Learning to Explore with Parameter-Space Noise: A Deep Dive into Parameter-Space Noise for Reinforcement Learning with Verifiable Rewards

Bizhe Bai [1 2] **Xinyue Wang** [1] **Peng Ye** [3 4] **Tao Chen** [2 1]

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) enhances large language model (LLM) reasoning, yet growing evidence indicates an *exploration ceiling*: they tend to reweight existing solution traces and produce low-diversity rollouts. This restricts the discovery of novel solution perspective and weakens out-of-distribution generalization. We address this limitation by directly perturbing the rollout policy's parameters while updating the clean policy model as usual, an approach that achieves better temporal consistency and yields more complex behavior patterns. To stabilize learning from these perturbed rollouts, which are inherently off-policy, we incorporate truncated importance sampling. Furthermore, we introduce a lightweight, dynamic noise scheduler based on semantic diversity and model self-certainty to optimally adjust the noise scale. Instantiated on GRPO, our method (PSN-GRPO) demonstrates its efficacy through four key contributions: (1) it improves high-budget pass@k, increases semantic and operational diversity, increases CoT consistency, and remains complementary to exploration-oriented RLVR techniques; (2) it validates efficacy across multiple models, including Qwen2.5, Qwen3, and Llama-3.1; (3) it shows robust performance on out-of-distribution science and coding benchmarks; and (4) it uncovers novel solution perspectives for problems the original model could not solve, as shown by qualitative analysis.

[1]College of Future Information Technology, Fudan University, Shanghai, China [2]Shanghai Innovation Institute, Shanghai, China [3]Shanghai AI Laboratory, Shanghai, China [4]The Chinese University of Hong Kong, Hong Kong, China. Correspondence to: Bizhe Bai <bizhe.bai@sii.edu.cn>, Tao Chen <eetchen@fudan.edu.cn>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) has become a central paradigm for improving the reasoning behavior of Large Language Models (LLMs), especially in domains with automatic correctness signals such as mathematics and code generation (Lu et al., 2024; Ouyang et al., 2022; Lambert et al., 2025). Policy-gradient algorithms—most notably Proximal Policy Optimization (PPO) (Schulman et al., 2017) and recent variants such as Group Relative Policy Optimization (GRPO) (Guo et al., 2025)—enable direct optimization against verifiers and have propelled systems such as DeepSeek-R1 (Guo et al., 2025) to strong empirical performance.

However, recent evidence suggests that current RLVR pipelines may face an *exploration ceiling*. RLVR often improves *sampling efficiency* (e.g., pass@1) more than it expands the high-budget reasoning capability boundary (e.g., pass@256): post-RLVR trajectories remain largely contained within, or close to, the base model's pretraining distribution (Yue et al., 2025; Wu et al., 2026). RLVR-trained models also tend to lose semantic and operation diversity relative to the original model (Dang et al., 2025), as shown in Figure 1. In this view, RLVR mainly reweights existing correct trajectories rather than discovering qualitatively new solution strategies, leaving an exploration gap in regions of the reasoning space that are unlikely under the initial policy but may contain more robust solutions.

Bridging this gap requires exploration that preserves the long-horizon dependencies needed for Chain-of-Thought (CoT) reasoning (Sutton & Barto, 1998). Existing approaches fall into three broad paradigms. **Action-space noise** such as temperature or nucleus sampling injects token-level stochasticity during decoding (Renze & Guven, 2024; Holtzman et al., 2019), but this noise is locally uncorrelated across time and can accumulate into incoherent reasoning. **Objective-level regularization**, including entropy bonuses and pass@k objectives (Zhan et al., 2025; Chen et al., 2025), explicitly rewards diversity but depends on proxy signals. **Data augmentation** broadens support through generated variants, offline data, or external teachers (Liang et al., 2025; Dong et al., 2025; Li et al., 2025), but typically adds compute, curation, or external supervision. Section 2.2 gives a

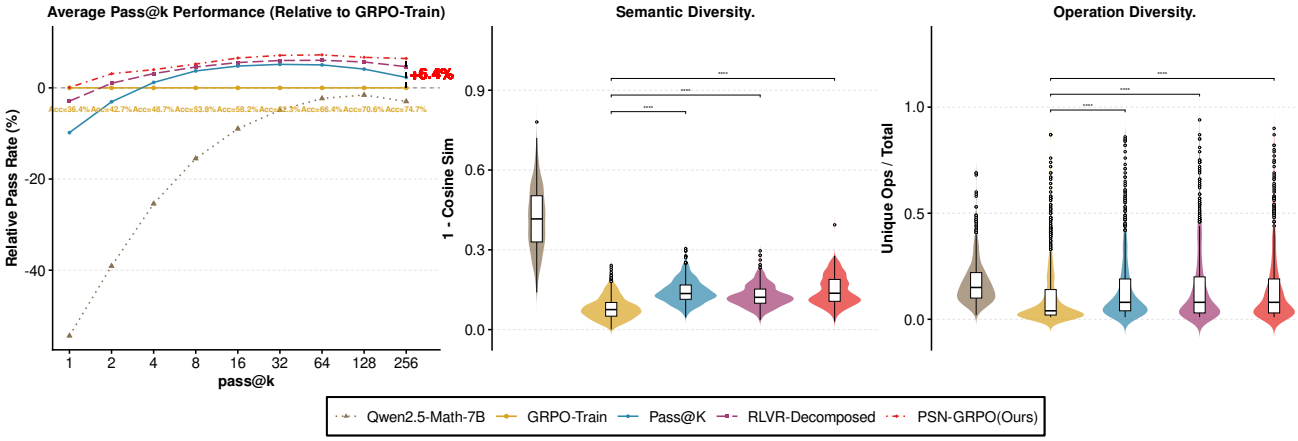

Figure 1. Comparison of reasoning capability boundaries across different RLVR paradigms. (1) Standard GRPO-trained models (GRPO-Train) exhibit reduced **semantic diversity** and **operation diversity** compared to the base model (Qwen2.5-Math-7B). (2) Our proposed method, **PSN-GRPO**, restores and enhances this diversity, achieving higher semantic and operation variance than the GRPO baseline. (3) In reasoning performance, PSN-GRPO outperforms exploration-focused methods such as **Pass@k** (Chen et al., 2025) and **RLVR-Decomposed** (Zhu et al., 2025), especially under large sampling budgets (e.g., $k = 256$).

compact comparison.

We instead study **Parameter-Space Noise** (PSN) (Plappert et al., 2018) for RLVR. PSN perturbs the parameters of the policy used to generate rollouts while maintaining a clean policy for learning. Unlike token-level perturbations, the same parameter perturbation governs an entire rollout, inducing temporally consistent, trajectory-level exploration that better matches long CoT reasoning. Applying PSN to RLVR raises two practical challenges: parameter perturbations create a sampling–update mismatch, and adaptive noise control based on perturbed-clean KL estimates is expensive for modern LLMs. We address these challenges with two lightweight modules: (i) **truncated importance sampling (TIS)** for stable learning from perturbed-policy rollouts, and (ii) a **real-time adaptive noise scheduler** that replaces expensive KL-based control with a surrogate combining semantic diversity and self-certainty.

To the best of our knowledge, this paper is the **first systematic study of parameter-space noise** for RLVR. Our main contributions are:

- **PSN-RLVR Framework.** We introduce a parameter-space noise framework for RLVR that injects noise directly into the parameters of the policy model responsible for rollout generation while updating the clean policy model. This induces temporally consistent, trajectory-level exploration. Instantiated on GRPO, we develop **PSN-GRPO**.
- **Two modules for RLVR-specific challenges.** To address the off-policy mismatch induced by parameter perturbations, we incorporate **truncated importance sampling (TIS)** for stable off-policy learning; to avoid expensive KL-based noise control, we propose a **lightweight real-time adaptive noise scheduler** driven by a surrogate that combines semantic diversity and model's self-certainty.
- **Comprehensive study of the PSN design space in RLVR.** We conduct extensive experiments and ablations that systematically answer key design questions—noise injection location, noise magnitude scaling, robustness across models, comparisons to action-space noise, OOD behavior, and composability with other exploration-oriented RLVR methods.

2. Related Work

2.1. Reinforcement Learning for Reasoning with Verifiable Rewards

RLVR has become a standard post-training paradigm for improving LLM performance in domains with objective ground truth, such as mathematics and coding (Ouyang et al., 2022; Lu et al., 2024; Guo et al., 2025). Appendix A gives the broader background on PPO, GRPO, and the exploration ceiling observed in recent RLVR systems.

2.2. The Exploration-Exploitation Boundary in LLMs

Existing exploration approaches for reasoning models fall into three categories: (1) action-space perturbations, (2) objective-level regularization, and (3) data augmentation. Action-space methods perturb decoding through temperature sampling (Renze & Guven, 2024), nucleus/top-k sampling (Holtzman et al., 2019), or prompt perturbations (Shur-Ofry et al., 2024). These heuristics can increase diversity, but they require careful tuning (Du et al., 2025), are sensi-

tive across domains and model families (Shi et al., 2024; Qiang et al., 2024; Chen et al., 2024), and introduce locally uncorrelated token noise that can disrupt long-horizon CoT coherence (Renze & Guven, 2024).

Training-time diversification instead modifies the RLVR objective, for example by replacing pass@1 optimization with pass@k objectives (Chen et al., 2025; Peng et al., 2025), using output entropy as an exploration signal (Cheng et al., 2025a; Cui et al., 2025), or incorporating negative samples (Zhu et al., 2025). These methods are promising but often depend on proxy signals whose utility varies with task difficulty and reward sparsity. Data and experience augmentation broaden exploration through self-generated task variants (Liang et al., 2025), offline data (Dong et al., 2025; Li et al., 2025), or additional LLM guidance (Jiang et al., 2025), typically at the cost of extra computation, data curation, or external signals.

2.3. Parameter-Space Noise

Parameter-Space Noise (PSN) offers a different route: perturbing policy weights rather than individual actions. Plappert et al. (Plappert et al., 2018) and Fortunato et al. (Fortunato et al., 2018) showed in conventional RL that PSN induces state-dependent exploration, where a perturbed policy behaves as a distinct agent that executes a consistent strategy over an episode. Recent LLM work has observed adjacent noise effects, such as QeRL's exploration gains from quantization noise (Huang et al., 2025). Our focus differs: we deliberately inject parameter-space noise into the RLVR rollout policy, study where and how strongly to inject it, and analyze how this structured exploration affects long-form reasoning. Appendix B provides additional PSN background and theoretical context.

