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

Bizhe Bai[2,1] **Xinyue Wang**[2] **Peng Ye**[3,4] **Tao Chen***[1,2]
bizhe.bai@sii.edu.cn, eetchen@fudan.edu.cn

[2] College of Future Information Technology, Fudan University, Shanghai, China
[1] Shanghai Innovation Institute, Shanghai, China
[3] Shanghai AI Laboratory, Shanghai, China
[4] The Chinese University of Hong Kong, Hong Kong, China

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 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) has become a central paradigm for improving the reasoning behaviors of Large Language Models (LLMs), delivering substantial gains on domains with automatic correctness signals such as mathematics and code generation [30, 35, 37]. In this setting, policy-gradient style algorithms—most notably Proximal Policy Optimization (PPO) [45] and its recent variants such as Group Relative Policy Optimization (GRPO) [18] propel systems like DeepSeek-R1 [18] to strong performance. However, emerging evidence suggests that current RLVR pipelines may face an *exploration ceiling*. Recent analyses [53, 59] indicate that RLVR tends to improve *sampling efficiency* (i.e., pass@1) rather than inducing genuinely new reasoning capability boundaries (i.e., pass@256) : the trajectories produced after RLVR are largely contained within (or near) the base model's pretraining distribution. In addition, the RLVR-trained LLM tend to have less semantic diversity and operation diversity compared to the original model [12] as shown in Figure 1 (right). Under this view, RLVR behaves primarily as a distributional *reweighting* mechanism—it amplifies pre-existing correct trajectories while rarely discovering qualitatively new solution strategies.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

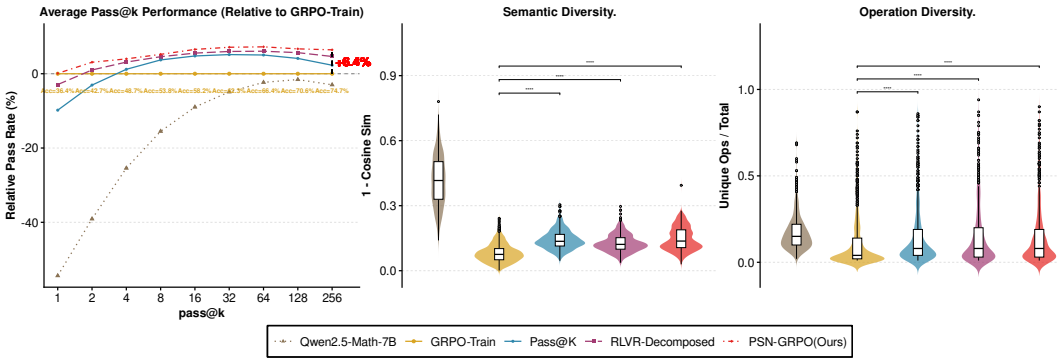


Figure 1: Comparison of reasoning capability boundaries across different RLVR paradigms. (1) Standard GRPO-trained models (GRPO-Train) exhibit a reduction in **semantic diversity** and **operation diversity** compared to the base model (Qwen2.5-Math-7B). (2) Our proposed method, **PSN-GRPO**, restores and enhances this diversity, achieving significantly higher semantic and operation variance compared to the GRPO baseline. (3) In terms of reasoning performance, PSN-GRPO is superior to other exploration-focused methods, such as **Pass@k** [7] and **RLVR-Decomposed** [62], consistently delivering higher **pass@k** metrics, particularly under large sampling budgets (e.g., $k = 256$).

This leaves an "exploration gap": the inability to reliably traverse regions of the reasoning space that are unlikely under the initial policy but may contain superior or more robust solutions.

Bridging this gap requires balancing exploration [49] and exploitation while preserving the long-horizon dependency essential for Chain-of-Thought (CoT) reasoning. Existing approaches generally fall into three paradigms, each with distinct limitations: **1. Action-Space noise (Decoding-Time):** Techniques such as temperature sampling [43] or nucleus sampling [24] inject token-level stochasticity at each step of the auto-regressive decoding process. However, this token-level noise is inherently *uncorrelated* across time steps. In multi-step reasoning, small perturbations in each step may accumulate into unstructured noise that degrades global coherence, harming long-horizon reasoning trajectories [43], rendering the CoT temporal-level inconsistent. **2. Objective-Level Regularization:** Methods that modify the training objective—such as entropy bonuses [61] or pass@k optimization [7]—attempt to force diversity explicitly. **3. Data Augmentation:** While experience augmentation (e.g., self-generated task variants, offline data, or external teacher models) [13, 33, 34] can broaden support but often introduces additional computational cost or reliance on external signals. More detailed discussion is provided in Section 2.2.

Departing from these three established paradigms, we advocate for a fundamentally different approach to exploration: Parameter-Space Noise (PSN) [39]. Originally introduced in conventional RL, PSN injects noise directly into the model responsible for generating rollouts, while maintaining and updating a clean policy model. Unlike token-level perturbations, PSN induces temporally consistent, trajectory-level exploration that naturally aligns with the long-horizon dependencies inherent in CoT reasoning. Nevertheless, integrating PSN into the RLVR framework presents two challenges: off-policy mismatches and the severe computational overhead of adaptive noise control. Specifically, computing the KL divergence between the perturbed and clean models—a trivial operation for the small models used in conventional RL—becomes prohibitively expensive at the scale of modern LLMs. We address these two limitations with two modules: (i) truncated importance sampling (TIS) to stabilize optimization , and (ii) a real-time adaptive noise scheduler that replaces expensive KL-based control with a low-overhead 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 paradigm for enhancing performance in domains with ground-truth, such as mathematics and coding [18, 35, 37]. Beyond RLVR, related efforts improve neural models through architecture-search regularization in β-DARTS and β-DARTS++ [55, 56], subnetwork supervision in Stimulative Training++ [57], and dynamic weight composition in MoW-Merging [58]. These methods provide broader context for improving model optimization and parameterization, whereas our work focuses on parameter-space perturbations for exploration in RLVR. A more detailed discussion is provided in Appendix A.

2.2 The Exploration-Exploitation Boundary in LLMs and current solutions

Existing approaches generally fall into three categories:(1) Action-Space Perturbations, (2) Objective-Level Regularization, and (3) Data Augmentation. We summarize these classes and their limitations below. First, a class of methods focus exploration at *decoding time* by perturbing the action space—e.g., temperature sampling [43], nucleus/top-k sampling [24], or prompt perturbations [48]. Although these heuristics can increase diversity, they typically demand careful hyperparameter tuning [14] and can be sensitive across domains and model families [6, 41, 47]. More fundamentally, token-level stochasticity is often *locally uncorrelated* across time [43], so small perturbations can accumulate into unstructured jitter that hinder long-horizon CoT coherence.

