UNDERSTANDING REASONING COLLAPSE IN MULTI-TURN AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In closed-loop multi-turn agent reinforcement learning, LLM agents exhibit reasoning collapse, where reasoning shift toward generic templates, weakly coupled to the inputs. We firstly identify that such collapse is easy to miss with entropy or surface diversity metrics since reasoning text still varies but becomes input-agnostic. We then propose an information-theoretic decomposition of reasoning variable Z's variation into conditional entropy $H(Z \mid X)$ (randomness under same input) and mutual information (MI) $I(X; Z)$ (input dependence). Template collapse occurs when $H(Z \mid X)$ stays high while $I(X; Z)$ drops, yielding diverse-looking but generic reasoning. To make $I(X; Z)$ a reproducible and sanity-checkable diagnostic, we further introduce an MI-style retrieval protocol treating each reasoning trace Z as a query to retrieve its source X from a minibatch; accuracy degrades toward chance under collapse. We thus provide a signal-to-noise ratio explanation for why $I(X; Z)$ drops: when within-input reward variance $\mathrm{Var}(R \mid X)$ is low, task gradients weaken and input-agnostic regularizers (KL, entropy) dominate, flattening cross-input differences. Finally, we propose reward-variance-aware filtering to prioritize high-signal updates. Across multi-turn environments, model scales, and modalities (including VLMs), this improves input dependence, stability, and performance while remaining competitive with state-of-the-art stabilization baselines.

1 INTRODUCTION

Multi-turn agent reinforcement learning is becoming a key route for training LLM agents. However, when training on self-sampled trajectories, we observe a common but overlooked failure mode: even when reward curves remain stable, the reasoning text gradually drifts toward task-agnostic templates (Figure 1). We find agentic settings are more prone to such templating than single-turn tasks: rewards are often sparse and weakly discriminative, so many incorrect trajectories can still receive similar returns, which reduces pressure to maintain input-specific reasoning.

Over the past year, a broad consensus has emerged around stabilizing LLM RL via KL constraints, entropy regularization, clipping, and PPO/GRPO variants (Schulman et al., 2017b; Ouyang et al., 2022; Rafailov et al., 2024; Xu et al., 2025). These techniques can control divergence and smooth training, but they do not guarantee that the policy preserves task-relevant, input-conditioned reasoning. As illustrated in Figure 2, when reward discrimination is weak (e.g., reward variance is small), stabilization terms exert relatively larger influence and act like global contractions that reduce across-input distinctions, accelerating templated reasoning. We observe that template collapse can still occur even with controlled KL and entropy.

Prior work quantifies reasoning collapse via conditional entropy or cross-entropy, asking whether outputs become more deterministic for the same prompt (Wei et al., 2025; Yao et al., 2025; Yun et al., 2025). In LLM agent RL, we argue that across-input dependence is equally critical: does reasoning remain diverse and driven by the input goal. We therefore decompose reasoning variation into conditional entropy $H(Z \mid X)$ (within-input variability) and mutual information $I(X; Z)$ (input dependence), and define template collapse as the regime where $H(Z \mid X)$ remains high while $I(X; Z)$ drops (Figure 1). Empirically, performance trajectories are more concentrated in Perf–MI space than in Perf–Entropy space (Figure 4), suggesting $I(X; Z)$ tracks task-relevant reasoning more reliably than entropy.

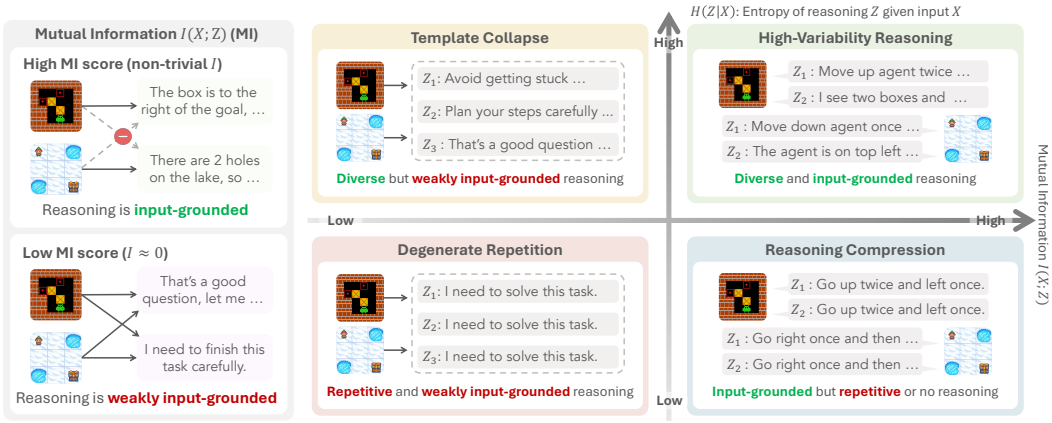

Figure 1: We characterize reasoning diversity in LLM-agent RL along two axes: conditional entropy $H(Z \mid X)$ (within-input variability) and mutual information $I(X;Z)$ (input dependence). The four quadrants correspond to: (i) low $H(Z \mid X)$, low $I(X;Z)$ (deterministic, input-agnostic template); (ii) low $H(Z \mid X)$, high $I(X;Z)$ (input-grounded but overly deterministic); (iii) high $H(Z \mid X)$, high $I(X;Z)$ (diverse and input-grounded; desired); (iv) high $H(Z \mid X)$, low $I(X;Z)$ (superficially diverse but input-agnostic; template collapse).

To make $I(X;Z)$ measurable and debuggable, we propose an MI-style retrieval diagnostic: within a minibatch, each reasoning trace Z serves as a query to retrieve its matching input X. When reasoning is input-driven, retrieval exceeds chance; under template collapse, accuracy degrades toward chance. We include chance-level baselines and sanity checks against confounds to turn "is input-conditioned reasoning collapsing" into a reproducible protocol.

We then ask why across-input distinguishability weakens during training. We provide an SNR mechanism view: effective policy updates rely on advantage differences among trajectories under the same input, and low within-prompt reward variance makes task gradients weak and noisy. In this regime, KL/entropy/clipping become relatively more prominent and can reduce across-input distinctions, lowering $I(X;Z)$ and encouraging templated reasoning. We support this with gradient decomposition across $\mathrm{Var}(R \mid X)$ buckets (Figure 3) and introduce reward-variance-aware filtering as a minimal control knob. The keep rate strengthens effective task signal by prioritizing high-RV samples and suppressing noise-dominated updates. Figure 5 further shows that RV also reflects how noisy the environment-facing learning loop is, which helps interpret when filtering is likely to be effective.

We evaluate our diagnostic and intervention across environments, model scales, and modalities, and compare against strong stabilization baselines (Table 1; Figures 5 and 6 and Table 3). Overall, we provide a closed-loop evidence chain from diagnostics to mechanism and a minimal control knob for understanding template collapse in agent RL.

Our contributions include: (1) a two-axis view that separates within-input variability from across-input dependence and isolates template collapse; (2) an MI-style retrieval diagnostic for monitoring input-conditioned reasoning during closed-loop RL; (3) an SNR-based mechanism supported by gradient decomposition; and (4) RV-aware filtering as a minimal control knob that improves input dependence and task performance.