3. Methodology

3.1. Preliminaries: Group Relative Policy Optimization

We build our framework upon Group Relative Policy Optimization (GRPO) (Guo et al., 2025), a widely used RLVR method, which optimizes a policy π_θ without a value function. For each query $q \sim P(Q)$, GRPO samples a group of outputs $\{o_i\}_{i=1}^{G}$ from the old policy $\pi_{\theta_{\text{old}}}$ and optimizes the following objective:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\, o\sim\pi_{\theta_{\text{old}}}(O|q)} \left[\frac{1}{|o|} \sum_{t=1}^{|o|} \ell_t^{\text{clip}}(\theta) \right], \tag{1}$$

where $\ell_t^{\text{clip}}(\theta) = \min(r_t(\theta)A_t,\ \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\, A_t)$, and $r_t(\theta) = \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}$ where A_t is the advantage computed from relative rewards within the group.

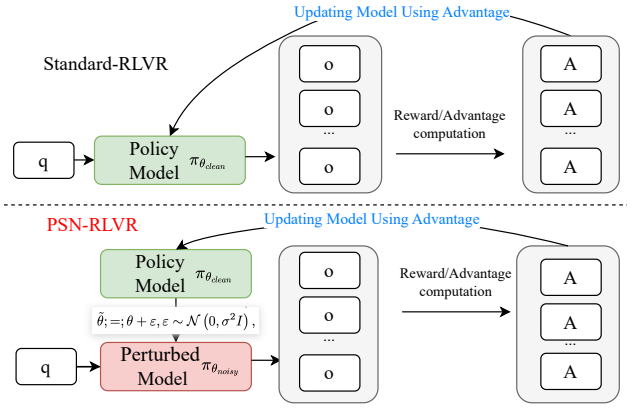

Figure 2. Overview of the PSN-RLVR framework compared to Standard-RLVR. The noise-perturbed model $\pi_{\tilde{\theta}}$ generates rollouts to induce temporally consistent, trajectory-level exploration. The resulting reward signals are used to update the clean policy π_θ

3.2. Parameter-Space Noise GRPO (PSN-GRPO)

We initialize PSN-RLVR with PSN-GRPO, an exploration-enhanced training framework that injects parameter-space noise into the rollout policy to induce temporally consistent exploration while updating the clean policy parameters via policy gradient learning. Because rollouts are generated by a noisy sampler policy, we correct the resulting off-policy mismatch using Truncated Importance Sampling (**TIS**). Finally, we propose a computationally efficient, real-time scheduler for noise. The main framework is illustrated in Figure 2.

3.3. Parameter-space noise exploration policy

Following Plappert et al. (2018), we explore by perturbing parameters rather than actions/tokens. At the beginning of each iteration, we apply additive Gaussian noise to the parameter vector of the current policy :

$$\tilde{\theta}; =; \theta + \varepsilon, \varepsilon \sim \mathcal{N}\left(0, \sigma^2 I\right), \tag{2}$$

which induces a *noisy exploration policy* $\pi_{\tilde{\theta}}$. Crucially, $\tilde{\theta}$ is held fixed for the entire rollout, yielding temporally consistent exploration: conditioned on the same prefix $(q, o_{<i})$.

3.4. Off-Policy Correction via Truncated Importance Sampling

A distribution mismatch arises because data is collected by $\pi_{\tilde{\theta}}$ but used to train π_θ. Ignoring this discrepancy yields biased gradient estimates. We correct this via Truncated Importance Sampling (TIS) (Ionides, 2008). We modify the standard GRPO objective by incorporating the importance ratio w_t into the gradient update. The corrected objective is:

$$\mathcal{J}_{\text{PSN}}(\theta) = \mathbb{E}_{q\sim P(Q), o\sim\pi_{\tilde{\theta}}} \left[\frac{1}{|o|} \sum_{t=1}^{|o|} w_t \cdot \ell_t^{\text{clip}}(\theta) \right]. \tag{3}$$

To prevent unbounded variance when $\pi_{\tilde{\theta}}$ diverges significantly from π_θ, we truncate the importance ratio:

$$w_t := \min\left(\frac{\pi_\theta(a_t)}{\pi_{\tilde{\theta}}(a_t)}, C\right) \tag{4}$$

where C is a clipping hyperparameter. This formulation allows the learner to leverage exploratory data from $\pi_{\tilde{\theta}}$ while maintaining training stability.

3.5. Adaptive Noise Scheduling

We would like the exploration policy to remain sufficiently close to the learner to keep off-policy correction stable, while still being different enough to encourage exploration. We therefore measure how far the noisy policy deviates from the clean policy on the *current* batch and adjust the noise for the **next** batch. We propose two adaptive scaling variants to dynamically adjust σ.

3.5.1. Variant I: Non-Real Scheduler

Following (Plappert et al., 2018), we adjust σ to maintain a target KL divergence, δ_{KL}, between the clean and noisy policies. After each batch, we compute:

$$d(\pi_\theta, \pi_{\tilde{\theta}}) = \mathbb{E}_s\left[\text{KL}\left(\pi_{\tilde{\theta}}(\cdot|s) \parallel \pi_\theta(\cdot|s)\right)\right]. \tag{5}$$

The noise scale for the next iteration, σ_{k+1}, is updated via:

$$\sigma_{k+1} = \begin{cases} \beta\sigma_k & \text{if } d(\pi_\theta, \pi_{\tilde{\theta}}) \le \delta_{\text{KL}} \\ \frac{1}{\beta}\sigma_k & \text{otherwise,} \end{cases} \tag{6}$$

where $\beta > 1$ is a step constant. However, this retrospective adaptation introduces feedback latency. Given the high variance in problem difficulty within RLVR datasets, this lag can lead to suboptimal noise scheduling: high noise levels necessitated by a difficult problem batch may be inappropriately applied to a subsequent simple problem batch, causing a mismatch that hinders efficient training.

3.5.2. Variant II: Real-Time Computationally Efficient Scheduler

To address the feedback lag and computational cost of full rollout sampling, we propose a real-time scheduler based on *semantic diversity* and *model self-certainty*. For each query q, we pre-generate two probe rollouts $o^{(1)}, o^{(2)}$ using the clean policy π_θ to gauge the model's current exploration needs.

Motivation. Ideally, accurate noise calibration for the current batch would require computing $d(\pi_{\text{clean}}, \pi_{\text{noisy}})$ *a priori*. However, this is computationally prohibitive. Since rollout generation accounts for 70–80% of total training time (Cheng et al., 2025b), a naive approach—sampling a set of rollouts (e.g., $N = 8$) to measure divergence, adjusting noise, and then resampling for backpropagation—would nearly double the computational overhead. To circumvent this, we propose a computationally efficient schedule that calibrates noise based on immediate signals of *semantic similarity* and *model self-certainty*. To this end, for each query q, we pre-generate two rollouts, $o^{(1)}$ and $o^{(2)}$, using π_{clean} to compute indicators based on semantic embeddings (Dang et al., 2025) and self-certainty (Kang et al., 2025). We explicitly avoid using $KL(\pi_{\text{clean}}||\pi_{\text{noisy}})$ for this adjustment because "KL between language models notoriously suffers from high variance" (Amini et al., 2025; Fang et al., 2025), particularly when estimated from only two samples.

Semantic similarity. Let $f(\cdot)$ be a fixed sentence embedding model; we compute

$$d_{\text{sem},i} = \cos\big(f(o^{(1)}), f(o^{(2)})\big), \bar{d}_{\text{sem},t} = \frac{\sum_1^B d_{\text{sem},i}}{B} \tag{7}$$

where B is batch size, and $\bar{d}_{\text{sem},t}$ is mean of semantic similarity of current batch. Higher similarity indicates models tend to generate similar rollouts, signaling a need for greater exploration (i.e., larger parameter noise).

Self-Certainty. We quantify *distributional self-certainty* by measuring how far the model's token-level predictive distribution departs from a uniform prior over the vocabulary. The key intuition is that a sharper (more concentrated) distribution corresponds to greater confidence. Concretely, for a query q and a generated completion $o = (o_1, \ldots, o_{|o|})$, let U be the uniform distribution on the vocabulary V. We define self-certainty as the mean, across decoding steps, of the KL divergence from U to the model distribution p_{π_θ}:

$$\text{Self-certainty}(o \mid q) = \frac{1}{|o|}\sum_{i=1}^{|o|} \text{KL}\big(U \,\big\|\, p_{\pi_\theta}(\cdot \mid q, o_{<i})\big), \tag{8}$$

where $o_{<i}$ are the previously generated tokens and $p(j \mid q, o_{<i})$ is the model's predicted probability for token j at step i. Higher self-certainty values indicate greater confidence. Larger values imply stronger concentration of probability mass, i.e., greater deviation from uniformity and thus requiring more exploration. We normalize the batch-averaged self-certainty SC_t to $[0, 1]$ using a global history buffer of running extrema $(S_{\min}, S_{\max})$:

$$\overline{\text{SC}}_t^{\text{norm}} = \text{clip}\left(\frac{\text{SC}_t - S_{\min}}{S_{\max} - S_{\min} + \epsilon}, 0, 1\right). \tag{9}$$

Update Rule. We define a composite indicator $\text{Ind}_t = \overline{\text{SC}}_t^{\text{norm}} + \bar{d}_{\text{sem}}$ for current batch. A high Ind_t implies the model is both confident and producing semantically similar outputs, signaling a need for stronger perturbations. We compare Ind_t to its historical mean $\overline{\text{Ind}}$ to update σ:

$$\sigma_k = \begin{cases} \beta\sigma_{k-1} & \text{if } \overline{\text{Ind}} \le \text{Ind}_t \\ \frac{1}{\beta}\sigma_{k-1} & \text{otherwise} \end{cases} \tag{10}$$

Compute overhead. Variant II uses two probe generations per query. Empirically, we observe a smaller end-to-end

throughput reduction of $\approx 8\%$ relative to fixed-σ PSN under identical hardware. We attribute the gap to *generation-time imbalance* (He et al., 2025). More discussion is illustrated in Appendix E.

4. Experiments

4.1. Experimental setup

Training, Models, and Datasets. We adopt GRPO (Guo et al., 2025) as the default RLVR training algorithm. Unless otherwise specified, all experiments utilize the standard configuration of SimpleRL-Zoo[1] (Zeng et al., 2025). More detailed experimental setup is stated in Appendix C.

Evaluation Protocol. Main evaluations are conducted on five reasoning benchmarks: AIME 2024, AIME 2025, AMC 2023 (aim, 2025), OlympiadBench (He et al., 2024), and Minerva (Lewkowycz et al., 2022). To ensure fair comparison, all evaluations are conducted based on `simpleRL-reason`. To obtain a comprehensive evaluation, we adopt the unbiased pass@K metric with K up to 256 following (Peng et al., 2025), computed as pass@$K := \mathbb{E}_{x\sim\mathcal{D}}[1 - \binom{n-c}{K}/\binom{n}{K}]$, where c denotes the number of correct completions out of n generated responses. To reduce variance, we set $n = 300$ for all datasets. All results report averages over five training seeds. The OOD benchmarks including LiveCodeBench (Jain et al., 2024) and GPQA:Diamond (Rein et al., 2023).