Second, training-time diversification can be pursued by modifying the RLVR objective, such as substituting the pass@1 objective with pass@k objectives [7, 38] in GRPO [18], using output entropy as a proxy for exploration control [8, 11], or incorporating negative samples to promote exploration [62]. While promising, objective-level regularization often depends on proxy signals whose effectiveness can be sensitive to task difficulty and reward sparsity.

Third, another line of work relies on data and experience augmentation to broaden exploration, such as creating new task variations from the model itself [34], leveraging off-line data [13, 33], or using additional LLM guidance [28]. Although these approaches can expand the effective support of training, they typically bring additional computation cost or require extra data curation.

2.3 Parameter-Space Noise

Parameter-Space Noise (PSN) offers another compelling alternative by perturbing the policy weights. Works such as [39] and [16] demonstrated in continuous control domains that PSN induces state-dependent exploration: a perturbed policy "agent" that executes a consistent strategy over an entire episode.Recent LLM work has begun to explore adjacent directions: QeRL observes exploration benefits as a side effect of quantization noise [25] and B-Trans samples sequence-level model instances through normalization-layer stochasticity.Our focus differs in that we deliberately inject parameter- space noise into the rollout policy for RLVR, study where and how strongly to inject it. More detailed discussion and theoretical foundation of PSN are provided in Appendi B.

3 Methodology

3.1 Preliminary: Group Relative Policy Optimization

We build our framework upon GRPO [18], 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.

3.2 Preliminary: Parameter-space noise (PSN) exploration policy

Following Plappert et al. [39], we drive exploration by perturbing parameters rather than actions or tokens. At the beginning of each iteration, we apply additive Gaussian noise to the parametesr of the policy that is used for generating rollouts:

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

Formally, the vector θ to be perturbed can represent either the full parameter space Θ or a targeted subset $\Theta_{\text{sub}} \subseteq \Theta$. This induces a *noisy exploration policy* $\pi_{\tilde{\theta}}$. Crucially, $\tilde{\theta}$ is held fixed for the entire episode (a full trajectory).

3.3 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.

3.4 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) [26]. We modify the standard GRPO objective by incorporating the importance ratio w_t into the gradient update. The corrected objective is:

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

w_t is used to to prevent unbounded variance when $\pi_{\tilde{\theta}}$ diverges significantly from π_θ. The C is a clipping hyperparameter. This truncation allows learning from exploratory data from $\pi_{\tilde{\theta}}$ while maintaining training stability.

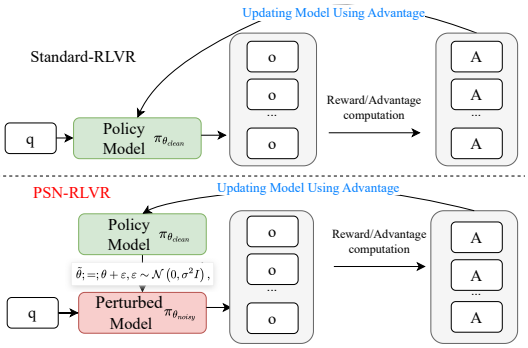


Figure 2: Overview of the PSN-RLVR (Bottom) framework compared to Standard-RLVR (Top). At beginning of each episode, the noise-perturbed model $\pi_{\tilde{\theta}}$ generates rollouts O_{noise} to induce temporally consistent, trajectory-level exploration. The resulting reward signals calculated from these rollouts, O_{noise}, are then used to compute the gradient and update the clean policy π_θ.

3.5 Adaptive Noise Scheduling

Parameter space noise requires us to pick a suitable scale σ. 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 [39], we adjust σ to maintain a target KL divergence, δ_{KL}, between the clean and noisy policies. After each batch, we compute:

$$d(\pi_\theta, \pi_{\tilde{\theta}}) = \mathbb{E}_s\left[\text{KL}\left(\pi_{\tilde{\theta}}(\cdot|s) \parallel \pi_\theta(\cdot|s)\right)\right]. \tag{4}$$

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

$$\sigma_{k+1} = \begin{cases} \beta\sigma_k & \text{if } d(\pi_\theta, \pi_{\tilde{\theta}}) \le \delta_{\text{KL}} \\ \frac{1}{\beta}\sigma_k & \text{otherwise,} \end{cases} \tag{5}$$

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 [9], 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 an efficient scheduler that combines *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 [12] and self-certainty [29]. We explicitly avoid using $KL(\pi_{\text{clean}}||\pi_{\text{noisy}})$ for this adjustment because "KL between language models notoriously suffers from high variance" [3, 15], particularly when estimated from only two samples.

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

$$d_{\text{sem},i} = \cos\big(f(o^{(1)}), f(o^{(2)})\big), \bar{d}_{\text{sem},t} = \frac{\sum_1^B d_{\text{sem},i}}{B}, \tag{6}$$

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 *self-certainty* by measuring how far the model's token-level predictive distribution departs from a uniform distribution over the vocabulary. The key intuition is that a sharper 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}:

$$\text{Self-certainty}(o \mid q) = \frac{1}{|o|} \sum_{i=1}^{|o|} \text{KL}\big(U \,\big\|\, p_{\pi_\theta}(\cdot \mid q, o_{<i})\big)\,, \tag{7}$$

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. The larger the value, the more exploration is needed. We normalize the batch-averaged self-certainty SC_t to $[0, 1]$ using a global history buffer of running extrema $(S_{\min}, S_{\max})$:

$$\overline{\text{SC}}_t^{\text{norm}} = \text{clip}\left(\frac{\text{SC}_t - S_{\min}}{S_{\max} - S_{\min} + \epsilon}, 0, 1\right). \tag{8}$$

Updating 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:

$$\sigma_k = \begin{cases} \beta\sigma_{k-1} & \text{if } \overline{\text{Ind}} \le \text{Ind}_t \\ \frac{1}{\beta}\sigma_{k-1} & \text{otherwise} \quad . \end{cases} \tag{9}$$

Compute overhead. Variant II utilizes two probe generations per query. Empirically, we observe a small end-to-end throughput increase of $\approx 8\%$ relative to fixed-σ PSN. This overhead is lower than a token-count estimate would predict; Appendix D discusses the effect in more detail.

