@32	M@16	M@32	M@16	M@32	M@16	M@32	M@16	M@32
DeepSeek-R1-Distill-Qwen-1.5B										
Base	36.7	40.0	33.3	33.3	82.5	82.5	90.0	91.0	60.6	61.7
GRPO	43.3	46.7	30.0	33.3	85.0	87.5	91.2	91.8	62.4	64.8
ES	36.7	43.3	36.7	33.3	90.0	92.5	91.8	92.2	63.8	65.3
ES→GRPO	46.7	53.3	33.3	33.3	85.0	92.5	91.6	92.6	64.2	**67.9**
GRPO→ES	46.7	46.7	40.0	36.7	90.0	85.0	91.4	92.2	**67.0**	65.1

Table 21: Maj@K results ($\times 100$) on mathematical benchmarks in the Hard Setting.

	GPQA		MBPP		CSQA		Countdown		Average	
Method	M@16	M@32	M@16	M@32	M@16	M@32	M@16	M@32	M@16	M@32
DeepSeek-R1-Distill-Qwen-1.5B										
Base	33.3	33.3	69.6	70.4	54.1	54.1	71.1	75.1	57.0	58.2
GRPO	35.9	35.4	69.6	70.8	53.7	52.7	68.5	72.8	56.9	57.9
ES	39.4	39.4	70.8	69.6	52.7	53.6	71.7	76.2	**58.6**	**59.7**
ES→GRPO	36.4	37.4	71.6	72.0	52.4	53.0	70.9	73.6	57.8	59.0
GRPO→ES	33.8	34.8	66.5	69.6	53.3	54.1	72.2	76.9	56.5	58.9

Table 22: Maj@K results ($\times 100$) on four held-out benchmarks in the one-epoch Hard Setting.

G PASS@K PROFILES OF INDIVIDUAL POST-TRAINING METHODS AND SEQUENTIAL ES–GRPO TRAINING

Figure 6 first compares the Pass@K profiles of ES and GRPO when applied individually, and then evaluates our sequential ES–GRPO training method in both orders: ES→GRPO and GRPO→ES. In panels (a–b), GRPO leads at Pass@1, whereas ES overtakes it for $K \geq 4$. ES exceeds Base at every reported K on MATH-500 and for $K \geq 2$ on Countdown. In panels (c–d), ES→GRPO gives the strongest AIME24 profile, while GRPO→ES leads on AIME25 for $K \geq 2$. Thus, our sequential ES–GRPO training method can better exploit the complementary strengths of ES and GRPO, although the effective training order is task-dependent.

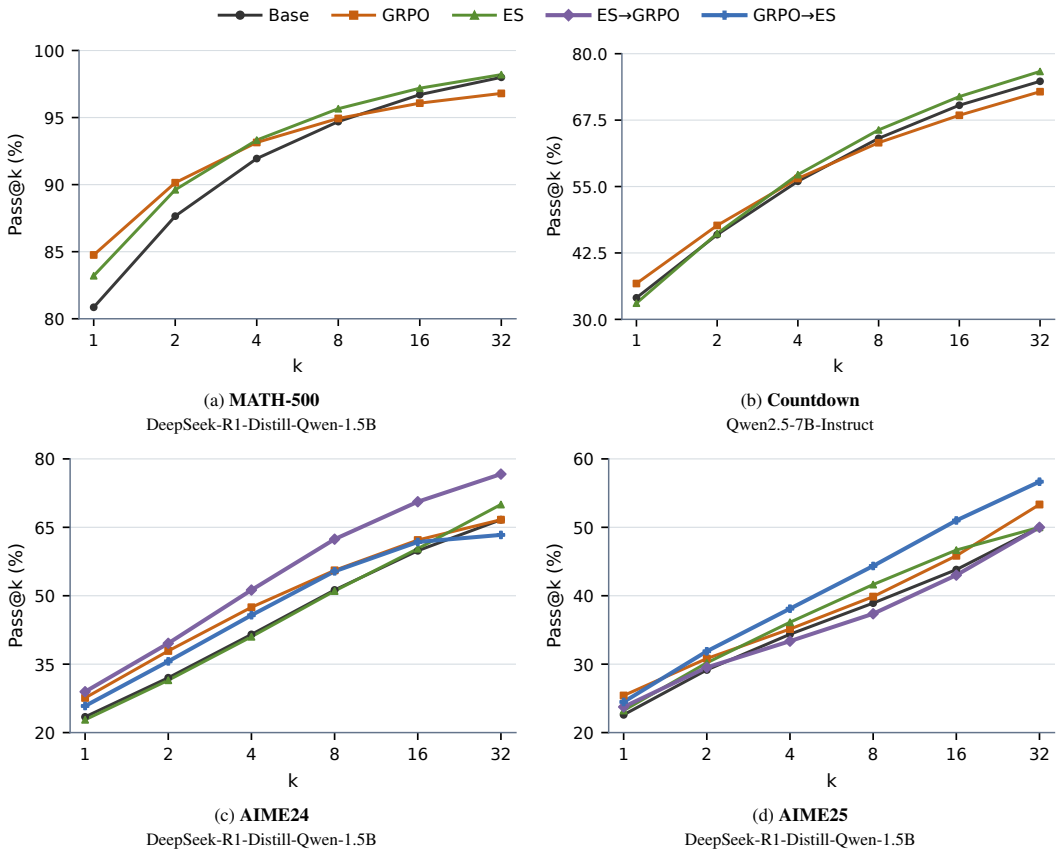


Figure 6: Pass@K profiles for individual post-training methods (a–b) and our sequential ES–GRPO training method (c–d).

H Additional RQ2 Results and Parameter Analyses

H.1 Parameter Changes Across Layers and Modules

Measurement We measure parameter displacement from the base model to each ES or GRPO endpoint, $\Delta\theta = \theta_{\text{trained}} - \theta_{\text{base}}$. For a module type, we pool its parameters across all 28 Transformer blocks and compute the root-mean-square (RMS) parameter change $R = \sqrt{\sum_j (\Delta\theta_j)^2 / n}$, where n is the number of pooled parameters. Token embeddings, final RMSNorm, and the LM head are measured separately; tied embedding and head weights are counted once. Figures 7 and 8 use a common piecewise-logarithmic color scale for update RMS at original checkpoint precision, preserving the absolute ES–GRPO comparison. Uncolored operations have no parameters. Llama-3.2-3B-Instruct has no Q/K/V bias.

Whole-block RMS uses the same coordinate-weighted calculation over all parameters within a Transformer block, including coordinates with zero change. In Figure 4b, attention projections pool Q, K, V, and O weights, and MLP projections pool gate, up, and down weights. For groups with parameter counts n_g and RMS values R_g, the pooled value is $\sqrt{\sum_g n_g R_g^2 / \sum_g n_g}$. Biases are excluded from these projection groups and reported separately in the architecture diagrams. All measurements in this subsection concern saved endpoint displacement and do not establish per-step update behavior or functional importance.

Module differences Across all four models, ES produces larger parameter update RMS values in **Pre-attention RMSNorm** and **Pre-MLP RMSNorm** than in the attention and MLP weight matrices. **Q and K biases** also have larger update RMS than these matrices in the three models that include such biases. GRPO shows different patterns across models: V bias stands out in Qwen2.5-1.5B-Instruct, and K/V weights and V bias stand out in DeepSeek-R1-Distill-Qwen-1.5B. Within-block module update RMS values are more similar in Llama-3.2-3B-Instruct and Qwen2.5-7B-Instruct. Final RMSNorm has higher update RMS under ES than GRPO in all four models, but the ES/GRPO ratio varies substantially: 1.11 for Qwen2.5-1.5B-Instruct, 10.22 for Qwen2.5-7B-Instruct, 192.52 for Llama-3.2-3B-Instruct, and 872.94 for DeepSeek-R1-Distill-Qwen-1.5B.

Consistency across layers The larger update RMS values in **Pre-attention RMSNorm** and **Pre-MLP RMSNorm** under ES recur across most layers. Pre-attention RMSNorm exceeds its own block update RMS in 27–28 of 28 layers, and pre-MLP RMSNorm does so in 24–28 layers. Q and K biases exceed their own block update RMS in all 28 layers in models that include these biases. For GRPO, V bias exceeds block update RMS in 27 layers of Qwen2.5-1.5B-Instruct; K weights, V weights, and V bias do so in 26, 28, and 26 layers of DeepSeek-R1-Distill-Qwen-1.5B, respectively. For Qwen2.5-7B-Instruct under GRPO, Pre-MLP RMSNorm and K weights exceed their block update RMS in 26 layers each. DeepSeek-R1-Distill-Qwen-1.5B GRPO pre-attention RMSNorm exceeds block update RMS only in the first layer, illustrating why pooled values need a layer-wise check. Tables 24 and 25 report the full range of within-layer update RMS ratios and the maximum absolute changes and their multiplicities.

Update magnitude distribution GRPO parameter displacements occur at substantially smaller absolute magnitudes than ES displacements. Table 23 shows that across the four models, approximately 97.26–100.00% of all GRPO parameters have $|\Delta\theta| \leq 10^{-4}$, compared with 10.01–13.51% for ES. Even after excluding unchanged parameters, over 99.99% of GRPO updates in the three Easy models and 86.13% in DeepSeek-R1-Distill-Qwen-1.5B do not exceed this threshold. These results describe displacement magnitudes in the saved checkpoints.

Table 23: Fraction of all parameters with absolute displacement at or below 10^{-4}, including unchanged parameters.

Model	ES	GRPO
Qwen2.5-1.5B-Instruct	13.51425%	99.99641%
Llama-3.2-3B-Instruct	11.32816%	99.99979%
Qwen2.5-7B-Instruct	10.00684%	99.99992%
DeepSeek-R1-Distill-Qwen-1.5B	11.88040%	97.25696%