Metrics representing reasoning capability boundary. We evaluate RLVR with pass@k for high-budget reasoning success (Yue et al., 2025; Peng et al., 2025; Chen et al., 2025), semantic and operation diversity for rollouts (Dang et al., 2025), and ReCEval for temporal consistency of long CoT (Prasad et al., 2023). Pass@k estimates whether at least one correct answer appears among k sampled rollouts, capturing the model's high-budget reasoning capability boundary (Yue et al., 2025; Peng et al., 2025; Chen et al., 2025). Semantic diversity measures variation among reasoning traces in embedding space, while operation diversity abstracts rollouts into arithmetic-operation sequences and measures procedural variation (Dang et al., 2025). ReCEval complements these diversity metrics by evaluating temporal consistency in long reasoning chains: we report entailment and contradiction-consistency scores, where higher values indicate that intermediate reasoning steps better support later conclusions and contain fewer internal contradictions (Prasad et al., 2023).

4.2. Results and Q&As

We present our results via the following Q&As.

[1]https://github.com/hkust-nlp/simpleRL-reason

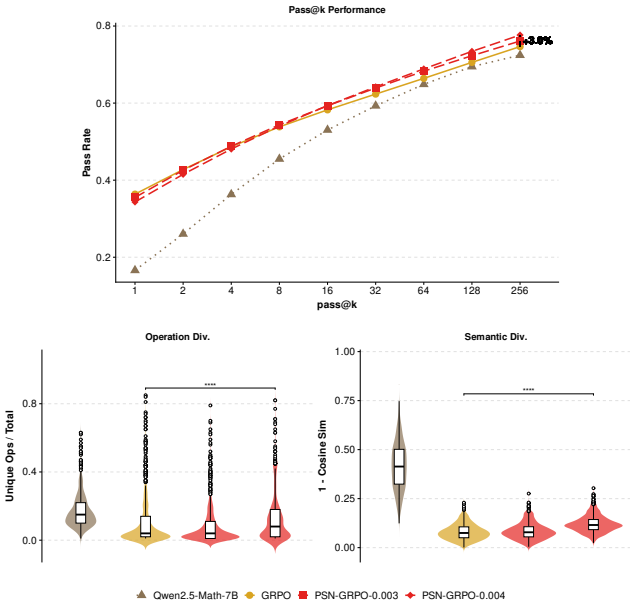

Figure 3. We compare the pass@k performance (top), semantic diversity , and operation diversity of PSN-GRPO against the standard GRPO baseline on Qwen2.5-Math-7B. PSN-GRPO achieves superior performance at large sampling budgets ($k \geq 16$). This gain is strongly correlated with increased semantic and operational diversity in generated trajectories (Dang et al., 2025).

Q1: Does PSN-RLVR expand the reasoning capability boundary?

A: Yes, PSN-RLVR achieves higher pass@k under large sampling budgets and increased rollouts semantic diversity and operation diversity.

We initialize the experiment with PSN-GRPO with **Qwen2.5-Math-7B** model. Figure 3 summarize the average performance across the evaluated datasets. We find that naive PSN-GRPO(without TIS and adaptive noise) outperforms the standard GRPO baseline when k is large (e.g., from $k = 16$ to $k = 256$). While standard RLVR improves selection efficiency among pre-existing trajectories, PSN effectively expands the reasoning search space. As shown in Figure 3, this performance boost correlates with significantly higher semantic and operation diversity compared to the baseline, confirming that PSN induces genuinely new reasoning modes rather than simply reweighting the pre-training distribution. Meanwhile, the naive PSN do hurt the pass rate when k is small. This will be mitigated through adaptive nosie mechanism and answered in Question 6. Appendix Table 8 further reports repeated-evaluation confidence intervals, where PSN-GRPO achieves the strongest pass@2, pass@4, and pass@256 means among the compared exploration methods.

Q2: Where should parameter-space noise be injected?

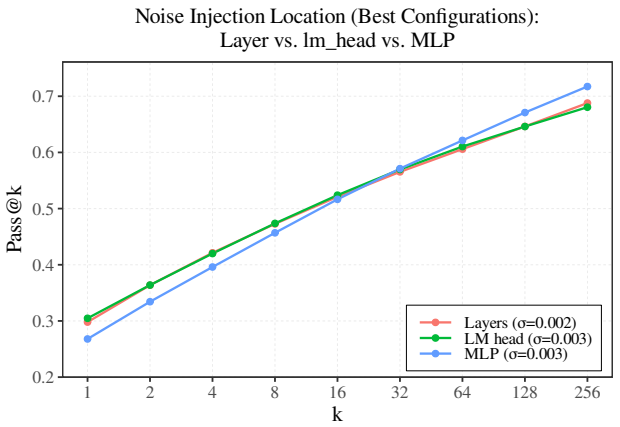

Figure 4. Noise injection location (best settings). Average pass@k for the best noise scale at each injection place (Whole Layers, `lm_head`, and MLP). MLP injection attains the largest gains at high k.

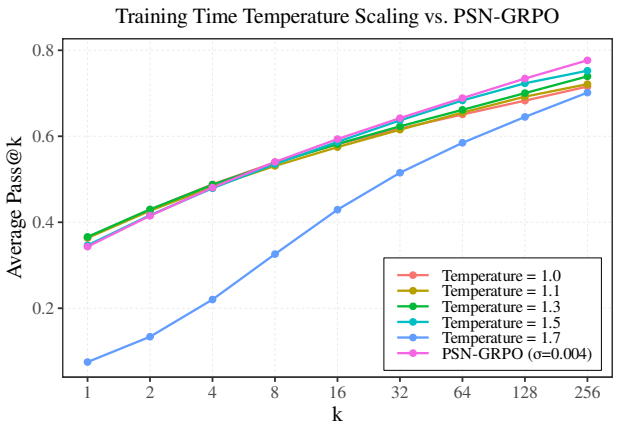

Figure 5. Average performance across benchmarks with different training time temperature. Increasing temperature beyond 1.5 degrades overall performance.

A: MLP/FFN layers are the most effective injection modules. They yield the strongest high-budget pass@k gains while preserving stable generation.

Theoretical work by Plappert et al. (Plappert et al., 2018) suggests that parameter-space noise is most effective when normalization separates perturbed layers. Standard Transformer architectures naturally satisfy this criterion in their MLP blocks. We therefore compare noise injection into full decoder layers, the language-modeling head (`lm_head`), and MLP/FFN blocks using the general-purpose **Qwen2.5-7B** model. As shown in Figure 4, MLP-only injection achieves the strongest high-budget pass@k performance, indicating a better stability–exploration trade-off than perturbing larger modules or the output head. Comprehensive results are provided in Appendix Figure 11.

Q3: Is Parameter space noise more effective than action-space noise with respect to long-trajectory(CoT) reasoning in RLVR method?

A: Yes, parameter-space noise outperforms both training and evaluation time action-space noise by preserving long-trajectory reasoning (CoT) consistency.

We evaluate the efficacy of parameter-space exploration against action-space noise baselines. First, we compare PSN-GRPO against training-time action-space noise, implemented via temperature scaling ($T \in \{1.0, \ldots, 1.7\}$) (Du et al., 2025). As shown in Figure 5, increasing temperature beyond 1.5 degrades performance. We attribute this failure mode to the local, unstructured nature of token-level perturbations: because action-space noise is typically uncorrelated across time steps, it accumulates into "logical drift" that disrupts the global coherence required for long-horizon Chain-of-Thought (CoT) reasoning.

In contrast, PSN perturbs the policy parameters prior to generation, inducing *trajectory-level consistency*. Consequently, PSN-GRPO demonstrates superior scaling behavior, with performance benefits becoming increasingly pronounced on long-horizon reasoning tasks, as shown in Table 1, and detailed results are shown in Appendix Figure 8. For example, on AIME 25 (average response length of 1,596 tokens), PSN-GRPO improves pass@256 over the best training-time and evaluation-time temperature baselines by $+3.8\%$ and $+4.1\%$, respectively. On AIME 24, PSN-GRPO also exceeds the corresponding baselines by $+1.9\%$ and $+2.9\%$. Furthermore, PSN-GRPO consistently exceeds the performance ceiling of evaluation-time temperature scaling (typically $T = 1.5$), which requires expensive tuning and lacks the coherent exploration necessary for difficult tasks as shown in Table 1, and detailed results are shown in Appendix Figure 9. ReCEval analysis (Prasad et al., 2023) provides a direct consistency check: PSN-GRPO achieves the strongest entailment and contradiction-consistency scores against base, temperature, and pass@k baselines (Appendix Table 9).

Q4: Is Truncated Importance Sampling (TIS) necessary?

A: Yes, TIS is crucial for stabilizing training by mitigating the off-policy mismatch.

Since PSN-RLVR generates rollouts using a perturbed policy π_{noisy} but updates a clean policy π_{clean}, a distribution mismatch arises. Table 2 compares standard GRPO and PSN-GRPO with and without TIS correction. We observe that while TIS provides negligible benefits to standard GRPO (where the sampling and training policies are identical), it significantly boosts the performance of PSN-GRPO

Dataset	Avg. Len (PSN)	Method	pass@256	
			Score	Gap
Olympiad	1246	PSN-GRPO	**78.9%**	-
		B.Train	77.9%	+1.0%
		B.Eval	78.7%	+0.2%
AIME 25	**1596**	PSN-GRPO	**65.4%**	-
		B.Train	61.6%	+3.8%
		B.Eval	61.3%	+4.1%
AIME 24	1375	PSN-GRPO	**74.6%**	-
		B.Train	72.7%	+1.9%
		B.Eval	71.7%	+2.9%

Table 1. **Performance comparison across mathematical benchmarks.** We report pass@256 scores (%) for PSN-GRPO ($\sigma = 0.004$) alongside the best training-time (B.Train) and best evaluation-time (B.Eval) temperature baselines. **Gap** denotes the absolute percentage point improvement of PSN-GRPO over the respective baseline. **Key Finding:** PSN-GRPO outperforms action-space noise, especially on long-horizon tasks such as AIME 25.

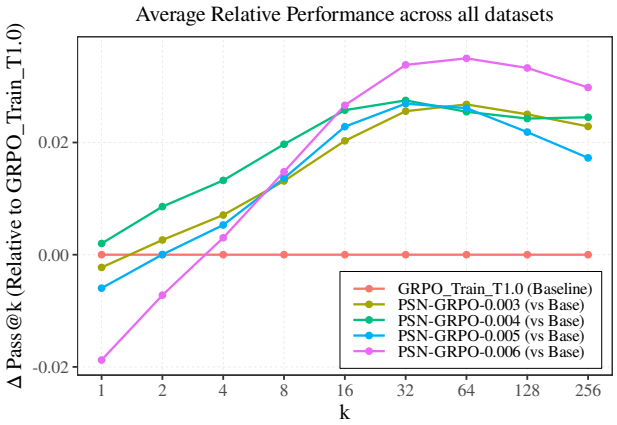

Figure 6. **Impact of Noise Magnitude σ on Exploration-Exploitation.** Increasing noise scale (e.g., $\sigma = 0.006$) maximizes the reasoning capability boundary at high k but degrades pass@1 performance. Moderate scales ($\sigma \in \{0.004, 0.005\}$) provide the optimal balance between exploitation reliability and exploratory diversity across benchmarks.