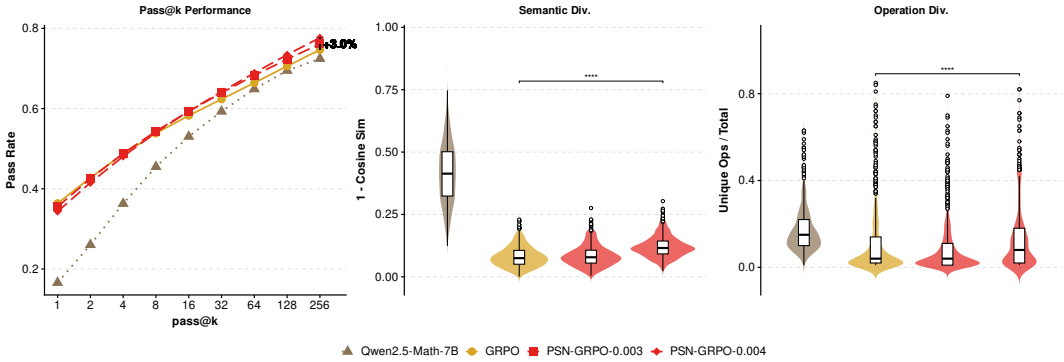


Figure 3: We compare the pass@k performance (top), semantic diversity , and operation diversity of PSN-GRPO against the standard GRPO baseline on Qwen2.5-Math-7B. PSN-GRPO achieves superior performance at large sampling budgets ($k \geq 16$). This gain is strongly correlated with increased semantic and operational diversity in generated trajectories [12].

4 Experiments Settings

4.1 Experimental setup

Training, Models, and Datasets. We adopt GRPO [18] as the default RLVR training algorithm. Unless otherwise specified, all experiments utilize the standard configuration of SimpleRL-Zoo[1] [60]. 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 [1], OlympiadBench [20], and Minerva [31]. 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 [38], 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 [27] and GPQA:Diamond [42].

Metrics representing reasoning capability boundary. We evaluate RLVR with pass@k for high-budget reasoning success [7, 38, 59], semantic and operation diversity for rollouts [12], and ReCEval for temporal consistency of long CoT [40]. More detailed discussionis stated in Appendix E.

5 Experiments Results

5.1 Results and Q&As

We present our results via the following Q&As. **PSN-GRPO** denotes the PSN-GRPO w/o TIS or adaptive noise scheduling; **PSN Var-I** uses fixed noise in Section 3.5.1, **PSN Var-II** uses real-time adaptive noise in Section 3.5.2, and both of them use TIS.

Q1: Does PSN-RLVR really 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 on **Qwen2.5-Math-7B** model. Figure 3 summarize the average performance across the five reasoning 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 $k = 256$). In addition, 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 noise do hurt small pass@k. This will be mitigated through adaptive noise mechanism stated in Question 6.

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

Q2: Where should parameter-space noise be injected ?

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. [39] suggests that parameter space noise yields optimal performance when using normalization between perturbed layers. In standard Transformer architectures, *the MLP blocks naturally satisfy this structural criterion*. To empirically validate this hypothesis and demonstrate the method's generalization beyond specific math-tuned models. We therefore compare noise injection into full decoder layers, the language-modeling head, and MLP/FFN blocks .We conducted ablation studies using the general-purpose **Qwen2.5-Math-7B** model. As shown in Figure 4, MLP-only injection achieves the best high-budget pass@k performance, with detailed results in Appendix Figure 10.

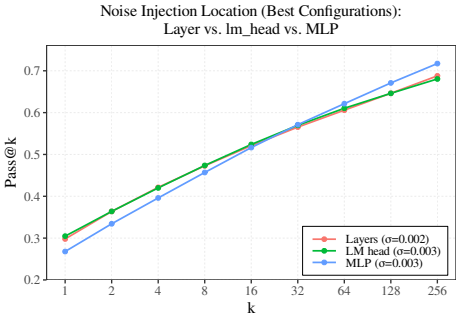


Figure 4: Noise injection location. MLP injection attains the largest gains at high k.

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 PSN-GRPO against action-space noise baselines, specifically training-time action-space noise via temperature scaling ($T \in \{1.0, \ldots, 1.7\}$) [14]. As Figure 5 shows, temperatures above 1.5 degrade performance. This occurs because uncorrelated, token-level perturbations accumulate into "logical drift," disrupting the global coherence required for long-horizon CoT reasoning.

In contrast, PSN perturbs policy parameters prior to generation, ensuring *trajectory-level consistency*. Consequently, PSN's benefits scale with reasoning length, becoming increasingly *pronounced as the length of reasoning trajectories grow* (Table 1, Appendix Figure 7). The gap widens significantly to $+8.9\%$ (pass@256) on the harder AIME 24 benchmark (average 1,978 tokens).

Furthermore, PSN-GRPO consistently surpasses the performance ceiling of evaluation-time temperature scaling (typically $T = 1.5$), avoiding its tuning costs and incoherent exploration (Table 1, Appendix Figure 8). ReCEval metrics [40] directly validate this consistency mechanism (Appendix Table 8); PSN-GRPO achieves peak entailment (0.8434) and contradiction-consistency (0.6918) scores, outperforming both training-time and evaluation-time temperature baselines.

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 PSN-GRPO with and without TIS correction. We observe that TIS significantly boosts the performance of PSN-GRPO (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).** TIS correction boosts PSN-GRPO (e.g., pass@256 from 74.33% to 76.94%) by mitigating the off-policy mismatch between perturbed sampling and clean training.

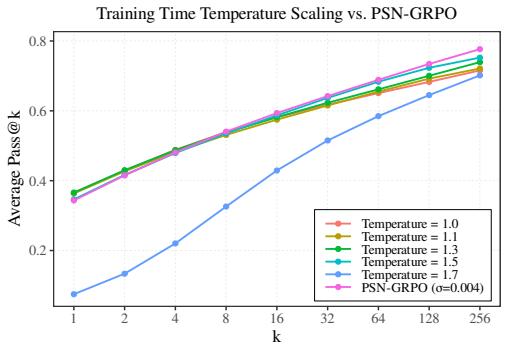


Figure 5: Average performance across reasoning benchmarks with different training temperature on naive GRPO . Increasing temperature beyond 1.5 degrades performance. Our proposed PSN-GRPO consistently outperforms standard GRPO, even when compared against its optimally hand-tuned temperature GRPO on large Pass@k.