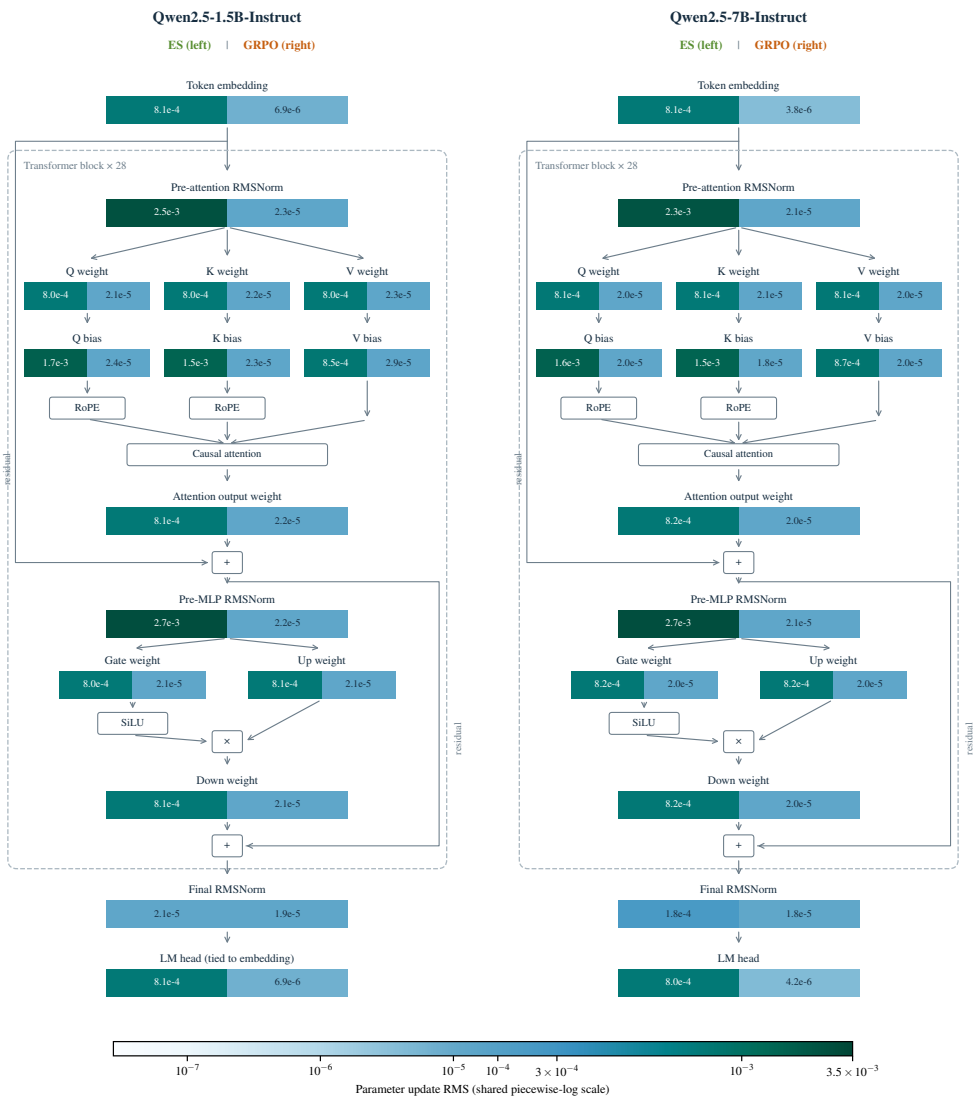


Figure 7: Module-wise parameter update RMS for Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct.

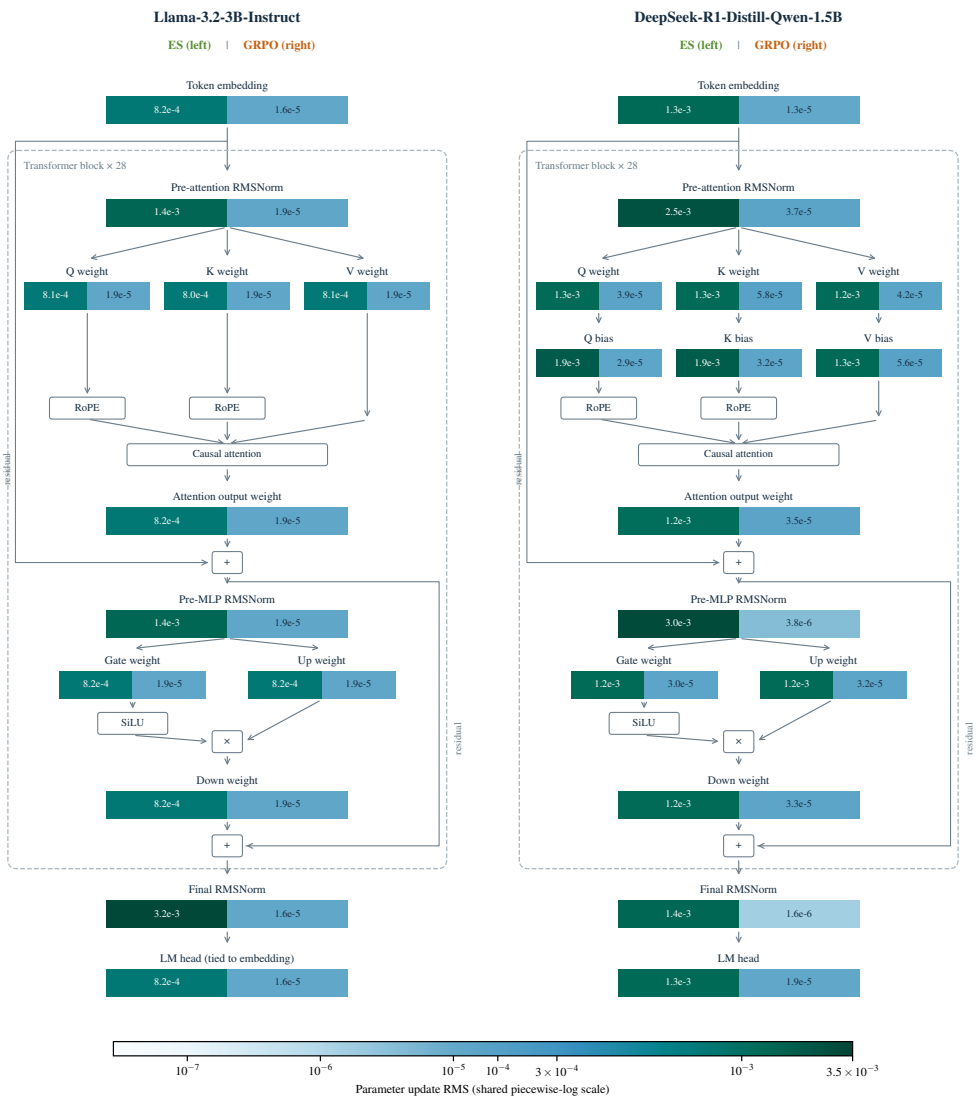


Figure 8: Module-wise parameter update RMS for Llama-3.2-3B-Instruct and DeepSeek-R1-Distill-Qwen-1.5B.

Table 24: Cross-layer update consistency across 28 layers. r is module update RMS divided by its layer's whole-block update RMS. Bold underlining marks patterns discussed in the text.

	ES		GRPO	
Module	Layers $r > 1$	Range of r	Layers $r > 1$	Range of r
Qwen2.5-1.5B-Instruct				
Pre-attention RMSNorm	**28/28**	1.79–3.77	23/28	0.95–1.27
Pre-MLP RMSNorm	**28/28**	2.81–4.00	20/28	0.94–1.12
Q bias	**28/28**	1.79–2.51	21/28	0.96–1.40
K bias	**28/28**	1.48–2.29	14/28	0.81–1.45
V bias	18/28	0.81–1.32	**27/28**	0.97–2.05
K weight	9/28	0.95–1.01	20/28	0.95–1.12
V weight	15/28	0.97–1.02	23/28	0.97–1.29
Llama-3.2-3B-Instruct				
Pre-attention RMSNorm	**27/28**	1.00–2.70	19/28	0.96–1.08
Pre-MLP RMSNorm	**24/28**	0.78–3.29	16/28	0.96–1.03
K weight	0/28	0.95–0.99	6/28	0.91–1.03
V weight	0/28	0.96–1.00	15/28	0.96–1.05
Qwen2.5-7B-Instruct				
Pre-attention RMSNorm	**28/28**	1.24–3.41	20/28	0.97–1.41
Pre-MLP RMSNorm	**28/28**	1.82–3.84	**26/28**	0.99–1.35
Q bias	**28/28**	1.77–2.46	15/28	0.92–1.40
K bias	**28/28**	1.51–2.39	2/28	0.82–1.15
V bias	16/28	0.92–1.43	14/28	0.88–1.39
K weight	3/28	0.96–1.02	**26/28**	0.95–1.21
V weight	1/28	0.98–1.03	20/28	0.99–1.30
DeepSeek-R1-Distill-Qwen-1.5B				
Pre-attention RMSNorm	**27/28**	0.98–2.62	1/28	0.00–3.80
Pre-MLP RMSNorm	**28/28**	1.04–3.75	0/28	0.00–0.36
Q bias	**28/28**	1.32–1.86	3/28	0.32–2.03
K bias	**28/28**	1.24–2.22	3/28	0.31–2.39
V bias	17/28	0.85–1.73	**26/28**	0.78–2.40
K weight	18/28	0.94–1.06	**26/28**	0.95–4.52
V weight	13/28	0.95–1.04	**28/28**	1.02–1.81

Maximum absolute changes Table 25 shows that ES has larger maximum absolute parameter changes than GRPO in every listed module across all four models. For example, the Pre-attention RMSNorm maximum in Llama-3.2-3B-Instruct is 1.172×10^{-2} under ES and 8.368×10^{-5} under GRPO. These extrema complement the module update RMS values shown in the architecture diagrams. Maximum values can be shared by many coordinates: in DeepSeek-R1-Distill-Qwen-1.5B, 929 GRPO K-weight parameters attain the same module maximum. Reporting the full tie count avoids characterizing extreme updates from an arbitrary subset of equal-valued coordinates.

Table 25: Module-wise maximum $|\Delta\theta|$ across 28 layers. Ties count all coordinates attaining each module maximum.

	ES		GRPO	
Module	Max $\|\Delta\|$	Ties	Max $\|\Delta\|$	Ties
Qwen2.5-1.5B-Instruct				
Pre-attention RMSNorm	1.562e-2	9	1.057e-4	1
Pre-MLP RMSNorm	1.562e-2	30	1.585e-4	1
Q bias	1.562e-2	6	1.154e-4	1
K bias	1.172e-2	1	9.260e-5	1
V bias	7.812e-3	4	1.397e-4	1
K weight	7.812e-3	3	1.224e-4	1
V weight	6.836e-3	2	1.612e-4	1
Llama-3.2-3B-Instruct				
Pre-attention RMSNorm	1.172e-2	5	8.368e-5	1
Pre-MLP RMSNorm	1.172e-2	1	9.608e-5	1
K weight	1.172e-2	1	1.174e-4	1
V weight	5.859e-3	5	1.135e-4	1
Qwen2.5-7B-Instruct				
Pre-attention RMSNorm	1.562e-2	28	8.926e-5	1
Pre-MLP RMSNorm	1.562e-2	75	1.018e-4	1
Q bias	1.562e-2	14	9.254e-5	1
K bias	1.562e-2	1	8.958e-5	1
V bias	1.562e-2	1	7.665e-5	1
K weight	6.348e-3	1	1.171e-4	1
V weight	7.812e-3	1	1.150e-4	1
DeepSeek-R1-Distill-Qwen-1.5B				
Pre-attention RMSNorm	1.562e-2	35	9.766e-4	4
Pre-MLP RMSNorm	1.562e-2	82	4.883e-4	1
Q bias	1.562e-2	19	9.766e-4	14
K bias	1.562e-2	5	4.883e-4	13
V bias	7.812e-3	17	4.883e-4	1
K weight	8.545e-3	1	9.766e-4	929
V weight	8.057e-3	1	4.883e-4	209

Whole-block variation and depth trends. We order the 28 Transformer blocks from input to output. For block RMS values R_ℓ, the cross-layer coefficient of variation is $\mathrm{std}(R_\ell)/\,\mathrm{mean}(R_\ell)$, using population standard deviation. We also report the maximum/minimum ratio, Spearman correlation ρ between layer index and RMS, and $100(\bar{R}_{\mathrm{last}}/\bar{R}_{\mathrm{first}} - 1)$, where the two means cover the last and first seven layers. Correlation describes the ordering of magnitudes, while the quarter-mean change quantifies the size of the depth trend. These are descriptive endpoint statistics. Layers are not treated as independent experimental repetitions.