2 Template Collapse in Multi-turn Agent RL

2.1 Setup and Preliminaries

This section introduces the closed-loop multi-turn agent RL setting and previews our core story: when optimizing a regularized policy objective on self-sampled trajectories, the agent can drift toward fluent but input-agnostic reasoning templates. At each turn t, the agent observes o_t and outputs reasoning tokens z_t plus an executable action a_t, producing trajectories $\tau = \{(o_t, z_t, a_t, r_t)\}_{t=1}^{T}$. We define X as the full prompt context before the current turn and Z as the current-turn reasoning tokens after removing action spans (Appendix D). We then (1) define template collapse and show why entropy can miss it; (2) introduce a length-normalized MI-style diagnostic for input dependence; (3) connect

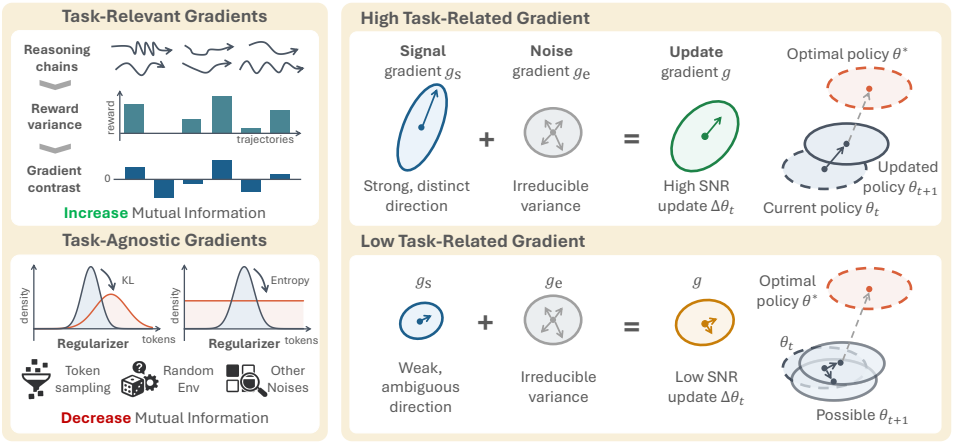

Figure 2: Schematic SNR view of closed-loop policy optimization. We write the update gradient as $g = g_s + g_e$, where g_s is task-related and g_e is residual noise. With high per-input reward variance $\mathrm{Var}(R \mid X)$, g_s has a clearer direction, yielding higher-SNR updates that preserve input dependence and increase $I(X;Z)$. With low $\mathrm{Var}(R \mid X)$, g_s weakens and updates are more shaped by task-agnostic factors (KL/entropy regularization, sampling noise, environment stochasticity), lowering $I(X;Z)$ and encouraging input-independent reasoning.

an SNR-based mechanism via gradient decomposition across reward-variance buckets (Appendix G, Appendix J); and (4) present RV-aware filtering as a minimal control knob with evidence across tasks, models, and modalities (Figures 3 to 6 and Tables 1 and 3).

2.2 Rethinking Reasoning Collapse from an Information-Theoretic Lens

Our key observation is that in closed-loop multi-turn agent RL, reasoning can lose *input dependence* without becoming less variable in a within-input sense: for the same input X, the model can still generate different reasoning Z (high conditional entropy), yet across different inputs the reasoning becomes hard to distinguish and increasingly template-like (low mutual information).

Conditional Entropy and Mutual Information Let X denote the per-turn prompt context and Z the corresponding per-turn reasoning tokens (Section 2.1). We use conditional entropy $H(Z \mid X)$ to measure within-input variability (diversity under the same X) (Tevet & Berant, 2021; Semeniuta et al., 2019). To capture across-input dependence, we use mutual information

$$I(X;Z) = H(Z) - H(Z \mid X),$$

which quantifies how informative Z is about X (Cover & Thomas, 2006). By the identity $H(Z) = I(X;Z) + H(Z \mid X)$, marginal diversity decomposes into within-input variability and cross-input dependence (Appendix D.2). Thus, diverse *and* input-driven reasoning requires both $H(Z \mid X)$ and $I(X;Z)$ to be high.

In Figure 1, we map reasoning behaviors onto the $H(Z \mid X)$–$I(X;Z)$ plane with four regimes: (i) *low H, low I* (Degenerate Repetition); (ii) *low H, high I* (Reasoning Compression); (iii) *high H, high I* (High-Variability Reasoning; desired); (iv) *high H, low I* (Template Collapse). ***Template Collapse*** is the MI-specific failure mode: outputs remain variable under a fixed input, yet become less distinguishable across inputs. Thus entropy can stay high while input dependence vanishes, motivating $I(X;Z)$ as a core diagnostic alongside entropy.

2.3 Measurement: an MI-style Retrieval Diagnostic

We use a mutual-information–style diagnostic computed *within a minibatch* to measure whether the agent's reasoning Z is input-dependent. The metric uses only teacher-forced log-likelihoods under the same LLM policy/scorer $p_\theta(\cdot \mid \cdot)$ and does not require any external model.

Setup Given a batch of N inputs $\{X_i\}_{i=1}^N$, we sample K reasoning segments per input (same sampling convention as Appendix E.1):

$$Z_{i,k} \sim p_\theta(\cdot \mid X_i), \quad k \in \{1, \ldots, K\},$$

where each $Z_{i,k}$ contains only the reasoning span (action tokens removed; Section 2.1).

Cross-scoring For each sampled reasoning $Z_{i,k}$, compute its teacher-forced log-likelihood under every candidate input (matched-pair scores in Appendix E.1). To avoid trivial dependence on reasoning length, we use per-token scores:

$$\tilde{\ell}_j(Z_{i,k}) = \frac{1}{|Z_{i,k}|} \log p_\theta(Z_{i,k} \mid X_j).$$

where $|Z_{i,k}|$ is the number of tokens in the reasoning span.

A Simple Diagnostic: Retrieval Accuracy Our primary diagnostic is in-batch retrieval accuracy, as it is directly interpretable and comparable across settings, with an explicit chance baseline. We define

$$\mathrm{Acc} = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}\Big[i = \arg\max_j \ell_j(Z_{i,k})\Big].$$

Under collapse, Acc approaches chance level $1/N$ (and recall@k approaches k/N).

Continuous Companion: MI-style Estimate We also report a continuous MI-style estimate that uses an in-batch approximation to the marginal $p_\theta(Z)$. We approximate the marginal by a uniform mixture over prompts (mixture score construction in Appendix E.1):

$$p_\theta(Z) \approx \frac{1}{N} \sum_{j=1}^{N} p_\theta(Z \mid X_j).$$

Define the length-normalized log marginal score

$$\tilde{\ell}_{\mathrm{mix}}(Z_{i,k}) = \frac{1}{|Z_{i,k}|} \log \Big(\frac{1}{N} \sum_{j=1}^{N} \exp\big(\ell_j(Z_{i,k})\big) \Big).$$

Note that longer Z can affect likelihood magnitudes and ranking. Therefore, we define the length-normalized log marginal score. Then we estimate input dependence via

$$\widehat{I}(X;Z) = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \Big(\tilde{\ell}_i(Z_{i,k}) - \tilde{\ell}_{\mathrm{mix}}(Z_{i,k}) \Big),$$

which increases when a reasoning segment is much more compatible with its source input than with the batch mixture. In template-collapse regimes, $\tilde{\ell}_i \approx \tilde{\ell}_{\mathrm{mix}}$ for many samples and thus $\widehat{I}(X;Z)$ approaches 0.