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: We report pass@256 scores (%) for PSN-GRPO ($\sigma = 0.004$) alongside the GRPO best training-time (B.Train) and best evaluation-time (B.Eval) temperature baselines. **Key Finding:** PSN-GRPO outperforms action-space noise, especially on long-horizon tasks like AIME 25 (1.5k tokens/response).

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.

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. This pattern suggests that σ mainly controls the breadth of trajectory exploration: too little perturbation keeps rollouts close to the clean policy and limits high-budget diversity, whereas too much perturbation discovers more diverse candidates but reduces single-sample reliability. Thus, the preferred noise magnitude depends on the target evaluation regime. For applications that prioritize reliable first attempts, a moderate scale is safer; for settings where many samples can be evaluated or filtered, a slightly larger scale can be useful because it expands the candidate solution set.

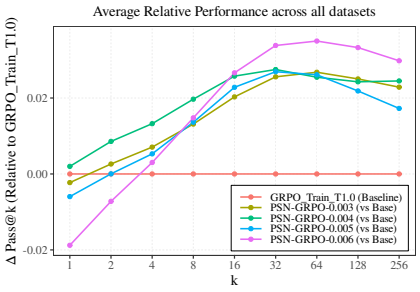


Figure 6: **Impact of Noise Magnitude σ.** Larger noise improves high pass$k$ because of high exploration but hurts pass@1; moderate scales balance reliability and diversity.

Q6: Is adaptive noise scheduling better than fixed noise ?

A: Yes. Real-time certainty-aware scheduling achieves the best overall trade-off by preserving pass@1 reliability while expanding the high-budget reasoning boundary (pass@256).

We benchmark our compute-aware adaptive noise schedules, Variant I and Variant II, against fixed-noise PSN and the GRPO baseline. The results in Table 3(and detailed result in Table 11) reveal a clear scheduling trade-off. **PSN Fixed** introduces no additional inference-time scheduling cost, but its static perturbation scale can hurt low-budget reliability, as reflected by its lower pass@1 compared with GRPO. **PSN Var-I** attempts to compensate for this instability by adapting the noise scale from delayed batch-level feedback; however, because the update is not

Dataset	Method	pass@1		pass@2		pass@256	
		Score	Gap	Score	Gap	Score	Gap
Average	PSN Var-II	**36.4%**	-	**44.1%**	-	**79.5%**	-
	GRPO Train	36.3%	+0.1%	42.7%	+1.3%	74.7%	+4.8%
	PSN Fixed	36.1%	+0.3%	43.6%	+0.5%	77.1%	+2.4%
	PSN Var-I	35.4%	+1.0%	42.5%	+1.5%	75.1%	+4.4%

Table 3: **Adaptive noise schedules.** PSN Var-II gives the best pass@k scores; gaps are absolute improvements over each compared method.

synchronized with the model's current rollout state, this lag weakens high-budget exploration and reduces pass@256 relative to PSN Fixed. In contrast, **PSN Var-II** updates the noise scale in real time using semantic-similarity and self-certainty signals. As a result, PSN Var-II achieves the best performance at both pass@1 and pass@256, indicating that real-time scheduling better balances single-sample reliability and broad solution-space exploration.
Appendix ablations provide further evidence. Table 10 shows that self-certainty and semantic similarity each help, with their additive combination performing best. Table 9 shows that TIS and scheduling only modestly improve GRPO, whereas PSN-GRPO gives the main high-budget gain; thus, parameter-space exploration is the primary driver, and the scheduler mainly stabilizes the reliability–exploration trade-off.
These findings suggest that the benefit of adaptive scheduling does not come merely from changing σ, but from changing it at the right temporal resolution. Delayed adaptation can partially regulate exploration, yet it may react to stale policy behavior and therefore over- or under-correct the next batch. Real-time scheduling mitigates this feedback lag, making the noise magnitude responsive to the current model state and yielding a more stable exploration–exploitation trade-off.

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

A: Yes. PSN-RLVR shows consistent gains across multiple model families and improves performance on OOD science and coding benchmarks.

We evaluate PSN-RLVR beyond the default Qwen2.5-Math-7B setting, including Qwen2.5-7B, Qwen3-4B-Base, and Llama-3.1-8B. Appendix Table 6 and 7 shows that PSN improves Qwen3-4B-Base average pass@256 from 70.70% to 74.56% with fixed-sigma PSN, while adaptive Variant II improves average pass@1/pass@2 from 33.14%/41.38% to 34.48%/42.28%. PSN also improves Llama-3.1-8B average pass@256 from 39.56% to 49.92% and gives gains on GPQA and LiveCodeBench, suggesting that the effect is not limited to the original math benchmarks.

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-GRPO against two mainstreams method that increase explore capability pass@K [7] and RLVR Decomposed [62] . 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 [50] 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 P.

6 Limitations

PSN-RLVR is most effective for long-horizon reasoning tasks requiring global consistency. In simpler, short-sequence tasks where token-level stochasticity suffices, PSN may yield diminishing returns. This limitation is mitigated by our real-time adaptive scheduler (Table 11), though further investigation is warranted.

7 Acknowledgments

This work is supported by National Key RD Program of China (No. 2026YFE0101200), National Natural Science Foundation of China (No. 62671175)and Shanghai General AI Foundation Models Program (No. 2025SHZDZX026D38). The computations in this research were performed using the CFFF platform of Fudan University.

8 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. Finally, our lightweight certainty- and semantics-aware scheduling achieves robust performance without additional rollout overhead, and PSN composes with existing RLVR exploration strategies to further extend the achievable pass@k frontier.

References

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

[2] Abhinav Pandey Abhishek Kadian Ahmad Al-Dahle Aiesha Letman Akhil Mathur Alan Schelten Amy Yang Angela Fan et al. Abhimanyu Dubey, Abhinav Jauhri. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[3] Afra Amini, Tim Vieira, and Ryan Cotterell. Better estimation of the kullback–leibler divergence between language models, 2025. URL https://arxiv.org/abs/2504.10637.

[4] Siddharth Aravindan and Wee Sun Lee. State-aware variational thompson sampling for deep q-networks. In *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems*, pages 124–132. ACM, 2021. doi: 10.5555/3463952.3463973.

[5] Bizhe Bai, Hongming Wu, Peng Ye, and Tao Chen. M-grpo: Stabilizing self-supervised reinforcement learning for large language models with momentum-anchored policy optimization, 2025. URL https://arxiv.org/abs/2512.13070.