Table 26 shows less whole-block variation under ES than GRPO for every model. ES quarter-mean changes range from -0.55% to $+1.74\%$, with no common decrease across models. Qwen2.5-1.5B-Instruct is nearly flat, Llama-3.2-3B-Instruct and Qwen2.5-7B-Instruct tend upward, and DeepSeek-R1-Distill-Qwen-1.5B decreases slightly. In the latter model, the corresponding GRPO decrease is substantially larger.

Table 26: Whole-block update variation and depth trends. CV and quarter-mean changes are percentages.

Method	Layers	Max/min	CV	ρ	Quarter change
Qwen2.5-1.5B-Instruct					
ES	28	1.0081	0.233	+0.106	-0.036
GRPO	28	1.1753	4.113	+0.036	-0.478
Llama-3.2-3B-Instruct					
ES	28	1.0079	0.169	+0.674	+0.264
GRPO	28	1.0949	2.156	+0.196	+2.405
Qwen2.5-7B-Instruct					
ES	28	1.0624	1.263	+0.354	+1.737
GRPO	28	1.4010	6.551	+0.147	+6.412
DeepSeek-R1-Distill-Qwen-1.5B					
ES	28	1.0169	0.345	-0.385	-0.548
GRPO	28	1.7150	11.779	-0.905	-19.009

Module-specific depth trends. Tables 27 and 28 report the same depth statistics separately for every within-block module type. Their change columns report percentage changes in quarter-mean RMS. Pre-attention RMSNorm ES changes increase with depth in all four models, with last-quarter means approximately 21–62% above first-quarter means. Pre-MLP RMSNorm also has higher last-quarter means, but Llama-3.2-3B-Instruct has only weak rank correlation ($\rho = 0.037$), illustrating a non-monotonic profile. Attention and MLP projections do not share a common decreasing trend. For example, DeepSeek-R1-Distill-Qwen-1.5B ES MLP up RMS decreases by about 1.90%, while K-projection RMS increases by about 6.25%. The GRPO endpoint for this model has lower last-quarter means for all 12 module types, whereas the other models exhibit mixed trends. Whole-block RMS can remain nearly constant because large projection matrices dominate the coordinate-weighted aggregate even when smaller normalization modules vary substantially.

Table 27: Module depth statistics for Qwen2.5-1.5B-Instruct and Llama-3.2-3B-Instruct.

Module	ES ρ	ES change	GRPO ρ	GRPO change
Qwen2.5-1.5B-Instruct				
Pre-attention RMSNorm	+0.412	+20.64	+0.593	+11.90
Pre-MLP RMSNorm	+0.356	+9.86	+0.531	+9.47
Q projection	+0.105	+0.16	+0.299	+4.60
K projection	+0.819	+2.21	+0.344	+4.09
V projection	-0.643	-1.90	+0.716	+12.89
O projection	-0.787	-1.18	+0.194	+2.02
MLP gate projection	+0.750	+1.42	-0.089	-1.89
MLP up projection	-0.697	-0.80	+0.027	-0.84
MLP down projection	-0.396	-0.56	+0.060	-0.56
Q bias	-0.488	-12.20	+0.620	+14.98
K bias	-0.112	-3.48	+0.678	+29.91
V bias	-0.012	+4.95	+0.889	+52.82
Llama-3.2-3B-Instruct				
Pre-attention RMSNorm	+0.812	+37.51	-0.117	+0.17
Pre-MLP RMSNorm	+0.037	+15.81	+0.176	+2.65
Q projection	+0.746	+1.35	+0.125	+4.43
K projection	+0.959	+2.02	-0.004	+2.37
V projection	+0.796	+2.04	-0.163	-0.46
O projection	+0.887	+1.42	-0.205	-0.90
MLP gate projection	-0.396	-0.27	+0.792	+4.86
MLP up projection	+0.331	+0.02	+0.229	+2.63
MLP down projection	-0.331	-0.20	-0.081	+0.66

Table 28: Module depth statistics for Qwen2.5-7B-Instruct and DeepSeek-R1-Distill-Qwen-1.5B.

Module	ES ρ	ES change	GRPO ρ	GRPO change
Qwen2.5-7B-Instruct				
Pre-attention RMSNorm	+0.825	+61.84	-0.164	-0.86
Pre-MLP RMSNorm	+0.437	+10.87	-0.196	-2.37
Q projection	-0.487	-0.36	+0.295	+6.30
K projection	-0.300	-0.97	+0.150	+2.43
V projection	+0.106	+1.17	-0.211	-1.75
O projection	+0.409	+0.78	-0.191	-1.61
MLP gate projection	+0.066	+1.78	+0.278	+10.85
MLP up projection	+0.815	+2.25	+0.097	+6.35
MLP down projection	+0.573	+1.83	+0.011	+4.57
Q bias	-0.065	-4.05	-0.274	-2.46
K bias	-0.054	-3.55	-0.010	+0.46
V bias	+0.400	+15.38	-0.574	-6.01
DeepSeek-R1-Distill-Qwen-1.5B				
Pre-attention RMSNorm	+0.507	+51.32	-0.542	-96.91
Pre-MLP RMSNorm	+0.632	+50.18	-0.448	-74.05
Q projection	+0.710	+1.58	-0.456	-36.11
K projection	+0.924	+6.24	-0.066	-51.73
V projection	-0.828	-4.27	-0.742	-37.26
O projection	-0.773	-3.06	-0.943	-39.46
MLP gate projection	+0.545	+1.44	-0.049	-2.87
MLP up projection	-0.982	-1.90	-0.935	-17.48
MLP down projection	-0.786	-1.20	-0.919	-14.35
Q bias	-0.488	-9.48	-0.482	-65.74
K bias	-0.383	-11.33	-0.771	-69.81
V bias	+0.180	+15.47	-0.783	-48.75

Complete module profiles. Figures 9–12 show every within-block module type in each model. Each curve is divided by its own across-layer mean, using the same normalization as Figure 4a. Panel scales vary to reveal module-specific shapes. These normalized curves cannot compare absolute ES and GRPO magnitudes, which are shown in the architecture diagrams. Non-block embedding, LM-head, and final-normalization parameters have no layer sequence and are excluded.

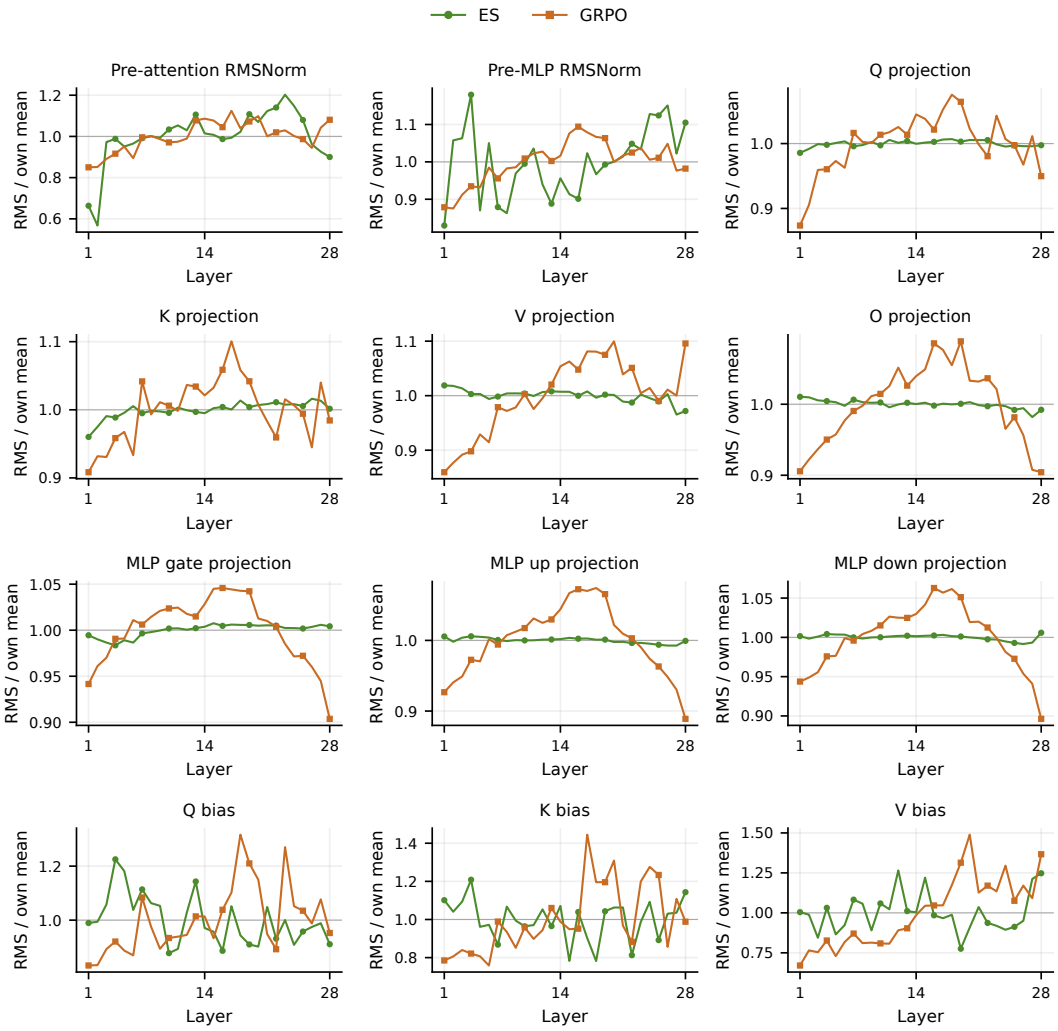


Figure 9: Module depth profiles for Qwen2.5-1.5B-Instruct.