3 Mechanics of Template Collapse in LLM Agent RL: An SNR View

3.1 SNR Mechanism via Gradient Decomposition

We explain template collapse through a signal-to-noise ratio (SNR) view of policy optimization (formal RV–signal and SNR bounds in Appendix G; see also Appendix J for the regularizer-dominance interpretation) (Figure 2). Intuitively, stable, input-conditioned learning requires that the *task-driven* gradient is both (i) sufficiently large and (ii) directionally informative, so that it can dominate more task-agnostic forces.

From advantage to task gradients In policy-gradient methods such as PPO (Schulman et al., 2017b) and GRPO (Shao et al., 2024), the policy update is driven by advantage-weighted log-probability gradients (notation in Appendix D).. The advantage is computed from reward signals and a baseline: in PPO it uses a learned critic/value function, while in GRPO it uses a group/average baseline. Therefore, when the within-prompt variability of returns is small, advantage-weighted gradients become weak: the per-prompt task gradient magnitude is upper bounded by $\sqrt{\mathrm{RV}(x)}$ (Appendix G, Theorem G.3).

SNR hypothesis We decompose the update gradient as $g_{total} = g_{task} + g_{reg}$, where g_{task} is task-related and g_{reg} aggregates task-agnostic terms (e.g., KL/entropy regularization) and other noise. We hypothesize that in low-SNR regimes, $\|g_{task}(x)\|$ weakens relative to $\|g_{reg}(x)\|$, causing updates be like a global contraction that reduces $I(X;Z)$ even when $H(Z \mid X)$ remains high (Appendix J).

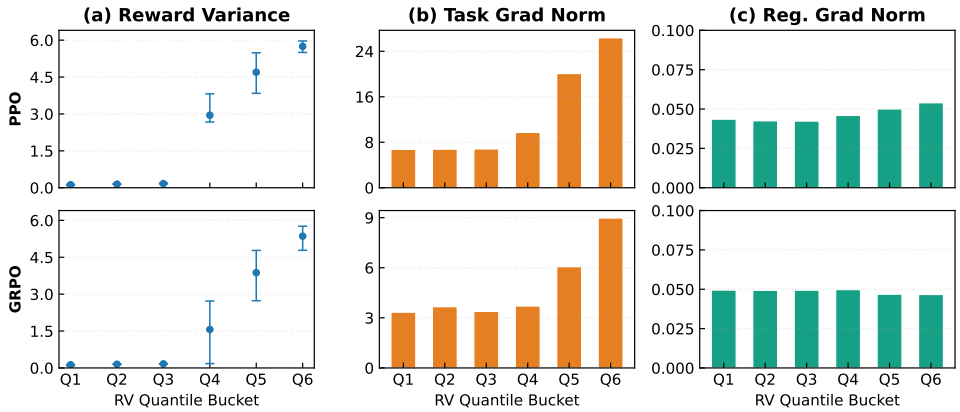

Figure 3: Gradient decomposition by reward-variance buckets. We sort tasks by Var(R|X) and split them into 6 equal-sized buckets. (a) Reward variance (RV) distributions per bucket. (b) The task/policy gradient norm increases with the bucket RV median. (c) The regularizer gradient norm (KL+entropy) is largely flat across RV. These trends are consistent with a Signal-to-Noise Ratio (SNR) view: low $\mathrm{RV}(x)$ upper-bounds task-gradient magnitude and degrades estimator SNR (Appendix G), which makes reward-agnostic regularizers relatively more dominant in the total update (Appendix J).

3.2 How To Separate Task-Relevant Signal from Noise

Reward variance as an SNR proxy To operationalize SNR in closed-loop agent RL, we use within-prompt reward variance $\mathrm{Var}(R \mid X)$ as a lightweight proxy (Appendix D, Appendix G) for the strength of task-discriminative signal. Intuitively, higher $\mathrm{Var}(R \mid X)$ indicates that different trajectories sampled under the same prompt can be meaningfully distinguished by reward, making advantage estimates more informative and increasing the chance that the resulting gradients align with task-relevant directions. In contrast, when $\mathrm{Var}(R \mid X)$ is small, reward provides weak discrimination, and the update is more likely to be shaped by task-agnostic factors such as KL/entropy regularization and sampling/environment noise.

Motivated by the SNR view, we propose reward-variance-aware (RV-aware) filtering (Appendix F): a simple intervention that prioritizes updates where the within-prompt return-variability suggests a stronger task signal. Specifically, at each iteration, we estimate $\mathrm{Var}(R \mid X)$ at the prompt level by sampling G trajectories for the same prompt X and computing the (unbiased) sample variance of the resulting episode returns:

$$\widehat{\mathrm{Var}}(R \mid X) = \frac{1}{G-1} \sum_{g=1}^{G} \big(R_g(X) - \overline{R}(X)\big)^2,$$

$$\overline{R}(X) = \frac{1}{G} \sum_{g=1}^{G} R_g(X).$$

We keep the top fraction of prompts by this reward-variance score with keep rate $\rho \in (0, 1]$ (a "top-P" selection analogous in spirit to nucleus/top-P sampling Holtzman et al. (2020), except the ranking is by per-prompt reward variance rather than token probability), then run the same PPO / GRPO-style update on the retained set (Appendix I).

We run a preliminary analysis (setup in Section 4) to test the SNR hypothesis and relate it to Figure 2. As shown in Figure 3, we sort prompts by $\widehat{\mathrm{Var}}(R \mid X)$, split them into equal-sized buckets, and decompose gradient norms within each bucket. We observe three trends: (i) RV distributions separate cleanly across buckets (Figure 3a); (ii) the task/policy-gradient norm grows with bucket RV (Figure 3b); (iii) regularizer-gradient norms (e.g., KL+entropy) remain relatively flat (Figure 3c). These patterns match the SNR view: $\mathrm{RV}(x)$ upper-bounds the task-gradient magnitude $\|g_{\mathrm{task}}(x)\|$ (Appendix G), and low $\mathrm{RV}(x)$ worsens the estimator SNR, making reward-agnostic regularizers comparatively more influential (Appendix J).

Table 1: Experimental matrix across tasks and experimental axes, including algorithms, model scales, model types, and modalities (%). Each entry reports both the peak performance without RV filtering and the impact of RV filtering, e.g., a (+Δ). For Qwen2.5-VL-3B, we evaluate both text- and image-conditioned input (T / V). Filtering improves performance on most tasks and consistently improves the score across all variants shown.