[6] Angelica Chen, Jason Phang, Alicia Parrish, Vishakh Padmakumar, Chen Zhao, Samuel R. Bowman, and Kyunghyun Cho. Two failures of self-consistency in the multi-step reasoning of llms, 2024. URL https://arxiv.org/abs/2305.14279.

[7] 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*, abs/2508.10751, 2025. URL https://api.semanticscholar.org/CorpusID:280649795.

[8] Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective, 2025. URL https://arxiv.org/abs/2506.14758.

[9] Rongxin Cheng, Kai Zhou, Xingda Wei, Siyuan Liu, Mingcong Han, Mingjing Ai, Yeju Zhou, Baoquan Zhong, Wencong Xiao, Rong Chen, and Haibo Chen. Fast llm post-training via decoupled and fastest-of-n speculation, 2025. URL https://arxiv.org/abs/2511.16193.

[10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

[11] 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.

[12] Xingyu Dang, Christina Baek, J. Zico Kolter, and Aditi Raghunathan. 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.

[13] Yihong Dong, Xue Jiang, Yongding Tao, Huanyu Liu, Kechi Zhang, Lili Mou, Rongyu Cao, Yingwei Ma, Jue Chen, Binhua Li, Zhi Jin, Fei Huang, Yongbin Li, and Ge Li. Rl-plus: Countering capability boundary collapse of llms in reinforcement learning with hybrid-policy optimization, 2025. URL https://arxiv.org/abs/2508.00222.

[14] Weihua Du, Yiming Yang, and Sean Welleck. Optimizing temperature for language models with multi-sample inference, 2025. URL https://arxiv.org/abs/2502.05234.

[15] Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling?, 2025. URL https://arxiv.org/abs/2410.23771.

[16] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Rémi Munos, Demis Hassabis, Shane Legg, et al. Noisy networks for exploration. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1706.10295.

[17] Benjamin Gravell and Tyler Summers. Robust learning-based control via bootstrapped multiplicative noise, 2021. URL https://arxiv.org/abs/2002.10069.

[18] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 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`.

[19] Abhishek Gupta, Russell Mendonca, Yuxuan Liu, P. Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *Neural Information Processing Systems*, 2018. URL `https://api.semanticscholar.org/CorpusID:3418899`.

[20] 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)*, pages 3828–3850, 2024.

[21] Jingkai He, Tianjian Li, Erhu Feng, Dong Du, Qian Liu, Tao Liu, Yubin Xia, and Haibo Chen. History rhymes: Accelerating llm reinforcement learning with rhymerl, 2025. URL `https://arxiv.org/abs/2508.18588`.

[22] 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.

[23] Jakob J. Hollenstein, Sayantan Auddy, Matteo Saveriano, Erwan Renaudo, and Justus H. Piater. 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`.

[24] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *ArXiv*, abs/1904.09751, 2019. URL `https://api.semanticscholar.org/CorpusID:127986954`.

[25] 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, 2025. URL `https://arxiv.org/abs/2510.11696`.

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

[27] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL `https://arxiv.org/abs/2403.07974`.

[28] Zishang Jiang, Jinyi Han, Tingyun Li, Xinyi Wang, Sihang Jiang, Jiaqing Liang, Zhaoqian Dai, Shuguang Ma, Fei Yu, and Yanghua Xiao. Selective expert guidance for effective and diverse exploration in reinforcement learning of llms, 2025. URL `https://arxiv.org/abs/2510.04140`.

[29] Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty, 2025. URL https://arxiv.org/abs/2502.18581.

[30] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

[31] 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.

[32] Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

[33] Jiazheng Li, Hongzhou Lin, Hong Lu, Kaiyue Wen, Zaiwen Yang, Jiaxuan Gao, Yi Wu, and Jingzhao Zhang. Questa: Expanding reasoning capacity in llms via question augmentation, 2025. URL https://arxiv.org/abs/2507.13266.

[34] Xiao Liang, Zhongzhi Li, Yeyun Gong, Yelong Shen, Ying Nian Wu, Zhijiang Guo, and Weizhu Chen. Beyond pass@1: Self-play with variational problem synthesis sustains rlvr, 2025. URL https://arxiv.org/abs/2508.14029.

[35] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, 2024. URL https://arxiv.org/abs/2310.02255.

[36] Ian Osband, Benjamin Van Roy, Daniel J. Russo, and Zheng Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

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

[38] Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. Simko: Simple pass@k policy optimization, 2025. URL https://arxiv.org/abs/2510.14807.

[39] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations*, 2018.

[40] Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. ReCEval: Evaluating reasoning chains via correctness and informativeness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10066–10086, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.622. URL https://aclanthology.org/2023.emnlp-main.622/.

[41] Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. Prompt perturbation consistency learning for robust language models, 2024. URL https://arxiv.org/abs/2402.15833.

[42] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof qa benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

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

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

[45] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[46] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[47] Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. 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.

[48] Michal Shur-Ofry, Bar Horowitz-Amsalem, Adir Rahamim, and Yonatan Belinkov. Growing a tail: Increasing output diversity in large language models. *ArXiv*, abs/2411.02989, 2024. URL https://api.semanticscholar.org/CorpusID:273821765.

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

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

[51] Qwen Team. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[52] Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[53] Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may or may not escape its origin, 2026. URL https://arxiv.org/abs/2507.14843.

[54] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

[55] Peng Ye, Baopu Li, Yikang Li, Tao Chen, Jiayuan Fan, and Wanli Ouyang. β-DARTS: Beta-decay regularization for differentiable architecture search. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 10864–10873. IEEE, 2022.

[56] Peng Ye, Tong He, Baopu Li, Tao Chen, Lei Bai, and Wanli Ouyang. β-DARTS++: Bi-level regularization for proxy-robust differentiable architecture search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[57] Peng Ye, Tong He, Shengji Tang, Baopu Li, Tao Chen, Lei Bai, and Wanli Ouyang. Stimulative training++: Go beyond the performance limits of residual networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6463–6479, 2025.

[58] Peng Ye, Chenyu Huang, Mingzhu Shen, Tao Chen, Yongqi Huang, and Wanli Ouyang. Dynamic model merging with mixture of weights. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(8): 7925–7939, 2025.

[59] 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?, 2025. URL https://arxiv.org/abs/2504.13837.

