Variance-Optimal State Resampling for Reinforcement Learning with Verifiable Rewards

Yibo Wang **Yuanxin Liu** **Zhaoran Wang**
Northwestern University
{yibowang2030, yuanxinliu2029}@u.northwestern.edu
zhaoranwang@gmail.com

Abstract

Reinforcement learning with verifiable rewards spends much of its rollout budget on states that yield no learning signal. We recast this as a question about the sampling distribution itself: under a fixed rollout budget, which distribution over states minimizes the variance of the policy-gradient estimator? The answer admits a closed-form Neyman allocation, $\mu^{\pi}_{\mathrm{var}}(s) \propto d^{\pi}(s)\,\sigma_{\pi}(s)$, which connects naturally to GRPO's group-relative advantage and induces an emergent reverse curriculum as the policy improves, with no externally imposed schedule. Building on this principle, we propose VSR (Variance-driven State Resampling), a plug-in modification to GRPO using a prefix replay buffer and a zero-cost variance proxy. On mathematical reasoning benchmarks and long-horizon agentic tasks (Sokoban, BabyAI), VSR outperforms GRPO and GiGPO on most benchmarks with fewer generated tokens per update.

1 Introduction

Training large language models with reinforcement learning from verifiable rewards has become a standard recipe for improving reasoning and agentic capabilities. In practice, however, a non-trivial fraction of training steps contribute little signal: in GRPO-style training, rollout groups that return uniform rewards—all-fail or all-success—yield a collapsed group-relative advantage and a near-zero update.

It can be helpful to look at this pathology at the level of state distribution, rather than prompt difficulty. Even on hard problems, not all intermediate states are equally uninformative. Consider a competition math problem that the current policy rarely solves from the prompt alone. Once the policy has produced a partial chain of thought that performs a key algebraic reduction, continuations from that prefix tend to succeed with noticeably higher probability, and different continuations produce genuinely mixed outcomes. A similar pattern shows up in long-horizon agentic tasks: from the initial query, a successful trajectory is uncommon, but once the agent has identified the relevant information source through a few tool calls, the remaining decisions can lead to meaningfully different outcomes. In both cases, the trajectory passes through intermediate states with rich gradient

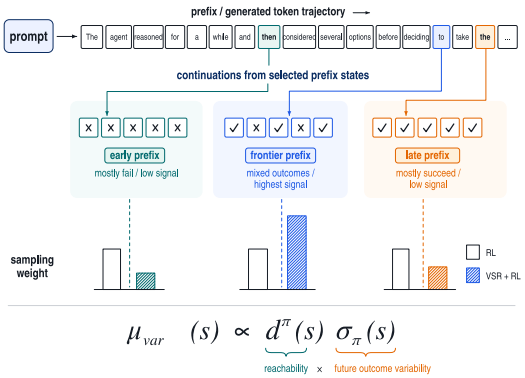

Figure 1: Continuations from early prefixes mostly fail and from late prefixes mostly succeed; both yield low signal. VSR samples states in proportion to $d^{\pi}(s)\,\sigma_{\pi}(s)$ to capture the highest signal.

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

signals, although rollouts generated from the prompt alone rarely reach it. The challenge is that such states tend to be visited with low probability under the current policy's state distribution d^π, so finite-sample training encounters them only occasionally.

Therefore, a complementary question which seems to have received comparatively less direct attention is: *Under a fixed rollout budget, which state distribution minimizes the variance of the policy-gradient estimator?*

In this paper we take up that question. Under a gradient uniformity assumption, we find that the answer takes a closed form with Neyman-allocation structure,

$$\mu^\pi_{\mathrm{var}}(s) \propto d^\pi(s)\sigma_\pi(s). \tag{1}$$

This proposal allocates training mass by the product of on-policy relevance and residual signal magnitude, and it evolves with the policy, giving rise to an emergent curriculum without a hand-designed difficulty schedule.

Implementing μ^π_{var} with exact proposal correction would involve an inverse-σ_π factor. GRPO's group-relative normalization provides a practical analogue of this factor by normalizing rewards with an empirical within-group standard deviation. We instantiate this idea as **VSR (Variance-driven State Resampling)**, a practical plug-in modification to GRPO-style algorithms that approximates the variance-optimal proposal. As a tractable algorithmic realization of the allocation principle, VSR uses a recent prefix replay buffer as a density-free surrogate for the current state-visitation distribution and a cheap proxy for σ_π to prioritize replayed prefixes. Experiments on math reasoning and long-horizon agentic tasks demonstrate the effectiveness of this approximation.

Contributions.

- We formulate the uniform-reward pathology in GRPO-style training as a variance-reduction problem over the state sampling distribution, and derive a closed-form $\mu^\pi_{\mathrm{var}} \propto d^\pi(s)\,\sigma_\pi(s)$, which matches the form of GRPO's group-relative advantage and induces an emergent reverse curriculum as the policy improves.
- Building on this principle, we propose VSR (Variance-driven State Resampling), a plug-in practical modification to GRPO that approximates $\mu^\pi_{\mathrm{var}} \propto d^\pi(s)\,\sigma_\pi(s)$ through a state replay buffer and a zero-cost variance proxy.
- On hard mathematical reasoning benchmarks and long-horizon agentic tasks, VSR outperforms GRPO and GiGPO baselines with fewer generated tokens per update, and ablations isolate the contributions of variance-driven scoring and recency-based buffer updates.

2 The variance-driven state proposal

2.1 Setup

A natural starting question for any policy-gradient method is: what distribution is training actually averaging over? For general RL the answer is the state visitation measure d^π, and the policy gradient takes the form

$$\nabla J(\pi) \;=\; \mathbb{E}_{s\sim d^\pi,\, a\sim\pi(\cdot|s)}[\nabla \log \pi(a \mid s)\, \tilde{r}(s,a)]\,, \quad d^\pi(s) \;\propto\; \sum_{t=0}^{T-1} \Pr\big(s_t = s \,\big|\, s_0 \sim \mathcal{D},\, \pi\big), \tag{2}$$

where $\tilde{r}(s,a) = r(s,a) - b(s)$ for a state-dependent baseline b. The expectation is taken with respect to d^π, which is therefore the distribution under which gradient signal is sampled.

LLM RL with verifiable rewards is most often presented as a contextual bandit under which the only input-side distribution is the prompt distribution $\mathcal{D}$. This framing is coarse relative to how generation actually proceeds: a single rollout on a reasoning problem spans thousands of tokens and passes through qualitatively distinct regimes such as restatement, algebraic reduction, and reflection, while on long-horizon agentic tasks a trajectory decomposes into a sequence of consequential intermediate decisions. We therefore adopt the standard state-action view throughout, in which

$$s_0 = x \sim \mathcal{D}, \qquad s_t = (x, y_{<t}), \qquad a_t = y_t, \qquad \tau = (s_0, a_0, s_1, a_1, \ldots, s_T). \tag{3}$$

This is the same structure realized by the triangular attention mask inside the transformer itself. Under this instantiation, d^π is a marginal over all intermediate contexts the policy passes through, not over prompts alone.

2.2 The sampling proposal as a variance-reduction lever

Reducing the variance of policy-gradient estimators is a long-standing concern, and most of the work has gone into the gradient baseline $b(s)$ [Greensmith et al., 2004]. The proposal μ from which states are sampled has not, by comparison, been treated as something to be optimized: it is set to d^π and any deviation is handled as an off-policy correction problem rather than as a variance-reduction choice. This section asks whether μ can be optimized alongside the baseline to minimize the variance.

The standard policy-gradient estimator with a state-dependent baseline takes the form

$$\widehat{\nabla J}_b(\pi) \;=\; \big(r(s,a) - b(s)\big)\,\nabla \log \pi(a \mid s), \qquad s \sim d^\pi,\; a \sim \pi(\cdot \mid s), \tag{4}$$

which is unbiased for $\nabla J(\pi)$ for every state-dependent b. For trace-covariance minimization, however, the optimal baseline generally depends on the action-gradient norm. Writing $g(s,a) := \nabla \log \pi(a \mid s)$ and $H(s,a) := \|g(s,a)\|^2$, the statewise optimum is

$$b_w^*(s) = \frac{\mathbb{E}[r(s,a)H(s,a) \mid s]}{\mathbb{E}[H(s,a) \mid s]}. \tag{5}$$

It reduces to the conditional mean $\bar{r}(s) := \mathbb{E}[r(s,a) \mid s]$ under the within-state conditional-moment condition stated below.

Crucially, unbiasedness of (4) does not require sampling from d^π. For any proposal μ with $\mathrm{supp}(\mu) \supseteq \mathrm{supp}(d^\pi)$, the importance sampling estimator

$$\widehat{\nabla J}_{b,\mu}(\pi) \;=\; \frac{d^\pi(s)}{\mu(s)}\big(r(s,a) - b(s)\big)\,\nabla \log \pi(a \mid s), \qquad s \sim \mu,\; a \sim \pi(\cdot \mid s), \tag{6}$$

is unbiased for every admissible pair (b, μ). The on-policy choice $\mu = d^\pi$ is the operational default in policy-gradient methods, but it is not, in general, the variance-minimizing choice. The reason is that states contribute different amounts of gradient signal per sample: a state whose continuations almost always succeed or fail yields little reward variation after baseline subtraction. A state whose continuations produce mixed outcomes yields per-sample signal proportional to that variation. Sampling states by visitation frequency alone does not reflect this difference.

We therefore treat μ, alongside b, as a variable to be optimized. Minimizing the trace covariance of (6) jointly over (b, μ) admits a closed-form solution.

Theorem 2.1 (Joint-optimal baseline and proposal; informal)**.** *Let $H(s,a) = \|\nabla \log \pi(a \mid s)\|^2$ and let $b_w^*(s)$ be given by* (5)*. The general joint minimizer of the trace covariance of* (6) *is*

$$\mu^\star(s) \;\propto\; d^\pi(s)\sqrt{M_w^*(s)}, \qquad M_w^*(s) := \mathbb{E}\big[(r(s,a) - b_w^*(s))^2 H(s,a) \mid s\big]. \tag{7}$$

If $\mathbb{E}[H \mid s, r] = G(s)$ is independent of the realized reward within each state, then $b_w^*(s) = \bar{r}(s)$ and

$$\mu^\star(s) \propto d^\pi(s)\sigma_\pi(s)\sqrt{G(s)}. \tag{8}$$

Under the additional across-state homogeneity condition $G(s) \equiv G_0$, this reduces to

$$\mu^\star(s) \;\propto\; d^\pi(s)\,\sigma_\pi(s) \;=:\; \mu^\pi_{\mathrm{var}}(s), \tag{9}$$

where $\sigma_\pi(s) := \sqrt{\mathrm{Var}_{a\sim\pi(\cdot|s)}(r(s,a))}$. On its effective support, the corresponding proposal-corrected estimator takes the standardized-residual form

$$\widehat{\nabla J}^{\star}(\pi) \;\propto\; \frac{r(s,a) - \bar{r}(s)}{\sigma_\pi(s)}\,\nabla \log \pi(a \mid s), \qquad s \sim \mu^\pi_{\mathrm{var}},\; a \sim \pi(\cdot \mid s). \tag{10}$$

Specialization to the GRPO form. Under the two conditions above, the exact proposal-corrected estimator (10) has the same standardized reward-residual form as GRPO's group-relative advantage [Shao et al., 2024]. This connection concerns the form of the normalization: standard GRPO does not sample states from μ^π_{var} or apply the exact state-density ratio d^π/μ. A formal statement, with an explicit expression for the variance gap, is given in Theorem B.5.

Beyond the regularity assumption. The two conditions that yield $\mu^\pi_{\mathrm{var}} \propto d^\pi\sigma_\pi$ are idealizations. In practice, action-gradient norms can depend on the realized reward within a state and their conditional scale can vary across states, so μ^π_{var} need not coincide with the exact joint optimum. We make this precise in Appendix B.4 (Assumption B.6 and Theorem B.7). That result is a proposal comparison under the stated regularity condition, not an unrestricted joint-optimality result.

Analysis of an idealized theoretical algorithm. Separately from practical VSR, Appendix C analyzes an idealized smoothed state-reweighting algorithm under oracle-score, coverage, critic-realizability, and exact-weighting assumptions. Consider sampling critic data from a smoothed version of $d^\pi \sigma_\pi$ at every iteration and updating the policy by KL-regularized mirror ascent. For any comparator policy μ covered by every iterate, the resulting theoretical algorithm satisfies

$$J(\mu) - J(\bar{\pi}_T) \;\le\; \widetilde{O}\left(\sqrt{\frac{\mathrm{KL}(\mu \,\|\, \pi_1)}{T}} \;+\; \sqrt{\frac{H^2}{\varepsilon N}}\right),$$

combining a $1/\sqrt{T}$ optimization rate with a critic-estimation term inflated only by the smoothing factor $1/\varepsilon$. The full statement, assumptions, and proof are deferred to Appendix C.