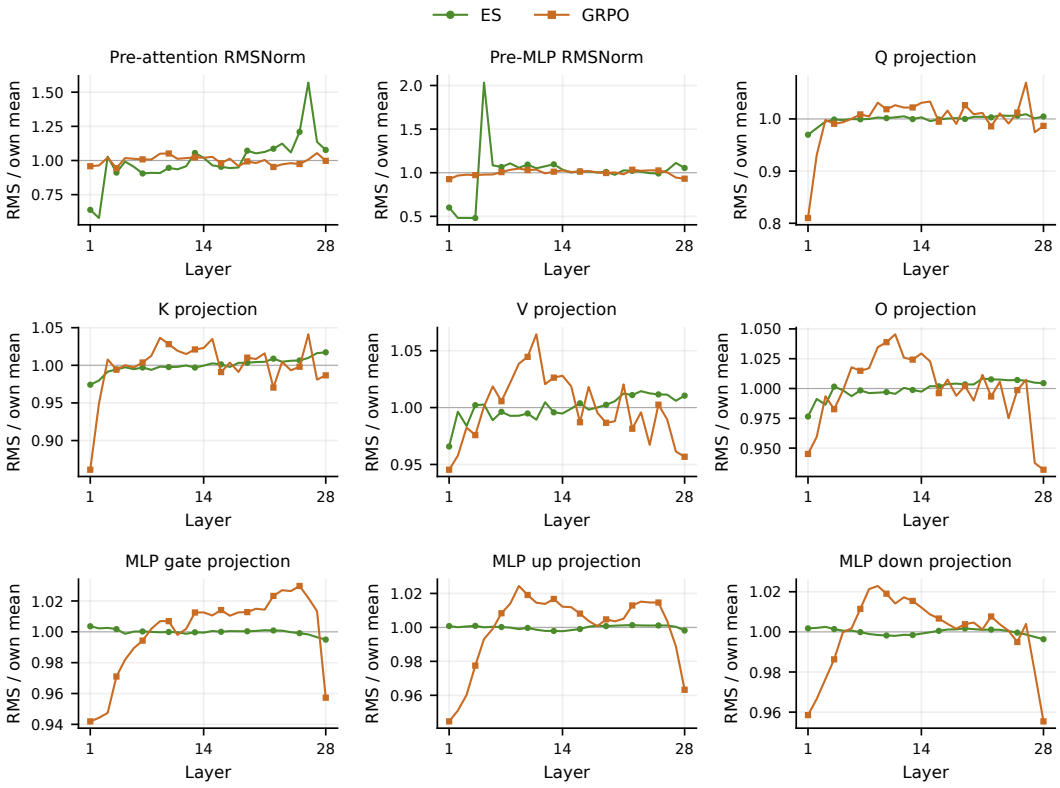


Figure 10: Module depth profiles for Llama-3.2-3B-Instruct.

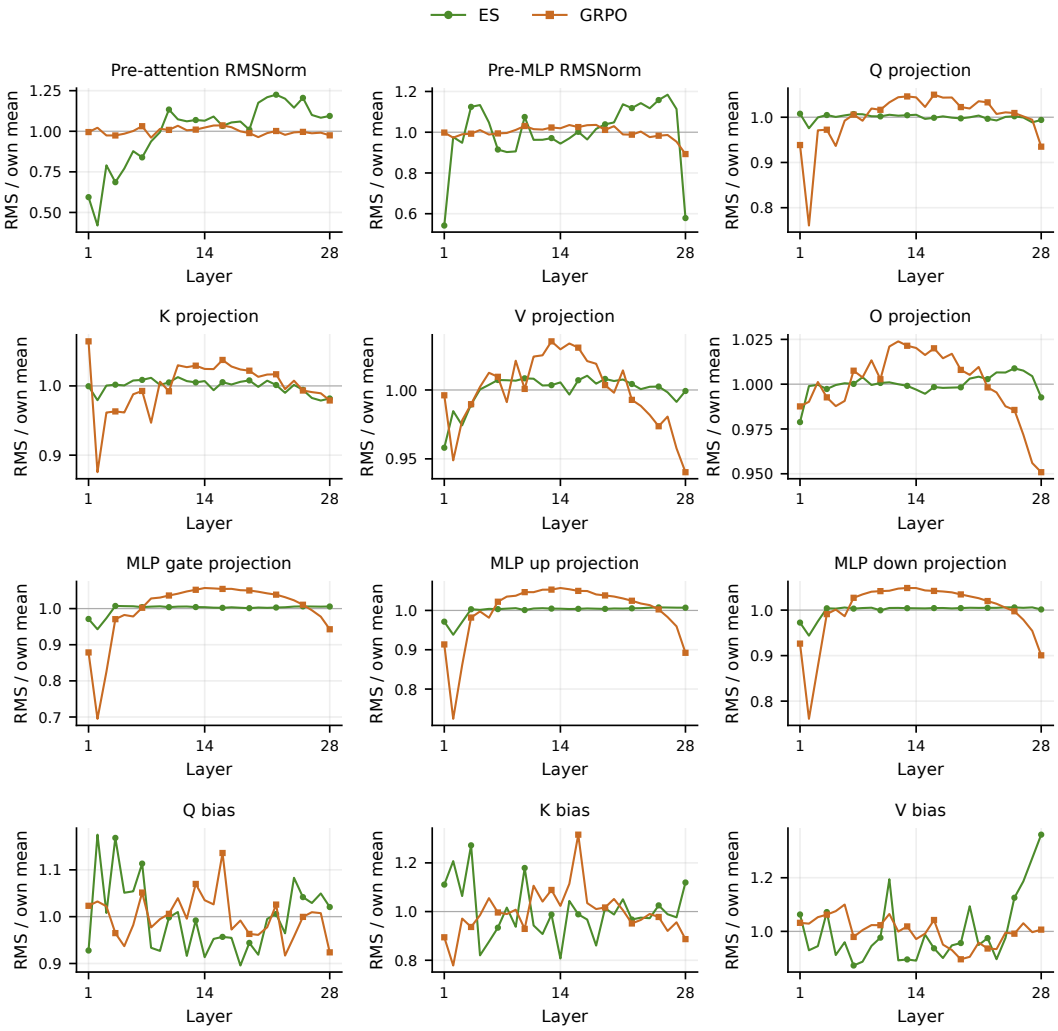


Figure 11: Module depth profiles for Qwen2.5-7B-Instruct.

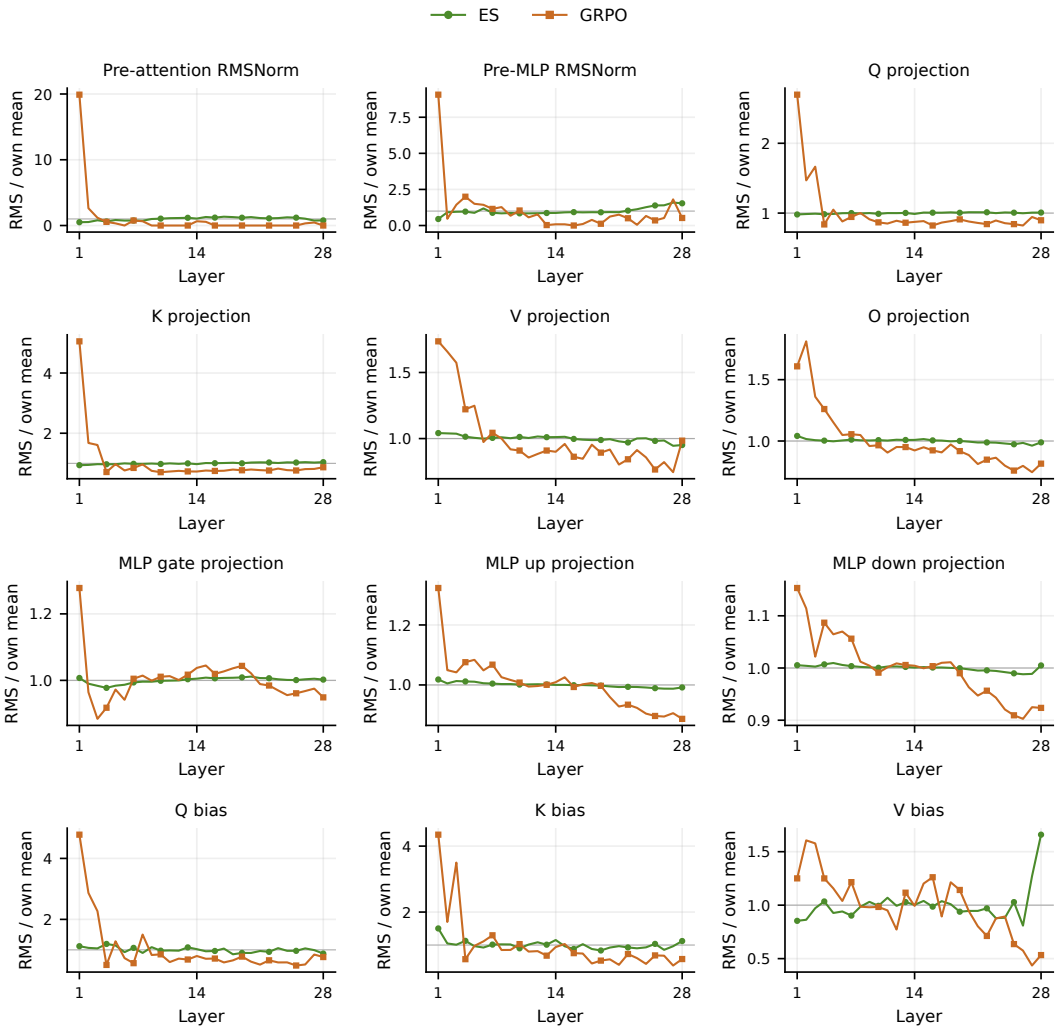


Figure 12: Module depth profiles for DeepSeek-R1-Distill-Qwen-1.5B.

H.2 Task Performance After Magnitude Thresholding

This subsection reports the complete target-task evaluation results for magnitude-thresholded ES. For a threshold τ, coordinates satisfying $0 < |\Delta\theta_i| \leq \tau$ are set to zero, equivalently resetting the corresponding parameters to their Base Model values, and update sparsity is computed over nonzero ES changes according to the definition in Section 4.2. Coordinates with zero change are excluded from both the sparsity denominator and the ablated set. For this thresholding experiment in both settings, Pass@1 is the empirical single-sample accuracy estimated from 32 retained responses per problem. The Δ vs. Full update column reports the signed difference in Pass@1 from the ES-trained model evaluated under the same settings, measured in percentage points.

Table 29: GSM8K Pass@1 for magnitude-thresholded ES in the Easy Setting.

Threshold τ	Update sparsity (%)	Pass@1 (%)	Δ vs. Full update (pp)
Qwen2.5-1.5B-Instruct			
Full update	0.00	73.199	+0.000
2.5×10^{-4}	27.82	73.462	+0.263
5.0×10^{-4}	50.77	73.079	−0.121
1.0×10^{-3}	79.11	73.045	−0.154
1.5×10^{-3}	92.47	72.458	−0.742
2.0×10^{-3}	97.57	71.138	−2.061
Base Model	100.00	70.271	–
Llama-3.2-3B-Instruct			
Full update	0.00	81.631	+0.000
2.5×10^{-4}	25.27	81.762	+0.130
5.0×10^{-4}	47.74	81.440	−0.192
1.0×10^{-3}	78.08	81.695	+0.064
1.5×10^{-3}	92.64	78.433	−3.198
2.0×10^{-3}	97.89	77.883	−3.748
Base Model	100.00	77.670	–
Qwen2.5-7B-Instruct			
Full update	0.00	91.016	+0.000
2.5×10^{-4}	25.43	91.087	+0.071
5.0×10^{-4}	47.51	91.028	+0.012
1.0×10^{-3}	78.41	91.504	+0.488
1.5×10^{-3}	93.02	91.736	+0.720
2.0×10^{-3}	98.11	91.824	+0.808
Base Model	100.00	91.774	–

Table 30: MATH-500 Pass@1 for magnitude-thresholded ES with DeepSeek-R1-Distill-Qwen-1.5B.