Experiment Variants	Sokoban	FrozenLake	MetaMathQA	Countdown	Average
Baseline					
PPO (Schulman et al., 2017b), Qwen2.5-3B (Qwen Team, 2024)	12.9 **(+16.0)**	67.0 **(+10.9)**	92.6 **(+0.6)**	97.9 (+0.0)	67.6 **(+6.9)**
Algorithm					
DAPO (Yu et al., 2025)	16.2 **(+5.1)**	66.8 **(+2.1)**	90.8 **(+2.8)**	95.7 **(+1.6)**	67.4 **(+2.9)**
GRPO (Shao et al., 2024)	12.1 **(+9.0)**	56.2 (-5.0)	91.2 **(+1.2)**	95.7 **(+2.2)**	63.8 **(+1.9)**
Dr. GRPO (Liu et al., 2025)	12.1 (-0.4)	23.2 **(+0.6)**	91.2 **(+1.4)**	96.5 **(+1.4)**	55.8 **(+0.8)**
Model Scale (PPO)					
Qwen2.5-0.5B (Qwen Team, 2024)	3.3 **(+22.9)**	19.5 **(+0.0)**	10.0 (-0.2)	23.0 (-0.7)	14.0 **(+5.5)**
Qwen2.5-1.5B (Qwen Team, 2024)	17.0 **(+6.2)**	36.5 **(+1.6)**	80.3 **(+7.0)**	56.6 **(+1.6)**	47.6 **(+4.1)**
Qwen2.5-7B (Qwen Team, 2024)	42.4 **(+4.9)**	85.0 (-0.6)	84.0 **(+11.7)**	97.7 **(+0.3)**	77.3 **(+4.1)**
Model Type					
Qwen2.5-3B-Instruct (Qwen Team, 2024)	22.5 **(+14.2)**	83.6 **(+2.3)**	91.2 **(+0.4)**	96.3 (-0.6)	73.4 **(+4.1)**
Llama3.2-3B (Meta Llama, 2024)	24.4 **(+18.8)**	84.1 (-4.1)	86.1 **(+3.7)**	99.2 (-1.2)	73.5 **(+4.3)**
Modality (Input Type)					
Qwen2.5-VL-3B (T) (Bai et al., 2025)	53.0 **(+6.0)**	0.0 **(+45.0)**	-	-	26.5 **(+25.5)**
Qwen2.5-VL-3B (V) (Bai et al., 2025)	65.0 **(+12.0)**	0.0 **(+45.0)**	-	-	32.5 **(+28.5)**

4 Experiments

4.1 Experimental Testbed

We adopt the RAGEN Wang et al. (2025b) testbed and evaluate LLM agents on four controllable tasks that stress complementary decision-making regimes: irreversible planning (Sokoban), sparse-reward long-horizon navigation under stochastic transitions (FrozenLake), and symbolic math reasoning (MetaMathQA, Countdown).

Environments and tasks. **Sokoban** is a grid puzzle where the agent pushes boxes onto target cells with effectively irreversible moves (no pulling) (Schrader, 2018). **FrozenLake** is a navigation task with sparse rewards and stochastic transitions (Brockman et al., 2016). **MetaMathQA** is a multi-attempt math QA task (Yu et al., 2023) where we allow answer revision and halve the reward after each retry. **Countdown** is a single-turn numbers game (Katz et al., 2025) where the agent constructs an arithmetic expression to reach a target.

Table 2: Summary of the features of the environments used.

Task	Stochastic	Multi-turn	State	Reward
Sokoban	✗	✓	Grid	Dense
FrozenLake	✓	✓	Grid	Binary
MetaMathQA	✗	✓	Text	Dense
Countdown	✗	✗	Text	Binary

Training and evaluation setup. We train Qwen2.5-3B (Qwen Team, 2024) with veRL/HybridFlow (Sheng et al., 2024), following RAGEN defaults unless noted (Wang et al., 2025b). We compare PPO (Schulman et al., 2017b), DAPO (Yu et al., 2025), GRPO (Shao et al., 2024), and Dr. GRPO (Liu et al., 2025) for up to 400 rollout–update iterations. Each iteration collects $K = 128$ trajectories per environment, organized as $P = 8$ prompts with $G = 16$ samples per prompt. With RV-aware filtering (keep rate ρ), we keep a ρ fraction of samples and scale the per-step loss by ρ to keep the effective step size comparable.

4.2 Large Scale evaluation across tasks, scales, modalities

Table 1 summarizes our experimental matrix over four tasks, multiple RL algorithms, model scales/types, and input modalities. Across this grid, RV-aware filtering shows two recurring effects. First, it improves peak task success in most settings (the reported $+\Delta$), indicating that selecting higher-signal updates improves learning efficiency. Second, these gains persist across optimizers (PPO/DAPO/GRPO/Dr. GRPO), model families and scales (Qwen2.5 0.5B–7B; Llama3.2-3B), and input modalities (text- and image-conditioned Qwen2.5-VL). DAPO and Dr. GRPO are recent

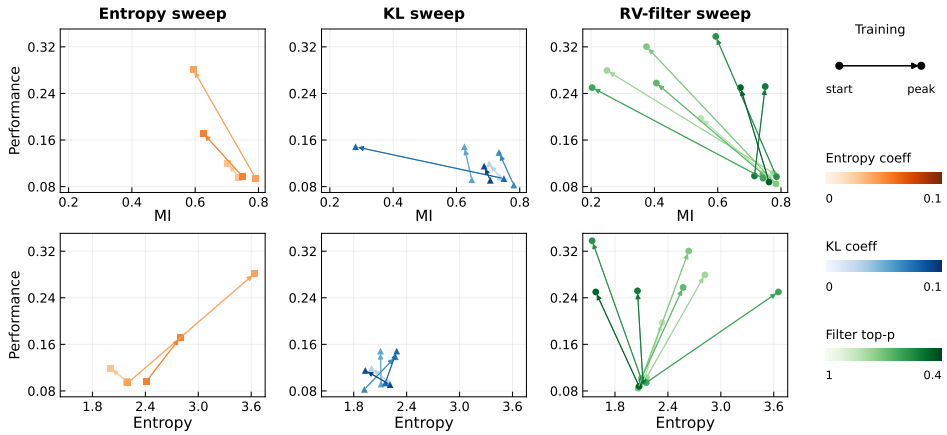

Figure 4: Training dynamics under three interventions. For each setting, we choose two checkpoints (steps 10/400) and connect them into a trajectory (arrows point to later steps). Color intensity indicates weaker to stronger intervention.

stabilization-oriented baselines; in Table 1, the DAPO "no-filter" entries match the original method without our filtering. While DAPO already performs a fixed acceptance step (a top-P rule with $P \to 1.0$), our RV-aware filtering provides an explicit, tunable SNR knob via the keep rate ρ, and complements standard regularizers such as KL and entropy.