2.3 Structure of the variance-optimal proposal

μ^π_{var} reweights the effective support of d^π. The unsmoothed proposal assigns zero mass to states with $\sigma_\pi(s) = 0$, i.e., states with no centered reward-variance signal, and therefore need not preserve the full support of d^π. When full support is required, one may use the smoothed proposal $\mu_\varepsilon = (1-\varepsilon)\mu_{\mathrm{var}} + \varepsilon d^\pi$, which retains an on-policy component while preserving the same variance-driven preference on the remaining mass.

μ^π_{var} trades off on-policy relevance and residual signal. The two factors capture distinct and complementary quantities: $d^\pi(s)$ measures how relevant a state is to the current policy's behavior, penalizing distributional shift away from on-policy sampling, while $\sigma_\pi(s)$ measures how much per-sample gradient signal a state carries, penalizing states whose outcomes are already deterministic. Their product is the Neyman allocation that minimizes estimator variance—what is upweighted is the intersection of states that are both reachable under π and informative about π's residual errors.

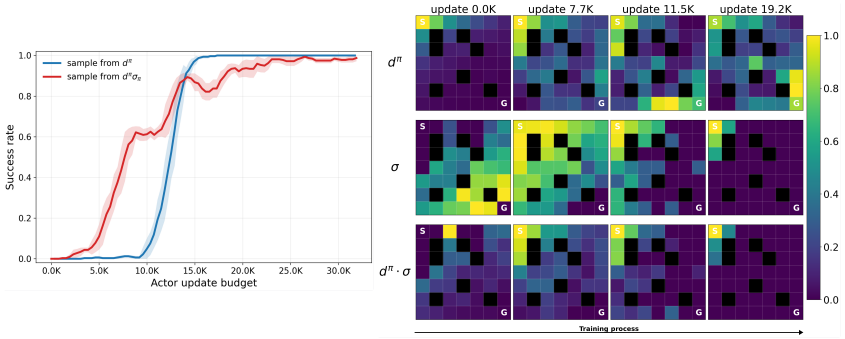

Figure 2: **An emergent curriculum from the variance-optimal proposal.** μ^π_{var} is concentrated near G early in training, and gradually walks backward through the maze toward S as cells closer to the goal are mastered one after another. This is the reverse curriculum: the training signal visits the state space in reverse without any externally imposed schedule. It is a built-in property of variance-optimal sampling: the variance-minimizing distribution by construction concentrates wherever the policy is reachable but still uncertain, and that frontier moves backward as the policy improves.

μ^π_{var} shifts with the policy, producing an emergent curriculum. Both factors depend on π: states that were once unreachable acquire visitation mass as π improves, and states that once produced mixed-outcome rollouts become near-deterministic and lose residual signal. The frontier of high μ^π_{var} therefore re-localizes itself as training proceeds, without any externally imposed schedule.

The dynamics above can be hard to visualize in settings where d^π and σ_π are intractable, so we examine μ^π_{var} in a setting where both can be computed exactly. A small MLP policy navigates a 7×7 gridworld with obstacles from start S to goal G under terminal reward $R \in \{0, 1\}$; the state space is small enough that $d^{\pi_t}(s)$ and $\sigma_{\pi_t}(s)$ can be enumerated, and $\mu^{\pi_t}_{\mathrm{var}} \propto d^{\pi_t}\sigma_{\pi_t}$ sampled from without approximation. Figure 2 shows the three quantities and a learning-curve comparison across training. Early on, d^{π_t} concentrates near S while σ_{π_t} peaks in a band of states near G that the policy can occasionally reach; their product picks out the intersection. As training progresses, the high-σ band recedes from G toward S as goal-adjacent states become near-deterministic successes, and μ_t

tracks it—until both factors collapse onto a small residual hard region near the start. We note that this curriculum is not designed; it is what minimizing estimator variance looks like when made visible.

3 VSR: A Practical Algorithm

The previous section characterized the variance-optimal proposal $\mu^{\pi}_{\mathrm{var}} \propto d^{\pi}\sigma_{\pi}$ as a structural object. However, both d^{π} and σ_{π} are intractable to compute exactly under realistic LLM rollout budgets, requiring cheap surrogates for implementation. We present **Variance-driven State Resampling** (VSR) in Algorithm 1, a plug-in modification of GRPO that approximates μ^{π}_{var} via a prefix replay buffer and a zero-cost variance proxy. The resulting algorithm should be understood as an approximation to the allocation principle of μ^{π}_{var} rather than as an unbiased estimator of the exact proposal-corrected gradient it induces. In particular, practical VSR does not estimate the density ratio d^{π}/μ: it changes the reset-state distribution from which rollout groups are launched and then applies the usual GRPO-style loss. Because the replay buffer and priorities evolve with the policy and replay history, practical VSR is a biased adaptive reset-state procedure motivated by the variance-allocation geometry, not optimization of one fixed reweighted objective.

Algorithm 1 VSR (Variance-driven State Resampling)

Require: Initial policy π_0, prompt distribution $\mathcal{D}$, group size G, recency window W, replay ratio $\rho \in [0, 1]$
Optional: expert successful trajectories E for cold-start bootstrapping

Initialize replay buffer $\mathcal{B}_0 \leftarrow \varnothing$ (or seed with prefix states from E if cold-starting)
for $t = 0, 1, \ldots, T-1$ **do**
  Draw fresh prompts $\{x_j\} \sim \mathcal{D}$ (fraction $1-\rho$) and replay prefixes $\{s_k\} \sim \mathcal{B}_t$ with probability $\propto \hat{\sigma}_t(s)$ (fraction ρ)
  Generate G rollouts under π_t from each x_j and each s_k
  Compute GRPO-style loss function $\mathcal{L}_{\mathrm{PG}}(\pi_t)$ over the rollouts
  Update $\pi_{t+1} \leftarrow \pi_t - \eta \nabla \mathcal{L}_{\mathrm{PG}}(\pi_t)$
  For each successful fresh trajectory τ generated from prompt x_j, insert every prefix state $s_i = (x_j, y_{<i})$ into $\mathcal{B}_{t+1}$ with score

$$\hat{\sigma}_{t+1}(s_i) = \sqrt{\hat{p}_i(1-\hat{p}_i)}, \quad \hat{p}_i = (1-\alpha_i)\, p_0(x_j) + \alpha_i\, R(\tau), \quad \alpha_i = i/|\tau|$$

  where $p_0(x_j)$ is the empirical success rate of x_j's fresh rollout group
  Evict from $\mathcal{B}_{t+1}$ all entries older than W steps
end for

3.1 From μ^{π}_{var} to a tractable surrogate

The two factors in μ^{π}_{var} are intractable in different ways. **(i) σ_{π} need to be approximated at zero rollout cost.** Estimating $\sigma_{\pi}(s)$ from conditional rollouts would reintroduce precisely the cost that state-level resampling is designed to amortize, so any usable proxy must be built from quantities already available in each GRPO step—the group success rate p_0 and terminal rewards $R(\tau)$. **(ii) d^{π} is not directly tractable.** We sidestep explicit density modeling via a recency-windowed prefix replay buffer $\mathcal{B}_t$ that collects states from recent rollouts, so that uniform draws from $\mathcal{B}_t$ approximate samples from d^{π_t}; recency-based eviction keeps the buffer aligned with the current policy.

Combining the two, we sample prefixes from $\mathcal{B}_t$ with probability proportional to $\hat{\sigma}_t(s)$ and launch fresh rollout groups from each (Algo-

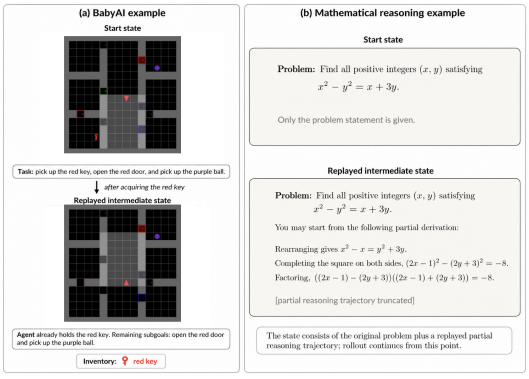

Figure 3: Illustration of intermediate states used by VSR. Left: a BabyAI agent after picking up the red key. Right: a math problem with a partial reasoning trace appended.

rithm 1). The design follows directly from μ^{π}_{var}: stay close to on-policy visitation through $\mathcal{B}_t$, and concentrate the sampling budget on states that still carry signal through $\hat{\sigma}_t$. Both surrogates are deliberately cheap rather than accurate.

3.2 Online prefix proposal construction

Recent prefix buffer. VSR maintains a recency-windowed replay buffer $\mathcal{B}_t$ of token-level prefix states extracted from recent rollout groups. For a trajectory $\tau = (s_0, a_0, \ldots, s_H)$, a replayable state has the form $s_i = (x, y_{<i})$, i.e., the original prompt together with the generated prefix up to position i. In agentic settings, the same state representation is applied to the serialized interaction history, including prior actions, tool outputs, and environment observations (Figure 3). Sampling from $\mathcal{B}_t$ gives a density-free empirical approximation to the current state-visitation measure d^{π_t}, while recency eviction keeps the proposal aligned with the evolving policy.

In our implementation, $\mathcal{B}_t$ stores prefixes from successful trajectories only. This is a deliberate departure from the exact empirical visitation distribution. This filtering is not theoretically required by the variance-optimal proposal. In the hard-problem regime studied here, longer failed prefixes are empirically more likely to yield another all-fail continuation group (Appendix D.5); we therefore use successful-prefix filtering as a practical budget-allocation choice that favors states known to lie on at least one completable trajectory.

Zero-cost variance score. Each buffered prefix is assigned a score intended to rank residual continuation uncertainty without additional conditional rollouts. For a prompt x, let $p_0(x)$ denote the empirical success rate of the rollout group generated from the original prompt, and let $R(\tau) \in \{0, 1\}$ denote the terminal reward of trajectory τ. For s_i being the i-th state along a trajectory with length $|\tau|$, we define its normalized position as $\alpha_i = i/|\tau|$.

VSR then estimates the continuation success probability at s_i by interpolating between the group-level success rate at the initial state and the realized terminal outcome:

$$\hat{p}_t(s_i \mid \tau) = (1 - \alpha_i)p_0(x) + \alpha_i R(\tau), \qquad \hat{\sigma}_t(s_i \mid \tau) = \sqrt{\hat{p}_t(s_i \mid \tau)\big(1 - \hat{p}_t(s_i \mid \tau)\big)}.$$

Early prefixes are therefore scored mostly by the prompt-level rollout statistics, while late prefixes are scored mostly by the observed outcome of the trajectory they belong to. Because the buffer stores only successful trajectories, $R(\tau) = 1$ for every stored prefix, so the implemented score is determined by the prompt-level success rate and normalized prefix position. We therefore use $\hat{\sigma}_t$ only as a zero-additional-rollout uncertainty-ranking proxy, not as a calibrated, unbiased, or consistent estimator of $\sigma_{\pi_t}(s)$. Appendix D.5 reports a statewise validation showing moderate positive association with empirical continuation uncertainty, including within fixed prefix positions.

Replay update. At each update, VSR forms a mixed batch of fresh prompts and replayed prefixes. A replay ratio $\rho \in [0, 1]$ determines the fraction of batch elements drawn from $\mathcal{B}_t$, with $\rho = 0$ recovering the base GRPO procedure. Replayed prefixes are sampled with probability proportional to $\hat{\sigma}_t$. For each selected prefix, the current policy generates a fresh group of continuations; the prefix itself is treated as fixed context and masked out of the policy-gradient loss. The GRPO-style objective and group-relative advantage computation are otherwise unchanged.

Cold start. When the base policy almost never reaches successful trajectories, the replay buffer may be initialized with prefixes from a small set of expert successful trajectories. These prefixes are used only as reset states for fresh rollouts under π_t; their tokens are masked from the loss exactly like ordinary replayed prefixes. Recency eviction then gradually replaces seeded prefixes with states produced by the current policy, so the proposal transitions from expert-supported states to on-policy states as training progresses.

Overall, the recent prefix buffer supplies the d^{π} factor by keeping replay states close to the current policy's visitation distribution, while variance-weighted sampling supplies the σ_{π} factor by concentrating rollout budget on uncertain states. Successful-prefix filtering and cold-start seeding bias the construction toward states that lie on completable trajectories, and recency eviction lets the resulting proposal move with the policy. The individual components are not independent replay heuristics, but practical approximations to the same variance-driven state allocation rule.

Table 1: Mathematical reasoning results (%) on AIME24, AIME25, AMC23, OlympiadBench, MATH500, and Minerva. Reinforce-Ada-Est and static uniform expert-prefix resets are additional single-run controls on Llama-3.2-3B-Instruct. Best result in each column is in **bold**.