Threshold τ	Update sparsity (%)	Pass@1 (%)	Δ vs. Full update (pp)
Full update	0.00	83.206	+0.000
2.5×10^{-4}	17.80	82.756	−0.450
5.0×10^{-4}	38.32	83.125	−0.081
1.0×10^{-3}	62.18	83.037	−0.169
1.5×10^{-3}	77.61	82.806	−0.400
2.0×10^{-3}	87.29	82.438	−0.769
2.5×10^{-3}	93.16	82.225	−0.981
3.0×10^{-3}	96.64	81.544	−1.663
Base Model	100.00	80.856	–

At 79.11%, 78.08%, and 77.61% update sparsity, Qwen2.5-1.5B-Instruct, Llama-3.2-3B-Instruct, and DeepSeek-R1-Distill-Qwen-1.5B change by -0.154, $+0.064$, and -0.400 percentage points relative to their corresponding ES-trained models, respectively. At higher sparsity, Qwen2.5-1.5B-Instruct and Llama-3.2-3B-Instruct degrade more sharply, whereas DeepSeek-R1-Distill-Qwen-1.5B declines more gradually. Qwen2.5-7B-Instruct improves across its tested range.

H.3 Random Coordinate Truncation and Displacement Norm Matching

These experiments examine whether the effectiveness of magnitude thresholding arises from the coordinates selected, the amount of displacement preserved, or both. We use Qwen2.5-1.5B-Instruct and Llama-3.2-3B-Instruct on GSM8K, and DeepSeek-R1-Distill-Qwen-1.5B on MATH-500, with the same base-model and ES-trained-model checkpoints and evaluation settings as the thresholding experiments above. Each condition uses 32 responses per problem. The complete matrix comprises 71 evaluations: three evaluations of the ES-trained models and four modifications at each of 17 model–threshold combinations.

Update construction. Let $\Delta = \theta_{\mathrm{ES}} - \theta_0$, let $S = \{i : \Delta_i \neq 0\}$, and define $T_\tau = \{i : |\Delta_i| > \tau\}$. A uniformly sampled subset $R_\tau \subseteq S$ has exactly $|T_\tau|$ coordinates. The random subsets are nested across thresholds, and the two random-truncation conditions share the same subset at each threshold. We compare

$$\begin{aligned} &\text{A:}\quad \Delta_A = \Delta, && \text{B:}\quad \Delta_B = \mathbf{1}_{T_\tau} \odot \Delta, \\ &\text{C:}\quad \Delta_C = \mathbf{1}_{R_\tau} \odot \Delta, && \text{D:}\quad \Delta_D = \alpha \Delta_C, \\ &\text{E:}\quad \Delta_E = \beta \Delta. \end{aligned}$$

Here A retains the full parameter update of the ES-trained model, B is magnitude thresholding, C is random coordinate truncation, D rescales C to B's displacement norm, and E rescales the full update to B's norm. Before accounting for rounding, $\alpha = \|\Delta_B\|_2 / \|\Delta_C\|_2$ and $\beta = \|\Delta_B\|_2 / \|\Delta\|_2$. Each modified model is evaluated at $\theta_0 + \Delta_g$. Sparsity is measured over S, with shared parameters counted once.

Core results. Table 31 reports the thresholds at which magnitude thresholding removes approximately 78–79% of nonzero updates. At these thresholds, B preserves 83.54%, 83.20%, and 86.26% of the original displacement norm for Qwen2.5-1.5B-Instruct, Llama-3.2-3B-Instruct, and DeepSeek-R1-Distill-Qwen-1.5B, respectively, whereas C preserves only 45.71%, 46.82%, and 47.31%. Thus, B–C comparisons alone cannot separate coordinate selection from retained displacement norm.

Table 31: Target-task Pass@1 (%) at the core thresholds. Qwen2.5-1.5B-Instruct and Llama-3.2-3B-Instruct use $\tau = 0.001$; DeepSeek-R1-Distill-Qwen-1.5B uses $\tau = 0.0015$. For each of B, C, and D, the removed fractions are 79.11% for Qwen2.5-1.5B-Instruct, 78.08% for Llama-3.2-3B-Instruct, and 77.61% for DeepSeek-R1-Distill-Qwen-1.5B. D and E target B's displacement norm; E does not impose coordinate truncation.

Update modification	Qwen2.5-1.5B-Instruct	Llama-3.2-3B-Instruct	DeepSeek-R1-Distill-Qwen-1.5B
A: Full update	73.199	81.631	83.206
B: Magnitude thresholding	73.045	81.695	82.806
C: Random coordinate truncation	72.318	78.435	81.906
D: Random truncation + norm matching	72.574	80.009	81.412
E: Full-update scaling	73.233	81.764	83.088

Magnitude-thresholded models achieve Pass@1 close to that of the corresponding ES-trained models and outperform randomly truncated models for all three models at the core thresholds (Table 32). For Llama-3.2-3B-Instruct, rescaling the random update improves Pass@1 by 1.573 points, yet magnitude thresholding remains 1.687 points higher; both intervals exclude zero. This provides evidence for effects of both displacement magnitude and its distribution across coordinates. For DeepSeek-R1-Distill-Qwen-1.5B, B exceeds D by 1.394 points, while D–C is inconclusive. For Qwen2.5-1.5B-Instruct, B–D is inconclusive. The evidence that displacement magnitude matters therefore comes primarily from Llama-3.2-3B-Instruct, whereas the evidence for differences beyond the norm comes from Llama-3.2-3B-Instruct and DeepSeek-R1-Distill-Qwen-1.5B. These comparisons do not independently isolate coordinate identity, update direction, and the allocation of amplitudes.

Table 32: Core-threshold Pass@1 differences in percentage points and paired 95% confidence intervals. Intervals use 10,000 bootstrap resamples of questions, preserving pairing across conditions. They are pointwise intervals without adjustment for multiple comparisons.

Model	Comparison	Difference	95% interval
Qwen2.5-1.5B-Instruct	B–A	−0.154	[−0.628, +0.315]
Qwen2.5-1.5B-Instruct	B–C	+0.727	[+0.251, +1.206]
Qwen2.5-1.5B-Instruct	B–D	+0.471	[−0.019, +0.960]
Qwen2.5-1.5B-Instruct	D–C	+0.256	[−0.194, +0.699]
Qwen2.5-1.5B-Instruct	B–E	−0.187	[−0.604, +0.228]
Llama-3.2-3B-Instruct	B–A	+0.064	[−0.313, +0.448]
Llama-3.2-3B-Instruct	B–C	+3.260	[+2.566, +3.957]
Llama-3.2-3B-Instruct	B–D	+1.687	[+1.106, +2.260]
Llama-3.2-3B-Instruct	D–C	+1.573	[+1.038, +2.132]
Llama-3.2-3B-Instruct	B–E	−0.069	[−0.422, +0.296]
DeepSeek-R1-Distill-Qwen-1.5B	B–A	−0.400	[−0.963, +0.163]
DeepSeek-R1-Distill-Qwen-1.5B	B–C	+0.900	[+0.237, +1.562]
DeepSeek-R1-Distill-Qwen-1.5B	B–D	+1.394	[+0.750, +2.044]
DeepSeek-R1-Distill-Qwen-1.5B	D–C	−0.494	[−1.113, +0.112]
DeepSeek-R1-Distill-Qwen-1.5B	B–E	−0.281	[−0.838, +0.269]

Full-update scaling also yields Pass@1 close to that of the corresponding ES-trained models at the core thresholds; B–E intervals include zero for all three models. At more aggressive truncation thresholds, E generally has higher point estimates than B (Figure 13). Consequently, magnitude thresholding provides evidence about task-relevant information in endpoint updates, without identifying a unique effective learning subspace or establishing that small updates have no role during training.

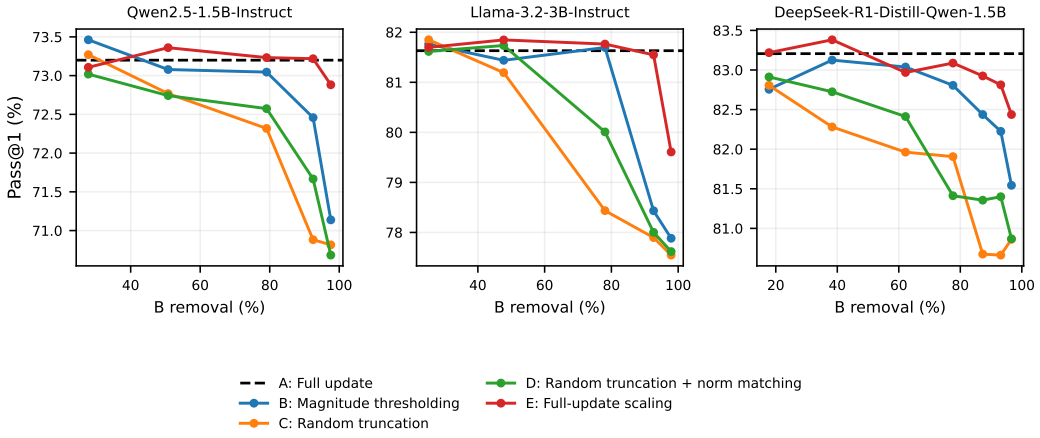


Figure 13: Target-task Pass@1 across all evaluated thresholds. The horizontal axis is the fraction of nonzero coordinates removed by B; C and D remove the same fraction. E uses B's corresponding norm budget, so its horizontal position does not indicate its own sparsity. Each panel has an independent vertical scale.

Additional sampling metrics. Table 33 reports Pass@16 and Pass@32 from the same responses. Core-threshold B is close to A for Qwen2.5-1.5B-Instruct and Llama-3.2-3B-Instruct; for DeepSeek-R1-Distill-Qwen-1.5B, B is lower by 0.717 and 0.800 points, respectively. The ordering under Pass@1 does not consistently carry over to Pass@K. These are descriptive point estimates; the paired intervals above concern Pass@1. No held-out evaluation of these modified endpoints is included.

Table 33: Pass@16 and Pass@32 (%) at the core thresholds, estimated from 32 responses per problem. Group definitions match Table 31.

Model	Group	Pass@16	Pass@32
Qwen2.5-1.5B-Instruct	A	94.896	96.588
Qwen2.5-1.5B-Instruct	B	94.960	96.816
Qwen2.5-1.5B-Instruct	C	95.248	96.740
Qwen2.5-1.5B-Instruct	D	94.976	96.740
Qwen2.5-1.5B-Instruct	E	94.993	96.664
Llama-3.2-3B-Instruct	A	96.381	97.877
Llama-3.2-3B-Instruct	B	96.613	97.726
Llama-3.2-3B-Instruct	C	96.679	97.953
Llama-3.2-3B-Instruct	D	96.396	97.574
Llama-3.2-3B-Instruct	E	96.116	97.422
DeepSeek-R1-Distill-Qwen-1.5B	A	97.188	98.200
DeepSeek-R1-Distill-Qwen-1.5B	B	96.471	97.400
DeepSeek-R1-Distill-Qwen-1.5B	C	96.120	97.200
DeepSeek-R1-Distill-Qwen-1.5B	D	96.191	97.400
DeepSeek-R1-Distill-Qwen-1.5B	E	96.399	97.200

Precision and achieved norm matching. We convert BF16 endpoint weights to FP32 for subtraction, scaling, and addition, then save the modified weights in BF16. We measure norms from the saved displacement by accumulating squared differences in FP64. The theoretical coefficients are calibrated using only weight statistics, with up to 18 bisection steps, to target a relative norm error of at most 0.1%. Because BF16 rounding yields discontinuous changes, 27 of the 34 D/E conditions meet this tolerance; the remaining seven are approximate matches (Table 34). They are the closest values found by this calibration procedure, without a claim of global optimality over all scaling coefficients. All three core D conditions meet the tolerance. These endpoint calculations do not test numerical restoration errors during ES training.