(e.g., increasing pass@256 from 74.33% to 76.94%). This confirms that properly handling the importance ratio clipping is essential when leveraging exploratory data from parameter-perturbed policies.

Method Name \ Avg of Datasets	pass@1	pass@128	pass@256
PSN-GRPO	36.01%	71.10%	74.33%
PSN-GRPO_with_TIS	**36.15%**	**73.07%**	**76.94%**

Table 2. **Impact of Truncated Importance Sampling (TIS).** The results show that TIS correction significantly boosts the performance of PSN-GRPO (e.g., increasing pass@256 from 74.33% to 76.94%) by effectively mitigating the off-policy mismatch between the perturbed sampling policy and the clean training policy.

Q5: How does performance scale with noise magnitude?

A: A moderate noise level (e.g., $\sigma \in \{0.004, 0.005\}$) tends to achieve the optimal exploration-exploitation balance. While larger noise (e.g., $\sigma = 0.006$) improves pass@k for larger k (e.g., $k = 128, 256$), it detrimentally affects performance at lower pass@k.

Figure 6 illustrates the trade-off inherent in noise magnitude selection. We find that larger noise scales (e.g., $\sigma = 0.006$) yield the highest pass@256 scores, indicating successful exploration of the outer reaches of the solution space. However, this comes at the cost of lower pass@1 performance compared to moderate noise levels (e.g., $\sigma \in \{0.004, 0.005\}$). For general applications, a moderate noise level provides the optimal balance between exploitation (reliability) and exploration (diversity).

Q6: Is adaptive noise scheduling better than fixed noise?

A: Yes, our proposed real-time certainty-aware scheduling achieves a superior exploration-stability trade-off.

We benchmark our compute-aware adaptive noise schedules (Variant I and Variant II, as detailed in Section 3.5) against fixed noise schedules and the GRPO baseline. The adaptive approach modulates σ based on semantic similarity and self-certainty signals (Equation 10). Our results highlight a critical trade-off: as shown in Table 3, and detailed result in Table 13. **Variant I** (non-real-time), despite incurring small inference overhead, suffers from feedback lag that degrades performance relative to standard GRPO. Conversely, **Variant II** (real-time), while introducing a marginal computational cost, effectively synchronizes noise scaling with the model's current state. This results in significant gains in both sample efficiency (e.g., pass@2) and the reasoning capability boundary (e.g., pass@256), consistently outperforming both fixed-noise baselines and naive GRPO. Appendix Table 10 further separates the source of the gains: adding TIS and the scheduler to a temperature baseline reaches only 73.2 pass@256, whereas PSN-GRPO without these modules already reaches 77.2 and the full method reaches 79.5. Appendix Table 11 shows that four probe rollouts and an alternative embedding backbone preserve strong performance, and Appendix Table 12 shows that self-certainty and semantic similarity are both useful scheduler signals, with their additive combination performing best.

Q7: Does PSN-RLVR generalize to other model families and OOD tasks?

Dataset	Method	pass@2		pass@4		pass@256	
		Score	Gap	Score	Gap	Score	Gap
Average	PSN Var-II	**44.1%**	-	**50.6%**	-	**79.5%**	-
	GRPO Train	42.7%	+1.3%	48.7%	+1.9%	74.7%	+4.8%
	PSN Fixed	43.6%	+0.5%	50.0%	+0.6%	77.1%	+2.4%
	PSN Var-I	42.5%	+1.5%	48.7%	+1.9%	75.1%	+4.4%

Table 3. **Performance comparison across mathematical benchmarks.** We report pass@k scores (%) for our adaptive variants (PSN Var-I/II) against the standard GRPO baseline and Fixed-Noise PSN ($\sigma = 0.004$). **Gap** denotes the absolute percentage point improvement of *PSN Var-II* over the compared method. Notably, the non-real-time schedule (Var-I) hurts performance due to feedback latency. In contrast, the proposed lightweight real-time schedule (Var-II) yields the best performance, improving both sample efficiency (e.g., +1.3% pass@2 vs. GRPO) and the reasoning capability boundary (e.g., +4.8% pass@256 vs. GRPO).

A: Yes. PSN-RLVR improves high-budget math reasoning across Qwen3 and Llama backbones and transfers to OOD science and coding benchmarks.

Beyond the default Qwen2.5-Math-7B setting, we evaluate PSN on **Qwen3-4B-Base** and **Llama-3.1-8B**. Appendix Tables 6 and 7 show consistent gains across model families: for Qwen3-4B-Base, fixed-sigma PSN improves average pass@256 from 70.70% to 74.56%, and Variant II reaches 76.10%; for Llama-3.1-8B, PSN improves the GRPO-trained backbone from 39.56% to 49.92% average pass@256. The same table also includes the raw Llama-3.1-8B base row and reports OOD gains on GPQA and LiveCodeBench, suggesting that the benefit is not limited to in-domain mathematical reasoning.

Q8: How does PSN-RLVR compared with other exploration methods and Is it orthogonal to these method?

A: Yes, PSN-RLVR outperforms other mainstream exploration baselines and is orthogonal to these methods, yielding additive performance gains when combined

First, we compare PSN-RLVR against two mainstreams method that increase explore capability pass@K (Chen et al., 2025) and RLVR Decomposed (Zhu et al., 2025) . As illustrated in Figure 1, PSN-RLVR not only achieves a superior reasoning capability boundary (higher pass@K at large budgets) but also exhibits significantly higher semantic and operation diversity compared to the baselines. This suggests that parameter-space perturbations induce novel reasoning modes that objective or data modifications alone fail to uncover. Second, we demonstrate that PSN acts as a complementary exploration mechanism. As shown in Table 4, combining PSN with pass@K training (PSN-GRPO-pass@K) further boosts performance, improving the average pass@256 from 76.37% to 79.12% over the strong GRPO-pass@K baseline. This result confirms that PSN is orthogonal to other strategies and can be effectively composed with them to maximize exploration.

Method Name \ Avg of Datasets	pass@64	pass@128	pass@256
GRPO-pass@K	69.77%	73.48%	76.37%
PSN-GRPO-pass@K	**70.35%**	**74.82%**	**79.12%**

Table 4. **PSN-GRPO is complementary to Pass@k training.** Combining Pass@k-based training with PSN-GRPO yields higher performance than either component alone.

Finding: PSN-RLVR discovers qualitatively new solution strategies. To probe whether the gains in *reasoning capability boundary* reflect genuine exploration (rather than mere reweighting), we conduct an Gemini-assisted (Team, 2025a) qualitative analysis on AIME 2024 under our standard evaluation protocol (30 problems; n=300 rollouts per problem). We focus on the subset of problems where the *base/original* model fails on all 300 rollouts, yet PSN-RLVR produces at least one correct rollout. Across these cases, we find that the successful PSN-RLVR traces typically employ solution perspective that are absent from the base model's rollout set, indicating that PSN-RLVR can access *new reasoning modes* rather than only improving selection among pre-existing trajectories. Representative examples are provided in Appendix R.

5. Conclusion

We introduced PSN-RLVR, the first systematic study of parameter-space noise for RLVR-trained language models, motivated by the observation that standard RLVR can saturate, primarily by improving selection among already-likely trajectories. By perturbing parameters rather than tokens to produce rollout-consistent exploration, PSN-RLVR improves long-horizon reasoning and yields larger gains under high sampling budgets. We showed that injecting noise into Transformer MLP/FFN blocks offers a favorable stability–exploration trade-off, and that truncated importance sampling is essential to reliably exploit exploratory data from perturbed policies. The expanded evaluation further shows that PSN transfers across Qwen and Llama model families, improves OOD science and coding results on GPQA and LiveCodeBench, and produces stronger ReCEval temporal-consistency scores. Finally, our lightweight certainty- and semantics-aware scheduling achieves robust performance without large rollout overhead, and PSN composes with existing RLVR exploration strategies to further extend the achievable pass@k frontier.

References

Aime. aime problems and solutions,, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Amini, A., Vieira, T., and Cotterell, R. Better estimation of the kullback–leibler divergence between language models, 2025. URL https://arxiv.org/abs/2504.10637.

Aravindan, S. and Lee, W. S. State-aware variational thompson sampling for deep q-networks. In *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems*, pp. 124–132. ACM, 2021. doi: 10.5555/3463952.3463973.

Chen, A., Phang, J., Parrish, A., Padmakumar, V., Zhao, C., Bowman, S. R., and Cho, K. Two failures of self-consistency in the multi-step reasoning of llms, 2024. URL https://arxiv.org/abs/2305.14279.

Chen, Z., Qin, X., Wu, Y., Ling, Y., Ye, Q., Zhao, W. X., and Shi, G. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models. *ArXiv*, abs/2508.10751, 2025. URL https://api.semanticscholar.org/CorpusID:280649795.

Cheng, D., Huang, S., Zhu, X., Dai, B., Zhao, W. X., Zhang, Z., and Wei, F. Reasoning with exploration: An entropy perspective, 2025a. URL https://arxiv.org/abs/2506.14758.

Cheng, R., Zhou, K., Wei, X., Liu, S., Han, M., Ai, M., Zhou, Y., Zhong, B., Xiao, W., Chen, R., and Chen, H. Fast llm post-training via decoupled and fastest-of-n speculation, 2025b. URL https://arxiv.org/abs/2511.16193.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Cui, G., Zhang, Y., Chen, J., Yuan, L., Wang, Z., Zuo, Y., Li, H., Fan, Y., Chen, H., Chen, W., Liu, Z., Peng, H., Bai, L., Ouyang, W., Cheng, Y., Zhou, B., and Ding, N. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL https://arxiv.org/abs/2505.22617.

Dang, X., Baek, C., Kolter, J. Z., and Raghunathan, A. Assessing diversity collapse in reasoning. In *ICLR 2025 Workshop on Scaling Self-Improving Foundation Models without Human Supervision (SSI-FM)*, 2025. URL https://openreview.net/forum?id=AMiKsHLjQh.

Dong, Y., Jiang, X., Tao, Y., Liu, H., Zhang, K., Mou, L., Cao, R., Ma, Y., Chen, J., Li, B., Jin, Z., Huang, F., Li, Y., and Li, G. Rl-plus: Countering capability boundary collapse of llms in reinforcement learning with hybrid-policy optimization, 2025. URL https://arxiv.org/abs/2508.00222.

