BEYOND POLICY ROLLOUTS: EMBEDDING-SPACE DIRECTIONAL EXPLORATION WITH ANCHORED NOISE FOR LLM RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) typically uses small rollout groups per prompt, where repeated sampling from the same policy can yield highly similar trajectories, reducing within-group reward variation and weakening group-relative advantage signals. Existing remedies such as higher sampling temperature, entropy bonuses, asymmetric clipping, and larger rollout budgets broaden sampling or modify policy updates but leave rollout conditioning unchanged. In this paper, we propose **EDEN** (*Embedding-space Directional Exploration via anchored Noise*), which intervenes at the *rollout source* through anchored input perturbations. EDEN explores beyond the policy's own rollouts. Within the same rollout budget, it replaces a small fraction of standard rollouts in each group with rollouts that the same policy generates from the prompt plus a single appended, perturbed embedding position, rather than from the prompt alone. Gaussian noise drives variation, while a shared anchor provides a common conditioning direction within each prompt group. Selected perturbed rollouts contribute to GRPO updates with downweighted advantages, and the model architecture, discrete decoding, and test-time inference remain unchanged. Across model scales and mathematical reasoning benchmarks, EDEN outperforms GRPO and the exploration controls evaluated in the same pipeline, with the largest gains in sampled solution coverage. Analyses show that more prompt groups provide non-zero group-relative learning signals and that noise and anchoring play complementary roles. Intervening at the rollout source is thus an effective lever for exploration in RLVR.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become the standard recipe for post-training reasoning LLMs, instantiated in algorithms such as PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024; Guo et al., 2025), and DAPO (Yu et al., 2025). Most of them share a group-rollout backbone: for each prompt, the policy samples G trajectories whose verifiable rewards drive the update, making rollout quality a direct constraint on learning. Rollout generation is expensive, so typical training uses small groups of 8–16 rollouts per prompt. At this budget, repeated sampling from the same policy can yield highly similar trajectories, and the gradient sees a narrow slice of the solution space (Wachi et al., 2026). The key constraint is therefore not rollout count alone, but the amount of useful variation within the rollout pool.

This rollout homogeneity can cause the group-relative advantage signal that GRPO (Shao et al., 2024) relies on to collapse. When all rollouts in a group receive the same reward, their advantages vanish, motivating dynamic-sampling filters in recent RLVR systems (Yu et al., 2025). Over training, policy entropy can collapse within a few hundred steps (Cui et al., 2025), and pass@k can flatten or even regress below the base model for large k (Yue et al., 2025; Wen et al., 2026; Yang et al., 2025; Wachi et al., 2026). These observations highlight exploration as a shared bottleneck: the rollout pool needs more diversity than the converging policy alone can supply (Zhu et al., 2025a), making exploration a central research focus in RLVR.

Existing exploration methods for RLVR address this through different leverage points. Higher sampling temperatures (Dang et al., 2026; Zhang et al., 2025b) broaden token-level sampling by

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.

flattening the output distribution, while entropy bonuses, asymmetric clipping (Clip-Higher) (Yu et al., 2025), and entropy-collapse regularizers such as Clip-Cov / KL-Cov (Cui et al., 2025) aim to prevent premature concentration or relax policy updates. These methods encourage exploration by modifying the training objective or the sampling distribution, while leaving rollout conditioning unchanged. Other methods apply *selection* (Lin et al., 2025; Zheng et al., 2025a; Yu et al., 2025), pruning low-information trajectories or skipping groups with uniform rewards. A distinct set of approaches directly alters rollout generation. Search-based methods branch from intermediate states (Xing et al., 2026), whereas perturbation-based methods modify soft-token representations, policy parameters, and logits, or repurpose the noise inherent in quantised execution (Butt et al., 2026; Bai et al., 2026; Huang et al., 2025; Zheng et al., 2025b). These works suggest that the locus of exploration matters, since changing the conditions under which trajectories are generated can alter the effective trajectory distribution beyond simply drawing more rollouts from the same distribution. Can a minimal input-embedding intervention enable useful trajectory-level exploration while retaining standard discrete decoding and leaving test-time inference unchanged?

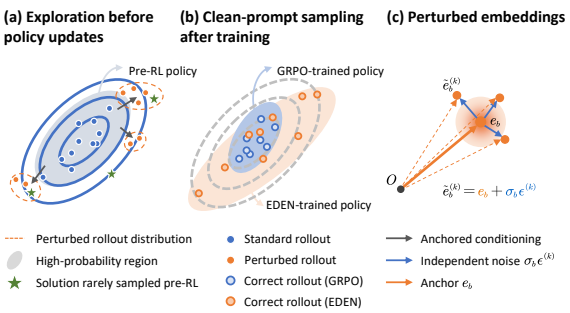


Figure 1: EDEN explores beyond the policy's rollouts. (a) Anchored conditioning yields rollouts from a shifted proposal. (b) After training, EDEN achieves broader coverage of correct trajectories under clean-prompt sampling. (c) Independent Gaussian noise perturbs a shared anchor in embedding space.

In this paper, we propose **EDEN** (*Embedding-space Directional Exploration via anchored Noise*), an RLVR exploration method that intervenes at the rollout source through anchored input perturbations. Without additional rollouts, EDEN explores beyond the policy's own rollouts (Fig. 1) by conditioning the same policy on perturbed inputs for a subset of each group, helping bring trajectories rarely obtained through standard sampling into training. Applied only during training, this conditioning is introduced at a single embedding position immediately after the prompt and before the first generated token, allowing it to influence the entire autoregressive rollout from the outset without repeatedly intervening during decoding. EDEN therefore requires neither policy-parameter perturbations nor specialized decoding, and leaves test-time inference unchanged. The perturbation combines Gaussian noise, which allows rollouts to deviate from the standard generation mode, with an anchor embedding shared by all perturbed rollouts for a prompt. The shared anchor provides directionality by defining a common embedding-space conditioning vector for all perturbed rollouts of a prompt, while independent isotropic Gaussian noise introduces local variation around it. To turn this additional exploration into useful training signals, EDEN retains perturbed trajectories with large absolute advantages and incorporates them into GRPO updates with downweighted advantages. Fig. 2 illustrates the training procedure. EDEN improves over GRPO across three model scales and also outperforms the exploration controls evaluated in the same pipeline. Mechanism analyses further support the effectiveness of input-side exploration. Our main contributions are summarized as follows:

- We propose **EDEN**, a new RLVR exploration method that intervenes at the rollout source through *anchored noise*: a small fraction of rollouts are perturbed via anchored Gaussian noise, and the most informative perturbed rollouts are incorporated into training via an advantage-gated filter and noise-aware advantage reweighting.
- EDEN realises *directional* exploration through the complementary roles of noise and anchoring, with Gaussian noise driving variation and a chosen anchor embedding supplying a shared conditioning direction within each prompt group. The design integrates with GRPO through a single appended embedding position, retains discrete decoding, and leaves the model architecture and inference unchanged.
- Across three model scales and eight mathematical reasoning benchmarks, EDEN improves average Pass@1 and Pass@8 over GRPO, with its advantage persisting under Pass@256 evaluation on AIME. Training dynamics show more prompt groups with nonzero training signals, while controlled ablations support the complementary roles of noise and anchoring.

2 Related Work

RLVR and GRPO foundations. RLVR trains reasoning LLMs with verifiable rewards (Lightman et al., 2024; Cobbe et al., 2021), using algorithms such as PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), and GRPO (Shao et al., 2024; Guo et al., 2025). GRPO replaces PPO's learned value function with a group-relative advantage signal computed across rollouts sampled per prompt, and has become the dominant choice for verifiable-reward LLM post-training. DAPO (Yu et al., 2025) extends GRPO with asymmetric clipping (Clip-Higher), Dynamic Sampling, token-level loss, and overlong reward shaping, which together stabilise large-scale training. Further variants refine advantage estimation, importance ratios, or training schedules while retaining the group-rollout backbone. Our work builds directly on GRPO and is compatible with these refinements.

Exploration approaches in RLVR. Exploration is a central challenge in RLVR, with policy concentration and limited solution coverage motivating a range of approaches (Yue et al., 2025; Wen et al., 2026; Yang et al., 2025; Wachi et al., 2026). Higher sampling temperatures (Dang et al., 2026; Zhang et al., 2025b), entropy bonuses, asymmetric clipping (Yu et al., 2025), and Clip-Cov / KL-Cov (Cui et al., 2025) broaden sampling or counter premature policy concentration. ProRL (Liu et al., 2025) extends training with KL resets to sustain exploration. Other methods reshape the optimisation *objective* itself. Pass-at-k Training (Chen et al., 2025), FlowRL (Zhu et al., 2025b), and Diversity-Aware Policy Optimization (Yao et al., 2025) adjust the reward objective to maintain diversity. These approaches modify sampling or optimization while leaving input conditioning unchanged. EDEN instead intervenes at the *rollout source* through anchored input perturbations.

Rollout methods for LLM RLVR. Some rollout methods improve efficiency by selecting informative rollouts or prompts, as in Dynamic Sampling (Yu et al., 2025), GRESO (Zheng et al., 2025a), CPPO (Lin et al., 2025), and related approaches (Hu et al., 2025; Qu et al., 2026; Tang et al., 2026; Nguyen et al., 2026), or by reusing historical rollouts to amortise generation cost (Huang & Wan, 2026; Zhang et al., 2025a;c; Sun et al., 2025). Other methods promote exploration by intervening before or during generation. Tree search (Xing et al., 2026) branches from intermediate states to diversify trajectories. Soft Tokens (Butt et al., 2026) perturbs input embeddings at each CoT step with soft-token decoding, PSN-GRPO (Bai et al., 2026) perturbs policy weights with truncated importance sampling, QeRL (Huang et al., 2025) exploits quantisation-induced noise, SofT-GRPO (Zheng et al., 2025b) applies Gumbel perturbations to logits with soft decoding, and REFT diversifies discrete first-token choices (Kim & No, 2026). EDEN instead uses a shared anchor to direct continuous perturbations at one appended input embedding for a subset of rollouts.