Model	Method	AIME24	AIME25	AMC23	Olympiad	MATH500	Minerva	Avg
Qwen2.5-7B -Instruct	Base model	10.0	7.9	53.2	**39.2**	**75.6**	37.3	37.2
	GRPO	8.8	4.8	53.4	38.3	72.5	38.6	36.1
	VSR+GRPO	**11.4**	**10.4**	**57.7**	33.5	73.7	**38.7**	**37.6**
Llama-3.2 -3B-Instruct	Base model	5.3	0.6	21.8	12.6	40.0	13.8	15.7
	SFT	0.3	0.4	7.2	5.0	23.1	5.4	6.9
	GRPO	4.4	**1.0**	18.7	12.6	39.2	16.5	15.4
	Reinforce-Ada	4.2	0.1	18.0	12.4	39.9	15.4	15.0
	Uniform expert-prefix	5.6	0.5	**22.7**	14.1	43.7	15.8	17.1
	VSR+GRPO	**6.2**	0.6	20.4	**15.3**	**54.0**	**17.8**	**19.1**

4 Experiments

Our experiments target the regime where a variance-driven proposal is predicted to matter most: tasks on which the base policy almost always fails from the initial state, but where some mid-trajectory states admit continuations with genuinely mixed outcomes. In this regime, σ_π is concentrated on intermediate states that on-policy rollouts rarely reach. We accordingly evaluate VSR in two settings where signals are rare: mathematical reasoning on a hard subset of OpenR1-Math-220k (filtered to prompts with low base-policy pass rate), and long-horizon agentic decision making on Sokoban and difficult BabyAI levels, where successful trajectories remain sparse under the base policies.

4.1 Mathematical reasoning

Setup. We train Qwen2.5-7B-Instruct [Yang et al., 2024] and Llama-3.2-3B-Instruct [Meta, 2024] on a 1K hard subset of OpenR1-Math-220k [Hugging Face, 2025]. The subset is constructed by filtering for prompts with low base-policy pass rate; for Qwen2.5-7B-Instruct, we retain prompts with at most 1 success out of 32 samples. VSR is initialized with replay prefixes extracted from successful DeepSeek-R1 trajectories [DeepSeek-AI et al., 2025]. These expert-seeded prefixes are used only as starting states for fresh rollouts, with prefix tokens masked from the policy-gradient loss, and are gradually replaced by recent on-policy prefixes through the replay buffer.

We compare VSR+GRPO against standard GRPO under the same training setup, and against SFT on the same expert trajectories used for cold-start initialization. We evaluate on AIME24 [MAA Committees, 2024], AIME25 [MAA Committees, 2025], AMC23, OlympiadBench [He et al., 2024], MATH500 [Hendrycks et al., 2021], and Minerva [Lewkowycz et al., 2022]. Final benchmark accuracy is reported in Table 1, and training dynamics, including reward, validation performance, and response length, are shown in Figure 4.

Additional controls. We include two additional single-run controls for the Llama-3.2-3B math setting. Static uniform expert-prefix resets use the same successful expert-trajectory source as VSR but sample uniformly from a fixed expert-prefix pool, probing how much of the gain comes from privileged reset states alone. Reinforce-Ada-Est[Xiong et al., 2025] reallocates a matched response budget across the original prompts without intermediate-state resets, testing whether prompt-level adaptive allocation alone reproduces the VSR gain. As shown in Table 1, static expert-prefix resets improve the average score from 15.4 for GRPO to 17.1, while Reinforce-Ada-Est obtains 15.0 and VSR reaches 19.1. Thus privileged resets explain part of the improvement, whereas prompt-level allocation alone does not reproduce the full VSR result. Detailed setups and limitations are provided in Appendix D.5.

4.2 Long-horizon agentic tasks

Setup. We implement VSR on Sokoban[Schrader, 2018] and BabyAI[Chevalier-Boisvert et al., 2018] using Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct[Yang et al., 2024]. For BabyAI, we use SynthLoc, OneRoom, and MiniBoss, which require multi-step planning and compositional decision making. We choose GiGPO [Feng et al., 2025a] as our base RL algorithm, and VSR is added as a

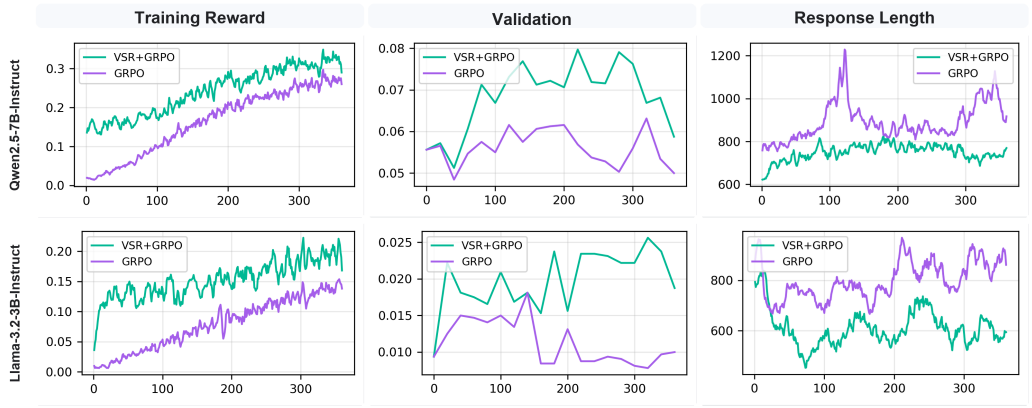

Figure 4: Training dynamics on the 1K hard subset of OpenR1-Math-220k for Qwen2.5-7B-Instruct (top) and Llama-3.2-3B-Instruct (bottom). Columns show training reward, validation accuracy, and response length over training steps. VSR+GRPO sustains higher training reward and validation accuracy than GRPO with shorter responses.

Table 2: Agentic-task success rates (%) on Sokoban and BabyAI. Best result is in **bold**.

Model	Method	Sokoban	BabyAI			
			SynthLoc	OneRoom	MiniBoss	Avg
Qwen2.5-1.5B-Instruct	Base model	9.4	11.6	9.3	21.4	14.1
	GiGPO	12.7	32.3	27.9	30.9	30.4
	VSR+GiGPO	**18.8**	**51.2**	**54.6**	**36.9**	**47.6**
Qwen2.5-3B-Instruct	Base model	17.2	9.3	18.6	16.7	14.9
	GiGPO	21.5	51.2	46.5	30.9	42.9
	VSR+GiGPO	**25.9**	**53.5**	**60.4**	**33.3**	**49.1**

plug-in state-resampling component. Unlike the math setting, the replay buffer is populated entirely from the model's own trajectories, without expert cold-start initialization.

We compare GiGPO+VSR against GiGPO under matched model, environment, and training settings. Final task success rates are reported in Table 2, and training dynamics, including reward, validation success, and newly generated continuation length after the selected reset state, are shown in Figure 5. The main training curves and most entries in Table 2 are from single training runs; evaluation success rates use 128 fixed map seeds per environment, while independent-training-run uncertainty is reported separately for the scoring ablation below.

4.3 Ablations

We isolate the two engineering components of VSR through controlled variants, both built on top of GiGPO and evaluated on BabyAI under the same setup as Section 4.2. **(i) Uniform vs. variance-weighted sampling over the replay buffer.** VSR(w/o scoring) retains the recency-updated replay buffer but samples prefixes uniformly rather than in proportion to $\hat{\sigma}_t$. This isolates the contribution of the variance-driven weighting from that of prefix replay itself. **(ii) Static vs. policy-tracking replay buffer.** VSR(w/o recency) retains $\hat{\sigma}_t$-weighted sampling but disables the recency-based buffer update, so the buffer is populated once at the start of training and held fixed thereafter. This isolates the contribution of tracking d^{π_t} from that of variance-driven weighting.

We additionally repeat the Qwen2.5-1.5B VSR-versus-uniform-replay scoring comparison across three independent training runs per variant. VSR reaches 48.2 ± 2.4 versus 44.0 ± 4.3 for uniform replay (mean $\pm$ SD), providing additional evidence for the benefit of variance-driven prioritization; full per-run results are reported in Appendix D.5. The original no-recency curve remains a single run.

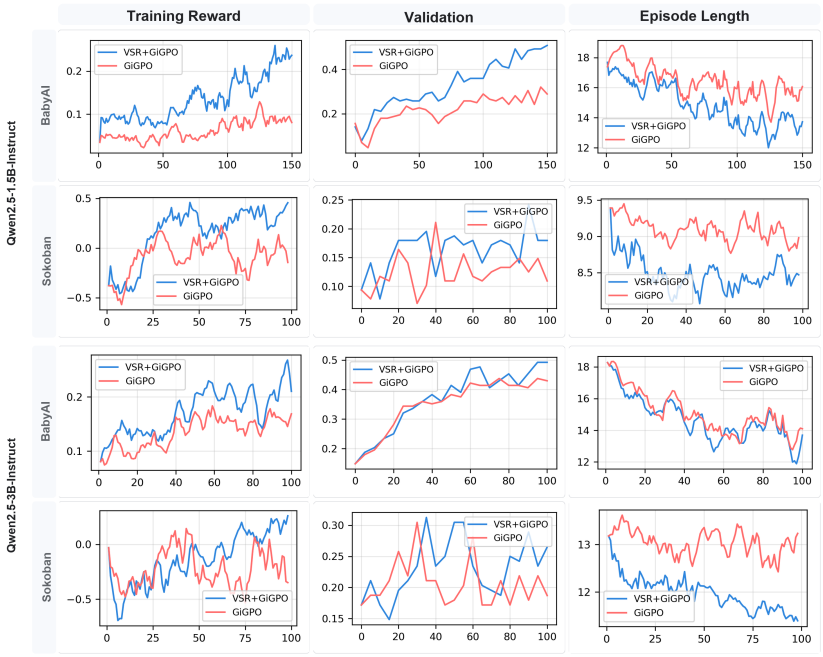

Figure 5: Training dynamics on BabyAI and Sokoban for Qwen2.5-1.5B-Instruct (top two rows) and Qwen2.5-3B-Instruct (bottom two rows). Columns show training reward, validation success rate, and newly generated continuation length after the selected reset state over training steps. The shorter continuation length partly follows mechanically from starting a fraction of rollouts at intermediate states and should not be read as shorter complete trajectories from the original task state.

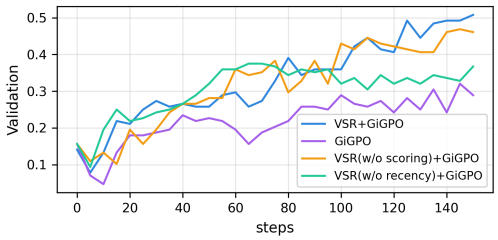

Figure 6: Ablation on BabyAI: validation success rate over training.

Table 3: Ablation on BabyAI (avg success rate, %). Best in **bold**.

Method	Avg
GiGPO	30.4
VSR(w/o scoring)	46.1
VSR(w/o recency)	36.7
VSR	**47.6**

4.4 Discussion

VSR yields stronger per-batch training signal. The training-reward curves in Figures 4 and 5 indicate that VSR yields a higher per-batch gradient signal: VSR consistently gains more training reward over the whole training. On the hard math subset, GRPO's training reward continues to climb while its validation accuracy stays essentially flat, and final benchmark performance ends up below the base model on both models. The reason is that training signal is concentrated on a narrow handful of prompts that occasionally produce mixed-outcome groups, and on-policy sampling repeatedly draws gradient signal from that same small set. VSR addresses this at the level predicted by μ^{π}_{var}: even when the prompt-level support of σ_{π} is narrow, its state-level support is much wider.

VSR reduces the rollout cost of each policy update. Because a non-trivial fraction of rollouts are initiated from mid-trajectory prefixes rather than from the original prompt, the expected number of newly generated tokens per trajectory is reduced; this is reflected in the consistently shorter newly generated continuation lengths in the agentic experiments and the analogous reduction in completion length observed in the mathematical reasoning experiments. Direct token accounting further supports this effect: at matched update counts, VSR+GiGPO generates 39.27M versus 81.30M

tokens for Qwen2.5-1.5B and 34.43M versus 36.26M for Qwen2.5-3B; full accounting is reported in Appendix D.5.

Variance-driven scoring improves training efficiency over uniform replay. The comparison between VSR and VSR(w/o scoring) in Figure 6 indicates that weighting buffered prefixes by $\hat{\sigma}_t$ yields higher validation performance on average over uniform sampling from the same buffer, particularly in the later phase of training. Across three independent training runs, VSR achieves 48.2 ± 2.4 versus 44.0 ± 4.3 for uniform replay (mean $\pm$ SD), further supporting the benefit of variance-driven prioritization; see Appendix D.5 for the individual runs. This indicates that the variance-driven weighting is not redundant with the replay mechanism itself: prioritizing prefixes with larger residual outcome variance translates into a more efficient use of the resampling budget.