Du, W., Yang, Y., and Welleck, S. Optimizing temperature for language models with multi-sample inference, 2025. URL https://arxiv.org/abs/2502.05234.

Fang, L., Wang, Y., Liu, Z., Zhang, C., Jegelka, S., Gao, J., Ding, B., and Wang, Y. What is wrong with perplexity for long-context language modeling?, 2025. URL https://arxiv.org/abs/2410.23771.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Legg, S., et al. Noisy networks for exploration. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1706.10295.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Gravell, B. and Summers, T. Robust learning-based control via bootstrapped multiplicative noise, 2021. URL https://arxiv.org/abs/2002.10069.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Xu, H., Ding, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Chen, J., Yuan, J., Tu, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., You, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Zhou, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y.,

Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., and Levine, S. Meta-reinforcement learning of structured exploration strategies. In *Neural Information Processing Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:3418899.

He, C., Luo, R., Bai, Y., Hu, S., Thai, Z., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., 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.

He, J., Li, T., Feng, E., Du, D., Liu, Q., Liu, T., Xia, Y., and Chen, H. History rhymes: Accelerating llm reinforcement learning with rhymerl, 2025. URL https://arxiv.org/abs/2508.18588.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Hollenstein, J. J., Auddy, S., Saveriano, M., Renaudo, E., and Piater, J. H. Action noise in off-policy deep reinforcement learning: Impact on exploration and performance. *ArXiv*, abs/2206.03787, 2022. URL https://api.semanticscholar.org/CorpusID:249461896.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. *ArXiv*, abs/1904.09751, 2019. URL https://api.semanticscholar.org/CorpusID:127986954.

Huang, W., Ge, Y., Yang, S., Xiao, Y., Mao, H., Lin, Y., Ye, H., Liu, S., Cheung, K. C., Yin, H., Lu, Y., Qi, X., Han, S., and Chen, Y. Qerl: Beyond efficiency – quantization-enhanced reinforcement learning for llms, 2025. URL https://arxiv.org/abs/2510.11696.

Ionides, E. L. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008. doi: 10.1198/106186008X320456.

Jain, N., Han, K., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Jiang, Z., Han, J., Li, T., Wang, X., Jiang, S., Liang, J., Dai, Z., Ma, S., Yu, F., and Xiao, Y. Selective expert guidance for effective and diverse exploration in reinforcement learning of llms, 2025. URL https://arxiv.org/abs/2510.04140.

Kang, Z., Zhao, X., and Song, D. Scalable best-of-n selection for large language models via self-certainty, 2025. URL https://arxiv.org/abs/2502.18581.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafjord, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

LI, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S. C., Rasul, K., Yu, L., Jiang, A., Shen, Z., Qin, Z., Dong, B., Zhou, L., Fleureau, Y., Lample, G., and Polu, S. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Li, J., Lin, H., Lu, H., Wen, K., Yang, Z., Gao, J., Wu, Y., and Zhang, J. Questa: Expanding reasoning capacity in llms via question augmentation, 2025. URL https://arxiv.org/abs/2507.13266.

Liang, X., Li, Z., Gong, Y., Shen, Y., Wu, Y. N., Guo, Z., and Chen, W. Beyond pass@1: Self-play with variational

problem synthesis sustains rlvr, 2025. URL https://arxiv.org/abs/2508.14029.

Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.-W., Galley, M., and Gao, J. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, 2024. URL https://arxiv.org/abs/2310.02255.

Osband, I., Van Roy, B., Russo, D. J., and Wen, Z. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

Ouyang, L. et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Peng, R., Ren, Y., Yu, Z., Liu, W., and Wen, Y. Simko: Simple pass@k policy optimization, 2025. URL https://arxiv.org/abs/2510.14807.

Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., and Andrychowicz, M. Parameter space noise for exploration. In *International Conference on Learning Representations*, 2018.

Prasad, A., Saha, S., Zhou, X., and Bansal, M. ReCEval: Evaluating reasoning chains via correctness and informativeness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10066–10086, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.622. URL https://aclanthology.org/2023.emnlp-main.622/.

Qiang, Y., Nandi, S., Mehrabi, N., Steeg, G. V., Kumar, A., Rumshisky, A., and Galstyan, A. Prompt perturbation consistency learning for robust language models, 2024. URL https://arxiv.org/abs/2402.15833.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Renze, M. and Guven, E. The effect of sampling temperature on problem solving in large language models. *ArXiv*, abs/2402.05201, 2024. URL https://api.semanticscholar.org/CorpusID:267547769.

Russo, D. Worst-case regret bounds for exploration via randomized value functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 14410–14420, 2019.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Shi, C., Yang, H., Cai, D., Zhang, Z., Wang, Y., Yang, Y., and Lam, W. A thorough examination of decoding methods in the era of llms. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:267627384.

Shur-Ofry, M., Horowitz-Amsalem, B., Rahamim, A., and Belinkov, Y. Growing a tail: Increasing output diversity in large language models. *ArXiv*, abs/2411.02989, 2024. URL https://api.semanticscholar.org/CorpusID:273821765.

Sutton, R. S. and Barto, A. G. Reinforcement learning: An introduction. *IEEE Trans. Neural Networks*, 9:1054–1054, 1998. URL https://api.semanticscholar.org/CorpusID:60035920.

Team, G. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *ArXiv*, abs/2507.06261, 2025a. URL https://api.semanticscholar.org/CorpusID:280151524.

Team, Q. Qwen2.5 technical report, 2025b. URL https://arxiv.org/abs/2412.15115.

Wu, F., Xuan, W., Lu, X., Liu, M., Dong, Y., Harchaoui, Z., and Choi, Y. The invisible leash: Why rlvr may or may not escape its origin, 2026. URL https://arxiv.org/abs/2507.14843.

Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., Lu, K., Xue, M., Lin, R., Liu, T., Ren, X., and Zhang, Z. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Yue, Y., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., and He, J. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

Zhan, G., Wang, L., Wang, P., Zhang, F., Duan, J., Tomizuka, M., and Li, S. E. Mind your entropy: From maximum entropy to trajectory entropy-constrained rl. *ArXiv*, abs/2511.11592, 2025. URL https://api.semanticscholar.org/CorpusID:283071650.

Zhu, X., Xia, M., Wei, Z., Chen, W.-L., Chen, D., and Meng, Y. The surprising effectiveness of negative reinforcement in llm reasoning, 2025. URL https://arxiv.org/abs/2506.01347.

A. Related works for Reinforcement Learning for Reasoning with Verifiable Rewards

The integration of Reinforcement Learning (RL) into the post-training pipeline of Large Language Models (LLMs) has become a standard paradigm for enhancing performance in domains with objective ground-truth, such as mathematics and coding (Ouyang et al., 2022; Lu et al., 2024). Early approaches primarily utilized Proximal Policy Optimization (PPO) (Schulman et al., 2017) to align models with reward signals derived from unit tests or symbolic solvers. More recently, Group Relative Policy Optimization (GRPO) (Guo et al., 2025) has gained prominence by eliminating the need for a separate value network, instead normalizing advantages within a group of sampled outputs. This efficiency has enabled massive-scale RL training, exemplified by models like DeepSeek-R1, which demonstrate emergent reasoning capabilities solely through reinforcement signals. However, emerging evidence suggests that current RLVR pipelines may face an *exploration ceiling*. Recent analyses (Yue et al., 2025; Wu et al., 2026) show that RLVR predominantly improves *selection efficiency* among pre-existing trajectories, with limited ability to generate genuinely new reasoning modes, leaving the learned policy largely constrained by the base model's pretraining distribution and exposing a critical exploration bottleneck.

B. Related works for parameter-space noise and theoretical foundation of PSN

While PSN is well-established in robotics (Fortunato et al., 2018; Gupta et al., 2018; Hollenstein et al., 2022; Gravell & Summers, 2021), its application to the discrete, high-dimensional reasoning space of RLVR remains largely unexplored. Subsequent theoretical work situates noise-injected policies within posterior sampling and randomized value functions, which can induce deep exploration and admit regret guarantees (Osband et al., 2019; Russo, 2019). Variational Thompson-sampling perspectives further interpret practical parameter-noise methods as tractable approximations to posterior sampling over value functions, providing a theoretical foundation for their empirical effectiveness (Aravindan & Lee, 2021). A concurrent direction is QeRL (Huang et al., 2025), which improves RLVR training efficiency through quantization and observes exploration benefits from quantization noise as a side effect. In contrast, we deliberately inject parameter-space noise into the rollout policy and systematically study the resulting noise dynamics, injection strategies, and scaling behavior for reasoning.

C. Training experiment setting

All experiments utilize the standard configuration of SimpleRL-Zoo[2] (Zeng et al., 2025) including learning rate, batch size, rollout count, and maximum sequence length; detailed hyperparameters are provided in Appendix Table 5. We test our method on Qwen2.5-Math-7B (Yang et al., 2024),Qwen2.5-7B (Team, 2025b),Qwen3-4B-Base (Yang et al., 2025) and Llama-3.1-8B (Grattafiori et al., 2024). Following the SimpleRL-Reason protocol (Zeng et al., 2025), training data is sampled from the NuminaMath dataset (LI et al., 2024), which is derived from GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). Unless explicitly stated otherwise, all reported results rely on the Qwen2.5-Math-7B model trained under these default settings. All training are done on H200 $\times$ 8 node, and inference are done on RTX4090. The training time is around 1 day, and evaluation time is around 2 days because large number(300) of rollouts for all dataset.

D. Detailed metric definitions.

Pass@k estimates whether at least one correct answer appears among k sampled rollouts, capturing the model's high-budget reasoning capability boundary (Yue et al., 2025; Peng et al., 2025; Chen et al., 2025). Semantic diversity measures variation among reasoning traces in embedding space, while operation diversity abstracts rollouts into arithmetic-operation sequences and measures procedural variation (Dang et al., 2025). ReCEval complements these diversity metrics by evaluating temporal consistency in long reasoning chains: we report entailment and contradiction-consistency scores, where higher values indicate that intermediate reasoning steps better support later conclusions and contain fewer internal contradictions (Prasad et al., 2023).

E. Compute overhead.

Variant II uses two probe generations per query under π_θ to estimate semantic similarity and self-certainty before sampling the G training rollouts under $\pi_{\tilde{\theta}}$. With $G = 8$, a naive upper-bound estimate suggests $2/8 = 0.25$ additional generations time; if on-policy rollout generation occupies at least 70% of wall-clock training time, this would imply an expected

[2]https://github.com/hkust-nlp/simpleRL-reason

Table 5. Hyperparameters used for SimpleRL-Zoo models.