4.3 Comparing RV-Filtering and KL/Entropy Regularizers

We evaluate the diagnostics and the SNR-motivated intervention from Section 3 by sweeping three knobs—entropy regularization, KL constraint strength, and the RV-aware filtering keep rate—and plotting training trajectories in Figure 4. Entropy- and KL-based stabilizers primarily change entropy and rarely move the model into a high-$I(X;Z)$ regime with clear performance gains. In contrast, RV-aware filtering consistently increases $I(X;Z)$ and improves final success even when entropy decreases (e.g., stronger filtering in the upper-right region raises both $I(X;Z)$ and returns). Moreover, entropy sweeps often yield fewer usable runs due to entropy blow-up, while KL mainly keeps the policy near its reference distribution, limiting movement in the diagnostic space. Overall, RV-aware filtering is a more direct knob for improving both performance and input-dependent reasoning than tuning entropy or KL.

4.4 Randomness sweeps and robustness

We sweep environment and policy randomness to probe the SNR mechanism under controlled stress. As randomness increases, task return drops, conditional entropy rises, and $I(X;Z)$ decreases (Figure 5). This pattern is consistent with the view that additional stochasticity injects noise into rollouts and dilutes task-discriminative gradients: action- and response-level randomness weakens input-grounded decisions, while stochastic environment transitions further blur the mapping from inputs to outcomes.

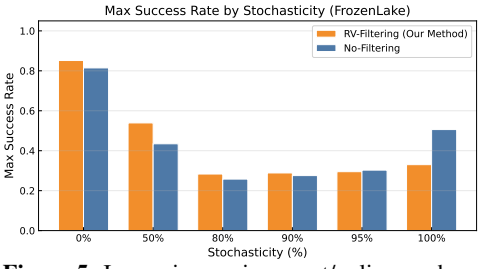

Figure 5: Increasing environment/policy randomness reduces return gain from RV-filtering.

5 Analysis and Implications

Across our experiments, we observe three recurring patterns. (i) Increasing environment/policy randomness tends to raise within-input variability (higher conditional entropy) while lowering input dependence (lower $I(X;Z)$), often accompanied by weaker task returns (Figure 5). (ii) RV-aware filtering tends to help most in higher-stochasticity settings and can improve time-to-performance by prioritizing higher-signal updates (Figure 6). (iii) Reward variance shows relatively weak correlation

with common proxies (e.g., length and entropy), suggesting it provides an additional, mechanism-aligned selection signal (Table 3).

Randomness can serve as an SNR knob. Figure 5 suggests that higher environment/policy randomness can increase within-input variability (higher conditional entropy) while reducing input dependence (lower $I(X;Z)$), often alongside weaker task returns. In practice, if a task becomes more stochastic (e.g., due to transition noise, exploration, or higher sampling temperature), it may help to use stronger stabilization or higher-SNR training signals.

Table 3: Correlations between RV and candidate proxies (higher ↑ indicates stronger positive correlation).

Variable	Spearman ↑	Pearson ↑
Reward	0.630	0.650
Retrieval Acc (traj)	0.130	0.150
Response Length	0.120	0.080
MI (traj, est)	-0.100	-0.170
Conditional Entropy (N)	-0.140	-0.180

Use RV-aware filtering to prioritize high-signal updates. Figure 6 suggests that stronger filtering is often favored in higher-stochasticity regimes and can improve time-to-performance in training-only wall-clock time. This is consistent with the SNR story: filtering shifts optimization weight toward prompts that provides clearer advantage separation, so updates are more likely to follow task-aligned directions.

Use RV as a mechanism-aligned selection signal. Table 3 suggests that reward variance has relatively weak correlation with common proxies (length, entropy, retrieval accuracy), supporting its role as a selection signal that targets update quality rather than surface statistics. This makes RV-aware filtering a clean control knob that can complement KL and entropy regularization.

6 Related Work

Reasoning collapse and policy degeneracy in closed-loop LM and Agent RL training. LLM-agent RL reports various collapse phenomena (DeepSeek AI, 2025; Wei et al., 2025): *reasoning collapse* (rationales becoming templated with weaker input correspondence) (Wei et al., 2025; Yao et al., 2025; Yun et al., 2025) and *policy-level degeneracy* (behavior concentrating on easy-to-reproduce patterns) (Feng et al., 2025; Wang & Ammanabrolu, 2025).

Evaluating reasoning diversity, input dependence, and reasoning faithfulness. Most diversity metrics do not test whether differences are *systematically driven by inputs* (Tevet & Berant, 2021; Yun et al., 2025). Common measures include lexical overlap (Li et al., 2016; Zhu et al., 2018), embedding dispersion (Pillutla et al., 2021; Tevet & Berant, 2021), and uncertainty analyses (Montahaei et al., 2019; Semeniuta et al., 2019), primarily capturing within-input variability and missing cross-input shifts (Semeniuta et al., 2019; Tevet & Berant, 2021).

Stabilizing multi-turn Agent RL under closed-loop sampling. Stability work spans KL control, entropy regularization, clipping, reward shaping, curricula, and replay mixtures (Schulman et al., 2017a;b; 2018; Haarnoja et al., 2019; Stiennon et al., 2022; Ouyang et al., 2022). However, these methods do not prevent drift toward input-agnostic templates: if rollouts receive similar rewards regardless of reasoning quality, gradients carry little information (Moskovitz et al., 2023; O'Mahony et al., 2024; Shumailov et al., 2024; Yun et al., 2025). We adopt a SNR view, using reward variance filtering low-signal samples to maintain effective SNR.

7 Conclusions and Limitations

We study a failure mode in closed-loop multi-turn agent RL where an LLM agent's reasoning drifts toward fluent but input-agnostic templates. We argue this *template collapse* can be missed by entropy-based diversity metrics, and propose an information-theoretic view that separates within-input variability $H(Z \mid X)$ from input dependence $I(X;Z)$. To make $I(X;Z)$ measurable, we introduce an in-batch MI-style retrieval diagnostic. We then give an SNR explanation for why $I(X;Z)$ drops when task-agnostic factors (e.g., KL/entropy regularization and noise) dominate. As a minimal control knob, we propose reward-variance-aware filtering to prioritize higher-signal updates. Across tasks, algorithms, and model scales, RV-aware filtering improves input dependence and often performance, suggesting that preserving effective task signal helps prevent input-agnostic drift in long-horizon training. Limitations include that we do not cover multi-agent settings, and benefits are limited in highly stochastic, low-SNR environments.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.

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

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, 2006.

DeepSeek AI. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. Re-rest: Reflection-reinforced self-training for language agents, 2025. URL https://arxiv.org/abs/2406.01495.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Lirong Gao, Ru Peng, Yiming Zhang, and Junbo Zhao. Dory: Deliberative prompt recovery for llm, 2024. URL https://arxiv.org/abs/2405.20657.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets, 2020. URL https://arxiv.org/abs/2004.02709.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data, 2024. URL https://arxiv.org/abs/2404.01413.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications, 2019. URL https://arxiv.org/abs/1812.05905.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Michael Katz, Harsha Kokel, and Sarath Sreedharan. Benchmarking llms on the game of countdown, 2025. URL https://arxiv.org/abs/2508.02900.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity, 2024. URL https://arxiv.org/abs/2310.06452.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam

McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

Hanqing Li and Diego Klabjan. Reverse prompt engineering, 2025. URL https://arxiv.org/abs/2411.06729.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models, 2016. URL https://arxiv.org/abs/1510.03055.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. URL https://arxiv.org/abs/2503.20783.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023. URL https://arxiv.org/abs/2303.17651.