Recency-based buffer updates are necessary to sustain learning. VSR(w/o recency) outperforms full VSR initially but plateaus thereafter. Without ongoing updates, the buffer's empirical state distribution drifts away from d^{π_t} as the policy evolves, and the policy is optimized against a proposal that no longer reflects its current visitation. The d^{π} factor in μ^{π}_{var} is precisely what guards against this distributional shift, consistent with the prediction in Section 2.3 that the variance-optimal proposal re-localizes with π.

5 Conclusion

We recast the signal-sparsity problem in RL with verifiable rewards as a variance-reduction problem over the state sampling distribution, and showed that the general trace-variance optimum depends on the action-gradient norm; under the stated within-state and across-state homogeneity conditions it reduces to the clean Neyman form $\mu^{\pi}_{\mathrm{var}} \propto d^{\pi}(s)\sigma_{\pi}(s)$, whose proposal-corrected estimator has the same standardized-residual form as GRPO's group-relative normalization. Building on this allocation principle, we proposed VSR, a practical prefix-replay method that improves over the compared GRPO/GiGPO baselines on most evaluated settings.

A main limitation is the gap between the exact proposal-corrected estimator and practical VSR. VSR does not estimate the exact density ratio, stores only successful prefixes, uses expert-prefix cold start in the math experiments, and relies on a coarse zero-additional-rollout uncertainty score. The additional controls and validations in Appendix D.5 clarify these effects but do not completely isolate every practical component; most main agentic results and the no-recency ablation also remain single-run. Tighter low-cost state scores and broader multi-seed evaluation remain important directions.

Acknowledgments and Disclosure of Funding

The authors declare no external funding.

The authors declare no competing interests.

References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Lili Chen, Mihir Prabhudesai, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Self-questioning language models. *arXiv preprint arXiv:2508.03682*, 2025.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. *arXiv preprint arXiv:1810.08272*, 2018.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, 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, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, 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, 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, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, 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, Yanhong Xu, 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: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025a.

Yunzhen Feng, Ariel Kwiatkowski, Kunhao Zheng, Julia Kempe, and Yaqi Duan. Pilaf: Optimal human preference sampling for reward modeling. *arXiv preprint arXiv:2502.04270*, 2025b.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pages 482–495. PMLR, 2017.

Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. Prompt curriculum learning for efficient llm post-training. *arXiv preprint arXiv:2510.01135*, 2025.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Patrick Haluptzok, Matthew Bowers, and Adam Tauman Kalai. Language models can teach themselves to program better. *arXiv preprint arXiv:2207.14502*, 2022.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng 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. *arXiv preprint arXiv:2402.14008*, 2024.

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

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*, 2025.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL `https://github.com/huggingface/open-r1`.

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.

Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, and Jiang Bian. Self-hinting language models enhance reinforcement learning. *arXiv preprint arXiv:2602.03143*, 2026.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, et al. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*, 2025.

Hongliang Lu, Yuhang Wen, Pengyu Cheng, Ruijin Ding, Jiaqi Guo, Haotian Xu, Chutian Wang, Haonan Chen, Xiaoxi Jiang, and Guanjun Jiang. Search self-play: Pushing the frontier of agent capability without supervision. *arXiv preprint arXiv:2510.18821*, 2025.

MAA Committees. AIME Problems and Solutions. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`, 2024.

MAA Committees. AIME Problems and Solutions. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`, 2025.

AI Meta. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2(5):6, 2024.

Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664*, 2020.

Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. Pope: Learning to reason on hard problems via privileged on-policy exploration. *arXiv preprint arXiv:2601.18779*, 2026.

Max-Philipp B. Schrader. gym-sokoban, January 2018. URL `https://github.com/mpschrader/gym-sokoban`.

Amrith Setlur, Matthew YR Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. e3: Learning to explore enables extrapolation of test-time compute for llms. *arXiv preprint arXiv:2506.09026*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Ruizhe Shi, Runlong Zhou, and Simon S Du. The crucial role of samplers in online direct preference optimization. *arXiv preprint arXiv:2409.19605*, 2024.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*, 2025.

Toby Simonds and Akira Yoshiyama. Ladder: Self-improving llms through recursive problem decomposition. *arXiv preprint arXiv:2503.00735*, 2025.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Yinjie Wang, Tianbao Xie, Ke Shen, Mengdi Wang, and Ling Yang. Rlanything: Forge environment, policy, and reward model in completely dynamic rl system. *arXiv preprint arXiv:2602.02488*, 2026.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

Alex Wilf, Pranjal Aggarwal, Bryan Parno, Daniel Fried, Louis-Philippe Morency, Paul Pu Liang, and Sean Welleck. Propose, solve, verify: Self-play through formal verification. *arXiv preprint arXiv:2512.18160*, 2025.

Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, Wei He, Yiwen Ding, Shichun Liu, Xin Guo, Junzhe Wang, et al. Training large language models for reasoning through reverse curriculum reinforcement learning. *arXiv preprint arXiv:2402.05808*, 2024.

Wei Xiong, Chenlu Ye, Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, Jiang Bian, Nan Jiang, and Tong Zhang. Reinforce-ada: An adaptive sampling framework under non-linear rl objectives. *arXiv preprint arXiv:2510.04996*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Jiarui Yao, Yifan Hao, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and rl. *arXiv preprint arXiv:2505.02391*, 2025.

Junkeun Yi, Damon Mosk-Aoyama, Baihe Huang, Ritu Gala, Charles Wang, Sugam Dipak Devare, Khushi Bhardwaj, Abhibha Gupta, Oleksii Kuchaiev, Jiantao Jiao, et al. Pivotrl: High accuracy agentic post-training at low compute cost. *arXiv preprint arXiv:2603.21383*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Zhiyuan Zeng, Hamish Ivison, Yiping Wang, Lifan Yuan, Shuyue Stella Li, Zhuorui Ye, Siting Li, Jacqueline He, Runlong Zhou, Tong Chen, et al. Rlve: Scaling up reinforcement learning for language models with adaptive verifiable environments. *arXiv preprint arXiv:2511.07317*, 2025.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. Bread: Branched rollouts from expert anchors bridge sft & rl for reasoning. *arXiv preprint arXiv:2506.17211*, 2025.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*, 2025.

Zirui Zhao, Hanze Dong, Amrita Saha, Caiming Xiong, and Doyen Sahoo. Automatic curriculum expert iteration for reliable llm reasoning. *arXiv preprint arXiv:2410.07627*, 2024.

A Related Work

A broad line of recent work on sparse-reward RL for LLMs aims, at varying levels of explicitness, to address the same underlying difficulty: when terminal reward is binary and success is rare, naively sampling rollouts under the current policy yields many uninformative updates. We organize the discussion around what each line of work effectively reshapes—the prompt distribution, the within-trajectory state at which generation is conditioned, the task itself, or the per-prompt sampling budget.

Selecting and scheduling existing prompts. A first family of methods operates at the prompt level. Filtering and dynamic sampling discard prompts whose rollout groups produce uniform reward, on the grounds that they contribute no group-relative signal [Yu et al., 2025, Cui et al., 2025, Wang et al., 2025, Gulcehre et al., 2023]. Curriculum-style schedulers go further and order prompts by difficulty or learnability, ranging from classical curriculum learning [Bengio et al., 2009, Portelas et al., 2020] to recent LLM-specific instantiations that adapt the prompt distribution online [Zhao et al., 2024, Shi et al., 2025, Gao et al., 2025, Team et al., 2025, Setlur et al., 2025]. RLVE [Zeng et al., 2025] makes the environment itself adaptive, generating verifiable problem instances calibrated to the current learner. A separate but related line studies the role of the sampler in preference-based post-training, where sampling distribution similarly affects gradient signal [Shi et al., 2024, Feng et al., 2025b]. The granularity of intervention in this family is the prompt: once a prompt is selected, the distribution of intermediate states encountered along its rollouts is left entirely to the current policy.

Conditioning on intermediate states within a trajectory. A second family conditions generation on intermediate states rather than only on prompts. Prefix-conditioned RL methods inject partial trajectories—either expert-derived or drawn from earlier policy iterates—as starting points for fresh rollouts [Zhang et al., 2025, Qu et al., 2026, Yi et al., 2026], extending an older RL tradition of reverse curricula style state resets [Andrychowicz et al., 2017, Florensa et al., 2017, Ecoffet et al., 2019, Xi et al., 2024] . Hint injection during rollout [Liao et al., 2026] and recursive problem decomposition [Simonds and Yoshiyama, 2025] can be viewed as variants of the same idea, supplying privileged context that places the policy at a more learnable state before continuation. ReST [Gulcehre et al., 2023] is a closer precedent on the replay side, iteratively training on successful trajectories, though as an offline SFT-then-RL procedure rather than as an online state-resampling mechanism. The criterion by which intermediate states are selected varies across these methods—difficulty, expert provenance, or a hand-designed schedule.

Generating new tasks: self-play schemes. A third family sidesteps the question of which existing state to train on by generating new tasks calibrated to the learner. Self-play and proposer–solver frameworks have the model—or a paired proposer model—propose problems that are subsequently solved and verified, with the proposal policy itself trained to produce informative tasks [Haluptzok et al., 2022, Zhao et al., 2025, Huang et al., 2025, Chen et al., 2025, Lu et al., 2025, Liu et al., 2025, Wilf et al., 2025, Wang et al., 2026] . The motivation of self-play schemes overlaps with curriculum learning and prefix conditioning, putting training mass on states from which learning signal is accessible

Variance-driven sampling budget reallocation. Closest in motivation are methods that explicitly target estimator variance. While a classical thread on variance reduction in policy-gradient estimators more broadly [Greensmith et al., 2004] focuses on baseline design rather than proposal design, some recent researches focus on prompt sampling. Reinforce-Ada [Xiong et al., 2025] adaptively allocates rollouts across prompts under non-linear RL objectives, and Yao et al. [2025] minimizes gradient variance through rejection sampling and rollout reallocation in chain-of-thought RL. Both are concerned with the same underlying inefficiency—sampling budget spent on prompts that yield no usable signal—and both reallocate rollouts across prompts, holding the per-prompt distribution of intermediate states fixed.

B Theoretical analysis of the variance-driven proposal

This appendix collects the full theoretical analysis underlying the variance-driven proposal $\mu_{\mathrm{var}} \propto d^{\pi}(s)\sigma_{\pi}(s)$ introduced in Section 2.2. Section B.1 establishes notation and the working form of the estimator variance. Section B.2 derives the exact variance-optimal proposal. Section B.3 specializes it to the standard gradient-uniformity convention, recovering μ_{var} and the GRPO-style

estimator. Section B.4 relaxes that convention and gives a quantitative condition—weaker than gradient uniformity—under which μ_{var} still strictly reduces variance relative to on-policy sampling.

B.1 Setup

Throughout, we use the state-action view of LLM RL described in Section 2.1: $s_t = (x, y_{<t})$, $a_t = y_t$, $s_0 = x \sim \mathcal{D}$, with terminal verifiable reward $R \in \{0, 1\}$. The state-visitation distribution is

$$d^\pi(s) \;\propto\; \sum_{t=0}^{T-1} \Pr\big[s_t = s \,\big|\, s_0 \sim \mathcal{D},\, \pi\big].$$

For a state-dependent baseline b and a proposal μ with $\mathrm{supp}(\mu) \supseteq \mathrm{supp}(d^\pi)$, the proposal-corrected policy-gradient estimator is

$$\widehat{\nabla J}_{b,\mu} \;=\; \frac{d^\pi(s)}{\mu(s)}\big(r(s,a) - b(s)\big)\nabla \log \pi(a \mid s), \qquad s \sim \mu,\; a \sim \pi(\cdot \mid s), \tag{11}$$

which is unbiased for $\nabla J(\pi)$ for every admissible pair (b, μ). We write $g(s,a) := \nabla \log \pi(a \mid s)$, $\delta(s,a) := r(s,a) - b(s)$, $\bar{r}(s) := \mathbb{E}_{a\sim\pi(\cdot|s)}[r(s,a)]$, and use $\sigma_\pi(s)$ for the within-state reward standard deviation, $\sigma_\pi(s)^2 := \mathrm{Var}_{a\sim\pi(\cdot|s)}\big(r(s,a)\big)$.