3 Method

3.1 Preliminaries

RLVR is widely used for post-training on tasks with automatically checkable solutions such as mathematical reasoning. Concretely, given a prompt $q \sim P(Q)$, the policy π_θ generates a response o by autoregressive sampling and receives a verifiable reward. We build on Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a critic-free policy-gradient algorithm that replaces PPO's learned value function with a group-relative baseline. For each prompt q, we sample a group of G rollouts $\{o_i\}_{i=1}^{G}$ from the old policy $\pi_{\theta_{\text{old}}}$ at sampling temperature τ_r and optimize

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left(\ell_{i,t}^{\text{clip}}(\theta) - \beta\,\hat{\mathbb{D}}_{\text{KL}}^{(i,t)}[\pi_\theta \,\|\, \pi_{\text{ref}}]\right)\right], \tag{1}$$

where $\ell_{i,t}^{\text{clip}}(\theta) = \min\big(r_{i,t}A_{i,t},\ \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon)A_{i,t}\big)$ and $r_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$, ϵ controls the clipping range. The Kullback–Leibler regularization term against a frozen reference policy π_{ref} is estimated by the non-negative unbiased form

$$\hat{\mathbb{D}}_{\text{KL}}^{(i,t)}[\pi_\theta \,\|\, \pi_{\text{ref}}] = \frac{\pi_{\text{ref}}(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta(o_{i,t} \mid q, o_{i,<t})} - \log\frac{\pi_{\text{ref}}(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta(o_{i,t} \mid q, o_{i,<t})} - 1. \tag{2}$$

The token-level advantage $A_{i,t} = (r_i - \mu_q)/\sigma_q$ standardizes the per-rollout rewards within the group, where μ_q and σ_q denote the mean and standard deviation of $\{r_j\}_{j=1}^{G}$. The same scalar advantage is shared by all tokens of o_i when computing the token-level loss.

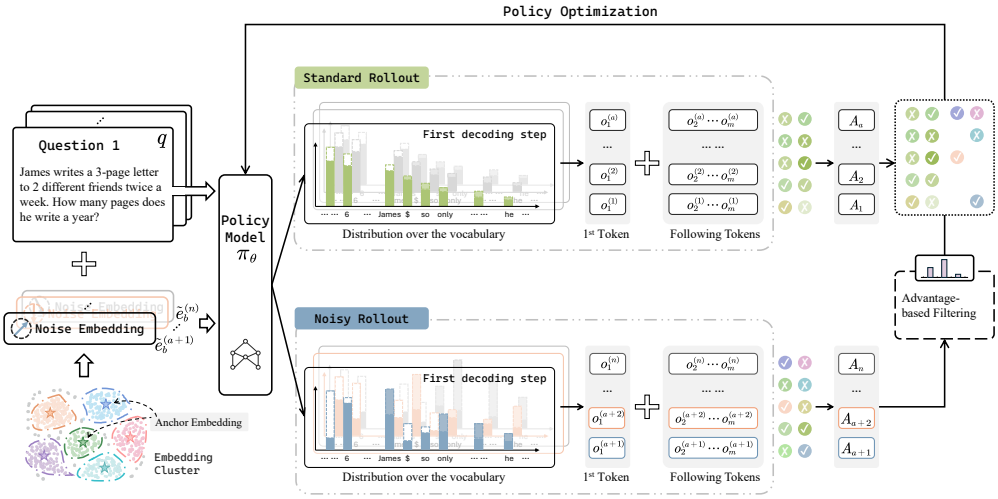


Figure 2: Overview of EDEN. EDEN augments RLVR with a small fraction of noise-perturbed rollouts to broaden trajectory-level exploration without increasing the total rollout budget.

3.2 Embedding-space Directional Exploration via Anchored Noise

EDEN reshapes the rollout distribution through controlled embedding-space perturbations. It has three parts. *Perturb* samples a fraction of each group from the same policy under perturbed input conditioning. *Regulate* controls exploration through a shared vocabulary-token anchor, a norm-relative noise scale, a screened anchor bank, and a temperature schedule. *Absorb* incorporates informative perturbed trajectories into updates of the unperturbed policy through advantage-based selection, advantage scaling, and evaluation under the clean prompt.

3.2.1 Noise-Perturbed Rollout Generation

We generate noise-perturbed rollouts by applying scaled noise to an anchor embedding drawn from a pre-constructed bank. Let $E(q) = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_L]$ denote the prompt embedding sequence. For each prompt, we append one embedding position to the prompt sequence and set it to a perturbed embedding $\tilde{\mathbf{e}}_b$, yielding $\tilde{E}(q) = \left[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_L, \tilde{\mathbf{e}}_b\right]$. Rollouts are then generated as $o \sim \pi_\theta(\cdot \mid \tilde{E}(q))$. This single-position intervention can influence the entire subsequent rollout through autoregressive conditioning, without further perturbations during decoding.

Noise bank construction. To give each prompt group's perturbed rollouts a shared conditioning centre while allowing exploration across different embedding-space directions, EDEN draws anchors from a fixed noise bank $\mathcal{B}$ of K vocabulary-token embeddings. The bank is constructed once before RL training and remains fixed throughout training. Anchors are drawn from the model's own vocabulary and screened for generation viability and complementary directions, aiming to induce non-trivial variation without substantially disrupting generation. Appendix C details the construction and screening procedure using 200 training-split prompts. A single fixed anchor already improves over GRPO, with the bank providing further gains mainly in greedy Pass@1 (Tables 4 and 6). For each prompt in each training batch, one anchor is sampled uniformly from $\mathcal{B}$ and shared across its K_n perturbed rollouts. Independent anchor sampling across prompts and batches allows exploration around different conditioning centres rather than a single fixed anchor.

Embedding-Space Perturbation. Given the shared anchor embedding $\mathbf{e}_b \in \mathbb{R}^d$, we independently sample Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for each perturbed rollout and construct $\tilde{\mathbf{e}}_b = \mathbf{e}_b + \rho \frac{\|\mathbf{e}_b\|_2}{\sqrt{d}} \boldsymbol{\epsilon}$, where ρ controls the relative perturbation scale. This yields per-dimension standard deviation $\sigma_b = \rho \|\mathbf{e}_b\|_2 / \sqrt{d}$ and root-mean-square perturbation magnitude $\sqrt{\mathbb{E}[\|\tilde{\mathbf{e}}_b - \mathbf{e}_b\|_2^2]} = \rho \|\mathbf{e}_b\|_2$. Thus, ρ scales perturbations relative to the anchor norm, stabilizing magnitudes across anchors and model sizes. The Gaussian component is isotropic and has no preferred direction. In high dimensions, the

scaled noise is nearly orthogonal to $\mathbf{e}_b$, with norm close to $\rho\|\mathbf{e}_b\|_2$, so the perturbed embedding forms an angle of approximately $\arctan\rho$ with the anchor. *Directional* therefore refers to two properties of the anchor. Within a prompt group, all K_n perturbed rollouts share one anchor and hence one displacement of the conditioning, with local variation from Gaussian noise (Table 4). Distinct anchor directions in the bank allow groups to explore different displacements (Table 6).

Exploration Control. We control exploration strength using a temperature schedule $\tau_n(s)$ that increases from τ_0 to τ^* over training, $\tau_n(s) = \tau_0 + (\tau^* - \tau_0)\frac{1}{2}\big(1 - \cos(\pi \cdot \mathrm{clip}(s/T_w, 0, 1))\big)$, where s is the training step and T_w is the warmup duration. This schedule limits early randomness for stability while enabling stronger exploration as the policy becomes more reliable.

3.2.2 Noise-Aware Trajectory Selection and Advantage Estimation

For each prompt q_i, we form a mixed group of K_r standard and K_n noise-perturbed rollouts, compute their rewards, and normalize advantages over $\mathcal{S}_{r,i} \cup \widetilde{\mathcal{S}}_{n,i}$ as in GRPO. Noisy rollouts are then selected once per prompt batch $\mathcal{P}$, rather than per prompt. From the batch-wide noisy rollout pool $\widetilde{\mathcal{S}}_n^{\mathcal{P}} = \bigcup_{q_i \in \mathcal{P}} \widetilde{\mathcal{S}}_{n,i}$, EDEN retains $\mathcal{S}_n^{\mathcal{P}} = \{s \in \widetilde{\mathcal{S}}_n^{\mathcal{P}} \mid |A_s| \text{ is among the top fraction } p \text{ of } \widetilde{\mathcal{S}}_n^{\mathcal{P}}\}$, where p is the noisy-rollout keep ratio. Group-wise normalization makes $|A_s|$ comparable across prompts, while batch-level selection allows each prompt to retain a variable number of noisy rollouts rather than a fixed quota. This criterion prioritizes perturbed trajectories that are most informative relative to their prompt groups. Under binary rewards, a non-degenerate group with G rollouts and m correct responses assigns advantages $\sqrt{(G-m)/m}$ to correct rollouts and $-\sqrt{m/(G-m)}$ to incorrect ones. The minority outcome thus has the largest magnitude, so batch-level $|A|$ selection favors rare within-group deviations that change the group-relative signal rather than reinforce the dominant outcome. Fully degenerate all-correct or all-incorrect groups are handled by the same numerical stabilizer used in GRPO normalization.

To control the influence of perturbed trajectories, we scale their advantages as $\tilde{A}_i = \gamma A_i$, where $\gamma \in (0, 1]$, while retaining $\tilde{A}_i = A_i$ for standard rollouts. Since perturbed rollouts are optimized under clean prompt conditioning, this scaling reduces their gradient contribution to account for the difference between generation and update conditions.