Settings	**SimpleRL-Zoo**
Framework	verl (Sheng et al., 2024)
Prompt Batch Size	512
Mini-batch Size	256
# Policy Rollout G	8
Max Rollout Length	8,192
Max Generation Length (Eval)	16,384
Clip Ratio	0.2
KL Loss Coefficient	1×10^{-4}
Training Temperature	1.0
Evaluation Temperature	0.9
Adaptive Noise Update Step β	1.01
Adaptive Noise Scale Range	$[0.8\sigma_{\text{init}}, 1.2\sigma_{\text{init}}]$
Target Divergence δ	0.03
C in TIS Equation 4	10

slowdown of roughly $0.25 \times 0.70 \approx 0.17$ (i.e., $\approx 17\%$). Empirically, however, we observe a smaller end-to-end throughput reduction of per-iteration $\approx 8\%$ relative to fixed-σ PSN under identical hardware, batch size, and maximum generation length. We attribute the gap to *generation-time imbalance* (He et al., 2025): in batched/parallel decoding, shorter sequences finish earlier and GPUs can become partially idle while waiting for the longest sequences (stragglers) to complete. Adding a small number of short probe generations reduces the variance in per-step generation workload and mitigates tail-latency effects, leading to a wall-clock overhead that is lower than the token-count-based estimate.

F. Detailed metric definitions

Pass@k estimates whether at least one correct answer appears among k sampled rollouts, capturing the model's high-budget reasoning capability boundary (Yue et al., 2025; Peng et al., 2025; Chen et al., 2025). Semantic diversity measures variation among reasoning traces in embedding space, while operation diversity abstracts rollouts into arithmetic-operation sequences and measures procedural variation (Dang et al., 2025). ReCEval complements these diversity metrics by evaluating temporal consistency in long reasoning chains: we report entailment and contradiction-consistency scores, where higher values indicate that intermediate reasoning steps better support later conclusions and contain fewer internal contradictions (Prasad et al., 2023).

G. Result of Qwen3-4B-Base

Benchmark	Clean ($\sigma = 0$)			PSN ($\sigma = 0.001$)			Gap (PSN−Clean)		
	pass@1	pass@2	pass@256	pass@1	pass@2	pass@256	pass@1	pass@2	pass@256
AIME 24	17.6%	24.2%	65.2%	18.8%	23.6%	65.5%	+1.2	-0.6	+0.3
AIME 25	15.7%	21.1%	49.9%	14.7%	20.0%	62.6%	-1.0	-1.1	+12.7
AMC 23	57.4%	69.9%	99.6%	57.2%	70.0%	99.6%	-0.2	+0.1	+0.0
Minerva-Math	31.2%	38.9%	61.6%	34.8%	42.8%	66.9%	+3.6	+3.9	+5.3
OlympiadBench	43.8%	52.8%	77.2%	45.7%	54.2%	78.2%	+1.9	+1.4	+1.0
Average	33.14%	41.38%	70.70%	34.24%	42.12%	74.56%	+1.10	+0.74	+3.86

Table 6. **Qwen3-4B-Base detailed results (clean vs. PSN).** Per-benchmark pass@k (%) for Qwen3-4B-Base comparing clean decoding ($\sigma = 0$) and PSN ($\sigma = 0.001$; MLP-only perturbation). The updated average row is computed over AIME 24, AIME 25, AMC 23, Minerva-Math, and OlympiadBench. Fixed-sigma PSN improves the average pass@1/pass@2/pass@256 over the clean baseline, with the largest average gain at pass@256; per-benchmark low-budget changes remain mixed.

H. Additional aggregate reasoning-boundary evidence

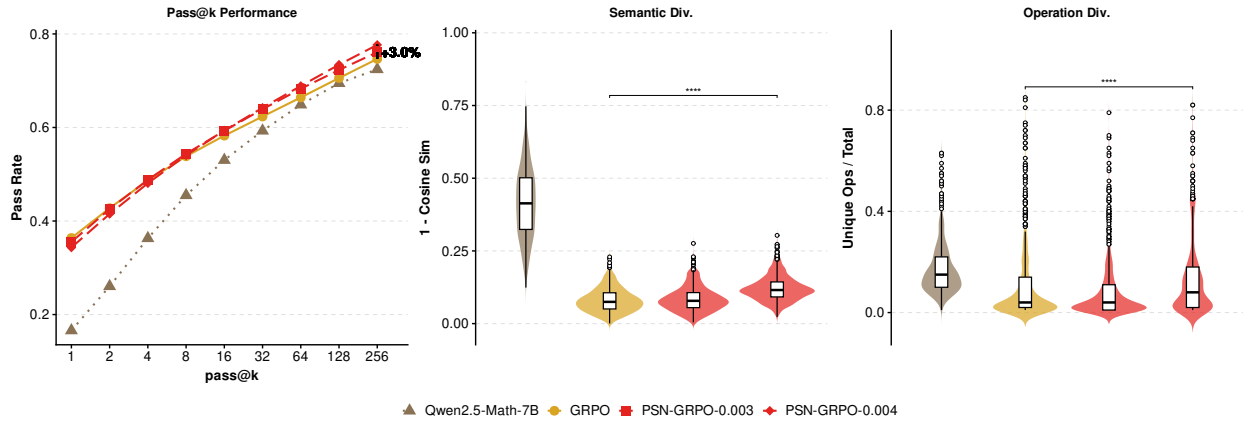

Figure 7. **Additional aggregate PSN-GRPO evidence.** This updated aggregate view complements Figure 3 by summarizing pass@k, semantic diversity, and operation diversity for PSN-GRPO against the GRPO baseline under the same reasoning-boundary evaluation protocol.

I. Additional evidence for model-family and OOD generalization

Method / backbone	Avg pass@1	Avg pass@2	Avg pass@256	GPQA pass@1	LCB pass@1	LCB pass@5	LCB pass@10
Qwen3-4B-Base-GRPO	33.14	41.38	70.70	0.3333	0.2215	0.3189	0.3469
Qwen3-4B-Base-PSN-GRPO-Fixed	34.24	42.12	74.56	0.3939	0.2222	**0.3566**	**0.3877**
Qwen3-4B-Base PSN-GRPO-Variant II	**34.48**	**42.28**	**76.10**	**0.4040**	**0.2442**	0.3269	0.3592
Llama-3.1-8B	2.00	3.74	46.56	0.0808	0.0960	0.1343	0.1488
Llama-3.1-8B-GRPO	3.60	4.70	39.56	0.0909	0.0883	0.1291	0.1460
Llama-3.1-8B-PSN-GRPO-Fixed	**4.02**	**6.84**	**49.92**	**0.1263**	**0.1001**	**0.1363**	**0.1507**

Table 7. **Model-family and OOD generalization evidence.** Additional experiments cover Qwen3-4B-Base and Llama-3.1-8B, with GPQA and LiveCodeBench (LCB) as OOD science and coding benchmarks. These results support evidence across multiple model families rather than a single-backbone claim. Fixed PSN-GRPO rows use $\sigma = 0.001$ without TIS or adaptive scheduling, while the Qwen3 Variant II row uses the adaptive schedule.

J. Additional evidence for temporal consistency and robustness

Method	pass@2 (95% CI)	pass@4 (95% CI)	pass@256 (95% CI)
GRPO	42.76 ([42.46, 43.03])	48.59 ([48.23, 48.94])	74.64 ([74.11, 75.17])
RLVR-Decomposed	42.96 ([42.69, 43.23])	50.12 ([49.75, 50.49])	78.14 ([77.64, 78.65])
GRPO-PASSK	41.35 ([41.05, 41.65])	49.19 ([48.88, 49.50])	76.01 ([75.55, 76.46])
PSN-GRPO	**44.10 ([43.82, 44.37])**	**50.49 ([50.12, 50.86])**	**79.53 ([79.00, 80.05])**

Table 8. **Repeated-evaluation confidence intervals.** We report pass@k means and 95% confidence intervals for pass@2/4/256 under the same unbiased pass@k evaluation protocol with $n = 300$. These intervals address high-budget and mid-budget reliability; no pass@1 statistical-significance claim is made here.

Metric	Baseline	Baseline mean	PSN-GRPO mean	p-value
entail_mean	Qwen2.5-Math-7B	0.7877	0.8434	< 0.0001
entail_mean	GRPO-Training_T1.3	0.8301	0.8434	< 0.0001
entail_mean	GRPO-Eval_T1.3	0.7456	0.8434	< 0.0001
entail_mean	pass@k training	0.7547	0.8434	< 0.0001
contradict_mean	Qwen2.5-Math-7B	0.6166	0.6918	< 0.0001
contradict_mean	GRPO-Training_T1.3	0.6604	0.6918	0.0017
contradict_mean	GRPO-Eval_T1.3	0.5603	0.6918	< 0.0001
contradict_mean	pass@k training	0.6660	0.6918	< 0.0001

Table 9. **ReCEval temporal-consistency comparison.** ReCEval (Prasad et al., 2023) evaluates rollout-level reasoning correctness and non-contradiction. PSN-GRPO obtains the highest entailment and contradiction-consistency scores against base, temperature, and pass@k baselines in these experiments.

K. Ablation study of TIS and scheduler with GRPO and PSN-GRPO

Method (Qwen2.5-Math-7B)	Avg pass@2	Avg pass@4	Avg pass@256
GRPO ($T = 1.5$)	36.4	44.1	74.7
GRPO ($T = 1.5$) + TIS	37.0	44.7	75.0
GRPO ($T = 1.5$) + TIS + scheduler	37.3	45.2	75.9
PSN-GRPO ($\sigma = 0.004$) w/o TIS/scheduler	43.7	49.0	77.2
PSN-GRPO + TIS + scheduler	**44.1**	**50.6**	**79.5**

Table 10. **Ablation separating PSN from stabilizing scaffolding.** Temperature plus TIS and the scheduler improves the baseline modestly, but does not match PSN-GRPO. This suggests that parameter-space perturbation is the primary source of the high-budget gain, while TIS and scheduling stabilize the trade-off.

Method	Avg pass@1	Avg pass@2	Avg pass@256
PSN-GRPO	44.07	50.62	79.46
PSN-GRPO (4 probes)	44.02	51.03	80.12
PSN-GRPO (Stella-400M-v5 embeddings)	42.91	50.14	78.81

Table 11. **Scheduler robustness.** Increasing from two to four probe rollouts yields only a small high-budget gain, and replacing the embedding backbone with Stella-400M-v5 preserves strong performance.

L. Ablation study of two modules in the real-time adaptive signal scheduler

Scheduler indicator	pass@2	pass@4	pass@256
Self-certainty only	43.5	49.8	77.8
Semantic similarity only	43.6	50.1	78.3
Additive combination (ours)	**44.1**	**50.6**	**79.5**

Table 12. **Scheduler component ablation.** Self-certainty and semantic similarity are useful independently, while their additive combination performs best in the ablation.