Meta Llama. Llama 3.2 3b model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B. Accessed 2026-01-28.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. Jointly measuring diversity and quality in text generation models, 2019. URL https://arxiv.org/abs/1904.03971.

John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. Language model inversion, 2023. URL https://arxiv.org/abs/2311.13647.

Ted Moskovitz, Aaditya K. Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca D. Dragan, and Stephen McAleer. Confronting reward model overoptimization with constrained rlhf, 2023. URL https://arxiv.org/abs/2310.04373.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022. URL https://arxiv.org/abs/2112.09332.

Laura O'Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. URL https://openreview.net/forum?id=3pDMYjpOxk.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers, 2021. URL https://arxiv.org/abs/2102.01454.

Qwen Team. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist, 2020. URL https://arxiv.org/abs/2005.04118.

Joshua Romoff, Peter Henderson, Alexandre Piché, Vincent Francois-Lavet, and Joelle Pineau. Reward estimation for variance reduction in deep reinforcement learning, 2018. URL https://arxiv.org/abs/1805.03359.

Max-Philipp B. Schrader. gym-sokoban, 2018. URL https://github.com/mpSchrader/gym-sokoban. Accessed 2026-01-29.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017a. URL https://arxiv.org/abs/1502.05477.

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

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL https://arxiv.org/abs/1506.02438.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. On accurate evaluation of gans for language generation, 2019. URL https://arxiv.org/abs/1806.04936.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework, 2024. URL https://arxiv.org/abs/2409.19256.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631:755–759, 2024. doi: 10.1038/s41586-024-07566-y. URL https://doi.org/10.1038/s41586-024-07566-y.

Noah Y. Siegel, Oana-Maria Camburu, Nicolas Heess, and Maria Perez-Ortiz. The probabilities also matter: A more faithful metric for faithfulness of free-text explanations in large language models, 2024. URL https://arxiv.org/abs/2404.03189.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. Fast best-of-n decoding via speculative rejection, 2024. URL https://arxiv.org/abs/2410.20290.

Leitian Tao, Ilia Kulikov, Swarnadeep Saha, Tianlu Wang, Jing Xu, Sharon Li, Jason E Weston, and Ping Yu. Hybrid reinforcement: When reward is sparse, it's better to be dense, 2025. URL https://arxiv.org/abs/2510.07242.

Guy Tevet and Jonathan Berant. Evaluating the evaluation of diversity in natural language generation, 2021. URL https://arxiv.org/abs/2004.02990.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023. URL https://arxiv.org/abs/2305.04388.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL https://arxiv.org/abs/2211.14275.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. URL https://arxiv.org/abs/2305.16291.

Jiawei Wang, Jiacai Liu, Yuqian Fu, Yingru Li, Xintao Wang, Yuan Lin, Yu Yue, Lin Zhang, Yang Wang, and Ke Wang. Harnessing uncertainty: Entropy-modulated policy gradients for long-horizon llm agents, 2025a. URL https://arxiv.org/abs/2509.09265.

Ruiyi Wang and Prithviraj Ammanabrolu. A practitioner's guide to multi-turn agentic reinforcement learning, 2025. URL https://arxiv.org/abs/2510.01132.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning, 2025b. URL https://arxiv.org/abs/2504.20073.

Tong Wei, Yijun Yang, Junliang Xing, Yuanchun Shi, Zongqing Lu, and Deheng Ye. Gtr: Guided thought reinforcement prevents thought collapse in rl-based vlm agent training, 2025. URL https://arxiv.org/abs/2503.08525.

Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N. Metaxas. Epo: Entropy-regularized policy optimization for llm agents reinforcement learning, 2025. URL https://arxiv.org/abs/2509.22576.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning, 2025. URL https://arxiv.org/abs/2505.23433.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2023. URL https://arxiv.org/abs/2309.12284.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Longfei Yun, Chenyang An, Zilong Wang, Letian Peng, and Jingbo Shang. The price of format: Diversity collapse in llms, 2025. URL https://arxiv.org/abs/2505.18949.

Kerem Zaman and Shashank Srivastava. Is chain-of-thought really not explainability? chain-of-thought can be faithful without hint verbalization, 2025. URL https://arxiv.org/abs/2512.23032.

Collin Zhang, John X. Morris, and Vitaly Shmatikov. Extracting prompts by inverting llm outputs, 2024. URL https://arxiv.org/abs/2405.15012.

Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. Promptbench: A unified library for evaluation of large language models, 2024. URL https://arxiv.org/abs/2312.07910.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models, 2018. URL https://arxiv.org/abs/1802.01886.

A Extended Related Work

Reasoning collapse and policy degeneracy in closed-loop LM and agent RL training. We study a family of degradation phenomena in closed-loop LLM-agent reinforcement learning that has not yet been uniformly defined, but has been repeatedly reported across settings (DeepSeek AI, 2025; Wei et al., 2025). After the model is updated on self-sampled trajectories over time, it may gradually exhibit *reasoning collapse* and *policy-level degeneracy* (DeepSeek AI, 2025; Wei et al., 2025). Here, *reasoning collapse* mainly refers to the rationales, plans, or explanations becoming increasingly templated and less diverse, while their correspondence to the input goal weakens (Wei et al., 2025; Yao et al., 2025; Yun et al., 2025). In contrast, *policy-level degeneracy* refers to behavioral choices concentrating on a small set of easy-to-reproduce action patterns that yield stable scores, with less exploration and less error correction (Feng et al., 2025; Wang & Ammanabrolu, 2025).

This family of phenomena echoes earlier findings in self-training, self-distillation, and iterative fine-tuning on synthetic or model-generated data. When a model repeatedly trains on its own generated distribution, the feedback loop can gradually narrow the effective data distribution, amplify a few high-probability modes, and suppress long-tail behaviors, even when average quality metrics appear stable (Gerstgrasser et al., 2024; Shumailov et al., 2024). In the agent RL setting, closed-loop optimization on on-policy trajectories introduces additional risks, but these risks do not necessarily appear first as an overt failure of the behavioral policy. Instead, a commonly reported pattern is that, even when the agent's external behavior remains effective or yields stable rewards, language-level reasoning expressions can become concentrated earlier. Plans and explanations may converge to a few reusable narrative skeletons, and their alignment with the specific input goal can weaken (Wei et al., 2025; Xu et al., 2025). In other words, reasoning-level degeneration can decouple from policy-level degeneracy, and in some settings it may precede it (Wang & Ammanabrolu, 2025). In multi-turn interaction, related work also describes several visible signatures of this degradation family, such as within-task convergence across repeated rollouts, cross-task templating where different prompts share the same planning or rhetorical skeleton, and late-stage degeneration where later turns become more mechanical or more conservative (Wang et al., 2025a; Xu et al., 2025).