Algorithm 1 EDEN Training

Require: policy π_θ, noise bank $\mathcal{B}$, temperatures τ_r, $\tau_n(s)$, rollout counts K_r, K_n, keep ratio p, advantage scale γ

for $s = 1, \ldots, S$ **do**
  Sample a prompt batch $\mathcal{P} = \{x_i\}_{i=1}^{B}$
  for each prompt $x_i \in \mathcal{P}$ **do**
    Sample standard rollouts $\mathcal{S}_{r,i} \sim \pi_\theta(\cdot \mid x_i; \tau_r)$
    Sample anchor $\mathbf{e}_b \sim \mathrm{Uniform}(\mathcal{B})$
    Sample noisy rollouts $\widetilde{\mathcal{S}}_{n,i} = \{o^{(k)} \sim \pi_\theta(\cdot \mid x_i, \tilde{\mathbf{e}}_b^{(k)}; \tau_n(s))\}_{k=1}^{K_n}$
    Compute rewards and group-relative advantages on $\mathcal{S}_{r,i} \cup \widetilde{\mathcal{S}}_{n,i}$
  end for
  $\mathcal{S}_n^{\mathcal{P}} \leftarrow \mathrm{Top}_p\Big(\bigcup_i \widetilde{\mathcal{S}}_{n,i};\, |A|\Big)$
  $\mathcal{D} \leftarrow \{(o, A(o)) : o \in \bigcup_i \mathcal{S}_{r,i}\} \cup \{(o, \gamma A(o)) : o \in \mathcal{S}_n^{\mathcal{P}}\}$
  Update θ on $\mathcal{D}$ via Eq. 3
end for

3.2.3 Noise-Aware Policy Update

EDEN updates the policy using standard and selected noise-perturbed rollouts. During the update, the appended perturbation position is removed, and token log-probabilities for all trajectories are evaluated under the clean prompt embedding $E(q)$. Thus, selected noisy trajectories are generated from $\pi_\theta(\cdot \mid \tilde{E}(q))$ but optimized under $\pi_\theta(\cdot \mid E(q))$. EDEN does not introduce an importance-sampling correction for this generation-update mismatch. Instead, advantage filtering determines which perturbed trajectories enter the update, and advantage scaling by γ reduces their gradient contribution. We use the following GRPO-style objective:

$$\mathcal{J}_{\mathrm{EDEN}}(\theta) = \mathbb{E}_q\left[\frac{1}{|\mathcal{S}(q)|}\sum_{i\in\mathcal{S}(q)}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big(\min\big(r_{i,t}\tilde{A}_i,\ \mathrm{clip}(r_{i,t}, 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}})\,\tilde{A}_i\big)\Big)\right] - \beta\,\mathbb{D}_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) + \alpha\,\mathbb{H}(\pi_\theta), \tag{3}$$

where $\mathcal{S}(q)$ denotes the set of rollouts used for training for prompt q, and $\tilde{A}_i$ is the scaled advantage. The detailed training procedure is summarized in Algorithm 1.

Table 1: Performance on mathematical reasoning benchmarks with Qwen2.5-Math-1.5B. We report Pass@1 and Pass@8. **Bold** indicates the best result per column.

Metric	Method	MATH	GSM8K	AMC	Olympiad	Gaokao	Minerva	Avg.
Pass@1	GRPO	76.60	**86.35**	53.01	40.50	64.94	**31.99**	58.90
	+ Higher Temp.	77.00	85.29	56.63	39.02	65.71	30.15	58.97
	+ Clip Higher	**77.40**	84.69	56.63	41.25	64.16	31.62	59.29
	EDEN	**77.40**	86.28	**63.86**	**41.84**	**67.01**	31.62	**61.34**
Pass@8	GRPO	86.60	94.92	69.88	55.34	77.14	41.54	70.90
	+ Higher Temp.	88.20	91.89	71.08	53.26	77.40	42.28	70.69
	+ Clip Higher	87.40	95.07	74.70	54.90	76.88	42.28	71.87
	EDEN	**89.60**	**95.60**	**79.52**	**55.93**	**78.96**	**44.12**	**73.96**

Table 2: Performance on mathematical reasoning benchmarks with Qwen2.5-Math-7B. We report Pass@1 and Pass@8. **Bold** indicates the best result per column.

Metric	Method	MATH	GSM8K	AMC	Olympiad	Gaokao	Minerva	Avg.
Pass@1	GRPO	**82.00**	88.93	**71.08**	44.81	66.75	35.66	64.87
	EDEN	81.80	**90.30**	68.67	**47.48**	**68.57**	**36.03**	**65.48**
Pass@8	GRPO	88.80	96.21	**83.13**	58.31	77.14	42.65	74.37
	EDEN	**93.00**	**97.04**	**83.13**	**64.39**	**80.52**	**50.00**	**78.01**

4 EXPERIMENTS

Models, Datasets, and Baselines. We evaluate on three Qwen2.5 models at different scales: Qwen2.5-0.5B (Yang et al., 2024a), Qwen2.5-Math-1.5B (Yang et al., 2024b), and Qwen2.5-Math-7B (Yang et al., 2024b). For Qwen2.5-0.5B, we train on GSM8K (Cobbe et al., 2021) and evaluate on GSM8K and MATH (Hendrycks et al., 2021). For the Qwen2.5-Math models, we train on MATH and DAPO-Math (Yu et al., 2025), and evaluate on GSM8K, MATH, AMC, AIME 2024, AIME 2025, OlympiadBench (He et al., 2024), Gaokao (Zhang et al., 2023), and MinervaMath (Lewkowycz et al., 2022). Beyond GRPO, we compare with the exploration controls in the same pipeline, none of which changes the rollout input conditioning: a higher sampling temperature, a larger entropy bonus, asymmetric clipping, and larger rollout groups.

Training Details. All experiments are conducted in the verl framework with vLLM for rollout generation. We use a binary reward, assigning 1 to correct responses and 0 otherwise. Each prompt produces 16 rollouts by default. For EDEN, these include 12 standard and 4 noise-perturbed rollouts. The noise bank size is 8. ρ is set to 0.12 for Qwen2.5-0.5B and 0.18 for the other models. We train with a batch size of 128 and a learning rate of 1×10^{-6}. We set $\beta = 0$ and $\alpha = 0.001$. All methods use this standard setting, and Table 7 reports seed robustness on Qwen2.5-0.5B. We train Qwen2.5-0.5B for 6 epochs and Qwen2.5-Math-1.5B and Qwen2.5-Math-7B for 10 epochs. For Pass@8 and Pass@256 evaluations, we set the sampling temperature to 0.6 and top-p to 0.95. All experiments are run on $8\times$H100 GPUs. Additional implementation details are provided in Appendix A.

4.1 MAIN RESULTS

EDEN outperforms GRPO on average at both 1.5B and 7B scales across English and Chinese mathematics benchmarks from grade-school to olympiad level. On Qwen2.5-Math-1.5B, we also compare with same-pipeline exploration controls, including **Higher Temperature** ($T = 1.2$) for increased sampling randomness and **Clip Higher** (Yu et al., 2025) for asymmetric PPO clipping. The entropy bonus and larger rollout groups are examined in Section 4.2. EDEN achieves the best average Pass@1 and Pass@8 in Table 1, exceeding GRPO by 2.44 and 3.06 points, respectively. Neither control alone yields comparable gains, highlighting the benefit of EDEN's exploration strategy. The improvement persists at 7B (Table 2). Across both scales, the larger gains on Pass@8 than Pass@1 are consistent with improved coverage of valid solutions under a fixed sampling budget.

Table 3: Pass@256 on AIME benchmarks. EDEN's advantage persists at high sampling budget.

Model	Method	AIME2024 Pass@256	AIME2025 Pass@256	Avg.
Qwen2.5-Math-1.5B	GRPO	53.33	50.00	51.67
	EDEN	**60.00**	**56.67**	**58.33**
Qwen2.5-Math-7B	GRPO	**73.33**	53.33	63.33
	EDEN	**73.33**	**66.67**	**70.00**

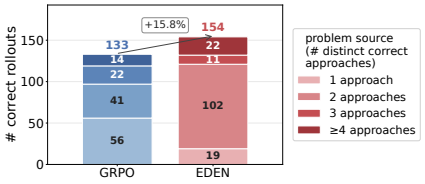


Figure 3: Correct AIME 2024/2025 rollouts (16 per problem, 60 problems), grouped by per-problem reasoning-pattern counts.

Table 4: Qwen2.5-0.5B performance under varying perturbations and anchor construction, averaged over MATH500 and GSM8K (16 rollouts/prompt). † Three anchor draws.

Method	Construction	Pass@1	Pass@8
GRPO	—	45.67	66.36
EDEN	curated bank (K=8), per prompt	**47.95**	**69.01**
– w/o noise	curated bank ($\rho = 0$), no perturbation	45.42	66.51
– w/o anchor	norm-matched random noise	43.39	65.45
– random anchor	random vocab anchor, per rollout	43.86	65.81
– random fixed anchor†	one random vocab anchor, fixed	46.37	67.78

4.2 Analysis

Exploration on Hard Problems To assess solution coverage at a larger sampling budget, we evaluate Pass@256 on the challenging AIME'24 and AIME'25 benchmarks. EDEN improves average Pass@256 over GRPO by 6.67 points across model scales, including a 13.33 point gain on AIME'25 on Qwen2.5-Math-7B (Table 3). These results suggest that EDEN's solution-coverage gains persist even at a large evaluation budget. Fig. 3 shows that after training EDEN produces more correct AIME rollouts, with more of them from problems solved by at least four distinct approaches and fewer from problems solved by a single approach. The case studies in Appendix F provide qualitative examples of how this diversity appears in complete reasoning traces.

Breaking the Rollout Budget Ceiling A simple alternative to EDEN's structured perturbation is to increase GRPO's rollout budget, potentially providing more gradient signal and broader sampling. We compare GRPO with 8, 16, and 32 rollouts per prompt against EDEN with 16. Increasing GRPO's budget from 8 to 32 improves Pass@1 by only 0.5 points and slightly reduces Pass@8, whereas EDEN substantially outperforms GRPO at both matched and larger GRPO budgets (Table 5). Fig. 5 shows that all GRPO configurations reach similarly low entropy by step 200, suggesting that sustaining exploration requires more than drawing additional samples from the same distribution. EDEN maintains higher entropy than every GRPO configuration after warm-up. Together, these results suggest that increasing GRPO's rollout count alone does not reproduce EDEN's sustained exploration or performance gains.

Role of Anchored Noise. Table 4 examines how noise, anchor choice, and anchor sharing affect EDEN. Removing noise reduces performance to the level of GRPO, showing that an unperturbed anchor alone is insufficient. Norm-matched random noise without an anchor also fails to reproduce EDEN's gains. A fixed random anchor recovers part of the improvement but remains below EDEN, showing that consistent anchoring can be useful even without bank diversity. Selecting a new random anchor for every rollout performs worse, which is consistent with increased within-group dispersion when rollouts from the same prompt use different conditioning centers. EDEN instead shares one curated-bank anchor within each prompt group and samples anchors across groups, combining a common within-group conditioning centre with variation across groups.