Table 34: Conditions exceeding the 0.1% norm-matching tolerance. Signed error is $100(\|\widehat{\Delta}_g\|_2/\|\Delta_B\|_2 - 1)$, where $\widehat{\Delta}_g$ is the displacement after BF16 saving.

Model	Group	Threshold τ	Relative norm error (%)
Qwen2.5-1.5B-Instruct	D	0.00025	$+0.3330$
Qwen2.5-1.5B-Instruct	D	0.0005	-0.1412
Qwen2.5-1.5B-Instruct	E	0.001	$+0.3412$
Llama-3.2-3B-Instruct	E	0.001	-0.3068
DeepSeek-R1-Distill-Qwen-1.5B	D	0.0005	$+0.1531$
DeepSeek-R1-Distill-Qwen-1.5B	E	0.002	-0.1518
DeepSeek-R1-Distill-Qwen-1.5B	E	0.003	-0.2599

H.4 Spectral Structure After Magnitude Thresholding

Scope and thresholding. We analyze the saved ES parameter changes in the four models used in RQ2. For each weight matrix, we subtract saved weights in FP32 and remove coordinates at or below an absolute threshold:

$$D = \mathrm{FP32}(W_{\mathrm{ES}}) - \mathrm{FP32}(W_{\mathrm{Base}}), \qquad (D_\tau)_{ij} = D_{ij}\mathbf{1}\{|D_{ij}| > \tau\}. \tag{35}$$

The lower/higher thresholds are $0.001/0.0015$ for the three Instruct models and $0.0015/0.0025$ for DeepSeek-R1-Distill-Qwen-1.5B. These settings come from the existing magnitude-thresholded performance evaluations. The spectral analysis masks coordinates in the difference matrix and introduces no further checkpoint rounding. It covers all 790 native two-dimensional matrices, including attention and MLP projections, embeddings, and independent LM heads. Tied embeddings and heads are counted once. Normalization and bias vectors are excluded from matrix-rank statistics.

Numerical rank and spectral concentration. For an $m \times n$ matrix, singular values σ_i are obtained from FP64 eigenvalues of the smaller Gram matrix, with tiny negative eigenvalues clamped to zero. We count numerical rank using the archived FP32-input tolerance:

$$r_{\mathrm{num}} = \#\{i : \sigma_i > \sigma_{\max} \max(m, n)\epsilon_{32}\}, \qquad \epsilon_{32} = 2^{-23}. \tag{36}$$

The rank fraction is $r_{\text{num}}/\min(m,n)$. For singular values in descending order, we also measure the fewest directions retaining 90% of squared displacement:

$$r_{90} = \min\left\{k : \sum_{i=1}^{k} \sigma_i^2 \geq 0.9 \sum_i \sigma_i^2\right\}. \tag{37}$$

Each setting uses its own squared displacement as the denominator. Zero matrices have rank and r_{90} equal to zero. Thus r_{90} quantifies spectral concentration of the parameter difference; its relationship to performance retention requires separate evaluation.

Results across thresholds. Table 35 connects coordinate retention with matrix rank. At the lower thresholds, only 20.89–22.39% of the originally nonzero coordinate updates remain. At the higher thresholds, this decreases to 6.84–7.53%. All 790 matrices remain full-rank or nearly full-rank in the unthresholded and both thresholded settings, with numerical-rank fractions of at least 99.87% across these settings. Coordinate retention and squared-displacement fractions cover all parameter coordinates, including vectors, whereas the rank ranges cover only the two-dimensional matrices.

Table 35: Retained coordinates, squared displacement, and numerical rank before and after thresholding. All fractions are percentages.

Setting	Coordinates	Squared displacement	Matrices	Rank fraction
Qwen2.5-1.5B-Instruct				
Unthresholded	100.00	100.00	197	99.93–100.00
$\tau = 0.001$	20.89	69.78	197	99.93–100.00
$\tau = 0.0015$	7.53	41.30	197	99.93–100.00
Llama-3.2-3B-Instruct				
Unthresholded	100.00	100.00	197	99.90–100.00
$\tau = 0.001$	21.92	69.22	197	99.90–100.00
$\tau = 0.0015$	7.36	38.28	197	99.90–100.00
Qwen2.5-7B-Instruct				
Unthresholded	100.00	100.00	198	99.92–100.00
$\tau = 0.001$	21.59	68.49	198	99.92–100.00
$\tau = 0.0015$	6.98	36.65	198	99.92–100.00
DeepSeek-R1-Distill-Qwen-1.5B				
Unthresholded	100.00	100.00	198	99.87–100.00
$\tau = 0.0015$	22.39	74.40	198	99.93–100.00
$\tau = 0.0025$	6.84	40.61	198	99.87–100.00

Tables 36 and 37 give complete module-wise rank and r_{90} ranges. Both quantities are expressed as percentages of $\min(m,n)$, with ranges spanning the matrices of each module type. U, L, and H denote unthresholded updates and the lower and higher thresholds defined above. For example, Qwen2.5-1.5B-Instruct MLP up matrices have maximum rank 1536. Their r_{90} is 1221 without thresholding, 1220–1221 at $\tau = 0.001$, and 1216–1220 at $\tau = 0.0015$. The other models likewise retain broad spectra after small-coordinate removal. The observed coordinate sparsity therefore does not produce an accompanying low-rank structure at these thresholds.

Table 36: Module-wise numerical rank and r_{90} fractions for Qwen2.5-1.5B-Instruct and Llama-3.2-3B-Instruct.

	Numerical rank			r_{90}		
Module	U	L	H	U	L	H
Qwen2.5-1.5B-Instruct						
Q projection	99.93–100.00	99.93–100.00	99.93–100.00	50.91–50.98	50.78–50.98	50.52–50.91
K projection	100.00	100.00	100.00	79.69–80.08	79.30–79.69	78.52–79.69
V projection	100.00	100.00	100.00	79.69–80.08	79.69	79.30–79.69
O projection	99.93–100.00	99.93–100.00	99.93–100.00	50.91–51.04	50.91–50.98	50.72–50.91
MLP gate projection	100.00	100.00	100.00	79.49	79.43–79.49	79.17–79.43
MLP up projection	100.00	100.00	100.00	79.49	79.43–79.49	79.17–79.43
MLP down projection	100.00	100.00	100.00	79.49	79.43–79.49	79.36–79.43
Embedding	100.00	100.00	100.00	88.15	88.15	88.15
Llama-3.2-3B-Instruct						
Q projection	99.93–100.00	99.90–99.97	99.90–99.97	50.85–51.01	50.65–50.98	49.93–50.91
K projection	100.00	100.00	100.00	72.66–72.75	72.46–72.75	71.88–72.66
V projection	100.00	100.00	100.00	72.66–72.75	72.36–72.75	71.58–72.66
O projection	99.90–99.97	99.90–99.97	99.93–99.97	50.94–51.01	50.91–50.98	50.68–50.91
MLP gate projection	100.00	100.00	100.00	71.06–71.09	71.06–71.09	70.96–71.06
MLP up projection	100.00	100.00	100.00	71.06–71.13	71.06–71.09	70.96–71.06
MLP down projection	100.00	100.00	100.00	71.06–71.09	71.03–71.09	70.83–71.06
Embedding	100.00	100.00	100.00	86.95	86.95	87.01

Table 37: Module-wise numerical rank and r_{90} fractions for Qwen2.5-7B-Instruct and DeepSeek-R1-Distill-Qwen-1.5B.

	Numerical rank			r_{90}		
Module	U	L	H	U	L	H
Qwen2.5-7B-Instruct						
Q projection	99.92–99.97	99.92–99.97	99.92–99.97	50.75–50.98	50.39–50.95	49.39–50.86
K projection	100.00	100.00	100.00	80.47–80.86	80.08–80.86	78.52–80.47
V projection	100.00	100.00	100.00	80.66–80.86	80.66–80.86	80.27–80.66
O projection	99.92–99.97	99.92–99.97	99.92–99.97	50.92–51.00	50.84–50.98	50.47–50.89
MLP gate projection	100.00	100.00	100.00	78.52–78.71	78.21–78.71	77.34–78.68
MLP up projection	100.00	100.00	100.00	78.54–78.71	78.24–78.71	77.43–78.68
MLP down projection	100.00	100.00	100.00	78.54–78.71	78.26–78.71	77.54–78.63
Embedding	100.00	100.00	100.00	87.05	87.05	87.03
LM head	100.00	100.00	100.00	87.03	86.97	86.75
DeepSeek-R1-Distill-Qwen-1.5B						
Q projection	99.87–100.00	99.93–100.00	99.87–100.00	50.78–50.91	50.78–50.91	50.72–50.85
K projection	100.00	100.00	100.00	79.30–79.69	79.30–79.69	79.30–79.69
V projection	100.00	100.00	100.00	79.30–80.08	79.30–79.69	79.30–79.69
O projection	99.93–100.00	99.93–100.00	99.93–100.00	50.85–50.98	50.78–50.98	50.78–50.91
MLP gate projection	100.00	100.00	100.00	79.49	79.43–79.49	79.43–79.49
MLP up projection	100.00	100.00	100.00	79.49	79.43–79.49	79.43–79.49
MLP down projection	100.00	100.00	100.00	79.43–79.49	79.36–79.49	79.43–79.49
Embedding	100.00	100.00	100.00	87.30	87.24	87.70
LM head	100.00	100.00	100.00	87.11	86.98	87.57

Numerical checks and interpretation. Table 38 reports the minimum rank fractions under alternative relative tolerances, counting singular values above $10^{-6}\sigma_{\max}$ or $10^{-4}\sigma_{\max}$ where recorded. These checks supplement numerical rank and do not constitute an exact algebraic-rank claim. The source analysis also checks one attention-output matrix per model using direct FP64 SVD and verifies thresholded coordinate counts and squared displacement against an independent scan. The milder-threshold checkpoints have target-task performance close to that of the corresponding ES-trained models, as reported above, while stronger thresholds degrade some models. No performance evaluation of low-rank replacements is available. Consequently, these spectra do not determine how many matrix directions are necessary to preserve performance.

Table 38: Minimum numerical-rank fractions under alternative relative singular-value tolerances, in percent. L and H use the thresholds in Table 35.