[60] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

[61] Guojian Zhan, Likun Wang, Pengcheng Wang, Feihong Zhang, Jingliang Duan, Masayoshi Tomizuka, and Shengbo Eben Li. Mind your entropy: From maximum entropy to trajectory entropy-constrained rl. *ArXiv*, abs/2511.11592, 2025. URL https://api.semanticscholar.org/CorpusID:283071650.

[62] Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 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 [35, 37]. Early approaches primarily utilized Proximal Policy Optimization (PPO) [45] to align models with reward signals derived from unit tests or symbolic solvers. More recently, Group Relative Policy Optimization (GRPO) [18] 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 [53, 59] 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. Beyond RLVR, several related studies investigate how changes to model optimization and parameterization can improve neural network performance. β-DARTS [55, 56] introduces Beta-Decay regularization on architecture parameters to stabilize differentiable architecture search and improve the transferability of searched architectures. Stimulative Training++ [57] addresses network loafing in residual networks by randomly sampling subnetworks and supervising them through consistency with the full network, with additional strategies for logit alignment, input scaling, and subnetwork sampling. MoW-Merging [58] dynamically combines task-specific model weights, adapting the merging process to input samples and accounting for differences in weight interference across layers. M_GRPO[5] using another EMA teacher to stabilize self-supervised RLVR.

These studies address architecture-search stability, subnetwork training, and multi-task model composition, respectively. Our work instead investigates exploration during RLVR: we perturb the rollout policy's parameters to generate temporally consistent reasoning trajectories while updating a clean policy, and address the resulting off-policy mismatch and adaptive noise-control challenges.

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

While PSN is well-established in robotics [16, 17, 19, 23], its application to the discrete, high-dimensional reasoning space of RLVR remains unexplored. On the other side, subsequent theoretical work situates such noise-injected policies within the broader paradigm of posterior sampling / randomized value functions, which can induce deep exploration and admits regret guarantees [36, 44]. In particular, variational Thompson-sampling perspectives interpret practical parameter-noise methods as tractable approximations to posterior sampling over value functions, providing a theoretical foundation for their empirical effectiveness [4].
A concurrent work is QERL [25], which focuses on improving RLVR training efficiency via quantization. While they report that the noise introduced by quantization *surprisingly* improved exploration, this phenomenon was observed as side effect of their efficiency optimization. Consequently, they did not systematically analyze the noise dynamics, optimal injection strategies, or scaling laws required to leverage this effect for reasoning.

C Experimental setup

All experiments utilize the standard configuration of SimpleRL-Zoo[2] [60] 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 [54],Qwen2.5-7B [51],Qwen3-4B-Base [52] and Llama-3.1-8B [2]. Following the SimpleRL-Reason protocol [60], training data is sampled from the NuminaMath dataset [32], which is derived from GSM8K [10] and MATH [22]. 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.

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

Table 5: Hyperparameters used for SimpleRL-Zoo models.

Settings	**SimpleRL-Zoo**
Framework	verl [46]
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	10

D 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 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* [21]: 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.

E 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 [7, 38, 59]. Semantic diversity measures variation among reasoning traces in embedding space, while operation diversity abstracts rollouts into arithmetic-operation sequences and measures procedural variation [12]. 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 [40].

F 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.

G Additional Evidence for additional model OOD evidence

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	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-GRPO	3.60	4.70	39.56	0.0909	0.0883	0.1291	0.1460
Llama-3.1-8B-PSN-GRPO	**4.02**	**6.84**	**49.92**	**0.1263**	**0.1001**	**0.1363**	**0.1507**

Table 7: **Model-family and OOD generalization evidence.** More 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. PSN-GRPO represent $\sigma = 0.001$, without TIS and scheduler in these experiments.

H Additional Evidence for temporal consistency

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 8: **ReCEval temporal-consistency comparison.** ReCEval [40] evaluates rollouts-level reasoning correctness and non-contradiction. PSN-GRPO obtains the highest entailment and contradiction-consistency scores against base, temperature, and pass@k baselines in the experiments.

I Ablation study of TIS and scheduler with GRPO and PSN-GRPO

Method (Qwen2.5-Math-7B)	Avg pass@1	Avg pass@2	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	77.1
PSN-GRPO + TIS + scheduler	**44.1**	**50.6**	**79.5**

Table 9: **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.

J Ablation study of two modules in 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 10: **Scheduler component ablation.** Self-certainty and semantic similarity are useful independently, while their additive combination performs best in the ablation.