Comparison with Exploration Controls We examine whether EDEN's gains can be explained by conventional exploration controls applied during generation or policy updates. Increasing the entropy bonus on Qwen2.5-Math-1.5B raises policy entropy throughout training without improving validation

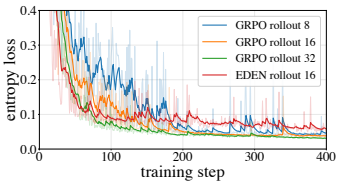


Figure 5: Training entropy on Qwen2.5-0.5B.

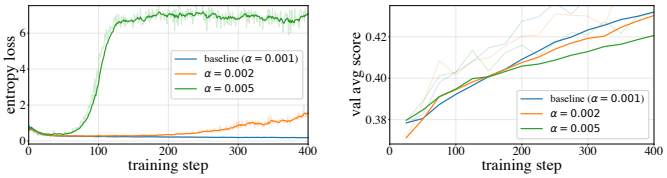
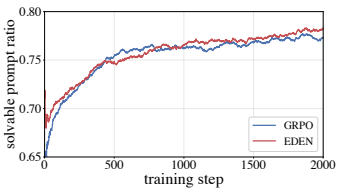
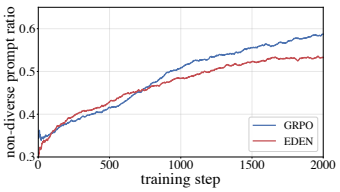


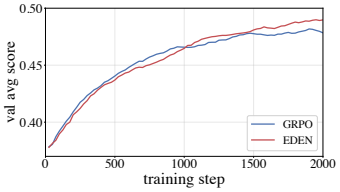
Figure 6: Effect of the entropy bonus coefficient. Left: Policy entropy. Right: Validation Avg@4.



(a) Fraction of prompts with at least one correct rollout.



(b) Fraction of prompts with zero advantages (all-correct/all-incorrect).



(c) Validation Avg@4 averaged across all eight benchmarks.

Figure 7: Training dynamics of EDEN vs. GRPO on Qwen2.5-Math-1.5B.

accuracy, which declines at larger bonuses (Fig. 6). Higher sampling temperatures yield inconsistent gains, and extreme temperatures destabilize training under the default settings (Appendix E). Clip Higher (Yu et al., 2025) provides limited and inconsistent gains over GRPO, whereas EDEN achieves stronger performance with the same asymmetric clipping, supporting the benefit of its additional rollout-side intervention. These results suggest that simply broadening the policy distribution does not reproduce EDEN's gains, supporting its use of structured perturbations.

Why EDEN Works: Expanding the Training Signal EDEN affects both trajectory diversity and group-relative training signals. Even before RL training, correct anchor-perturbed trajectories spread more widely than standard rollouts in aligned 4-gram space (Fig. 4), with an RMS radius about 1.6 times as large and more samples outside the standard-rollout hull. Anchored noise thus changes the surface form of correct trajectories before any policy update. Visualization details are provided in Appendix D. Fig. 7b shows fewer zero-advantage groups under EDEN from about step 750 onward, reaching 0.53 versus 0.58 for GRPO at step 2000. Unlike post-hoc filtering of a fixed rollout distribution, EDEN changes the distribution itself, allowing otherwise uniform-reward groups to yield mixed outcomes and provide non-zero group-relative training signals.

4.3 Ablation Studies

Effect of Advantage Filtering. We study the role of advantage-based filtering in EDEN by comparing against a variant that removes this mechanism and uses all noisy rollouts during training. As shown in Fig. 8, removing the filter leads to unstable updates, with frequent spikes in gradient norms due to high-variance noisy rollouts. In contrast, by selectively retaining trajectories with large absolute advantages, EDEN yields smoother gradient dynamics and consistently better performance across the evaluated benchmarks.

Table 5: Performance on Qwen2.5-0.5B under varying rollout budgets.

Method		Pass@1			Pass@8		
		MATH500	GSM8K	Avg.	MATH500	GSM8K	Avg.
GRPO	8	31.20	59.36	45.28	59.20	76.50	67.85
	16	31.60	59.74	45.67	58.20	74.53	66.36
	32	30.00	**61.56**	45.78	57.60	74.83	66.21
EDEN	16	**34.80**	61.11	**47.95**	**59.40**	**78.62**	**69.01**

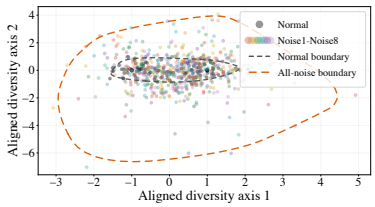
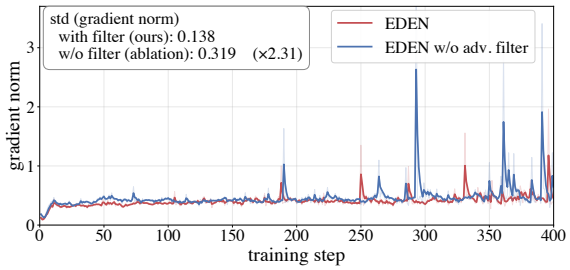


Figure 4: Diversity of correct trajectories in aligned 4-gram space.



Method	MATH	GSM8K	Avg.
EDEN	**34.10**	**59.95**	**47.03**
- adv. filter	31.85	59.38	45.62

Figure 8: Effect of advantage filtering. Left: training gradient norm. Right: Avg@4 performance.

Table 6: Effect of varying K. Results averaged over all benchmarks.

Method	K	**Pass@1**	**Pass@8**
GRPO	–	48.34	59.43
EDEN	1	48.45	61.96
EDEN	4	48.58	62.02
EDEN	8	**49.33**	**62.13**

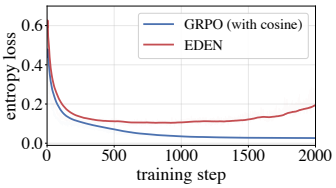


Figure 9: Qwen2.5-Math-1.5B training entropy with temperature warmup on 4 rollouts.

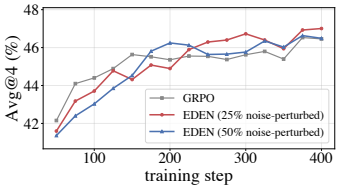


Figure 10: Effect of noisy rollout fraction on Qwen2.5-0.5B (Avg@4).

Anchor Bank Size and Noisy Rollout Fraction. We compare $K \in \{1, 4, 8\}$ on Qwen2.5-Math-1.5B in Table 6. A single anchor from the constructed bank recovers nearly all of the Pass@8 gain over GRPO, showing that anchored perturbation remains effective in the minimal setting. Larger banks provide modest additional gains, especially on Pass@1, motivating our choice of $K = 8$. EDEN thus benefits from complementary anchor centers without requiring a large bank. We also vary the noisy rollout fraction on Qwen2.5-0.5B, keeping 16 rollouts per prompt and comparing 0% (GRPO), 25% (12 standard + 4 noisy), and 50% (8 standard + 8 noisy). The 25% setting performs best, while 50% matches GRPO (Fig. 10). A modest noisy fraction thus suffices, while larger fractions leave fewer standard rollouts to provide reliable training signals.

Cosine schedule for exploration control strength. EDEN warms up the sampling temperature of noisy rollouts with a cosine schedule, as using the target temperature from the start leads to high entropy and near-zero rewards. With the same temperature warmup but no embedding perturbations, the baseline's entropy quickly collapses, whereas EDEN maintains higher entropy (Fig. 9). This suggests that the sustained exploration is not explained by the schedule alone. Warmup lengths of 200 and 400 steps yield similar accuracy, indicating limited sensitivity to duration (Appendix E).

5 Conclusion

RLVR faces limited exploration under small rollout budgets, with concentrated trajectories and weak learning signals. We propose EDEN, which explores beyond the policy's own rollouts through directional input-embedding perturbations at the rollout source. It generates a fraction of each group under perturbed conditioning and absorbs informative trajectories into the unperturbed policy through selective incorporation and advantage reweighting. This enables structured exploration without changing the model architecture, rollout budget, or test-time inference path. Across model scales and benchmarks, EDEN outperforms GRPO and the exploration controls in the same pipeline. These results suggest gains from input-side perturbations beyond output-side exploration controls alone.

Limitations and Future Work. Several directions may further extend EDEN's capabilities. First, the current noise bank is constructed from a set of anchor embeddings. More adaptive or learnable constructions, such as dynamic selection or joint optimization with the policy, may enable more targeted exploration. Second, EDEN applies Gaussian perturbations centered at anchor embeddings. More structured schemes, such as anisotropic noise or task-dependent directions, may further improve coverage of the trajectory space.

AI USE STATEMENT

We used LLM-based tools for language polishing, manuscript editing and organization, LaTeX formatting, and checks of completeness and consistency, including discussions of how experimental results were presented. These tools were not used to develop the core research ideas or methodology. The authors reviewed the AI-assisted revisions and take responsibility for the final content.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Bizhe Bai, Xinyue Wang, Peng Ye, and Tao Chen. Learning to explore with parameter-space noise: A deep dive into parameter-space noise for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2602.02555*, 2026.

Natasha Butt, Ariel Kwiatkowski, Ismail Labiad, Julia Kempe, and Yann Ollivier. Soft tokens, hard truths. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9JjKTp8Jmy.

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

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

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, 2025. URL https://arxiv.org/abs/2505.22617.

Haoran Dang, Cuiling Lan, Hai Wan, Xibin Zhao, and Yan Lu. Temperature as a meta-policy: Adaptive temperature in LLM reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AoTHU2OmS6.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Zengjie Hu, Jiantao Qiu, Tianyi Bai, Haojin Yang, Binhang Yuan, Qi Jing, Conghui He, and Wentao Zhang. Vade: Variance-aware dynamic sampling via online sample-level difficulty estimation for multimodal rl. *arXiv preprint arXiv:2511.18902*, 2025.

Baizhou Huang and Xiaojun Wan. PROS: Towards compute-efficient RLVR via rollout prefix reuse. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=lz1SRTcnUb.

Wei Huang, Yi Ge, Shuai Yang, Yicheng Xiao, Huizi Mao, Yujun Lin, Hanrong Ye, Sifei Liu, Ka Chun Cheung, Hongxu Yin, Yao Lu, Xiaojuan Qi, Song Han, and Yukang Chen. Qerl: Beyond efficiency. quantization-enhanced reinforcement learning for llms. *arXiv preprint arXiv:2510.11696*, 2025.