M. Detailed performance of training time temperature scaling

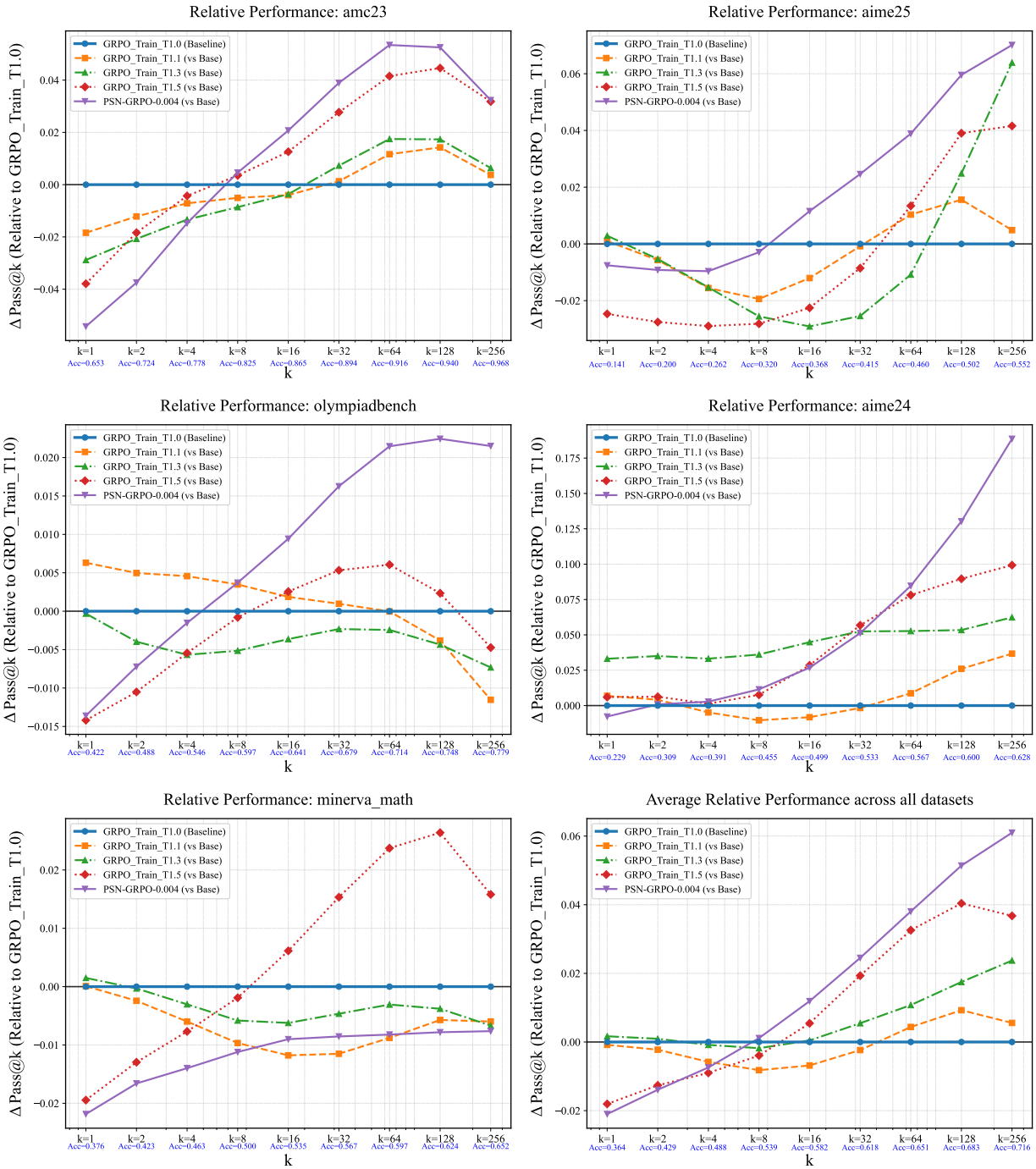

Figure 8. Performance across benchmarks. We do not plot T=1.7 in these plots since its overall performance collapsed as shown in Figure 5.

N. More experiment result of evaluation temperature scaling

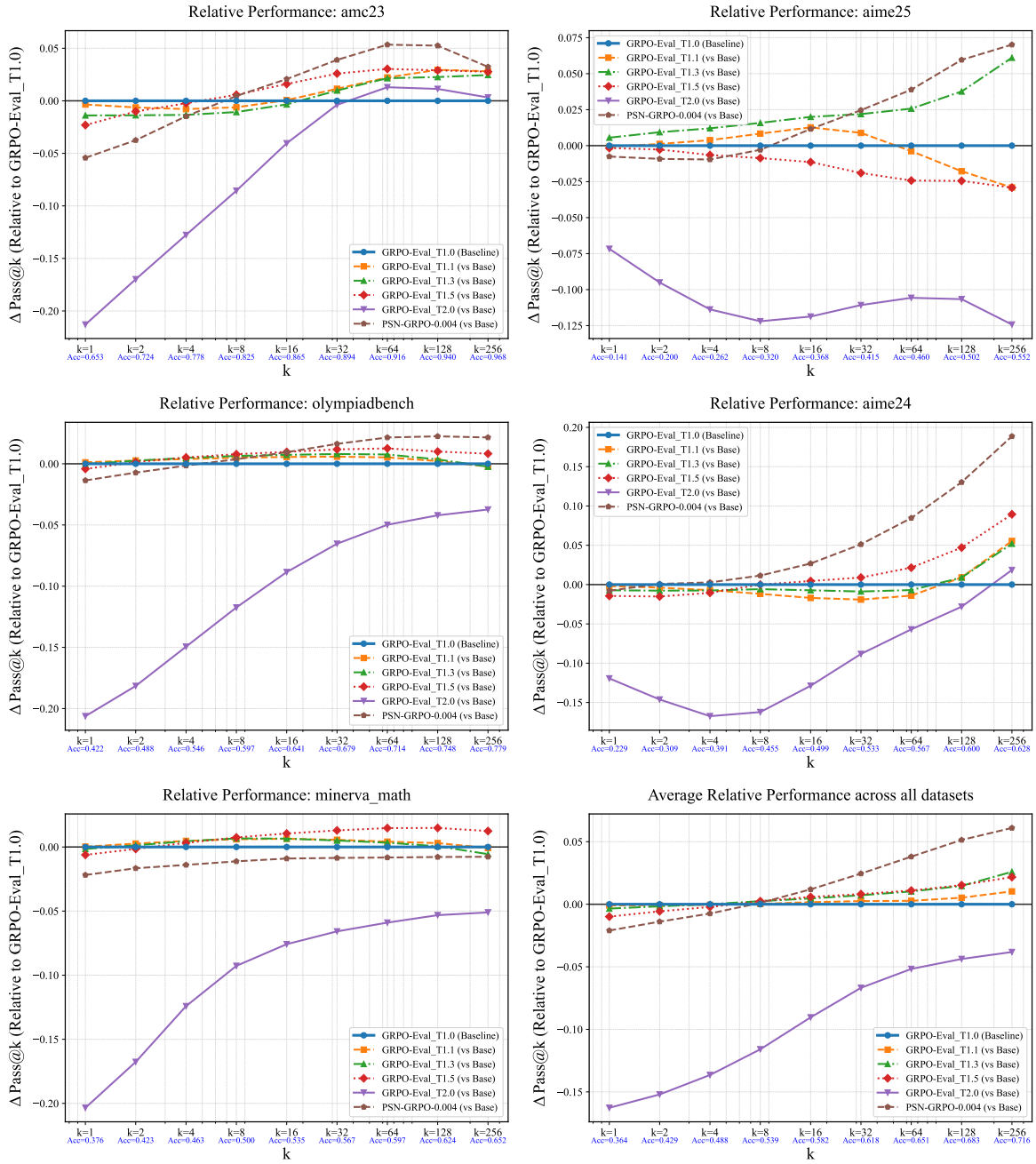

Figure 9. Detailed performance analysis across five datasets.

O. More detailed experiment result of evaluation temperature scaling

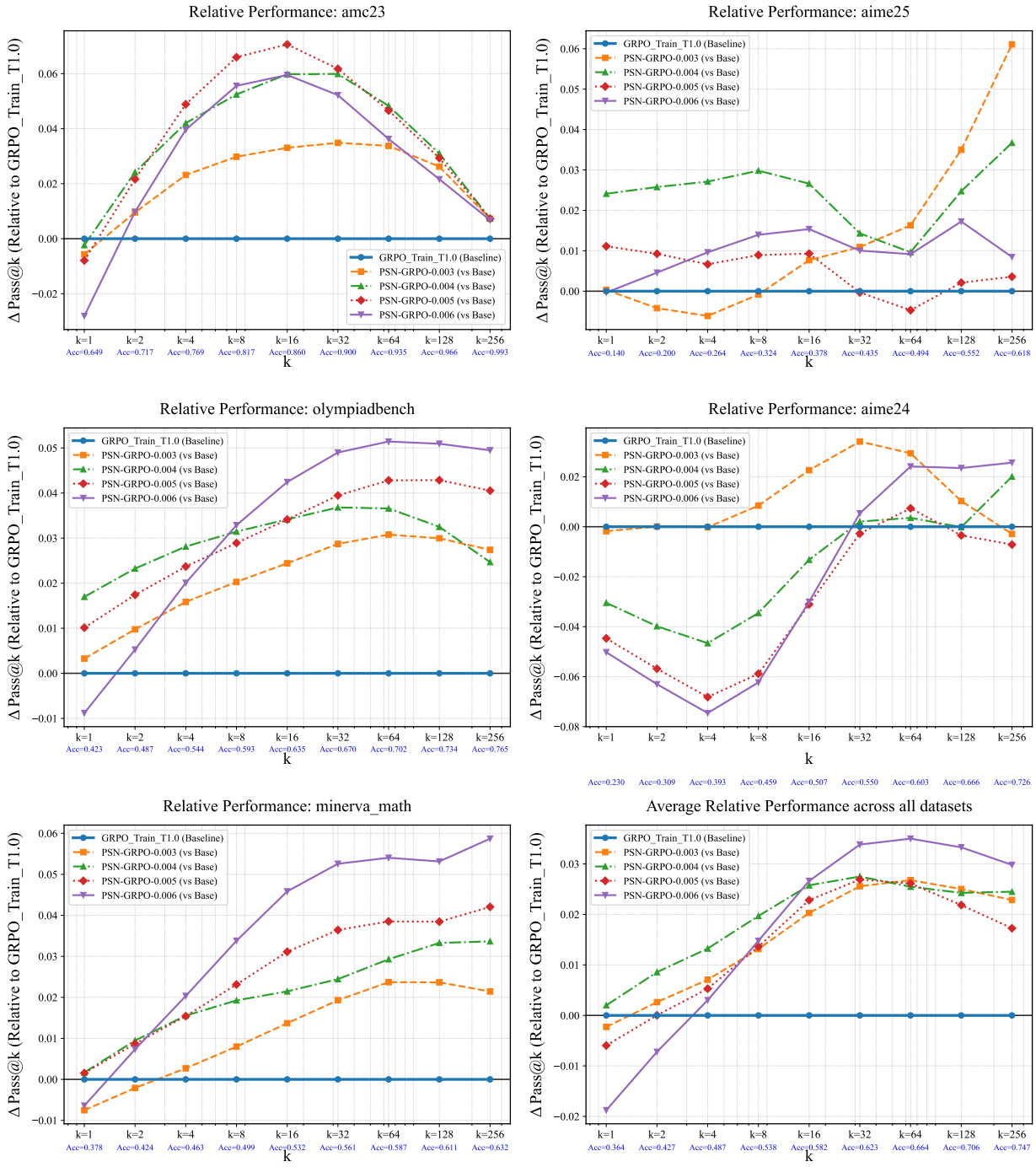

Figure 10. Detailed performance analysis across five datasets. In general, a larger noise scale σ tends to yield higher pass@k for large k. Specifically, on datasets of moderate difficulty (e.g., OlympiadBench and Minerva Math), increased σ improves reasoning capability boundary, yield higher pass@k for large k. Conversely, on more challenging benchmarks such as AIME24 and AIME25, a larger noise scale tends to degrade overall performance.