Evaluating reasoning diversity, input dependence, and reasoning faithfulness. Prior work on evaluating *reasoning diversity* often answers how different the outputs are, but less directly answers whether these differences are *systematically driven by the input goal*, which can blur the interpretation of template-like degeneration under closed-loop training (Tevet & Berant, 2021; Yun et al., 2025). Concretely, common metrics range from lexical measures such as n-gram statistics and self-BLEU (Li et al., 2016; Zhu et al., 2018), to embedding-based dispersion and distributional distances (Pillutla et al., 2021; Tevet & Berant, 2021), as well as token-level uncertainty proxies and multi-sample coverage or consistency analyses (Montahaei et al., 2019; Semeniuta et al., 2019). These metrics primarily capture overall randomness or within-input variability, and they are often less sensitive to whether the reasoning distribution changes coherently *across* inputs (Semeniuta et al., 2019; Tevet & Berant, 2021). Other evaluation protocols rely on model scoring or human preference judgments to compare overall response quality, but they are not designed to isolate input-conditioned reasoning differences, and they may conflate prompt-coupled variation with prompt-agnostic surface diversity, especially when outputs converge to shared formats (Kirk et al., 2024; Yun et al., 2025). This leaves a gap for scalable evaluation of whether reasoning is *diagnostic of the input*, which is particularly salient in multi-turn, stochastic environments where a fixed agent policy can produce diverse yet reusable templates (Wang & Ammanabrolu, 2025). Recent work has started to probe input dependence via behavioral tests and local boundary checks (Gardner et al., 2020; Ribeiro et al., 2020), prompt robustness benchmarks (Zhu et al., 2024), and retrieval-style output–input matching or prompt reconstruction signals (Morris et al., 2023; Gao et al., 2024; Zhang et al., 2024; Li & Klabjan, 2025). However, a unified and scalable treatment tailored to closed-loop agent RL remains limited, even as algorithmic work continues to address long-horizon stability and collapse (Feng et al., 2025; Yao et al., 2025).

A closely related line studies *reasoning faithfulness* (explanation faithfulness), which asks whether a rationale reflects the true basis of a decision rather than a plausible post-hoc story (Lanham et al., 2023; Turpin et al., 2023; Siegel et al., 2024; Zaman & Srivastava, 2025). Our question is related but not equivalent: faithfulness emphasizes whether reasoning causally supports a particular decision, while we focus on a different degeneration risk in closed-loop optimization, namely whether reasoning

gradually becomes *less sensitive to the input* and drifts toward reusable templates, even when local explanations remain self-consistent (Kirk et al., 2024). This motivates our decomposition of reasoning diversity into within-input variability and cross-input dependence, and our scalable proxy for the latter through an information-theoretic lens.

Stabilizing multi-turn Agent RL under closed-loop sampling. To improve training stability when aligning LLMs and LLM-based agents, prior work has proposed a broad set of algorithmic and system-level techniques. These include KL control or trust-region style constraints, entropy regularization, clipping and normalization in policy-gradient updates, reward shaping and credit assignment, curriculum design, replay or offline–online mixtures, as well as rejection sampling and best-of-N selection (Schulman et al., 2017a;b; 2018; Haarnoja et al., 2019; Stiennon et al., 2022; Ouyang et al., 2022; Rafailov et al., 2024; Sun et al., 2024; Feng et al., 2025; Wang & Ammanabrolu, 2025; Wang et al., 2025a; Xu et al., 2025; Yao et al., 2025). For multi-step agents, researchers have also explored stepwise rewards and intermediate supervision, imitation-to-RL pipelines, and self-correction or reflection signals to support longer-horizon planning and reduce brittle behaviors (Cobbe et al., 2021; Nakano et al., 2022; Uesato et al., 2022; Madaan et al., 2023; Shinn et al., 2023; Wang et al., 2023; Yao et al., 2023; Dou et al., 2025; Wei et al., 2025).

Despite these advances, many stabilization methods are tuned to prevent optimization collapse or to improve overall reward. When the effective learning signal in the closed loop becomes weak or noisy, these methods do not necessarily prevent drift toward prompt-agnostic templates. For example, if most rollouts for the same prompt receive similar rewards regardless of reasoning quality, then the gradient update carries little information about which reasoning path matters (Moskovitz et al., 2023; O'Mahony et al., 2024; Shumailov et al., 2024; Yun et al., 2025). This motivates methods that explicitly manage the balance between task-specific signal and task-agnostic pressure. We adopt a signal-to-noise view of closed-loop updates: we use within-prompt reward variance as a proxy for signal strength, and we filter low-signal samples to maintain an effective SNR, so that exploration and input-conditioned reasoning are less likely to be washed out over long-horizon multi-turn optimization (Romoff et al., 2018; Shao et al., 2024; Tao et al., 2025; Feng et al., 2025; Yao et al., 2025).

B Detailed Experimental Settings

B.1 Environments and Tasks

We construct a diverse four-environment testbed to evaluate LLM agents across complementary axes of decision-making complexity, including planning under irreversible dynamics (Sokoban), long-horizon control with deterministic transitions (Frozen Lake), and multi-step symbolic reasoning in mathematics (MetaMathQA and Countdown). All environments are synthetic and fully controllable, enabling clean analysis of RL learning from scratch without relying on real-world priors.

Sokoban. We use the puzzle Sokoban Schrader (2018) to study multi-turn agent interaction with irreversible dynamics. The agent must push boxes to designated target locations within a grid-based warehouse. Unlike standard navigation tasks, Sokoban is characterized by irreversibility: boxes can only be pushed, not pulled, meaning a single misstep can create unsolvable dead-ends where boxes become permanently stuck against walls or corners. This requires the agent to reason ahead and plan multi-step sequences before committing to actions. The reward signal encourages both efficiency and accuracy: $+1$ for each box successfully placed on a target, -1 for moving a box off a target, $+10$ upon task completion, and -0.1 per action as a step penalty. We use procedurally generated puzzles with configurable room dimensions and box counts to ensure diverse training scenarios.

Frozen Lake. This environment of FrozenLake Brockman et al. (2016) combines long-horizon decision-making with deterministic transitions. The agent navigates a grid of frozen tiles to reach a goal while avoiding holes that terminate the episode. We use the 2% random rate variant of Frozen Lake, where each intended action is executed at a 98% probability. Rewards are sparse: only successful goal-reaching trials receive a reward of $+1$, with all other outcomes yielding 0. The combination of sparse rewards and long-horizon planning makes this environment challenging for credit assignment.

MetaMathQA. To evaluate mathematical reasoning capabilities, we include MetaMathQA Yu et al. (2023), a question-answering task drawn from the MetaMathQA dataset. Each episode presents the

agent with a mathematical problem requiring multi-step reasoning—ranging from arithmetic and algebra to word problems and geometry. The agent must produce a final answer, and correctness is determined by exact match with the ground truth. To encourage efficient reasoning, we employ a diminishing reward scheme: correct answers on the first attempt receive full reward (1.0), with rewards halving for each subsequent attempt (0.5, 0.25, ...).

Countdown. Inspired by the numbers game from the TV show "Countdown" Katz et al. (2025), this environment tests compositional arithmetic reasoning. The agent is given a target number and a set of source numbers, and must construct an arithmetic expression using each source number at most once to reach the target exactly. For example, given target 24 and numbers $[1, 5, 6, 7]$, a valid solution is $6 \times (7 - 5 + 1) + 6$. Rewards distinguish between format correctness and solution correctness: full reward (1.0) for correct solutions, partial reward (0.1) for expressions that use the correct numbers but yield incorrect results, and zero for malformed expressions.