Soeun Kim and Albert No. Where rollouts begin: Low-load, high-leverage first-token diversification for rlvr, 2026. URL https://arxiv.org/abs/2605.28295.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.

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

Hieu Trung Nguyen, Bao Nguyen, Wenao Ma, Yuzhi Zhao, Ruifeng She, and Viet Anh Nguyen. Adaptive rollout allocation for online reinforcement learning with verifiable rewards. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Z5sWYACAop.

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating rl finetuning of reasoning models? In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1240–1250, 2026.

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

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

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*, 2025.

Xinyu Tang, Zhenduo Zhang, Yurou Liu, Xin Zhao, zujie wen, Zhiqiang Zhang, and JUN ZHOU. Towards high data efficiency in reinforcement learning with verifiable reward. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=sruA4AZmZI.

Akifumi Wachi, Hirota Kinoshita, Shokichi Takakura, Rei Higuchi, and Taiji Suzuki. A relative-budget theory for reinforcement learning with verifiable rewards in large language model reasoning. *arXiv preprint arXiv:2602.01523*, 2026.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jGbRWwIidy.

Shangyu Xing, Siyuan Wang, Chenyuan Yang, Xinyu Dai, and Xiang Ren. Lookahead tree-based rollouts for enhanced trajectory-level exploration in reinforcement learning with verifiable rewards. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4nLvUk8edu.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024a. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024b.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Yongxin Wang, Dongchun Xie, Hanhui Li, Yiwei Wang, Xiaodan Liang, and Jing Tang. Depth-breadth synergy in rlvr: Unlocking llm reasoning gains with adaptive exploration. *arXiv preprint arXiv:2508.13755*, 2025.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 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 D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 57654–57689. Curran Associates, Inc., 2025. doi: 10.52202/085713-1933. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/537d5aa768c2d534016a4d06f87bc8fb-Paper-Conference.pdf.

Hongzhi Zhang, Jia Fu, Jingyuan Zhang, Kai Fu, Qi Wang, Fuzheng Zhang, and Guorui Zhou. Rlep: Reinforcement learning with experience replay for llm reasoning. *arXiv preprint arXiv:2507.07451*, 2025a.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*, 2023.

Xuechen Zhang, Zijian Huang, Chenshun Ni, Ziyang Xiong, Jiasi Chen, and Samet Oymak. Making small language models efficient reasoners: Intervention, supervision, reinforcement. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*, 2025b. URL https://openreview.net/forum?id=aPtDOzJ56p.

Yuheng Zhang, Wenlin Yao, Changlong Yu, Yao Liu, Qingyu Yin, Bing Yin, Hyokun Yun, and Lihong Li. Improving sampling efficiency in rlvr through adaptive rollout and response reuse. *arXiv preprint arXiv:2509.25808*, 2025c.

Haizhong Zheng, Yang Zhou, Brian R Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*, 2025a.

Zhi Zheng, Yu Gu, Wei Liu, Yee Whye Teh, and Wee Sun Lee. Soft-grpo: Surpassing discrete-token llm reinforcement learning via gumbel-reparameterized soft-thinking policy optimization. *arXiv preprint arXiv:2511.06411*, 2025b.

Qinglin Zhu, Runcong Zhao, Hanqi Yan, Yulan He, Yudong Chen, and Lin Gui. Soft reasoning: Navigating solution spaces in large language models through controlled embedding exploration. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=4gWE7CMOlH.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. Flowrl: Matching reward distributions for llm reasoning. *arXiv preprint arXiv:2509.15207*, 2025b.

A Detailed Experimental Setup

Datasets. We evaluate our method across a diverse set of mathematical reasoning datasets, covering a wide range of difficulty levels.

For smaller models (Qwen2.5-0.5B), we train on the GSM8K training set and evaluate on both GSM8K and MATH. GSM8K focuses on grade-school arithmetic reasoning, while MATH contains more challenging competition-style problems.

For math-specialized models (Qwen2.5-Math-1.5B and Qwen2.5-Math-7B), we train on a combination of the MATH training set and DAPO-Math, which together provide diverse and high-difficulty mathematical reasoning data. Evaluation is conducted on a broad suite of benchmarks, including GSM8K and MATH (standard reasoning tasks), AMC, AIME 2024, and AIME 2025 (competition-level problems), as well as OlympiadBench, Gaokao, and MinervaMath, which further extend coverage to olympiad-style and advanced mathematical reasoning tasks.

Models. We conduct experiments on three Qwen2.5 models spanning different scales: Qwen2.5-0.5B[1], Qwen2.5-Math-1.5B[2], and Qwen2.5-Math-7B[3].

The Qwen2.5-0.5B model serves as a general-purpose base model, while Qwen2.5-Math-1.5B and Qwen2.5-Math-7B are specialized for mathematical reasoning through domain-specific pretraining. This setup allows us to evaluate the effectiveness of our method across both general and math-specialized models, as well as across different model scales.

Training Details. All experiments are implemented using the `verl` framework with vLLM for rollout generation. We use a binary outcome reward that assigns 1 to correct responses and 0 otherwise. Across methods, the global batch size is 128 and the learning rate is 1×10^{-6}. GRPO uses symmetric PPO clipping with $\epsilon = 0.2$. EDEN and the Clip Higher baseline use asymmetric clipping with $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. We set the KL coefficient to 0 and the entropy coefficient to 0.001. The maximum prompt and response lengths are 1536 and 2560 tokens, respectively. Qwen2.5-0.5B is trained for approximately 400 steps, corresponding to 6 epochs, while the math-specialized models are trained for approximately 2000 steps, corresponding to 10 epochs. All experiments are run on $8\times$H100 GPUs.

GRPO and EDEN use the same generation budget of 16 rollouts per prompt. GRPO generates all 16 rollouts from the unmodified input, whereas EDEN generates 12 standard rollouts and 4 noise-perturbed rollouts. For each prompt, EDEN samples one anchor from a bank of 8 candidates and shares it across the 4 perturbed rollouts, while applying an independent Gaussian perturbation to the shared anchor for each rollout. We set the relative perturbation scale ρ to 0.12 for Qwen2.5-0.5B and 0.18 for the math-specialized models.

For GRPO and EDEN, standard rollouts use a fixed sampling temperature of $\tau_r = 1.0$. The temperature for noise-perturbed rollouts follows a cosine schedule from 0.2 to 1.0 over the first 250 training steps for Qwen2.5-0.5B, 200 steps for Qwen2.5-Math-1.5B, and 400 steps for Qwen2.5-Math-7B, and remains at 1.0 thereafter. Rewards and group-relative advantages are computed jointly over all 16 rollouts for each prompt. All standard rollouts are retained for policy updates. We rank the perturbed rollouts globally across each prompt batch by absolute advantage and retain the top $p = 0.5$ fraction, so the number retained for an individual prompt can range from 0 to K_n. The advantages of the retained perturbed rollouts are scaled by $\gamma = 0.5$.

Anchor Assignment. Sharing one anchor across the perturbed rollouts of a prompt keeps the conditioning direction fixed within the group, while independent Gaussian perturbations introduce local variation around that direction. The anchor is resampled for each prompt, exposing different rollout groups to different directions from the fixed bank.

[1] `https://huggingface.co/Qwen/Qwen2.5-0.5B`
[2] `https://huggingface.co/Qwen/Qwen2.5-Math-1.5B`
[3] `https://huggingface.co/Qwen/Qwen2.5-Math-7B`

B Full Training Procedure of EDEN

We provide the full training procedure of EDEN in Algorithm 2. Compared to the simplified version presented in the main text, this algorithm explicitly details the construction of mixed rollout batches, including both standard and noise-perturbed rollouts, as well as the subsequent advantage computation and policy updates. In particular, for each prompt, EDEN first samples a set of standard rollouts, followed by a set of noise-anchored rollouts constructed via embedding-space perturbations. These rollouts are combined and filtered based on advantage scores, after which the policy is updated using the selected samples.

This formulation highlights the integration of structured noise injection with standard policy optimization, illustrating how EDEN balances exploration and stability during training.

Algorithm 2 Full Training Procedure of EDEN

Require: policy π_θ, noise bank $\mathcal{B}$, rollout temperature τ_r, noise-perturbed temperature schedule $\tau_n(s)$, rollout counts K_r, K_n, keep ratio p, advantage scale γ

for $s = 1$ to S **do**
  Sample a prompt batch $\mathcal{P} = \{x_i\}_{i=1}^{B}$
  for each prompt $x_i \in \mathcal{P}$ **do**
    Sample standard rollouts $\mathcal{S}_{r,i} = \{o^{(k)}\}_{k=1}^{K_r} \sim \pi_\theta(\cdot \mid x_i; \tau_r)$
    Sample anchor embedding $\mathbf{e}_b \sim \mathrm{Uniform}(\mathcal{B})$
    for $k = 1$ to K_n **do**
      Construct perturbed embedding $\tilde{\mathbf{e}}_b^{(k)}$ from $\mathbf{e}_b$
      Sample noise-perturbed rollout $o^{(k)} \sim \pi_\theta(\cdot \mid x_i, \tilde{\mathbf{e}}_b^{(k)}; \tau_n(s))$
    end for
    $\widetilde{\mathcal{S}}_{n,i} \leftarrow \{o^{(k)}\}_{k=1}^{K_n}$
    Compute rewards and group-relative advantages on $\mathcal{S}_{r,i} \cup \widetilde{\mathcal{S}}_{n,i}$
  end for
  $\mathcal{S}_n^{\mathcal{P}} \leftarrow \mathrm{Top}_p\big(\bigcup_i \widetilde{\mathcal{S}}_{n,i};\, |A|\big)$
  $\mathcal{D} \leftarrow \{(o, A(o)) : o \in \bigcup_i \mathcal{S}_{r,i}\} \cup \{(o, \gamma A(o)) : o \in \mathcal{S}_n^{\mathcal{P}}\}$
  Update θ on $\mathcal{D}$ via Eq. 3
end for

C Additional Details on Noise Bank Construction

The noise bank $\mathcal{B}$ is constructed separately for each model once before RL training and remains fixed throughout policy optimization. Candidate anchors are drawn from the model's own vocabulary and organized as anchor-relative directions in the embedding space. Given a reference token a and another token $i \neq a$, we define

$$\mathbf{u}_i = \frac{\Delta_i}{\|\Delta_i\|_2}, \qquad \Delta_i = \mathbf{e}_i - \mathbf{e}_a. \tag{4}$$

Normalizing the embedding displacement removes the influence of scale and retains the direction from the reference token to each candidate anchor.