P. Detailed performance of where to inject noise

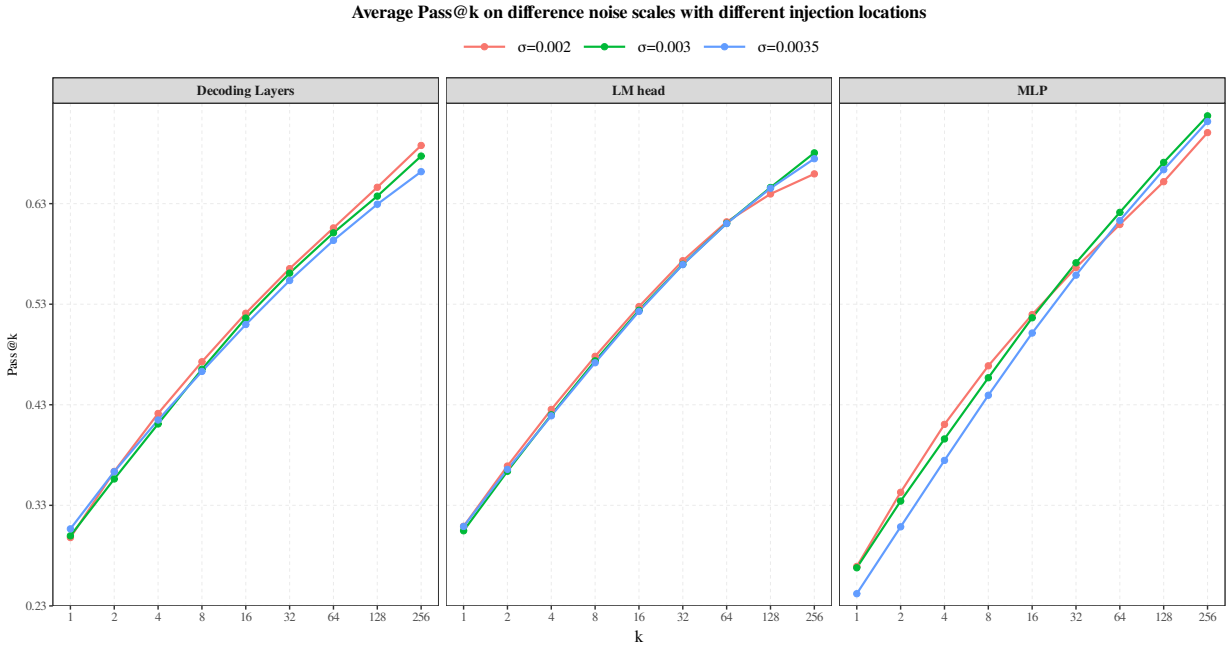

Figure 11. Average Pass@k under parameter-space noise injected whole transformer layers, the `lm_head`, or all MLP sublayers, sweeping noise scale σ. MLP injection yields the strongest and most consistent improvements, especially at large k.

Q. Detailed performance of adaptive noise

Dataset	Method	Pass@2		Pass@4		Pass@256	
		Score	Gap	Score	Gap	Score	Gap
AMC 23	PSN Var-II	**74.3%**	-	**81.8%**	-	99.6%	-
	GRPO Train	71.7%	+2.6%	76.9%	+4.8%	99.3%	+0.4%
	PSN Fixed	74.1%	+0.2%	81.1%	+0.6%	**100.0%**	-0.4%
	PSN Var-I	72.6%	+1.7%	79.2%	+2.6%	100.0%	-0.4%
Olympiad	PSN Var-II	**51.3%**	-	**57.7%**	-	**81.1%**	-
	GRPO Train	48.7%	+2.6%	54.4%	+3.3%	76.5%	+4.7%
	PSN Fixed	51.1%	+0.3%	57.2%	+0.5%	78.9%	+2.2%
	PSN Var-I	50.3%	+1.0%	56.4%	+1.3%	78.6%	+2.6%
AIME 25	PSN Var-II	**22.9%**	-	**29.2%**	-	**65.8%**	-
	GRPO Train	20.0%	+2.9%	26.4%	+2.8%	61.8%	+4.0%
	PSN Fixed	22.6%	+0.3%	29.1%	+0.1%	65.4%	+0.3%
	PSN Var-I	20.7%	+2.2%	26.9%	+2.3%	55.1%	+10.7%
AIME 24	PSN Var-II	28.1%	-	37.1%	-	**81.7%**	-
	GRPO Train	**30.9%**	-2.8%	**39.3%**	-2.2%	72.6%	+9.1%
	PSN Fixed	27.0%	+1.1%	34.6%	+2.4%	74.6%	+7.1%
	PSN Var-I	25.5%	+2.6%	32.9%	+4.2%	75.0%	+6.7%
Minerva	PSN Var-II	**43.7%**	-	47.4%	-	**69.1%**	-
	GRPO Train	42.4%	+1.4%	46.3%	+1.0%	63.2%	+5.9%
	PSN Fixed	43.3%	+0.4%	47.9%	-0.5%	66.6%	+2.5%
	PSN Var-I	43.5%	+0.3%	**48.0%**	-0.7%	66.7%	+2.4%
Average	PSN Var-II	**44.1%**	-	**50.6%**	-	**79.5%**	-
	GRPO Train	42.7%	+1.3%	48.7%	+1.9%	74.7%	+4.8%
	PSN Fixed	43.6%	+0.5%	50.0%	+0.6%	77.1%	+2.4%
	PSN Var-I	42.5%	+1.5%	48.7%	+1.9%	75.1%	+4.4%

Table 13. **Performance comparison across mathematical benchmarks.** We report Pass@k scores (%) for our adaptive variants (PSN Var-I/II) against the standard GRPO baseline and Fixed-Noise PSN ($\sigma = 0.004$). **Gap** denotes the absolute percentage point improvement of *PSN Var-II* over the compared method. Notably, the non-real-time schedule (Var-I) hurts performance due to feedback latency.

R. Detailed example of PSN-GRPO discovers qualitatively new solution strategies

Question (AIME I 2024, Problem 6)

Each vertex of a regular octagon is independently colored either red or blue with equal probability. The probability that the octagon can then be rotated so that all of the blue vertices end up at positions where there were originally red vertices is $\frac{m}{n}$, where m and n are relatively prime positive integers. What is $m + n$?

✗ Baseline Error Logic 1: Invariance Trap

The model incorrectly simplifies the problem by assuming the coloring must be invariant under a non-trivial rotation.

- It identifies $k \in \{0, 4, 8\}$ as the only counts allowing rotation.
- For $k = 4$, it only counts 2 symmetric patterns (alternating positions $\{1, 3, 5, 7\}$ and $\{2, 4, 6, 8\}$).

Total colorings = $1(k = 0) + 2(k = 4) + 1(k = 8) = 4$.

Diversity Collapse: The policy has collapsed to a small subset of globally symmetric solutions. It fails to recognize that "rotatable to red" simply requires the set of blue vertices B to satisfy $B \cap R_\theta(B) = \emptyset$ for at least one θ.

✗ Baseline Error Logic 2: Local Symmetry Constraint

For $k = 2$, the model argues that if vertices are at positions $\{1, 3\}$, they cannot be rotated because "the 90° separation does not align with the octagon's fundamental 45° or 180° symmetry axis." It erroneously prunes all non-opposite pairs.

Exploration Ceiling: This demonstrates a failure to simulate even a single-step 45° rotation ($1 \to 2, 3 \to 4$). The model incorrectly filters out valid configurations that do not satisfy its rigid internal symmetry priors, leading to a massive undercount.

✗ Baseline Error Logic 3: Counting Pruning Error

For $k = 3$ and $k = 4$, the model bypasses enumeration entirely. It claims that since $k \geq 3$ creates "too many constraints on adjacent slots," it is statistically impossible for a rotation to map all blue vertices to red positions unless the distribution is perfectly uniform (i.e., the $k = 4$ case in Error 1).

Reasoning Failure: The model uses pigeonhole-style heuristics incorrectly. It fails to explore asymmetric but valid sparse placements (e.g., $k = 3$ with no two adjacent). PSN-GRPO finds 40 and 46 cases for $k = 3, 4$, which the baseline misses due to premature search pruning.

✔ PSN-GRPO Answer (Ours)

PSN-GRPO identifies the problem's core requirement: finding all $B \subseteq V$ such that $\exists \theta \in \{45°, \ldots, 315°\}, R_\theta(B) \cap B = \emptyset$.

Comprehensive Strategy Discovery:

- $k = 0, 1$: All $\binom{8}{0} + \binom{8}{1} = 9$ colorings work.
- $k = 2$: Valid if not adjacent ($B = \{1, 2\}$ fails for all θ). Valid = $\binom{8}{2} - 8 = 20$.
- $k = 3$: PSN explores non-symmetric triplets. It correctly excludes triplets like $\{1, 2, 3\}$ and $\{1, 2, x\}$ where all rotations are blocked. Valid = $\binom{8}{3} - 16 = 40$.
- $k = 4$: Discovers 46 valid cases by maintaining diverse reasoning paths that avoid the "greedy trap" of symmetry.

Strategic Discovery: PSN-GRPO's parameter-space perturbation allows it to sustain multiple valid reasoning trajectories for complex counts ($k = 3, 4$), eventually summing to 115 favorable colorings.

Total Probability = $115/256$. Result: $m + n = 115 + 256 = \mathbf{371}$.