K Detailed performance of training time temperature scaling

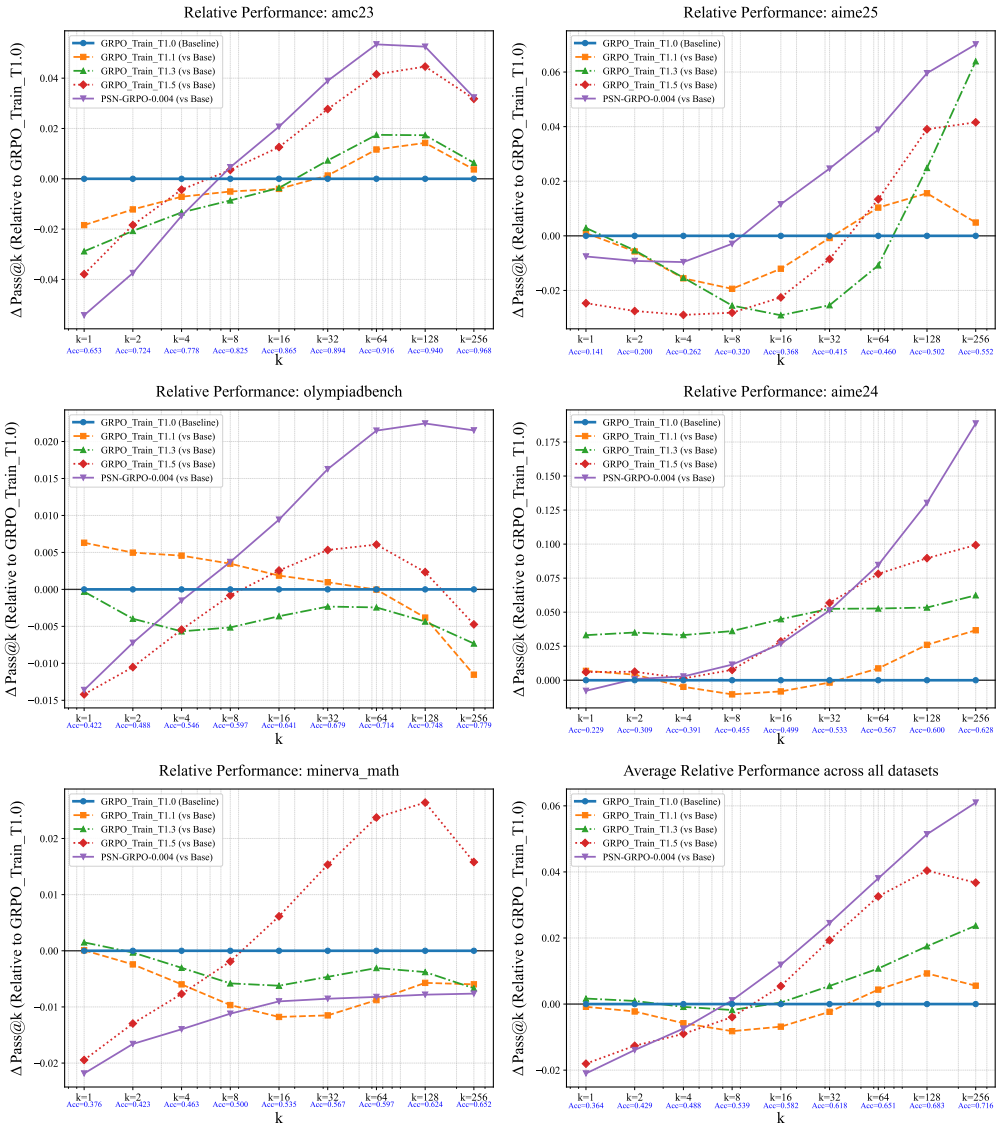


Figure 7: Performance across benchmarks. We do not plot T=1.7 in these plots since its overall performance collapsed as shown in Figure 5.

L More experiment result of evaluation temperature scaling

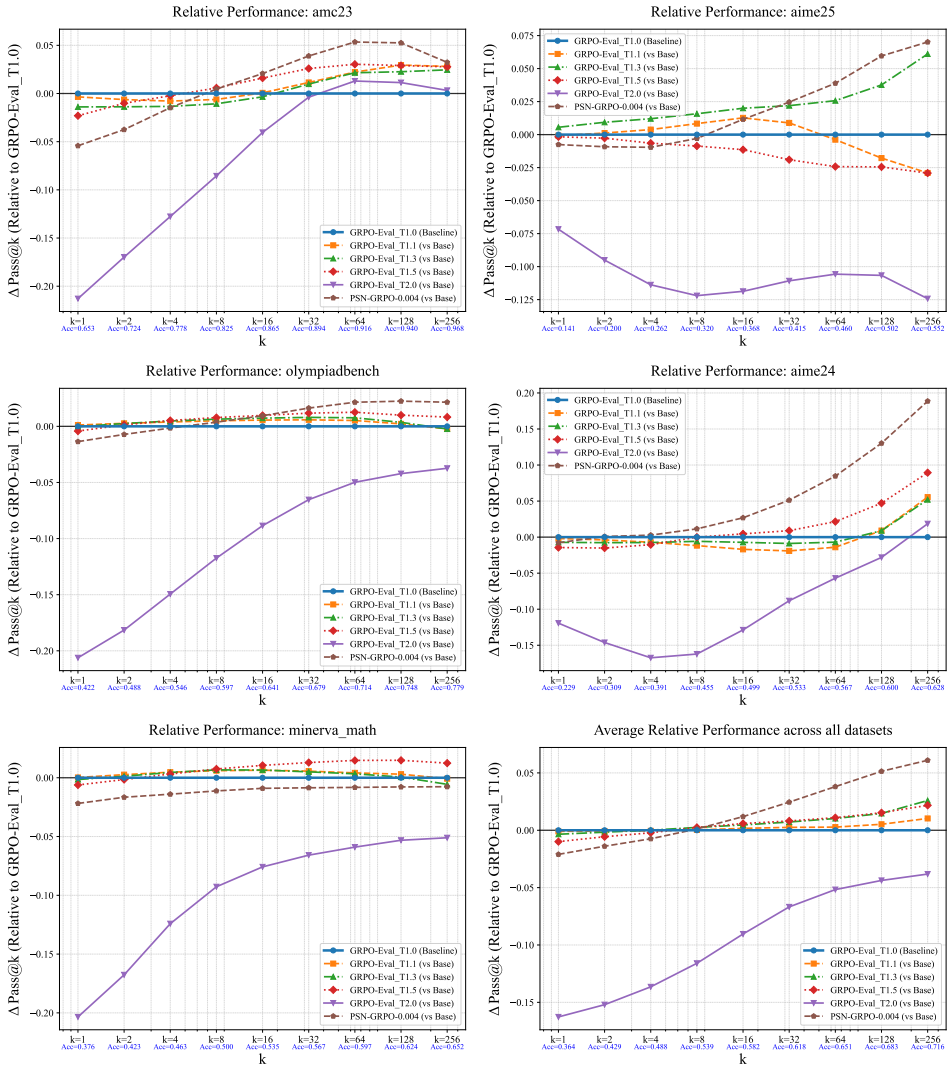


Figure 8: Detailed performance analysis across five datasets.