We cluster the resulting directions and retain 32 representative candidate anchors for offline screening. For Qwen2.5-0.5B, the screening set contains 200 prompts from the GSM8K training set. For Qwen2.5-Math-1.5B and Qwen2.5-Math-7B, it contains 200 prompts from the DAPO-Math training data. For each candidate and screening prompt, we generate eight perturbed rollouts using the same perturbation construction as in training.

The final bank is selected so that its perturbed rollouts complement the standard rollout distribution. We use the lexicographic objective

$$\mathcal{B}^\star = \arg\max_{\mathcal{B}} \big(\mathrm{Cov}(\mathcal{B}),\, \mathbf{1}[\mathrm{LenRatio}(\mathcal{B}) \geq \eta_{\mathrm{len}}],\, \mathrm{DisAns}(\mathcal{B}),\, \mathrm{DisReason}(\mathcal{B}),\, \mathrm{TotCorrect}(\mathcal{B})\big), \tag{5}$$

Here, $\mathrm{Cov}(\mathcal{B})$ is the fraction of prompts with at least one correct perturbed rollout, and $\mathrm{LenRatio}(\mathcal{B})$ enforces response-length feasibility. $\mathrm{DisAns}(\mathcal{B})$ is the mean number of distinct extracted final-answer

values per prompt. $\mathrm{DisReason}(\mathcal{B})$ is one minus the mean pairwise Jaccard similarity between token-level 6-gram sets of the normalized reasoning traces, computed within the eight rollouts for each prompt and averaged over the screening prompts; normalization lowercases the traces, removes final-answer patterns, and collapses whitespace. $\mathrm{TotCorrect}(\mathcal{B})$ is the total number of correct rollouts. We prioritize coverage and length feasibility, and use rollout diversity and correctness to rank the remaining candidates. The final bank contains eight anchors.

This construction is a one-time screening heuristic rather than an objective optimized during RL training. Its purpose is to exclude anchors whose perturbations either have little effect on generation or substantially disrupt coherent responses, rather than to identify a unique optimal bank.

D Additional Experimental Results

Pre-RL Rollout Diversity Visualization. We analyze rollouts generated from the Qwen2.5-Math-1.5B checkpoint before RL fine-tuning to isolate the direct effect of anchored noise on rollout generation. We represent each correct trajectory by its set of token-level 4-grams and compute pairwise distances as one minus Jaccard similarity. For each of the 28 prompts with at least two correct standard rollouts, we obtain a prompt-specific two-dimensional MDS embedding over the standard and anchor-perturbed trajectories, center and align it using the standard-rollout centroid and principal axis, and normalize it by the standard-rollout RMS radius before pooling the coordinates across prompts. The plotted boundaries are smoothed convex hulls of the resulting points.

Post-RL AIME Rollout Diversity. For Fig. 3, we sample 16 responses per problem from the final Qwen2.5-Math-1.5B GRPO and EDEN checkpoints on 60 AIME 2024 and AIME 2025 problems and retain the correct responses. Each response is assigned a deterministic approach signature defined by the set of predefined reasoning markers detected in its complete trace, including summation, multiplication, variable substitution, system solving, exponential reformulation, polynomial division, remainder-based reasoning, case analysis, and modular arithmetic. Correct rollouts are grouped according to the number of distinct signatures observed for their source problem.

Training-Seed Robustness on Qwen2.5-0.5B. We repeat GRPO and EDEN with three independent training seeds under the same configuration. Table 7 reports the mean and sample standard deviation. EDEN improves the macro average for all three paired seeds under both greedy and Pass@8 evaluation.

Table 7: Training-seed robustness on Qwen2.5-0.5B. Results are reported as mean $\pm$ sample standard deviation over three independent training runs. Pass@1 uses greedy decoding with $T = 0$ and top-$p = 1.0$; Pass@8 uses $T = 0.6$ and top-$p = 0.95$.

Metric	Method	MATH500	GSM8K	Avg.
Pass@1	GRPO	31.40 ± 1.11	59.59 ± 0.40	45.50 ± 0.76
	EDEN	$\mathbf{34.80 \pm 1.00}$	$\mathbf{61.13 \pm 0.42}$	$\mathbf{47.96 \pm 0.71}$
Pass@8	GRPO	57.80 ± 1.25	74.32 ± 0.56	66.06 ± 0.90
	EDEN	$\mathbf{59.40 \pm 0.80}$	$\mathbf{78.60 \pm 0.27}$	$\mathbf{69.00 \pm 0.53}$

Average Performance Across Rollouts. In addition to Pass@k, which measures whether at least one of k sampled responses is correct, we report Avg@4 as the mean correctness over four independently sampled responses per prompt. Pass@1 uses greedy decoding ($T = 0$), and all trained models are evaluated using their final training checkpoints.

As shown in Fig. 11, Table 8 and Table 9, EDEN consistently improves Avg@4 across both small-scale and math-specialized models. Notably, these gains are observed under a relatively high sampling temperature ($T = 1$), where standard rollouts typically exhibit higher variance. This suggests that the improvements from EDEN are not limited to occasional high-quality samples, but extend to average correctness across sampled responses.

Method	MATH	GSM8K	Avg
GRPO (rollout 8)	32.35	58.24	45.30
GRPO (rollout 16)	33.10	58.70	45.90
EDEN (rollout 16)	**34.10**	**59.95**	**47.03**

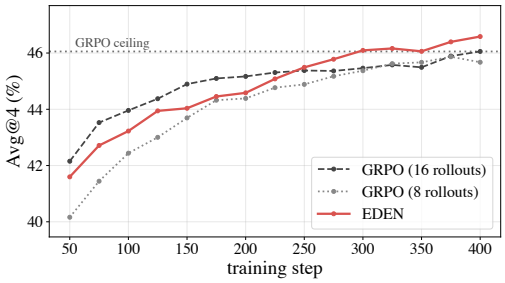


Figure 11: Average performance (Avg@4) comparison on Qwen2.5-0.5B under T=1, top-p=0.7. Left: numerical results. Right: visualization of Avg@4 performance.

Table 8: Average performance (Avg@4) comparison on Qwen2.5-Math-1.5B models under T=1, top-p=0.7.

Method	MATH	GSM8K	AMC	AIME24	AIME25	Olympiad	Gaokao	Minerva	Avg
GRPO	**78.00**	**85.58**	56.33	**16.67**	6.67	40.47	64.03	31.53	47.41
EDEN	77.30	84.95	**57.83**	**16.67**	**11.67**	**41.88**	**66.75**	**32.81**	**48.73**

E Extended Ablation Studies

Effect of Temperature Schedule. We study the role of the cosine temperature schedule in EDEN by analyzing both its necessity and the effect of its duration.

As shown in Fig. 12, removing the temperature schedule leads to unstable training. The gradient norm grows rapidly and eventually diverges, resulting in numerical instability (NaNs), while validation accuracy initially improves but subsequently collapses to zero. This demonstrates that gradually increasing the sampling temperature is essential for stabilizing noise-perturbed rollouts during early training.

We further investigate the impact of the schedule duration by comparing cosine schedules of 0–200 and 0–400 steps. As shown in Fig. 13, both configurations achieve comparable performance across most benchmarks and consistently outperform the GRPO baseline under both Pass@1 (greedy) and Pass@8 (sampling) settings. While extending the schedule to 0–400 steps yields slight improvements on more challenging tasks, the overall differences remain modest.

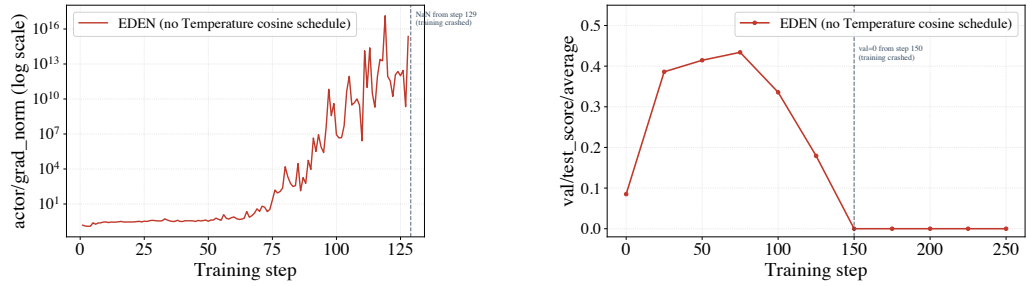
These results indicate that temperature scheduling is necessary for stable optimization, while performance is relatively insensitive to the schedule duration. This suggests that the primary performance gains of EDEN stem from the proposed noise-based exploration mechanism rather than fine-tuned temperature scheduling.

Effect of Extreme Temperature Sampling. We observe that increasing temperature alone does not lead to consistent improvements across benchmarks. While moderate temperature (e.g., $T = 1.2$) can improve performance on certain tasks, the gains are unstable and vary significantly across benchmarks.

At extreme temperature ($T = 1.4$), training becomes highly unstable under the default setting due to excessively large entropy and stochasticity. To enable a meaningful comparison, we set the entropy coefficient to zero in both GRPO and EDEN, as the exploration induced by high temperature alone is already sufficient. Without this adjustment, training frequently collapses.

Table 9: Average performance (Avg@4) comparison on Qwen2.5-Math-7B under T=1, top-p=0.7.