For a fixed baseline b, define the gradient-signal second moment

$$M_b(s) \;:=\; \mathbb{E}_{a\sim\pi(\cdot|s)}\big[(r(s,a) - b(s))^2\, H(s,a)\big], \qquad H(s,a) := \|g(s,a)\|^2. \tag{12}$$

Lemma B.1 (Working form of the variance)**.** *For any admissible (b, μ),*

$$V(\mu; b) \;:=\; \mathrm{tr}\,\mathrm{Cov}\big(\widehat{\nabla J}_{b,\mu}\big) \;=\; \mathbb{E}_{s\sim d^\pi}\left[\frac{d^\pi(s)}{\mu(s)}\, M_b(s)\right] \;-\; \|\nabla J(\pi)\|^2. \tag{13}$$

Proof. By unbiasedness, $\mathrm{tr}\,\mathrm{Cov}(\widehat{\nabla J}_{b,\mu}) = \mathbb{E}\,\|\widehat{\nabla J}_{b,\mu}\|^2 - \|\nabla J(\pi)\|^2$. Expanding $s \sim \mu$ gives

$$\mathbb{E}\,\|\widehat{\nabla J}_{b,\mu}\|^2 = \mathbb{E}_{d^\pi}\left[\frac{d^\pi(s)}{\mu(s)} M_b(s)\right]. \qquad \square$$

B.2 The exact joint optimum

Theorem B.2 (Exact joint-optimal baseline and proposal)**.** *For each state, the trace-variance-minimizing baseline is*

$$b^*_w(s) = \frac{\mathbb{E}[r(s,a)H(s,a) \mid s]}{\mathbb{E}[H(s,a) \mid s]}. \tag{14}$$

Defining

$$M^*_w(s) := \mathbb{E}[(r(s,a) - b^*_w(s))^2 H(s,a) \mid s], \tag{15}$$

the joint-optimal proposal on the effective support is

$$\mu^{\pi,\star}(s) \propto d^\pi(s)\sqrt{M^*_w(s)}. \tag{16}$$

If full support is required, the proposal can be smoothed by mixing with d^π.

Proof. For fixed μ, the statewise factor multiplying $M_b(s)$ in (13) is positive, so minimizing over $b(s)$ is equivalent to minimizing $\mathbb{E}[(r-b)^2 H \mid s]$. Differentiating in b gives (14). Substituting b^*_w into (13), Cauchy–Schwarz yields the minimizing proposal $\mu(s) \propto d^\pi(s)\sqrt{M^*_w(s)}$ exactly as in the standard Neyman-allocation argument. $\square$

B.3 Clean case: conditions yielding $d^\pi\sigma_\pi$

The exact joint optimum couples reward variability and action-gradient norms. The following two conditions separate these quantities.

Assumption B.3 (Within-state conditional-moment condition)**.** *For each state s, there exists $G(s) > 0$ such that*

$$\mathbb{E}[H(s,a) \mid s, r] = G(s). \tag{17}$$

That is, conditional on the state, the expected squared score-function norm does not depend on the realized reward.

Under Assumption B.3, $b_w^*(s) = \bar{r}(s)$ and

$$M_w^*(s) = G(s)\sigma_\pi(s)^2, \qquad \mu^{\pi,\star}(s) \propto d^\pi(s)\sigma_\pi(s)\sqrt{G(s)}. \tag{18}$$

Assumption B.4 (Across-state gradient-scale homogeneity)**.** *$G(s) \equiv G_0$ on the effective support.*

Theorem B.5 (Clean Neyman form)**.** *Under Assumptions B.3 and B.4, the joint-optimal proposal on the effective support reduces to*

$$\mu^{\pi,\star}(s) \;\propto\; d^\pi(s)\,\sigma_\pi(s) \;=:\; \mu_{\mathrm{var}}(s), \tag{19}$$

and the exact proposal-corrected estimator is proportional to

$$\frac{r(s,a) - \bar{r}(s)}{\sigma_\pi(s)} \nabla \log \pi(a \mid s), \qquad s \sim \mu_{\mathrm{var}},\ a \sim \pi(\cdot \mid s). \tag{20}$$

Moreover,

$$V(d^\pi; \bar{r}) - V(\mu_{\mathrm{var}}; \bar{r}) = G_0 \operatorname{Var}_{s\sim d^\pi}(\sigma_\pi(s)) \geq 0, \tag{21}$$

with strict inequality whenever σ_π is non-constant on the effective support.

Proof. Assumption B.3 gives $b_w^* = \bar{r}$ and $M_w^* = G(s)\sigma_\pi^2(s)$. Assumption B.4 then gives $M_w^* = G_0\sigma_\pi^2$, so (16) reduces to (19); substituting in the proposal-corrected estimator yields (20). The variance-gap expression follows from the same Jensen calculation as in the original clean-case derivation. □

B.4 Weak version: variance reduction beyond gradient uniformity

The clean conditions above are idealizations that fail to varying degrees in practice: different states have different conditional gradient-norm scales, and these scales may co-vary with σ_π across states. The natural question is whether $\mu_{\mathrm{var}} \propto d^\pi\sigma_\pi$—which no longer coincides with the exact optimum (16) once Assumption B.4 is dropped—is still a strict improvement over on-policy sampling, and under what condition.

To state this cleanly we factor $M(s)$ through $\sigma_\pi(s)$. On $\{s : \sigma_\pi(s) > 0\}$, define

$$K(s) \;:=\; \frac{M(s)}{\sigma_\pi(s)^2}. \tag{22}$$

Assumption B.4 is equivalent to $K \equiv G_0$ being a state-independent constant; relaxing it amounts to allowing K to vary across s.

For notational simplicity, we assume throughout this subsection that $\sigma_\pi(s) > 0$ for d^π-almost every state. States with $\sigma_\pi(s) = 0$ carry no reward-variance signal under $b = \bar{r}$ and can be removed by working on the effective support, with the corresponding normalization factors tracked explicitly.

The relaxed assumption is a quantitative bound on the heterogeneity of K, measured by its coefficient of variation, relative to the spread of σ_π:

Assumption B.6 (Gradient regularity)**.** *On the set of states $\{s : \sigma_\pi(s) > 0\}$, the function $K(s)$ satisfies*

$$\frac{\sqrt{\operatorname{Var}_{d^\pi}(K)}}{\mathbb{E}_{d^\pi}[K]} \;<\; \frac{2}{3}\sqrt{\operatorname{Var}_{d^\pi}(\sigma_\pi)}, \tag{23}$$

where the moments are taken with respect to $s \sim d^\pi$ restricted to that set.

Assumption B.6 is strictly weaker than Assumption B.4: gradient uniformity is the limit $\operatorname{Var}(K) = 0$, in which the left-hand side is zero and (23) is vacuous (provided σ_π is non-constant). It admits a transparent reading: μ_{var} remains beneficial as long as the relative state-to-state variation of K is small compared to the absolute spread of σ_π on the policy's reachable set—i.e., as long as the policy's reachable-but-uncertain frontier is wide enough to outweigh whatever heterogeneity K exhibits across states.

Theorem B.7 (Variance reduction under gradient regularity)**.** *Suppose $r \in \{0, 1\}$ and $b = \bar{r}$. Under Assumption B.6,*

$$V(\mu_{\text{var}}; b^*) \;<\; V(d^\pi; b^*). \tag{24}$$

The proof proceeds via an exact decomposition of the variance gap, isolating a non-positive Jensen-type term and a coupling correction that vanishes under Assumption B.4.

Lemma B.8 (Variance gap decomposition)**.** *Let $b = \bar{r}$. Then, with all expectations, variances, and covariances taken with respect to $s \sim d^\pi$ restricted to $\{\sigma_\pi > 0\}$,*

$$V(\mu_{\text{var}}; b^*) - V(d^\pi; b^*) \;=\; -\,\mathbb{E}[K] \cdot \text{Var}(\sigma_\pi) \;+\; \Delta_{\text{coup}}, \tag{25}$$

where $\Delta_{\text{coup}} := \mathbb{E}[\sigma_\pi]\,\text{Cov}(\sigma_\pi, K) - \text{Cov}(\sigma_\pi^2, K)$.

Proof. Let $Z := \mathbb{E}_{d^\pi}[\sigma_\pi] > 0$, so $\mu_{\text{var}} = d^\pi \sigma_\pi / Z$. By Lemma B.1, the $\|\nabla J\|^2$ terms cancel and

$$V(\mu_{\text{var}}; b^*) - V(d^\pi; b^*) = Z\,\mathbb{E}_{d^\pi}\left[\frac{M(s)}{\sigma_\pi(s)}\right] - \mathbb{E}_{d^\pi}[M(s)].$$

Substituting $M = \sigma_\pi^2 K$,

$$= \mathbb{E}[\sigma_\pi]\,\mathbb{E}[\sigma_\pi K] - \mathbb{E}[\sigma_\pi^2 K].$$

Apply $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] + \text{Cov}(X, Y)$ to both products:

$$\mathbb{E}[\sigma_\pi]\,\mathbb{E}[\sigma_\pi K] = \mathbb{E}[\sigma_\pi]^2\,\mathbb{E}[K] + \mathbb{E}[\sigma_\pi]\,\text{Cov}(\sigma_\pi, K),$$
$$\mathbb{E}[\sigma_\pi^2 K] = \mathbb{E}[\sigma_\pi^2]\,\mathbb{E}[K] + \text{Cov}(\sigma_\pi^2, K).$$

Subtracting and using $\text{Var}(\sigma_\pi) = \mathbb{E}[\sigma_\pi^2] - \mathbb{E}[\sigma_\pi]^2$ gives (25). □

Lemma B.9 (Variance-of-square bound)**.** *For any random variable X with values in $[0, M]$,* $\text{Var}(X^2) \le 4M^2\,\text{Var}(X)$.

Proof. Let X' be an independent copy of X. Then

$$\text{Var}(X^2) = \tfrac{1}{2}\,\mathbb{E}[(X^2 - X'^2)^2] = \tfrac{1}{2}\,\mathbb{E}[(X+X')^2(X-X')^2] \le 2M^2\,\mathbb{E}[(X-X')^2] = 4M^2\,\text{Var}(X),$$

where the inequality uses $0 \le X + X' \le 2M$ almost surely. □

Proof of Theorem B.7. By Lemma B.8, it suffices to show $\mathbb{E}[K]\,\text{Var}(\sigma_\pi) > \Delta_{\text{coup}}$. We bound $|\Delta_{\text{coup}}|$ from above.

By Cauchy–Schwarz,

$$|\text{Cov}(\sigma_\pi, K)| \le \sqrt{\text{Var}(\sigma_\pi)\text{Var}(K)}, \qquad |\text{Cov}(\sigma_\pi^2, K)| \le \sqrt{\text{Var}(\sigma_\pi^2)\text{Var}(K)}.$$

For binary rewards, $\sigma_\pi(s) \le 1/2$ structurally, so Lemma B.9 with $M = 1/2$ gives $\text{Var}(\sigma_\pi^2) \le \text{Var}(\sigma_\pi)$ and hence $|\text{Cov}(\sigma_\pi^2, K)| \le \sqrt{\text{Var}(\sigma_\pi)\text{Var}(K)}$. Combined with $\mathbb{E}[\sigma_\pi] \le 1/2$,

$$|\Delta_{\text{coup}}| \;\le\; \tfrac{1}{2}\sqrt{\text{Var}(\sigma_\pi)\text{Var}(K)} + \sqrt{\text{Var}(\sigma_\pi)\text{Var}(K)} \;=\; \tfrac{3}{2}\sqrt{\text{Var}(\sigma_\pi)\text{Var}(K)}. \tag{26}$$

Plugging (26) into (25),

$$V(\mu_{\text{var}}; b^*) - V(d^\pi; b^*) \;\le\; -\,\mathbb{E}[K]\,\text{Var}(\sigma_\pi) + \tfrac{3}{2}\sqrt{\text{Var}(\sigma_\pi)\text{Var}(K)}.$$

The right-hand side is negative iff $\mathbb{E}[K]\sqrt{\text{Var}(\sigma_\pi)} > \tfrac{3}{2}\sqrt{\text{Var}(K)}$, i.e. iff $\sqrt{\text{Var}(K)}/\mathbb{E}[K] < \tfrac{2}{3}\sqrt{\text{Var}(\sigma_\pi)}$, which is exactly Assumption B.6. □

Remark B.10 (Beyond binary rewards)**.** *For $r \in [0, R]$, $\sigma_\pi(s) \le R/2$ and Lemma B.9 gives* $\text{Var}(\sigma_\pi^2) \le R^2\text{Var}(\sigma_\pi)$. *The proof of Theorem B.7 carries over with constants scaled accordingly:* $|\Delta_{\text{coup}}| \le \frac{3R}{2}\sqrt{\text{Var}(\sigma_\pi)\text{Var}(K)}$, *and Assumption B.6 becomes* $\sqrt{\text{Var}(K)}/\mathbb{E}[K] < \frac{2}{3R}\sqrt{\text{Var}(\sigma_\pi)}$.

Algorithm 2 State-Visitation Policy Mirror Descent (SVMD)

Require: Initial policy π_1, horizon H, iterations T, per-stage state budgets $\{N_h\}_{h=1}^H$, step size $\eta > 0$, smoothing parameter $\varepsilon \in (0, 1]$, state-score oracle $\sigma_h^\pi(\cdot)$, stagewise critic classes $\{\mathcal{F}_h\}_{h=1}^H$.