M More detailed experiment result of evaluation temperature scaling

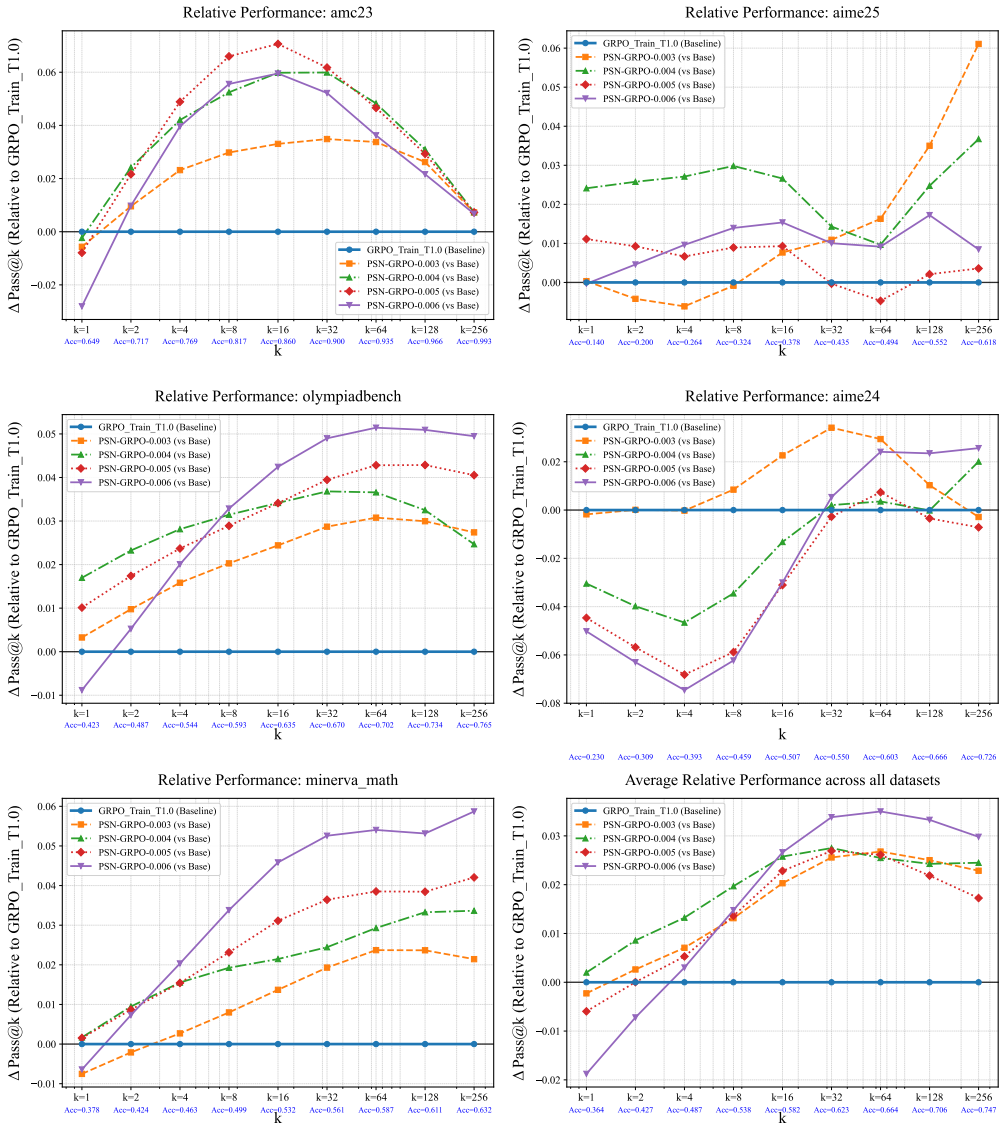


Figure 9: Detailed performance analysis across five datasets. In general, a larger noise scale σ tends to yield higher pass@k for large k. Specifically, on datasets of moderate difficulty (e.g., OlympiadBench and Minerva Math), increased σ improves reasoning capability boundary, yield higher pass@k for large k. Conversely, on more challenging benchmarks such as AIME24 and AIME25, a larger noise scale tends to degrade overall performance.

N Detailed performance of where to inject noise

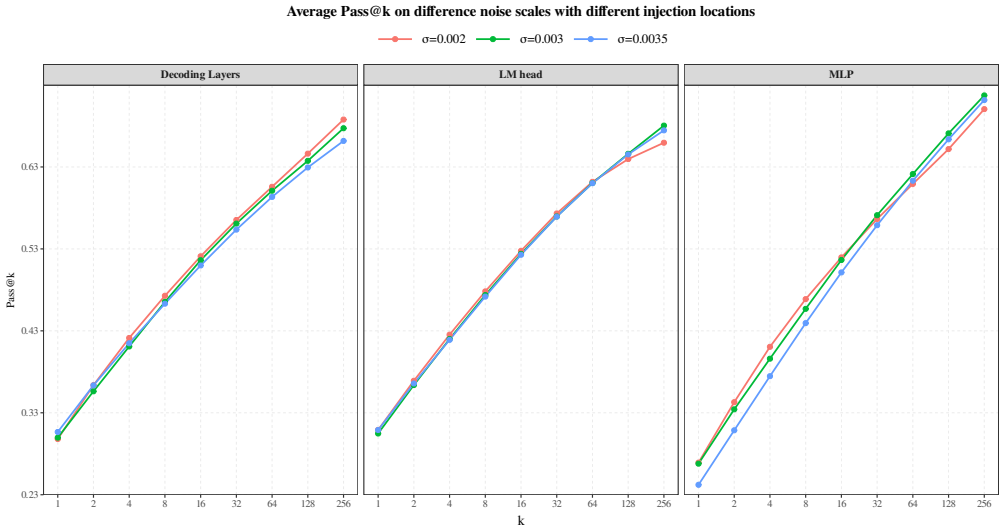


Figure 10: Average Pass@k under parameter-space noise injected whole transformer layers, the `lm_head`, or all MLP sublayers, sweeping noise scale σ. MLP injection yields the strongest and most consistent improvements, especially at large k.

O 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%
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 11: **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.

P 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: $m + n = 115 + 256 = \mathbf{371}$.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the PSN-RLVR contribution, TIS correction, adaptive scheduler, and empirical scope. The claims are supported by the main experiments and bounded by the stated limitations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The paper includes a limitations discussion covering short-horizon tasks, the tested 4B–8B model scale, and unmeasured costs when scaling parameter perturbations to much larger models.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper proposes algorithmic objectives and empirical analyses, but does not claim new theoretical results or formal theorems.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: The paper reports the training framework, model families, datasets, rollout budgets, pass@k estimator, noise settings, TIS clipping, scheduler design, and evaluation protocol.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An anonymized supplement provides code/data access and reproduction instructions for the main experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The main text and appendix specify the datasets, model backbones, training setup, evaluation temperature, pass@k computation, rollout counts, adaptive-scheduler constants, and TIS clipping parameter.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports averages over multiple training seeds test the statistical significance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: The discusses include GPU and amount of compute required for each of the individual experimental.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public benchmarks and open model families for algorithmic evaluation, preserves anonymity, and does not involve human-subject data collection.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Answer: [Yes]

    Justification: Existing models, datasets, and benchmarks are credited through citations in the main text and appendix, and the supplement documents the assets used for reproduction.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new high-risk pretrained model or scraped dataset; it studies an RL training method using existing model backbones and benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Existing models, datasets, and benchmarks are credited through citations in the main text and appendix, and the supplement documents the assets used for reproduction.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The anonymized supplement documents the released experimental code and scripts needed to reproduce the proposed PSN-GRPO method and evaluations.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject experiments, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.