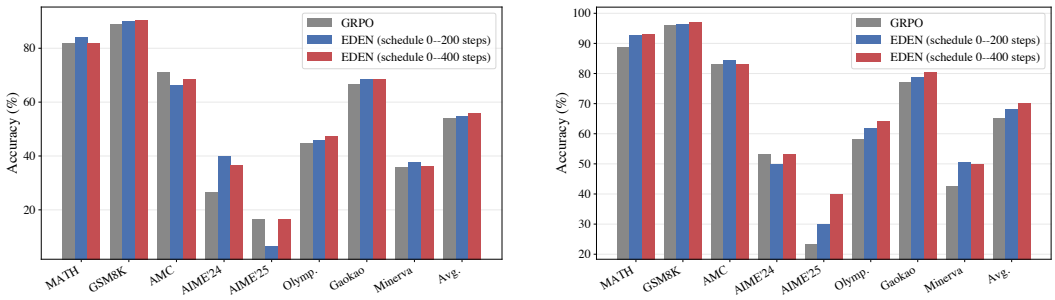
Method	MATH	GSM8K	AMC	AIME24	AIME25	Olympiad	Gaokao	Minerva	Avg
GRPO	81.65	88.70	**70.18**	29.17	**15.83**	**46.66**	67.99	36.12	54.54
EDEN	**83.25**	**90.09**	65.06	**37.50**	12.50	45.14	**68.77**	**36.67**	**54.87**



(a) Gradient norm (log scale)

(b) Validation accuracy (Avg@4)

Figure 12: Training instability without temperature scheduling on Qwen2.5-0.5B. Left: gradient norm (log scale) rapidly increases and eventually diverges, leading to numerical instability (NaNs). Right: validation accuracy initially improves but subsequently collapses to zero as training becomes unstable. These results demonstrate that temperature scheduling is essential for stabilizing noise-perturbed rollouts in EDEN.



(a) Pass@1 (greedy)

(b) Pass@8 (T=0.6, top-p=0.95)

Figure 13: Effect of temperature schedule duration in EDEN on Qwen2.5-Math-7B. We compare cosine schedules with durations of 0–200 and 0–400 steps. Both configurations yield comparable performance and consistently outperform the GRPO baseline across most benchmarks. Left: Pass@1 with greedy decoding. Right: Pass@8 with T=0.6 and top-p=0.95.

Under this controlled setting, we find that extreme temperature does not consistently improve performance and often leads to degradation, particularly for Pass@8. In contrast, EDEN achieves the best overall performance at $T = 1$, suggesting that structured noise-based exploration provides more effective and stable exploration than simply increasing sampling temperature.

Table 10: Effect of temperature on mathematical reasoning performance (Qwen2.5-Math-1.5B). We compare GRPO and EDEN under different temperature settings. We report Pass@1 (greedy) and Pass@8 (sampling). **Bold** indicates the best result within each block.

Metric	Method	MATH	GSM8K	AMC	AIME'24	AIME'25	Olympiad	Gaokao	Minerva	Avg.
Pass@1	GRPO (T=1)	76.60	**86.35**	53.01	13.33	**20.00**	40.50	64.94	31.99	48.34
	GRPO (T=1.2)	77.00	85.29	56.63	13.33	10.00	39.02	65.71	30.15	47.14
	GRPO (T=1.4)	76.60	84.91	59.04	13.33	16.67	41.10	66.49	31.99	48.77
	EDEN (T=1)	77.40	86.28	**63.86**	13.33	13.33	**41.84**	**67.01**	31.62	**49.33**
	EDEN (T=1.4)	**78.60**	85.75	57.83	**20.00**	13.33	39.17	64.94	**33.09**	49.09
Pass@8	GRPO (T=1)	86.60	94.92	69.88	**33.33**	16.67	55.34	77.14	41.54	59.43
	GRPO (T=1.2)	88.20	91.89	71.08	26.67	**33.33**	53.26	77.40	42.28	60.51
	GRPO (T=1.4)	87.80	91.74	75.90	**33.33**	26.67	54.75	75.58	41.18	60.87
	EDEN (T=1)	**89.60**	**95.60**	**79.52**	**33.33**	20.00	**55.93**	**78.96**	**44.12**	**62.13**
	EDEN (T=1.4)	87.80	94.77	71.08	30.00	20.00	55.79	75.58	42.28	59.66

F CASE STUDIES.

We present a case study on AIME to examine how post-training rollout diversity appears in complete reasoning traces. For a selected problem, we compare correct rollouts generated by GRPO and EDEN from 16 samples per prompt using T=0.6 and top-p=0.95. These examples provide qualitative context for the aggregate AIME rollout diversity in Fig. 3. Here, "correct" refers to final-answer correctness, not the validity of every intermediate reasoning step.

Problem. Find the sum of all positive integers n such that $n+2$ divides the product $3(n+3)(n^2+9)$

Correct answer. 49

Analysis of Reasoning Traces. We compare how the correct rollouts formulate the underlying divisibility argument.

For GRPO, the two correct solutions follow essentially the same derivation. Both introduce the substitution $k = n + 2$, rewrite the expression in terms of k, and reduce the divisibility condition to $k \mid 39$ through algebraic simplification and coprimality arguments. Their differences are largely limited to the order of expansion and intermediate steps.

EDEN, by contrast, produces a wider range of derivations: (i) polynomial division by $(n + 2)$ to obtain the constant remainder 39; (ii) parameterization via $n = kx - 2$ followed by modular simplification; (iii) direct use of the remainder theorem through $n \equiv -2 \pmod{n+2}$; and (iv) substitution $k = n + 2$ followed by algebraic expansion and modular reduction. Although these derivations share the same core reduction $n + 2 \mid 39$, they reach it through different algebraic formulations and sequences of intermediate steps.

This example illustrates that EDEN produces greater variation in complete reasoning traces than GRPO, spanning several derivations of the same core divisibility argument. This qualitative pattern complements the aggregate post-training rollout diversity shown in Fig. 3.

EDEN — Rollout 1 (correct, ✓).

We need to find all positive integers n such that $n + 2$ divides the product $3(n + 3)(n^2 + 9)$.
This means that $\frac{3(n+3)(n^2+9)}{n+2}$ must be an integer.

First, let's perform polynomial long division on $\frac{3(n+3)(n^2+9)}{n+2}$.
We start by expanding the numerator:

$$3(n+3)(n^2+9) = 3(n^3+9n+3n^2+27) = 3(n^3+3n^2+9n+27) = 3n^3+9n^2+27n+81.$$

Now, we need to divide $3n^3 + 9n^2 + 27n + 81$ by $n + 2$. Using polynomial long division: 1. Divide the leading term $3n^3$ by n to get $3n^2$. 2. Multiply $3n^2$ by $n + 2$ to get $3n^3 + 6n^2$. 3. Subtract $3n^3 + 6n^2$ from $3n^3 + 9n^2 + 27n + 81$ to get $3n^2 + 27n + 81$. 4. Divide the leading term $3n^2$ by n to get $3n$. 5. Multiply $3n$ by $n + 2$ to get $3n^2 + 6n$. 6. Subtract $3n^2 + 6n$ from $3n^2 + 27n + 81$ to get $21n + 81$. 7. Divide the leading term $21n$ by n to get 21. 8. Multiply 21 by $n + 2$ to get $21n + 42$. 9. Subtract $21n + 42$ from $21n + 81$ to get 39.
So, we have:

$$\frac{3n^3 + 9n^2 + 27n + 81}{n + 2} = 3n^2 + 3n + 21 + \frac{39}{n + 2}.$$

For $\frac{3(n+3)(n^2+9)}{n+2}$ to be an integer, $\frac{39}{n+2}$ must be an integer. This means $n + 2$ must be a divisor of 39. The divisors of 39 are $\pm 1, \pm 3, \pm 13, \pm 39$. Since n is a positive integer, $n + 2$ must be one of the positive divisors of 39, i.e., $1, 3, 13, 39$.
So, we solve for n: 1. $n + 2 = 1$ gives $n = -1$ (not a positive integer). 2. $n + 2 = 3$ gives $n = 1$. 3. $n + 2 = 13$ gives $n = 11$. 4. $n + 2 = 39$ gives $n = 37$.
Thus, the possible values for n are $1, 11$, and 37. The sum of these values is:

$$1 + 11 + 37 = 49.$$

So, the sum of all positive integers n such that $n + 2$ divides $3(n + 3)(n^2 + 9)$ is $\boxed{49}$.

EDEN — Rollout 2 (correct, ✓).

We need to find all positive integers n such that $n+2$ divides the product $3(n+3)(n^2+9)$. This means that:

$$3(n+3)(n^2+9) \equiv 0 \pmod{n+2}.$$

To simplify this, we will use the fact that if $n+2 \equiv 0 \pmod{k}$, then $n \equiv -2 \pmod{k}$, or $n = kx-2$ for some integer x. Substituting $n = kx-2$ into the expression $3(n+3)(n^2+9)$, we get:

$$n+3 = (kx-2)+3 = kx+1,$$

and

$$n^2+9 = (kx-2)^2+9 = k^2x^2-4kx+4+9 = k^2x^2-4kx+13.$$

So the product becomes:

$$3(n+3)(n^2+9) = 3(kx+1)(k^2x^2-4kx+13).$$

We need this to be divisible by $k = n+2$. Therefore:

$$3(kx+1)(k^2x^2-4kx+13) \equiv 0 \pmod{k}.$$

Since k divides k^2x^2 and $4kx$, it must also divide $kx+1$ and $k^2x^2-4kx+13$. So:

$$kx+1 \equiv 1 \pmod{k},$$

and

$$k^2x^2-4kx+13 \equiv 13 \pmod{k}.$$

Thus, we need:

$$3 \cdot 1 \cdot 13 \equiv 0 \pmod{k},$$

which simplifies to:

$$39 \equiv 0 \pmod{k}.$$

This means k must be a divisor of 39. The divisors of 39 are $1, 3, 13, 39$. We will check each case:

1. $k = 1$:

$$n+2 = 1 \Rightarrow n = -1,$$

which is not a positive integer.

2. $k = 3$:

$$n+2 = 3 \Rightarrow n = 1.$$

We substitute $n = 1$ into the original expression:

$$3(n+3)(n^2+9) = 3(1+3)(1^2+9) = 3 \cdot 4 \cdot 10 = 120.$$

And $120 \div 3 = 40$, which is an integer. So $n = 1$ works.

3. $k = 13$:

$$n+2 = 13 \Rightarrow n = 11.$$

We substitute $n = 11$ into the original expression:

$$3(n+3)(n^2+9) = 3(11+3)(11^2+9) = 3 \cdot 14 \cdot (121+9) = 3 \cdot 14 \cdot 130 = 5460.$$

And $5460 \div 13 = 420$, which is an integer. So $n = 11$ works.

4. $k = 39$:

$$n+2 = 39 \Rightarrow n = 37.$$

We substitute $n = 37$ into the original expression:

$$3(n+3)(n^2+9) = 3(37+3)(37^2+9) = 3 \cdot 40 \cdot (1369+9) = 3 \cdot 40 \cdot 1378 = 165360.$$

And $165360 \div 39 = 4240$, which is an integer. So $n = 37$ works.