1: **for** $t = 1, \dots, T$ **do**

2: **for** $h = 1, \dots, H$ **do**

3: Define the proposal

$$\widetilde{\mu}_h^t(s) \propto d_h^{\pi_t}(s)\sigma_h^{\pi_t}(s), \qquad \mu_h^t(s) := (1-\varepsilon)\widetilde{\mu}_h^t(s) + \varepsilon\, d_h^{\pi_t}(s),$$

and importance weights $w_h^t(s) := d_h^{\pi_t}(s)/\mu_h^t(s) \in [0, 1/\varepsilon]$.

4: Sample i.i.d. data $\{(s_{h,i}, a_{h,i}, G_{h,i})\}_{i=1}^{N_h}$: $s_{h,i} \sim \mu_h^t$, $a_{h,i} \sim \pi_{t,h}(\cdot \mid s_{h,i})$, and $G_{h,i} := \sum_{\ell=h}^H r_\ell(s_\ell, a_\ell)$ is the Monte Carlo return-to-go obtained by rolling out π_t from $(s_{h,i}, a_{h,i})$.

5: *Weighted critic fit:*

$$\widehat{Q}_h^t \in \arg\min_{f \in \mathcal{F}_h} \sum_{i=1}^{N_h} w_h^t(s_{h,i})\big(f(s_{h,i}, a_{h,i}) - G_{h,i}\big)^2.$$

6: *KL-regularized mirror ascent:* for each $s \in \mathcal{S}$,

$$\pi_{t+1,h}(\cdot \mid s) \leftarrow \arg\max_{p \in \Delta(\mathcal{A})} \Big\{\eta\, \langle \widehat{Q}_h^t(s, \cdot),\, p\rangle - \mathrm{KL}\big(p \,\big\|\, \pi_{t,h}(\cdot \mid s)\big)\Big\}.$$

7: **end for**

8: **end for**

9: **return** uniform mixture $\bar{\pi}_T := \mathrm{Unif}\{\pi_1, \dots, \pi_T\}$.

C State-Visitation Policy Mirror Descent: Theory

This appendix analyzes an idealized algorithm—State-Visitation Policy Mirror Descent (SVMD)—that samples states from a smoothed version of $d^\pi\sigma$ and uses an explicit critic fitted by importance-weighted regression. SVMD is *not* the GRPO-based algorithm of the main text: it assumes an oracle for $\sigma_h^\pi(s)$ and replaces GRPO's group-relative advantage with a fitted Q. We include it as theoretical evidence that proposals of the form $d^\pi\sigma$ are well-behaved in a tractable PMD setting; a direct finite-sample analysis of the practical algorithm is left to future work.

C.1 Setup

We consider a finite-horizon episodic MDP with horizon H, state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernels $P_h(\cdot \mid s, a)$, reward functions $r_h(s, a) \in [0, 1]$, and initial state distribution ρ. A (non-stationary) policy is $\pi = (\pi_1, \dots, \pi_H)$. Define

$$V_h^\pi(s) := \mathbb{E}_\pi\left[\sum_{\ell=h}^H r_\ell(s_\ell, a_\ell) \,\middle|\, s_h = s\right], \qquad Q_h^\pi(s, a) := r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)}[V_{h+1}^\pi(s')],$$

$$A_h^\pi(s, a) := Q_h^\pi(s, a) - V_h^\pi(s), \qquad J(\pi) := \mathbb{E}_{s_1 \sim \rho}[V_1^\pi(s_1)].$$

Let d_h^π denote the stage-h state occupancy of π, and let $\nu_h^\pi(s, a) := d_h^\pi(s)\pi_h(a \mid s)$ be the stage-h state-action occupancy. Throughout the appendix, $\sigma_h^\pi : \mathcal{S} \to \mathbb{R}_{\geq 0}$ denotes a fixed nonnegative *state-score function* associated with policy π at stage h (e.g. $\sigma_h^\pi(s) = \sqrt{\mathrm{Var}_{a \sim \pi_h(\cdot|s)}(Q_h^\pi(s, a))}$). Our analysis does not depend on the specific form of σ.

C.2 Algorithm

The mirror update has the closed form $\pi_{t+1,h}(a \mid s) \propto \pi_{t,h}(a \mid s)\exp(\eta \widehat{Q}_h^t(s, a))$, i.e. a softmax NPG step.

C.3 Assumptions

Assumption C.1 (Bounded rewards and bounded critic class)**.** *For all h, s, a, $r_h(s,a) \in [0,1]$, hence $0 \le Q_h^\pi(s,a) \le H$ for all π, h, s, a. Moreover $\mathcal{F}_h \subseteq \{f : \mathcal{S} \times \mathcal{A} \to [0,H]\}$.*

Assumption C.2 (Stagewise realizability)**.** *For every iteration $t \in [T]$ and stage $h \in [H]$, $Q_h^{\pi_t} \in \mathcal{F}_h$.*

Assumption C.3 (Comparator coverage)**.** *There exist $\kappa \ge 1$ and a comparator policy μ such that for all $t \in [T]$, $h \in [H]$, (s,a),*

$$\nu_h^\mu(s,a) \le \kappa\, \nu_h^{\pi_t}(s,a).$$

Assumption C.4 (Proposal smoothing)**.** *For every $t \in [T]$, $h \in [H]$, $s \in \mathcal{S}$, $\mu_h^t(s) \ge \varepsilon\, d_h^{\pi_t}(s)$, equivalently $d_h^{\pi_t}(s)/\mu_h^t(s) \le 1/\varepsilon$. (This holds by construction of μ_h^t in Algorithm 2.)*

C.4 Main Theorem

Theorem C.5 (Finite-sample guarantee for SVMD)**.** *Under Assumptions C.1–C.4, fix $\delta \in (0,1)$. With probability at least $1-\delta$, the policy mixture $\bar\pi_T$ returned by Algorithm 2 satisfies*

$$J(\mu) - J(\bar\pi_T) \;\le\; \frac{D_\mu}{\eta T} \;+\; \frac{\eta H^3}{2} \;+\; 2\kappa \sum_{h=1}^H \beta_h,$$

where

$$D_\mu := \sum_{h=1}^H \mathbb{E}_{s\sim d_h^\mu}\big[\mathrm{KL}\big(\mu_h(\cdot \mid s) \,\big\|\, \pi_{1,h}(\cdot \mid s)\big)\big], \qquad \beta_h := \sqrt{C \cdot \frac{H^2}{\varepsilon} \cdot \frac{\log(HT|\mathcal{F}_h|/\delta)}{N_h}},$$

for a universal constant $C > 0$. In particular, choosing $\eta = \sqrt{2D_\mu/(H^3T)}$ yields

$$J(\mu) - J(\bar\pi_T) \;\le\; \sqrt{\frac{2D_\mu H^3}{T}} \;+\; 2\kappa \sum_{h=1}^H \beta_h.$$

C.5 Proof Outline

The proof has three modular parts.

1. **Per-step mirror descent inequality** (Lemma C.7). The KL-regularized update yields, stagewise and per iteration, a bound on $\eta\langle \widehat{Q}_h^t, \mu_h - \pi_{t,h}\rangle$ in terms of the progress $\mathrm{KL}(\mu_h\|\pi_{t,h}) - \mathrm{KL}(\mu_h\|\pi_{t+1,h})$ plus a quadratic drift term.
2. **Telescoping and averaging.** Summing the per-step inequality over t and h and dividing by T controls the average inner product between $\widehat{Q}_h^t$ and $\mu_h - \pi_{t,h}$.
3. **Critic error control** (Lemma C.8). The performance difference lemma reduces $J(\mu) - J(\pi_t)$ to the same inner product plus a critic error $\mathcal{E}_{h,t}$. Coverage (Assumption C.3) bounds $|\mathcal{E}_{h,t}|$ by an on-policy L^1 critic error, which is in turn controlled by weighted ERM analysis under realizability and proposal smoothing.

C.6 Lemmas

Lemma C.6 (Performance difference)**.** *For any policies π, μ,*

$$J(\mu) - J(\pi) = \sum_{h=1}^H \mathbb{E}_{s\sim d_h^\mu}\Big[\big\langle Q_h^\pi(s,\cdot),\, \mu_h(\cdot \mid s) - \pi_h(\cdot \mid s)\big\rangle\Big].$$

Proof. Standard finite-horizon performance difference lemma; see e.g. Kakade and Langford (2002). □

Lemma C.7 (Mirror descent per-step inequality)**.** *Fix $t \in [T]$ and $h \in [H]$, and any state s. Let $\pi_{t+1,h}(\cdot \mid s)$ be the mirror update in Algorithm 2. Then for any $p \in \Delta(\mathcal{A})$,*

$$\eta\,\big\langle \widehat{Q}_h^t(s,\cdot),\, p - \pi_{t,h}(\cdot \mid s)\big\rangle \;\le\; \mathrm{KL}\big(p \,\big\|\, \pi_{t,h}(\cdot \mid s)\big) - \mathrm{KL}\big(p \,\big\|\, \pi_{t+1,h}(\cdot \mid s)\big) + \frac{\eta^2 H^2}{2}.$$

Proof. Fix s and write π_t, π_{t+1} for $\pi_{t,h}(\cdot \mid s), \pi_{t+1,h}(\cdot \mid s)$, $\widehat{Q}$ for $\widehat{Q}_h^t(s,\cdot)$.

Step 1: First-order optimality. Since π_{t+1} maximizes $\eta\langle \widehat{Q}, p\rangle - \mathrm{KL}(p\|\pi_t)$ over $p \in \Delta(\mathcal{A})$ in the relative interior, the closed-form solution is $\pi_{t+1}(a) \propto \pi_t(a)\exp(\eta\widehat{Q}(a))$, equivalently

$$\log \pi_{t+1}(a) - \log \pi_t(a) = \eta\widehat{Q}(a) - Z, \tag{27}$$

where Z is the (state-dependent) log-normalizer.

Step 2: Three-point identity. For any $p \in \Delta(\mathcal{A})$, direct computation gives the well-known three-point identity for KL:

$$\mathrm{KL}(p\|\pi_t) - \mathrm{KL}(p\|\pi_{t+1}) - \mathrm{KL}(\pi_{t+1}\|\pi_t) = \big\langle p - \pi_{t+1},\, \log(\pi_{t+1}/\pi_t)\big\rangle.$$

Substituting (27), the right-hand side equals $\eta\langle p - \pi_{t+1}, \widehat{Q}\rangle$ (the Z term vanishes since p and π_{t+1} are both probabilities). Therefore

$$\eta\big\langle \widehat{Q},\, p - \pi_{t+1}\big\rangle = \mathrm{KL}(p\|\pi_t) - \mathrm{KL}(p\|\pi_{t+1}) - \mathrm{KL}(\pi_{t+1}\|\pi_t). \tag{28}$$

Step 3: Bound the drift $\langle \widehat{Q}, \pi_{t+1} - \pi_t\rangle$. By Hölder and Pinsker,

$$\eta\big\langle \widehat{Q}, \pi_{t+1} - \pi_t\big\rangle \le \eta\|\widehat{Q}\|_\infty \|\pi_{t+1} - \pi_t\|_1 \le \eta H\sqrt{2\,\mathrm{KL}(\pi_{t+1}\|\pi_t)},$$

where $\|\widehat{Q}\|_\infty \le H$ uses Assumption C.1. By $ab \le \frac{1}{2}a^2 + \frac{1}{2}b^2$ with $a = \eta H$, $b = \sqrt{2\,\mathrm{KL}(\pi_{t+1}\|\pi_t)}$,

$$\eta\big\langle \widehat{Q}, \pi_{t+1} - \pi_t\big\rangle \le \frac{\eta^2 H^2}{2} + \mathrm{KL}(\pi_{t+1}\|\pi_t). \tag{29}$$

Step 4: Combine. Adding (28) and (29),

$$\eta\big\langle \widehat{Q},\, p - \pi_t\big\rangle \le \mathrm{KL}(p\|\pi_t) - \mathrm{KL}(p\|\pi_{t+1}) + \frac{\eta^2 H^2}{2},$$

where the two $\mathrm{KL}(\pi_{t+1}\|\pi_t)$ terms cancel exactly. This is the claim. □

Lemma C.8 (Weighted critic error)**.** *Fix $t \in [T]$ and $h \in [H]$. Under Assumptions C.1, C.2, C.4, and conditional on π_t, with probability at least $1 - \delta'$ over the sampling of $\{(s_{h,i}, a_{h,i}, G_{h,i})\}_{i=1}^{N_h}$,*

$$\mathbb{E}_{(s,a)\sim\nu_h^{\pi_t}}\big[(\widehat{Q}_h^t(s,a) - Q_h^{\pi_t}(s,a))^2\big] \le C \cdot \frac{H^2}{\varepsilon} \cdot \frac{\log(|\mathcal{F}_h|/\delta')}{N_h},$$

for a universal constant $C > 0$. Consequently, by Cauchy–Schwarz,

$$\mathbb{E}_{(s,a)\sim\nu_h^{\pi_t}}\big[|\widehat{Q}_h^t(s,a) - Q_h^{\pi_t}(s,a)|\big] \le \sqrt{C \cdot \frac{H^2}{\varepsilon} \cdot \frac{\log(|\mathcal{F}_h|/\delta')}{N_h}}.$$

Proof. Step 1: Reweighting to on-policy loss. For any $f \in \mathcal{F}_h$, the weighted population squared loss equals the on-policy loss to a constant offset:

$$\mathbb{E}_{s\sim\mu_h^t, a\sim\pi_{t,h}, G}\big[w_h^t(s)(f(s,a) - G)^2\big] = \mathbb{E}_{(s,a)\sim\nu_h^{\pi_t}, G}\big[(f(s,a) - G)^2\big],$$

because $w_h^t(s) = d_h^{\pi_t}(s)/\mu_h^t(s)$ exactly cancels the change of state measure, while $a \mid s \sim \pi_{t,h}$ and the conditional distribution of G given (s,a) are the same under both samplings.