	U (10^{-6})	L (10^{-6})	L (10^{-4})	H (10^{-6})	H (10^{-4})
Qwen2.5-1.5B-Instruct					
Minimum rank fraction	100.00	100.00	99.93	100.00	99.93
Llama-3.2-3B-Instruct					
Minimum rank fraction	100.00	100.00	99.97	100.00	99.93
Qwen2.5-7B-Instruct					
Minimum rank fraction	100.00	100.00	99.97	100.00	99.97
DeepSeek-R1-Distill-Qwen-1.5B					
Minimum rank fraction	100.00	99.93	99.93	100.00	99.87

I Detailed Held-Out Performance Changes

Table 39 reports changes in Pass@1 and Maj@K, averaged equally over all held-out benchmarks: CommonsenseQA, HotpotQA, Countdown, GPQA Diamond, and MBPP in the Easy Setting, and GPQA Diamond, MBPP, CommonsenseQA, and Countdown in the Hard Setting. Table 40 reports changes in Pass@16 and Pass@32, averaged equally over HotpotQA, Countdown, and MBPP in the Easy Setting, and Countdown and MBPP in the Hard Setting. All changes are post-training minus Base, in percentage points. P@K and M@K denote Pass@K and Maj@K, respectively. Table 41 provides the per-task Pass@K results on MBPP and Countdown. The corresponding CommonsenseQA and GPQA Diamond results appear in Appendix E.

Table 39: Held-out Pass@1 and majority-vote accuracy changes, in percentage points. Averages cover all held-out benchmarks in each setting.

Method	ΔP@1	ΔM@16	ΔM@32
Easy: Qwen2.5-1.5B-Instruct			
GRPO	+0.72	+1.31	+1.52
ES	+0.02	+1.02	+1.96
Easy: Llama-3.2-3B-Instruct			
GRPO	+2.26	+1.29	+0.56
ES	+1.43	−0.01	−0.03
Easy: Qwen2.5-7B-Instruct			
GRPO	+0.90	−0.47	−0.41
ES	−0.45	+0.30	+0.65
Hard: DeepSeek-R1-Distill-Qwen-1.5B			
GRPO	+2.77	−0.11	−0.31
ES	+1.15	+1.61	+1.47

Table 40: Held-out Pass@16 and Pass@32 changes, in percentage points. Averages cover HotpotQA, Countdown, and MBPP in the Easy Setting, and Countdown and MBPP in the Hard Setting.

Method	ΔP@16	ΔP@32
Easy: Qwen2.5-1.5B-Instruct		
GRPO	−0.30	−0.51
ES	+0.34	+0.46
Easy: Llama-3.2-3B-Instruct		
GRPO	−0.74	−1.16
ES	+2.00	+1.87
Easy: Qwen2.5-7B-Instruct		
GRPO	−0.99	−0.86
ES	+0.21	+0.43
Hard: DeepSeek-R1-Distill-Qwen-1.5B		
GRPO	−0.49	−1.37
ES	−0.09	−0.88

Table 41: Pass@K results (×100) on MBPP and Countdown in the one-epoch Hard Setting. Average gives equal weight to the two benchmarks.

	MBPP			Countdown			Average		
Method	P@1	P@16	P@32	P@1	P@16	P@32	P@1	P@16	P@32
DeepSeek-R1-Distill-Qwen-1.5B									
Base	62.9	85.6	87.5	37.5	79.9	85.7	50.2	82.7	86.6
GRPO	66.6	87.0	88.3	36.6	77.5	82.2	51.6	82.3	85.3
ES	63.5	85.9	87.2	38.5	79.4	84.4	51.0	82.7	85.8
ES→GRPO	64.6	86.6	87.9	38.8	80.6	86.8	**51.7**	**83.6**	**87.4**
GRPO→ES	64.2	85.2	87.2	38.1	79.9	85.3	51.2	82.5	86.2

J Population-Size Scaling Trajectories

Figure 14 reports the complete per-update mean-reward trajectories summarized by Table 6.

For Qwen2.5-0.5B-Instruct, rewards decline over later updates with $N = 8$ and $N = 16$, whereas they continue improving with $N = 32$ and $N = 64$. For Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct, the $N = 16$, $N = 32$, and $N = 64$ curves remain substantially closer throughout training.

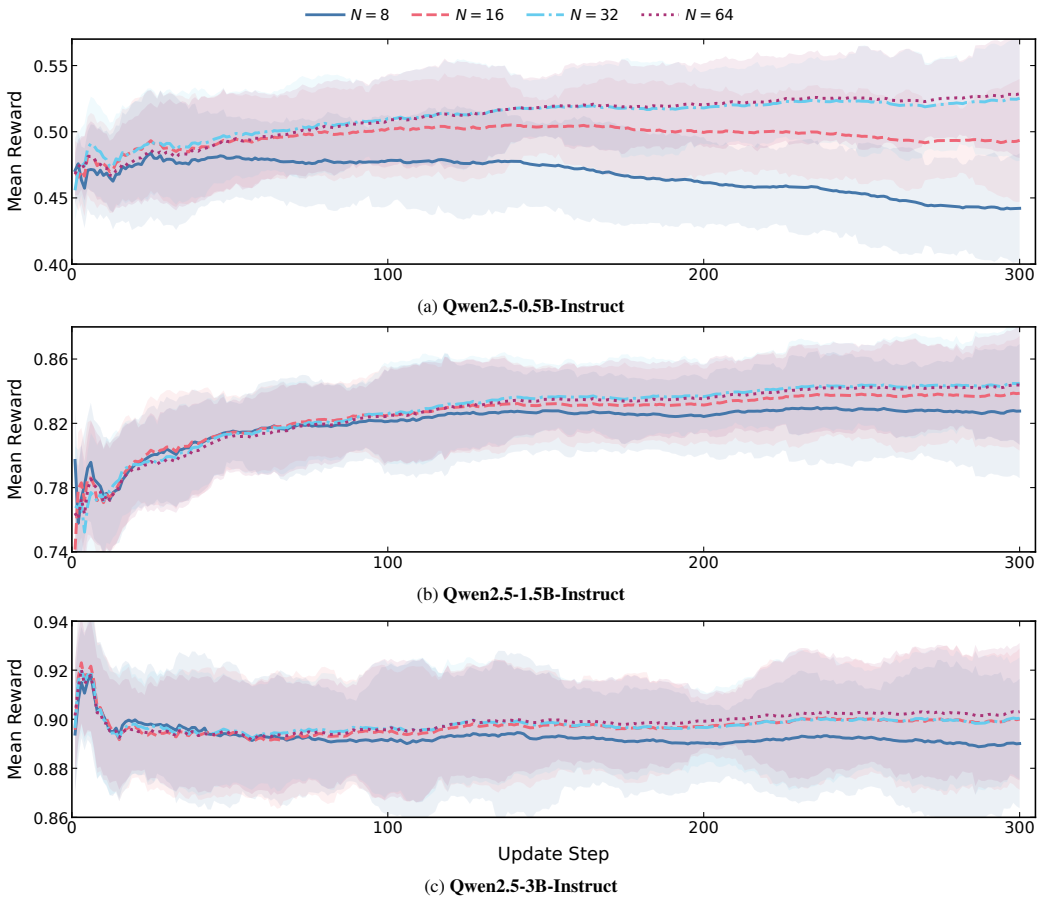


Figure 14: GSM8K reward trajectories by population size. Bands show ± 1 local standard deviation of raw-minus-smoothed residuals within each trajectory. Y-axis ranges differ across models.

K ES–ZO Estimator Equivalence and Its Finite-Sample Boundary

This appendix examines why two-point estimation may offer no advantage over one-point ES in reasoning training. We first show how ES and zeroth-order optimization estimate the gradient of the same Gaussian-smoothed objective (Nesterov & Spokoiny, 2017; Salimans et al., 2017). We then examine when pairing positive and negative perturbations reduces variance and test whether this benefit persists when reasoning responses are regenerated.

K.1 A Unified Smoothed-Objective View

For a measurable scalar objective F, we use a maximization convention throughout. For reward maximization, we take $F(\theta) = R(\theta)$. For loss minimization, we take $F(\theta) = -\mathcal{L}(\theta)$, so that all estimators below are written in ascent form. Assume that some neighborhood U of θ satisfies $\sup_{\theta' \in U} \mathbb{E}_\epsilon[|F(\theta' + \sigma\epsilon)|(1 + \|\epsilon\|)] < \infty$, and define its Gaussian smoothing as

$$F_\sigma(\theta) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \left[F(\theta + \sigma\epsilon) \right]. \tag{38}$$

The Gaussian score-function identity gives

$$\nabla F_\sigma(\theta) = \frac{1}{\sigma} \mathbb{E}_\epsilon \left[F(\theta + \sigma\epsilon)\epsilon \right]. \tag{39}$$

The resulting one-point estimator over N directions is

$$\widehat{g}^{1p} = \frac{1}{N\sigma} \sum_{i=1}^{N} F(\theta + \sigma\epsilon_i)\epsilon_i, \qquad \mathbb{E}\big[\widehat{g}^{1p}\big] = \nabla F_\sigma(\theta). \tag{40}$$

The multi-direction two-point estimator is

$$\widehat{g}^{2p} = \frac{1}{2N\sigma} \sum_{i=1}^{N} \left[F(\theta + \sigma\epsilon_i) - F(\theta - \sigma\epsilon_i) \right] \epsilon_i. \tag{41}$$

Because ϵ and $-\epsilon$ have the same distribution,

$$\mathbb{E}_\epsilon \left[F(\theta - \sigma\epsilon)\epsilon \right] = -\mathbb{E}_\epsilon \left[F(\theta + \sigma\epsilon)\epsilon \right]. \tag{42}$$

Substitution into Equation (41) yields

$$\mathbb{E}\big[\widehat{g}^{2p}\big] = \nabla F_\sigma(\theta) = \mathbb{E}\big[\widehat{g}^{1p}\big]. \tag{43}$$

If ES uses the same positive–negative perturbation pairs, its antithetic estimator is exactly Equation (41). Thus, antithetic ES and two-point ZO are the same estimator when they share the objective, perturbation distribution, scale, and raw paired-difference rule. This identity does not imply equal finite-budget variance because one pair consumes two function evaluations, whereas one one-point sample consumes one. Thus, N counts perturbation directions in both estimators, while their perturbed-evaluation counts are $N_{\text{eval}} = N$ and $N_{\text{eval}} = 2N$, respectively.

K.2 Covariance Required by Paired Subtraction

Suppose an objective evaluation is stochastic because it includes prompt sampling, autoregressive generation, or verifier noise. Let ξ collect this evaluation randomness and let $\widehat{F}(\vartheta; \xi)$ be an unbiased scalar evaluation of the objective, such that

$$\mathbb{E}_\xi \left[\widehat{F}(\vartheta; \xi) \right] = F(\vartheta) \tag{44}$$

for every evaluated parameter point ϑ. For fixed $(\theta, \epsilon, \sigma)$, the repeated evaluations are

$$\begin{aligned} X_+ &= \widehat{F}(\theta + \sigma\epsilon; \xi_+), \\ X_0 &= \widehat{F}(\theta; \xi_0), \\ X_- &= \widehat{F}(\theta - \sigma\epsilon; \xi_-). \end{aligned} \tag{45}$$

Table 42: Local subtraction diagnostic at the Qwen2.5-1.5B-Instruct base checkpoint. Raw scalar variance ratios κ_{base} and κ_{pair} (Equation (49)) below 1 indicate variance reduction.