So the positive integers n that satisfy the condition are $1, 11,$ and 37. The sum of these integers is:

$$1+11+37 = 49.$$

Thus, the final answer is:

$$\boxed{49}$$

EDEN — Rollout 3 (correct, ✓).

We need to find all positive integers n such that $n+2$ divides the product $3(n+3)(n^2+9)$. This means that:

$$3(n+3)(n^2+9) \equiv 0 \pmod{n+2}$$

To simplify this, we will use the fact that if $n+2 \mid k$, then $n \equiv -2 \pmod{n+2}$. So, we substitute $n=-2$ into the expression $3(n+3)(n^2+9)$:

$$3((-2)+3)((-2)^2+9) = 3(1)(4+9) = 3 \cdot 1 \cdot 13 = 39.$$

This means that $n+2$ must divide 39. The divisors of 39 are $\pm 1, \pm 3, \pm 13, \pm 39$. Since n is a positive integer, $n+2$ must be one of the positive divisors of 39, i.e., $1, 3, 13, 39$. Therefore, we have:

$$n+2=1 \implies n=-1 \quad \textit{(not a positive integer)},$$
$$n+2=3 \implies n=1,$$
$$n+2=13 \implies n=11,$$
$$n+2=39 \implies n=37.$$

So, the possible values for n are $1, 11$, and 37. Now, we need to check if these values satisfy the original condition: 1. For $n=1$:

$$n+2=3,$$
$$3(n+3)(n^2+9) = 3(1+3)(1^2+9) = 3 \cdot 4 \cdot 10 = 120.$$

Since $3 \mid 120$, $n=1$ is a valid solution. 2. For $n=11$:

$$n+2=13,$$
$$3(n+3)(n^2+9) = 3(11+3)(11^2+9) = 3 \cdot 14 \cdot (121+9) = 3 \cdot 14 \cdot 130 = 5460.$$

Since $13 \mid 5460$, $n=11$ is a valid solution. 3. For $n=37$:

$$n+2=39,$$
$$3(n+3)(n^2+9) = 3(37+3)(37^2+9) = 3 \cdot 40 \cdot (1369+9) = 3 \cdot 40 \cdot 1378 = 165360.$$

Since $39 \mid 165360$, $n=37$ is a valid solution.
So, the valid values for n are $1, 11$, and 37. The sum of these values is:

$$1+11+37=49.$$

Thus, the sum of all positive integers n such that $n+2$ divides the product $3(n+3)(n^2+9)$ is $\boxed{49}$.

EDEN — Rollout 4 (correct, ✓).

We need to find all positive integers n such that $n+2$ divides the product $3(n+3)(n^2+9)$. This means that $3(n+3)(n^2+9) \equiv 0 \pmod{n+2}$.
First, let's substitute $n=k-2$ to simplify the expression. We need to find k such that k divides $3(k+1)((k-2)^2+9)$.
So, we substitute $n=k-2$:

$$3(n+3)(n^2+9) = 3((k-2)+3)((k-2)^2+9) = 3(k+1)((k-2)^2+9).$$

Now, we simplify $(k-2)^2+9$:

$$(k-2)^2+9 = k^2-4k+4+9 = k^2-4k+13.$$

So, we need k to divide:

$$3(k+1)(k^2-4k+13).$$

We need to find the values of k such that k divides $3(k+1)(k^2-4k+13)$. This means:

$$3(k+1)(k^2-4k+13) \equiv 0 \pmod{k}.$$

Since k divides k^2 and $4k$, we have:

$$3(k+1)(k^2-4k+13) \equiv 3(k+1)(13) \equiv 39(k+1) \pmod{k}.$$

This simplifies to:

$$39(k+1) \equiv 0 \pmod{k}.$$

This means k must divide $39(k+1)$. Since k divides $39(k+1)$, it must divide $39(k+1) - 39k = 39$. So, k must be a divisor of 39. The divisors of 39 are $1, 3, 13, 39$.

We now need to check each of these values for k:

1. If $k = 1$, then $n = k - 2 = 1 - 2 = -1$, which is not a positive integer. 2. If $k = 3$, then $n = k - 2 = 3 - 2 = 1$. We check:

$$3(n+3)(n^2+9) = 3(1+3)(1^2+9) = 3 \cdot 4 \cdot 10 = 120.$$

And $n + 2 = 1 + 2 = 3$, which divides 120. So $n = 1$ works. 3. If $k = 13$, then $n = k - 2 = 13 - 2 = 11$. We check:

$$3(n+3)(n^2+9) = 3(11+3)(11^2+9) = 3 \cdot 14 \cdot (121+9) = 3 \cdot 14 \cdot 130 = 5460.$$

And $n + 2 = 11 + 2 = 13$, which divides 5460. So $n = 11$ works. 4. If $k = 39$, then $n = k - 2 = 39 - 2 = 37$. We check:

$$3(n+3)(n^2+9) = 3(37+3)(37^2+9) = 3 \cdot 40 \cdot (1369+9) = 3 \cdot 40 \cdot 1378 = 165360.$$

And $n + 2 = 37 + 2 = 39$, which divides 165360. So $n = 37$ works.

So, the positive integers n that satisfy the condition are $1, 11$, and 37. The sum of these integers is:

$$1 + 11 + 37 = 49.$$

Therefore, the final answer is:

$$\boxed{49}$$

GRPO Baseline — Rollout 1 (correct, ✓).

To solve the problem, we need to find all positive integers n such that $n + 2$ divides the product $3(n+3)(n^2+9)$.

Let's denote $k = n + 2$. Then $n = k - 2$. We need to find n such that k divides $3(k-2+3)((k-2)^2+9)$. Simplifying the expression inside the divisors, we get:

$$k \div 3(k-2+3)((k-2)^2+9) = k \div 3(k+1)((k-2)^2+9).$$

Now, let's simplify $(k-2)^2 + 9$:

$$(k-2)^2 + 9 = k^2 - 4k + 4 + 9 = k^2 - 4k + 13.$$

So, the condition becomes:

$$k \div 3(k+1)(k^2-4k+13).$$

This means that $3(k+1)(k^2-4k+13)$ must be divisible by k. Therefore, we need:

$$3(k+1)(k^2-4k+13) \equiv 0 \pmod{k}.$$

Since k divides $k(k^2-4k+13)$, we only need to check the divisibility by 3:

$$3(k+1)(k^2-4k+13) \equiv 0 \pmod{k} \implies 3(k+1)\left(k^2-4k+13\right) \equiv 0 \pmod{k}.$$

This simplifies to checking the divisibility of $3(k+1)\left(k^2-4k+13\right)$ by k. Since k and $k+1$ are consecutive integers, they are coprime, so k must divide $3(k^2-4k+13)$. Therefore, we need:

$$3(k^2-4k+13) \equiv 0 \pmod{k}.$$

This further simplifies to:

$$3(k^2-4k+13) = 3k^2 - 12k + 39.$$

We need:

$$3k^2 - 12k + 39 \equiv 0 \pmod{k}.$$

Since $3k^2$ and $-12k$ are both divisible by k, we are left with:

$$39 \equiv 0 \pmod{k}.$$

This means that k must be a divisor of 39. The divisors of 39 are $1, 3, 13, 39$. However, since $k = n + 2$, k must be greater than 2. Therefore, the possible values for k are $3, 13, 39$.

Now, we can find the corresponding values of n: 1. If $k = 3$, then $n = k - 2 = 3 - 2 = 1$. 2. If $k = 13$, then $n = k - 2 = 13 - 2 = 11$. 3. If $k = 39$, then $n = k - 2 = 39 - 2 = 37$.

The positive integers n that satisfy the condition are $1, 11$, and 37. The sum of these integers is:

$$1 + 11 + 37 = 49.$$

Therefore, the sum of all positive integers n such that $n + 2$ divides the product $3(n+3)(n^2+9)$ is $\boxed{49}$.

GRPO Baseline — Rollout 2 (correct, ✓).

To solve the problem, we need to find all positive integers n such that $n+2$ divides the product $3(n+3)(n^2+9)$.
Let's denote $k = n+2$. Then $n = k-2$. We need to find n such that k divides $3(k-2+3)((k-2)^2+9)$. Simplifying the expression inside the divisors, we get:

$$k \div 3(k-2+3)((k-2)^2+9) = k \div 3(k+1)((k-2)^2+9).$$

Expanding $(k-2)^2+9$, we get:

$$(k-2)^2+9 = k^2-4k+4+9 = k^2-4k+13.$$

So, the condition becomes:

$$k \div 3(k+1)(k^2-4k+13).$$

This means that $3(k+1)(k^2-4k+13)$ must be divisible by k. Therefore, we need:

$$3(k+1)(k^2-4k+13) \equiv 0 \pmod{k}.$$

Since k divides $k(k^2-4k+13)$, we only need to check the divisibility by 3:

$$3(k+1)(k^2-4k+13) \equiv 0 \pmod{k} \implies 3(k+1)\left(k^2-4k+13\right) \equiv 0 \pmod{k}.$$

This simplifies to checking the divisibility of $3(k+1)\left(k^2-4k+13\right)$ by k. Since k and $k+1$ are coprime, k must divide $3(k^2-4k+13)$. So, we need:

$$3(k^2-4k+13) \equiv 0 \pmod{k}.$$

This further simplifies to:

$$3(k^2-4k+13) = 3k^2-12k+39.$$

We need:

$$3k^2-12k+39 \equiv 0 \pmod{k}.$$

Since $3k^2$ and $-12k$ are both divisible by k, we are left with:

$$39 \equiv 0 \pmod{k}.$$

This means that k must be a divisor of 39. The divisors of 39 are $1, 3, 13, 39$. However, since $k = n+2$, k must be greater than 2. Therefore, the possible values for k are $3, 13, 39$.
Now, we can find the corresponding values of n: 1. If $k = 3$, then $n = k-2 = 3-2 = 1$. 2. If $k = 13$, then $n = k-2 = 13-2 = 11$. 3. If $k = 39$, then $n = k-2 = 39-2 = 37$.
So, the positive integers n that satisfy the condition are $1, 11$, and 37. The sum of these integers is:

$$1+11+37 = 49.$$

Therefore, the sum of all positive integers n such that $n+2$ divides the product $3(n+3)(n^2+9)$ is $\boxed{49}$.