Step 2: Realizability and noise structure. By construction, $G_{h,i}$ is a return-to-go obtained by rolling out π_t from $(s_{h,i}, a_{h,i})$, so

$$\mathbb{E}[G_{h,i} \mid s_{h,i}, a_{h,i}] = Q_h^{\pi_t}(s_{h,i}, a_{h,i}), \qquad G_{h,i} \in [0, H].$$

By Assumption C.2, $Q_h^{\pi_t} \in \mathcal{F}_h$, so $Q_h^{\pi_t}$ minimizes the on-policy population squared loss over $\mathcal{F}_h$.

Step 3: Weighted ERM fast rate. The per-sample weighted squared loss is bounded: $w_h^t(s)(f(s,a) - G)^2 \in [0, H^2/\varepsilon]$ by Assumption C.4 and Assumption C.1. Standard realizable-case ERM analysis (e.g. Bartlett–Mendelson local Rademacher, or the offset-Rademacher bound of Liang–Rakhlin–Sridharan 2015) for finite classes then gives

$$\mathbb{E}_{(s,a)\sim\nu_h^{\pi_t}}\big[(\widehat{Q}_h^t(s,a) - Q_h^{\pi_t}(s,a))^2\big] \le C \cdot \frac{B\log(|\mathcal{F}_h|/\delta')}{N_h},$$

with $B = H^2/\varepsilon$ the per-sample loss range. The Cauchy–Schwarz step $\mathbb{E}|X| \le \sqrt{\mathbb{E}X^2}$ then converts L^2 to L^1. □

C.7 Proof of Theorem C.5

Step 1: Per-step bound, telescoping. Apply Lemma C.7 with $p = \mu_h(\cdot \mid s)$, take expectation over $s \sim d_h^\mu$, and sum over $h \in [H]$ and $t \in [T]$. The KL terms telescope in t:

$$\eta \sum_{t=1}^{T} \sum_{h=1}^{H} \mathbb{E}_{s\sim d_h^\mu}\big[\langle \widehat{Q}_h^t(s,\cdot), \mu_h - \pi_{t,h}\rangle\big] \le D_\mu + \frac{T\eta^2 H^3}{2},$$

since each per-step term contributes $\eta^2 H^2/2$ and there are TH of them. Dividing by ηT,

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{h=1}^{H} \mathbb{E}_{s\sim d_h^\mu}\big[\langle \widehat{Q}_h^t(s,\cdot), \mu_h - \pi_{t,h}\rangle\big] \le \frac{D_\mu}{\eta T} + \frac{\eta H^3}{2}. \tag{30}$$

Step 2: Performance difference and critic error. By Lemma C.6 and adding/subtracting $\widehat{Q}_h^t$,

$$J(\mu) - J(\pi_t) = \sum_{h=1}^{H} \mathbb{E}_{s\sim d_h^\mu}\big[\langle \widehat{Q}_h^t(s,\cdot), \mu_h - \pi_{t,h}\rangle\big] + \sum_{h=1}^{H} \mathcal{E}_{h,t},$$

where, with $e_{h,t}(s,a) := \widehat{Q}_h^t(s,a) - Q_h^{\pi_t}(s,a)$,

$$\mathcal{E}_{h,t} := \mathbb{E}_{s\sim d_h^\mu}\big[\langle Q_h^{\pi_t}(s,\cdot) - \widehat{Q}_h^t(s,\cdot), \mu_h - \pi_{t,h}\rangle\big].$$

Step 3: Coverage transfers $\mathcal{E}_{h,t}$ to on-policy L^1. By the triangle inequality,

$$|\mathcal{E}_{h,t}| \le \big|\mathbb{E}_{s\sim d_h^\mu, a\sim\mu_h}[e_{h,t}(s,a)]\big| + \big|\mathbb{E}_{s\sim d_h^\mu, a\sim\pi_{t,h}}[e_{h,t}(s,a)]\big|.$$

Assumption C.3 gives $\nu_h^\mu \le \kappa \nu_h^{\pi_t}$, hence

$$\big|\mathbb{E}_{s\sim d_h^\mu, a\sim\mu_h}[e_{h,t}]\big| \le \kappa\, \mathbb{E}_{(s,a)\sim\nu_h^{\pi_t}}[|e_{h,t}|].$$

Summing $\nu_h^\mu(s,a) \le \kappa \nu_h^{\pi_t}(s,a)$ over a also gives $d_h^\mu \le \kappa d_h^{\pi_t}$ pointwise, so

$$\big|\mathbb{E}_{s\sim d_h^\mu, a\sim\pi_{t,h}}[e_{h,t}]\big| \le \kappa\, \mathbb{E}_{(s,a)\sim\nu_h^{\pi_t}}[|e_{h,t}|].$$

Therefore

$$|\mathcal{E}_{h,t}| \le 2\kappa\, \mathbb{E}_{(s,a)\sim\nu_h^{\pi_t}}[|e_{h,t}(s,a)|]. \tag{31}$$

Step 4: Apply Lemma C.8 with a union bound. Apply Lemma C.8 with confidence $\delta/(HT)$ for each (t,h). A union bound over the HT such events implies that, with probability at least $1-\delta$, simultaneously for all t, h,

$$\mathbb{E}_{(s,a)\sim\nu_h^{\pi_t}}[|e_{h,t}(s,a)|] \le \beta_h.$$

Combined with (31), $|\mathcal{E}_{h,t}| \le 2\kappa\beta_h$ on this event.

Step 5: Assemble. On the high-probability event of Step 4,

$$J(\mu) - J(\pi_t) \le \sum_{h=1}^{H} \mathbb{E}_{s\sim d_h^\mu}\big[\langle \widehat{Q}_h^t, \mu_h - \pi_{t,h}\rangle\big] + 2\kappa \sum_{h=1}^{H} \beta_h.$$

Averaging over $t \in [T]$ and using $J(\bar\pi_T) = \frac{1}{T}\sum_{t=1}^{T} J(\pi_t)$ (since $\bar\pi_T$ is the uniform mixture), then plugging in (30),

$$J(\mu) - J(\bar\pi_T) \le \frac{D_\mu}{\eta T} + \frac{\eta H^3}{2} + 2\kappa \sum_{h=1}^{H} \beta_h.$$

The choice $\eta = \sqrt{2D_\mu/(H^3 T)}$ minimizes the first two terms, giving the second form of the bound. □

D Experiments

This appendix provides additional details for the experiments in Section 4. The purpose of these experiments is to test whether variance-driven state resampling is most useful in the regime predicted by our analysis: tasks in which rollouts from the initial state often produce uniform rewards, but intermediate states along successful or partially successful trajectories can still produce mixed continuation outcomes. Across all experiments, VSR changes only the state from which rollout groups are initialized; the underlying policy-gradient objective, reward function, and evaluation protocol are otherwise kept the same as the corresponding baseline.

D.1 Toy gridworld

We use the gridworld experiment in Section 2.3 as a controlled setting where both the on-policy state-visitation distribution d^{π_t} and the within-state reward standard deviation σ_{π_t} can be computed exactly. The environment is a 7×7 gridworld with fixed obstacles, a start state S, and a goal state G. The agent receives a binary terminal reward $R \in \{0, 1\}$ indicating whether it reaches the goal. The policy is a one-hidden-layer MLP with 32 hidden units and tanh activations, followed by a four-action output layer.

At each training iteration, we enumerate the finite state space to compute the current occupancy distribution $d^{\pi_t}(s)$ and the continuation success probability $p_t(s)$ from every reachable state. We then compute

$$\sigma_{\pi_t}(s) = \sqrt{p_t(s)(1 - p_t(s))}$$

and form the variance-driven proposal

$$\mu_t(s) \propto d^{\pi_t}(s)\sigma_{\pi_t}(s).$$

We compare policy-gradient updates that sample states from the standard on-policy proposal d^{π_t} against updates that sample from μ_t. This experiment is intended as a diagnostic visualization of the variance-optimal proposal rather than as a benchmark of practical replay heuristics: unlike the LLM and agent experiments, no approximation to either d^{π_t} or σ_{π_t} is used.

D.2 Mathematical reasoning

Training data. For mathematical reasoning, we train on a hard subset of OpenR1-Math-220k. The subset contains 1K prompts selected to have low pass rate under the base policy. We filter for prompts with pass@32 at most 1 using Qwen2.5-7B-Instruct. This filtering focuses training on problems for which prompt-level on-policy rollouts are usually uninformative, creating a setting where intermediate successful prefixes can provide useful continuation states.

Models and baselines. We run experiments with Qwen2.5-7B-Instruct and Llama-3.2-3B-Instruct. For each model, we compare the base model, supervised fine-tuning on expert trajectories, standard GRPO, and VSR+GRPO. The SFT baseline uses the same successful expert trajectories that are used to initialize the VSR replay buffer, so that the comparison separates direct imitation of expert completions from using expert prefixes only as reset states for RL rollouts. All experiments are performed on 4 A100 GPUs.

VSR initialization. In the math experiments, VSR uses cold-start initialization because successful trajectories are rare under the base policy. We seed the replay buffer with prefix states extracted from successful DeepSeek-R1 trajectories. A seeded prefix consists of the original prompt concatenated with a partial reasoning trace. During VSR training, the prefix tokens are treated as fixed context and are masked out of the policy-gradient loss; only newly generated continuation tokens contribute to the GRPO objective. As training proceeds, the replay buffer is updated with successful prefixes generated by the current policy, and recency-based eviction gradually replaces expert-seeded prefixes with on-policy states. For every prompt, we only keep the most recent single successful trajectory.

Replay construction and scoring. For every successful fresh trajectory τ generated during training, we insert prefix states $s_i = (x, y_{<i})$ into the replay buffer. Each inserted prefix receives the zero-cost variance score

$$\hat{\sigma}(s_i) = \sqrt{\hat{p}_i(1 - \hat{p}_i)}, \qquad \hat{p}_i = (1 - \alpha_i)p_0(x) + \alpha_i R(\tau), \qquad \alpha_i = i/|\tau|.$$

Here $p_0(x)$ is the empirical success rate of the rollout group generated from the original prompt, and $R(\tau)$ is the terminal reward of the trajectory from which the prefix was extracted. At each update, a fraction ρ of batch elements is drawn from the replay buffer with probability proportional to $\hat{\sigma}$, while the remaining batch elements are fresh prompts from the training set. For replayed prefixes, we generate a fresh GRPO rollout group from the prefix using the current policy.

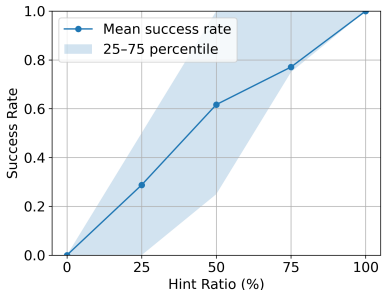

Figure 7: **Empirical continuation success rate as a function of hint ratio.** For each of the 1K hard prompts in the training subset, we truncate the corresponding successful DeepSeek-R1 trajectory to a fraction $\alpha \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$ of its length and sample 8 continuations under Qwen2.5-7B-Instruct. The blue curve shows the mean per-prompt success rate across the 1K prompts; the shaded band shows the 25–75 percentile across prompts. The mean is approximately linear in α, supporting the linear interpolation used in $\hat{p}_t(s_i \mid \tau) = (1 - \alpha_i)\, p_0(x) + \alpha_i\, R(\tau)$.