(a) Objective comparison at $\sigma = 10^{-3}$

Objective	Evaluation coupling	$\text{Corr}(X_+, X_0)$	κ_{base}	$\text{Corr}(X_+, X_-)$	κ_{pair}
GSM8K regenerated-rollout reward	Common seed	0.2777	1.4445	0.1304	1.9870
GSM8K regenerated-rollout reward	Independent seed	0.0561	1.3577	0.0224	2.7847
SST-2 supervised CE	Fixed batch	0.9776	0.0447	0.9196	0.1557
Mean token log-probability	Fixed rollout	0.9939	0.0125	0.9857	0.0285
Policy sequence loss	Fixed rollout	0.9939	0.0167	0.9903	0.0194
GRPO surrogate	Fixed rollout	0.9998	0.0006	0.9997	0.0007

(b) Perturbation-scale sweep for GSM8K regenerated-rollout reward

σ	Seed coupling	$\text{Corr}(X_+, X_0)$	κ_{base}	$\text{Corr}(X_+, X_-)$	κ_{pair}
10^{-4}	Common	0.8797	0.2288	0.5537	0.8194
10^{-4}	Independent	-0.0245	1.4080	0.0450	2.3822
3×10^{-4}	Common	0.5607	1.0424	0.3026	1.9434
3×10^{-4}	Independent	-0.0303	1.6687	0.0532	2.4379
10^{-3}	Common	0.2777	1.4445	0.1304	1.9870
10^{-3}	Independent	0.0561	1.3577	0.0224	2.7847

The random variables (ξ_+, ξ_0, ξ_-) may be coupled through common random numbers or sampled independently. Their raw difference variances satisfy

$$\text{Var}(X_+ - X_0) = \text{Var}(X_+) + \text{Var}(X_0) - 2\,\text{Cov}(X_+, X_0), \tag{46}$$

$$\text{Var}(X_+ - X_-) = \text{Var}(X_+) + \text{Var}(X_-) - 2\,\text{Cov}(X_+, X_-). \tag{47}$$

In particular, $\text{Var}(X_+ - X_-) < \text{Var}(X_+)$ if and only if

$$\text{Cov}(X_+, X_-) > \frac{1}{2}\,\text{Var}(X_-). \tag{48}$$

We report the correlations and the raw variance ratios

$$\kappa_{\text{base}} = \frac{\text{Var}(X_+ - X_0)}{\text{Var}(X_+)}, \qquad \kappa_{\text{pair}} = \frac{\text{Var}(X_+ - X_-)}{\text{Var}(X_+)}. \tag{49}$$

A ratio below one means that subtraction reduces the variance of the scalar objective values in this diagnostic.

K.3 Local Perturbation Diagnostic

Protocol. We conduct a local diagnostic around the Qwen2.5-1.5B-Instruct base checkpoint. For each sampled full-parameter direction, we evaluate Equation (45) at several perturbation scales. The reported statistics pool scalar observations across prompts, generation seeds, and directions.

Objectives. We compare three evaluation protocols. First, *GSM8K regenerated-rollout reward* regenerates a response separately at each of the positive, center, and negative parameter points and assigns a terminal correctness reward. Second, *SST-2 supervised CE* evaluates a differentiable cross-entropy objective on the same fixed batch at all three parameter points. Third, *GSM8K fixed rollout* generates responses once at the base checkpoint, freezes them, and recomputes mean token log-probability, policy sequence loss, and a GRPO surrogate at the perturbed points. Fixed rollouts isolate rescoring from autoregressive generation noise.

Randomness Coupling. For regenerated rollouts, *common* uses the same generation seed at the three parameter points, whereas *independent* uses different seeds. Common seeds implement common random numbers, but they do not guarantee identical autoregressive trajectories after a parameter perturbation changes an early token distribution.

K.4 Diagnostic Results

Table 42(a) compares the three objective classes at $\sigma = 10^{-3}$. The repeated-direction regenerated-rollout run uses 16 prompts, 16 generation seeds per prompt, and 4 perturbation directions. The

fixed-rollout run uses 16 prompts, 8 seeds per prompt, and 4 directions. The supervised control contains 128 pooled scalar observations.

GSM8K regenerated-rollout rewards are weakly correlated across the positive and negative perturbations at this scale, and paired subtraction increases their raw scalar variance. The same perturbation implementation yields strong correlations and ratios below one for supervised CE and all fixed-rollout objectives. This contrast suggests that regenerating responses and evaluating terminal rewards contribute to the weaker cancellation of evaluation fluctuations. The very small fixed-rollout GRPO ratio further shows that a GRPO surrogate is not intrinsically incompatible with paired ZO evaluation when the responses are held fixed.

Table 42(b) shows that coupling regenerated-rollout reward evaluations with common random numbers is scale dependent. At $\sigma = 10^{-4}$, common seeds produce $\kappa_{\mathrm{pair}} = 0.8194$, whereas the corresponding ratio with independent seeds remains above one. At the two larger scales, paired subtraction does not reduce raw variance under either seed mode.

These local results complement the training comparison under matched evaluation budgets and reward normalization in Figure 1(c), item (3).

L Related Work

L.1 Evolution Strategies for LLM Reasoning Post-Training

Evolution strategies (ES) are population-based black-box optimizers that estimate parameter updates from forward-evaluated perturbations and naturally distribute candidate evaluations across workers (Salimans et al., 2017). Recent work has adapted this paradigm to LLM reasoning post-training. Qiu et al. demonstrate that full-parameter ES can operate directly in billion-dimensional parameter spaces and optimize LLMs with outcome-level rewards (Qiu et al., 2026). ESSA restricts evolutionary search to low-rank adapters and further reduces the search space by optimizing their singular values, enabling quantized forward-only alignment on mathematical reasoning and instruction-following tasks (Korotyshova et al., 2025). EGGROLL structures perturbations as low-rank matrices to improve the arithmetic intensity of population evaluation and applies the resulting system to reasoning post-training at large population sizes (Sarkar et al., 2026). ESSAM combines full-parameter ES with a sharpness-aware objective and evaluates this design on GSM8K, emphasizing generalization and memory-efficient reasoning fine-tuning (Sun et al., 2026). These methods establish several viable implementations of ES for LLM post-training, spanning direct full-parameter search, parameter-efficient search, and structured perturbations.

A related line of work asks whether the geometry of ES updates predicts capability retention. Abdi et al. report that ES updates have larger norms and affect a larger fraction of parameter elements than GRPO updates (Abdi et al., 2026). They associate these update properties with held-out degradation. However, their forgetting analysis uses a small training set, one model, one training task, and one held-out benchmark, limiting the scope of conclusions about broad capability retention. Hoy et al. provide complementary evidence that larger parameter movement does not necessarily lead to consistent behavioral degradation: even when ES and GRPO attain similar target-task accuracy, ES travels much farther in weight space, while the resulting accuracy changes vary across held-out benchmarks and ES iteration budgets (Hoy et al., 2026). Schweighofer et al. (2026) mitigate ES forgetting observed when training on particular tasks by increasing the population size or applying anchored weight decay. Taken together, these studies show that large parameter displacement can accompany capability degradation. The predictive value of weight-space distance for forgetting, and the mechanisms linking parameter drift to capability degradation, remain open questions.

ESMA studies ES-based metacognitive alignment in factual question answering and uses magnitude-based weight patching to show that a subset of larger updates accounts for much of the observed improvement (Park et al., 2026). We adopt a similar approach to evaluate target-task accuracy after removing small updates in reasoning post-training, and additionally analyze parameter displacement and the spectral structure of retained updates.

L.2 Reinforcement Learning with Verifiable Rewards

Reinforcement learning with verifiable rewards (RLVR) improves reasoning models using automatically checked outcome rewards from domains such as mathematics and code. Group Relative Policy Optimization (GRPO) removes the learned critic by normalizing rewards within a group of responses and optimizing a clipped token-level policy objective (Shao et al., 2024). Although this recipe can improve single-sample accuracy, recent work identifies policy-entropy collapse as a recurring limitation of RLVR. The entropy-mechanism analysis relates entropy change to the covariance between action log-probabilities and logit updates, and observes sustained entropy decrease during reasoning training (Cui et al., 2025). Subsequent studies identify contributions from positive-advantage tokens (Jin et al., 2026) and imbalanced token-level entropy flows (Xu et al., 2026), examine how clipping configurations and repeated off-policy updates affect entropy dynamics (Petrenko et al., 2026; Jin et al., 2026), and trace collapse to premature overconfidence at a small set of structurally important decision points (Jang et al., 2026).

A second line of work evaluates coverage behaviorally through large-budget sampling and asks whether RLVR expands the set of correct reasoning paths accessible from the base model. Large-budget sampling studies find that RLVR can improve Pass@1 while underperforming the base model at larger K, indicating more efficient exploitation of high-probability correct paths but weaker coverage of less likely solutions (Yue et al., 2025). Controlled pretraining experiments further show that RL post-training tends to amplify behaviors already represented in the pretraining distribution (Zhao

et al., 2025). Support-based analyses similarly find that RLVR may shrink access to some correct solutions even as it raises single-attempt accuracy (Wu et al., 2025). These findings motivate jointly evaluating Pass@1 and Pass@K to assess single-sample accuracy and reasoning coverage.

L.3 Zeroth-Order Optimization for LLMs

Zeroth-order (ZO) optimization replaces backpropagated derivatives with a gradient-shaped estimate constructed solely from scalar objective values. Given a Gaussian direction $u \sim \mathcal{N}(0, I)$ and perturbation radius μ, two forward evaluations first estimate the directional derivative as $\delta_u f(\theta) = [f(\theta + \mu u) - f(\theta - \mu u)]/(2\mu)$, then map this scalar back to parameter space as the pseudo-gradient $\widehat{g}_{\mathrm{ZO}}(\theta; u) = \delta_u f(\theta) u$. Averaging these pseudo-gradients across directions estimates the gradient of a Gaussian-smoothed objective, enabling parameter updates without differentiating through the model (Nesterov & Spokoiny, 2017). MeZO adapts simultaneous perturbation-based ZO optimization to operate in place, reducing memory to approximately the inference footprint and supporting both full-parameter and parameter-efficient language-model fine-tuning (Malladi et al., 2023). Variance-reduced extensions such as MeZO-SVRG improve the stability and convergence of forward-only language-model fine-tuning by periodically constructing lower-variance update estimates (Gautam et al., 2024). ZO-Bench broadens the comparison across model families, task types, and fine-tuning schemes, and studies blockwise descent, hybrid optimization, forward-gradient estimators, and gradient sparsity (Zhang et al., 2024).

ES and ZO estimator families overlap. With the same objective, perturbation distribution, scale, and raw positive–negative difference, antithetic ES is algebraically identical to a two-point ZO estimator. A relevant distinction is the evaluation protocol: fixed-data or fixed-rollout objectives can be rescored at perturbed parameters, while regenerated-rollout objectives require new responses at each parameter point and may introduce additional generation noise. As analyzed in Appendix K, resampling responses at the perturbed parameter points can weaken the covariance required for paired subtraction to reduce variance. The relative performance of one- and two-point estimators therefore requires empirical evaluation under rollout rewards.