Empirical justification of the linear interpolation. The score $\hat{\sigma}_t$ in Algorithm 1 estimates the continuation success probability at an intermediate prefix s_i by linearly interpolating between the prompt-level rollout success rate $p_0(x)$ and the terminal reward $R(\tau)$ of the trajectory the prefix is drawn from. We adopt the linear form because it is the simplest functional form consistent with the boundary conditions $\hat{p}_0 = p_0(x)$ and $\hat{p}_{|\tau|} = R(\tau)$, and we did not tune it. To check that this choice is not grossly mis-specified in our training setting, we measure the empirical continuation success rate as a function of the hint ratio $\alpha = i/|\tau|$. Concretely, for each of the 1K hard prompts in our training subset—on which the base policy's pass rate is essentially zero—we take the corresponding successful DeepSeek-R1 trajectory used to seed the replay buffer, truncate it to $\alpha \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$ of its length, and sample 8 continuations under Qwen2.5-7B-Instruct from each truncation. Figure 7 reports the mean and 25–75 percentile of the resulting per-prompt success rate across the 1K prompts. The mean success rate rises approximately linearly in α from ≈ 0 at $\alpha = 0$ to ≈ 1 at $\alpha = 100\%$, with no strong concavity or saturation that would favor a sigmoid or quadratic interpolant. The interquartile spread is widest near the middle of the trajectory, which is exactly the regime where $\sigma_\pi(s)$ is largest under the proxy and where VSR is designed to concentrate sampling mass. We therefore retain the linear form throughout; tighter low-cost surrogates remain a natural target for future work.

Evaluation. We evaluate final models on AIME24, AIME25, AMC23, OlympiadBench, MATH500, and Minerva. For each benchmark, we report accuracy under the same decoding and evaluation protocol across all methods. We sample 16 rollouts over 2 different seeds for all the prompts under `temperature=0.7` and report the mean results. In addition to final benchmark accuracy, we track training reward, validation accuracy, and response length over training. Response length is included because VSR can reduce the number of newly generated tokens per update by starting a fraction of rollouts from mid-trajectory prefixes.

D.3 Long-horizon agentic tasks

Environments. We evaluate VSR on both Sokoban and BabyAI. These environments require multi-step decision making and produce sparse terminal rewards, making them a natural testbed for state-level resampling. For BabyAI, we use the SynthLoc, OneRoom, and MiniBoss levels. The policy observes a serialized interaction history consisting of the task instruction, previous actions, environment feedback, and current observations. A replay state is therefore represented as a prefix of this serialized trajectory, analogous to a token prefix in the math experiments.

Table 4: Hyperparameters for mathematical reasoning experiments.

Hyperparameter	Qwen2.5-7B-Instruct	Llama-3.2-3B-Instruct
Replay ratio ρ	0.5	0.5
Training steps	360	360
Max Prompt Length	2048	2048
Max Response Length	4096	4096
Rollout Group Size	8	8
Training batch size	128	128
Mini batch size	64	64
Learning rate	1×10^{-6}	1×10^{-6}
KL coefficient	0	0

Table 5: Hyperparameters for agentic-task experiments.

Hyperparameter	Qwen2.5-1.5B-Instruct	Qwen2.5-3B-Instruct
Replay ratio	0.3	0.3
Recency window	5	5
Max prompt length	2048	2048
Max response length	256	256
Rollout group size	8	8
Training batch size	32	32
Learning rate	1×10^{-6}	1×10^{-6}
KL coefficient	0	0

Models and baselines. We use Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct. The base RL algorithm is GiGPO, and VSR is added as a plug-in state-resampling component. We compare the base model, GiGPO, and VSR+GiGPO under matched model, environment, reward, and training settings. Unlike the math experiments, we do not use expert successful trajectories for cold-start initialization. The replay buffer is populated entirely from trajectories generated by the current policy. All experiments are performed on a single H200 GPU.

VSR in agentic environments. At each training iteration, the policy generates rollout groups either from the initial environment state or from replayed intermediate states. When a successful trajectory is observed, we insert its intermediate serialized states into the replay buffer. Each replayed state includes all information needed to resume interaction from that point, including prior actions and environment observations. As in the math setting, prefix tokens or history tokens preceding the replay point are treated as fixed context and are masked from the policy-gradient loss. The model is optimized only on newly generated actions after the replay state.

Prefix scores are computed with the same zero-cost interpolation rule used in the math experiments. For a prefix at normalized position α_i along a successful trajectory, the estimated continuation success probability interpolates between the rollout-group success rate from the initial state and the realized terminal reward of the trajectory. Replay prefixes are sampled in proportion to the resulting $\hat{\sigma}$ score. The recency window is used to keep the replay buffer aligned with the current policy, preventing the proposal from becoming dominated by states collected under earlier, weaker policies.

Evaluation. For Sokoban and BabyAI, we report task success rate. For Sokoban, we sample 128 times over a fixed set of map seeds and report the average success rate. For each level of Baby AI, we sample 128 times over a fixed set of map seeds and report the average success rate. The environment hyperparameters remain the same as what we use in training. For training dynamics, we additionally track training reward, validation success rate, and episode length. Episode length is reported because state replay changes the average number of newly generated steps per rollout: when training starts from intermediate states, the model can receive useful gradient signal while generating fewer new actions.

D.4 Ablation details

We conduct ablations on BabyAI using the same training and evaluation setup as the main agentic-task experiments. All ablation variants are built on top of GiGPO and differ only in the replay component.

Uniform replay. The variant VSR w/o scoring keeps the same replay buffer construction and recency-based updates as VSR, but samples prefixes uniformly from the buffer rather than in proportion to $\hat{\sigma}$. This ablation tests whether the variance-driven score contributes beyond the benefit of simply restarting rollouts from previously visited successful prefixes.

Static replay buffer. The variant VSR w/o recency keeps $\hat{\sigma}$-weighted prefix sampling but disables recency-based buffer updates. The buffer is populated early in training and then held fixed. This ablation tests whether tracking the current policy's state-visitation distribution is necessary for sustained learning. A static buffer can be useful early in training, but as the policy improves, its state distribution changes; without recency updates, replayed prefixes may no longer approximate the current d^{π_t} factor in the variance-driven proposal.

Full VSR. Full VSR combines both components: recent successful-prefix replay approximates the current state-visitation factor, while $\hat{\sigma}$-weighted sampling prioritizes prefixes whose continuations are estimated to have high residual uncertainty. The ablations therefore isolate the two practical approximations to the proposal $\mu^{\pi}_{\mathrm{var}} \propto d^{\pi}\sigma_{\pi}$.

D.5 Controls and validations

Table 6: Additional mathematical-reasoning controls on Llama-3.2-3B-Instruct. These are single training runs; the original benchmark evaluation protocol is otherwise unchanged.

Method	AIME24	AIME25	AMC23	Olympiad	MATH500	Minerva	Avg.
GRPO	4.4	1.0	18.7	12.6	39.2	16.5	15.4
Reinforce-Ada-Est	4.2	0.1	18.0	12.4	39.9	15.4	15.0
Static uniform expert-prefix resets	5.6	0.5	22.7	14.1	43.7	15.8	17.1
VSR	6.2	0.6	20.4	15.3	54.0	17.8	19.1

Static expert-prefix reset control. For Llama-3.2-3B-Instruct, we added a control using the same successful expert-trajectory source as VSR but sampling uniformly from a fixed expert-prefix pool and never adding prefixes generated by the current policy. The benchmark results (AIME24/AIME25/AMC23/OlympiadBench/MATH500/Minerva/Avg.) are 4.4/1.0/18.7/12.6/39.2/16.5/15.4 for GRPO, 5.6/0.5/22.7/14.1/43.7/15.8/17.1 for static uniform expert-prefix resets, and 6.2/0.6/20.4/15.3/54.0/17.8/19.1 for VSR. Thus privileged reset states themselves explain a material part of the gain. Because full VSR also adds online policy-generated replay, recency updates, and priority-weighted sampling, the remaining difference cannot be attributed specifically to the priority score. This is one training run per method.

Prompt-level adaptive allocation. We also compare with Reinforce-Ada-Est on Llama-3.2-3B-Instruct. Each update uses 128 prompts and exactly $128 \times 8 = 1024$ generated responses. Reinforce-Ada-Est reallocates these responses across prompts subject to $N_x \in [4, 32]$ and $\sum_x N_x = 1024$, so its average group size is 8, matching GRPO and VSR. It uses no expert trajectories, expert prefixes, or intermediate reset states; every rollout begins from the original problem prompt. The training subset, number of updates, optimization hyperparameters, and evaluation protocol are otherwise matched. Under this matched response budget, its benchmark results are 4.2/0.1/18.0/12.4/39.9/15.4 (Avg. 15.0), compared with GRPO at 15.4 and VSR at 19.1. This supports the narrower conclusion that prompt-level response reallocation alone does not reproduce the full VSR result in this hard-prompt setting; it does not isolate state-level allocation as the sole source of the difference, and it is a single run.

Independent-run scoring ablation. On Qwen2.5-1.5B BabyAI, the three uniform-replay runs are 46.1, 39.1, and 46.9 (mean 44.0 ± 4.3 SD), while the three VSR runs are 47.6, 50.8, and 46.1 (mean 48.2 ± 2.4 SD). We report all runs and do not perform formal significance testing.

Table 7: Independent-run BabyAI scoring ablation on Qwen2.5-1.5B. Values are average success rates (%).

Method	Run 1	Run 2	Run 3	Mean $\pm$ SD
Uniform replay	46.1	39.1	46.9	44.0 ± 4.3
VSR	47.6	50.8	46.1	48.2 ± 2.4

Failed-prefix analysis and same-checkpoint continuations. For failed trajectories, the mean continuation success rate decreases as the prefix becomes longer (0.396, 0.260, and 0.130 at the three tested prefix positions), while the fraction of continuation groups with no successful rollout increases from 26% to 42% to 62%. For successful trajectories from the same problems, the corresponding all-fail rates are 0%, 2%, and 4%. At the latest tested failed-prefix position, 36% of groups still contain mixed outcomes, so this evidence does not imply that every failed prefix is unrecoverable; it supports successful-only filtering as a budget-allocation choice in the sparse-reward regime.

Statewise validation of the priority score. We directly validate the implemented score on MATH-500. For each policy, we select 50 problems with mixed initial outcomes, retain one self-generated successful trajectory per problem, extract prefixes at 25%, 50%, and 75% of the response, and generate 16 new continuations from each prefix, yielding 150 prefix states per policy. For Qwen2.5-7B-Instruct, predicted versus empirical continuation standard deviation has pooled Pearson correlation 0.441 [95% CI: 0.277, 0.591] and Spearman correlation 0.433 [0.270, 0.585]; Pearson correlations at the three fixed prefix positions are 0.446/0.387/0.342 and Spearman correlations are 0.494/0.305/0.329. For an intermediate training checkpoint, pooled Pearson is 0.484 [0.291, 0.641] and pooled Spearman is 0.499 [0.314, 0.642], with within-position Pearson 0.587/0.432/0.307 and Spearman 0.620/0.424/0.315. These moderate positive associations, including within fixed prefix positions, show that the score contains state-dependent information beyond prefix position alone. The validation uses selected successful MATH-500 states rather than the full online replay-buffer distribution, and the score remains an uncalibrated heuristic rather than an unbiased or consistent variance estimator.

Generated-token accounting. Across all model calls in the agentic experiments, Qwen2.5-1.5B VSR+GiGPO generates 39.27M tokens over 150 matched updates compared with 81.30M for GiGPO; Qwen2.5-3B generates 34.43M versus 36.26M over 100 matched updates. These totals establish lower generated-token usage at matched update counts. We do not claim lower wall-clock cost or performance superiority under an exactly matched total-token budget.

E Broader Impacts

Our work on variance-driven state resampling for RL fine-tuning of LLMs offers several potential benefits to society: the improved sample efficiency in policy optimization could enhance applications in mathematical reasoning and long-horizon agentic decision making, reduced rollout cost contributes to environmental sustainability through decreased computational resource consumption, and more efficient use of training compute may improve the accessibility of RL post-training for researchers with limited resources. However, as with any advancement in AI capabilities, potential risks should be considered: while our method is evaluated on public reasoning benchmarks and standard agentic environments for research purposes, the enhanced training efficiency could potentially be misused if not properly secured, and we recommend implementing appropriate access controls and maintaining transparency about the system's limitations and intended use cases. We encourage future work to continue exploring responsible deployment strategies that maximize beneficial impacts while minimizing potential risks.

NeurIPS Paper Checklist

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 main claims reflect the paper's contributions and scope.

   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: See Section 5 for discussion on limitations.

   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: [Yes]

Justification: See Appendix B and Appendix C.

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: We provide the details of our experiments in Appendix D.

   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: We will submit our code in Supplemental Material and will open-source code and models on GitHub.

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: See Section 4 and Appendix D.

   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: See Table 7.

   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: See Appendix D.

   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: Yes, the research in the manuscript is compliant with the NeurIPS Code of Ethics.

   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**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix E

    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: Our research poses no risk of misuse.

    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: Yes, the creators/original owners of assets used in the paper are properly credited.

    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: [N/A]

    Justification: This paper does not release new assets.

    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: This paper does not involve crowdsourcing nor 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: This paper does not involve crowdsourcing nor research with human subjects.

    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: [N/A]

Justification: The core method development in this research does not involve LLMs.

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.