B.2 Training and Evaluation Setup

We conduct our main experiments using Qwen2.5-3B and train with four policy-gradient variants—PPO, DAPO, GRPO, and Dr.GRPO—for up to 400 rollout–update iterations on NVIDIA GPUs using the veRL framework, with early stopping enabled as described below. Each iteration collects $K = 128$ trajectories per environment, organized as $P = 8$ prompt groups with $G = 16$ parallel samples per prompt.

Episode horizons. To match task structure, the interactive environments (Sokoban, Frozen Lake) use up to 5 interaction turns with 2 actions per turn (10 total actions per trajectory). The single-step reasoning tasks (Countdown, MetaMathQA) use 1 turn with 1 action.

Optimization. We use an update batch size of 32 and a per-GPU minibatch size of 4. Policy optimization uses GAE with $(\gamma, \lambda) = (1.0, 1.0)$ and Adam with $(\beta_1, \beta_2) = (0.9, 0.999)$. The actor learning rate is 1×10^{-6} and the critic learning rate is 1×10^{-5}. We apply entropy regularization with coefficient $\beta = 0.001$. For PPO-based methods, we use asymmetric clipping with $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. We additionally impose a format penalty of -0.1 when the agent fails to output a valid structured response (e.g., missing `<think>` or `<answer>` tags).

Early stopping. We stop training if either (i) reward-variance collapse is detected—the reward variance drops below 10% of the baseline variance (defined as the mean variance over the first 10 training iterations) for 5 consecutive iterations—or (ii) the validation success rate remains below 1% for 5 consecutive evaluation checkpoints.

Filtering ablation. We compare filtered rollouts with `top_p` $= 0.9$ (keeping the top 90% of trajectory groups ranked by reward variance) against an unfiltered setting.

Evaluation. We evaluate on a fixed set of 512 validation prompts per environment and decode with temperature $T = 0.5$ using stochastic sampling. We report success rate as the primary metric across all environments.

C Filtering Ablation Results

We conduct our filtering experiments using Qwen2.5-3B model on Sokoban environment. We summarize the filtering ablation results in Table 4. Each row reports the absolute value of each metric, with the change relative to a section-specific baseline shown in parentheses. Within each block, the first row labeled *baseline* defines the reference point, and all deltas are computed relative to that baseline. We report four metrics: **Task Performance**, defined as the maximum validation success rate attained during training; **MI Proxy**, measured as retrieval accuracy at the training step where task performance peaks; **Entropy**, an estimate of reasoning entropy at the same step; and **Collapse**, a binary indicator of whether validation success ever falls below 0.01 during training.

Sampling Settings. We first study the interaction between filtering and sampling by varying sampling thresholds while holding the reward-variance (RV) filter fixed. Relative to the `top_p` $= 1.0$ baseline, reducing `top_p` or `min_p` generally improves task performance while reducing entropy, but with heterogeneous effects on MI retention. In contrast, `top_k` sampling induces a sharper trade-off: MI proxy is often preserved or improved, while gains in task performance are less consistent. These

Table 4: Ablation results for sampling settings, filtering metrics, and keep strategy. Values in parentheses denote the change relative to the corresponding baseline in each block. A crossmark in the Stable column means training collapsed.

EXPERIMENT SETUP	PERFORMANCE	MI PROXY	ENTROPY	STABLE
Sampling Settings				
Top-p=1.0 (baseline)	0.17	0.54	2.76	✗
Top-p=0.9	0.38 **(+0.20)**	0.84 **(+0.29)**	1.64 (-1.12)	✓
Top-p=0.5	0.29 **(+0.12)**	0.83 **(+0.29)**	1.88 (-0.88)	✓
Min-p=0.05	0.42 **(+0.25)**	0.67 **(+0.13)**	1.64 (-1.12)	✓
Min-p=0.2	0.45 **(+0.27)**	0.36 (-0.18)	3.01 **(+0.26)**	✓
Top-k=0.25	0.22 **(+0.05)**	0.86 **(+0.32)**	1.28 (-1.48)	✓
Top-k=0.5	0.44 **(+0.27)**	0.89 **(+0.35)**	1.47 (-1.29)	✓
Filtering Metrics				
No filter (baseline)	0.17	0.54	2.76	✗
RV	0.38 **(+0.20)**	0.84 **(+0.29)**	1.64 (-1.12)	✓
Entropy	0.20 **(+0.02)**	0.41 (-0.14)	2.20 (-0.56)	✗
Entropy-variance	0.23 **(+0.06)**	0.70 **(+0.16)**	2.94 **(+0.18)**	✗
Length	0.16 (-0.02)	0.91 **(+0.36)**	1.65 (-1.10)	✗
Keep Strategy				
Keep largest (baseline)	0.44	0.89	1.47	✓
Keep smallest	0.29 (-0.15)	0.47 (-0.42)	5.31 **(+3.84)**	✓

results indicate that filtering behavior is strongly modulated by the sampling regime, even when the underlying filter metric is unchanged.

Filtering Metrics. Next, we fix the sampling scheme and vary the filtering criterion. Switching between RV, entropy-based, entropy-variance, and length-based filters leads to substantial differences in both peak task performance and MI proxy. In particular, RV filtering consistently achieves strong task performance while better preserving MI compared to entropy-based alternatives. Entropy- and length-based filters either suppress MI or fail to prevent collapse, suggesting that RV provides a more stable and informative signal for selecting useful rollouts.

Keep Strategy. Finally, we compare *keep-largest* and *keep-smallest* strategies under the same `top_k` configuration. As expected, retaining high-variance trajectory groups yields substantially higher task performance and MI proxy, while keeping the smallest-variance groups degrades both and markedly increases entropy. This asymmetry supports the hypothesis that high-variance rollouts contain more informative training signal, whereas low-variance rollouts are largely uninformative or noisy.

Summary. Overall, the ablation reveals strong interactions between sampling strategy and filtering choice. More aggressive filtering is not universally beneficial, and the choice of filtering metric is critical: reward-variance filtering consistently improves task performance while maintaining information content, whereas entropy-based heuristics are less reliable and more prone to collapse.

D NOTATION AND BASIC IDENTITIES

D.1 RANDOM VARIABLES AND DISTRIBUTIONS

[Prompts, trajectories, and rollouts] Let X denote an input prompt and Z a reasoning trajectory. A rollout sample is

$$\xi = (x, z, r),$$

with x the prompt, z the realized trajectory, and $r \in \mathbb{R}$ the scalar reward.

We write $\pi_\theta(z \mid x)$ as the policy and $P(X)$ the prompt distribution. Rollouts are generated by

$$x \sim P(X), \qquad z \sim \pi_\theta(\cdot \mid x), \qquad r = R(z; x),$$

where $R(z; x)$ is the reward function.

[Baseline and advantage] Let $b(x)$ be any function of x only. Define the advantage

$$A(z; x) := R(z; x) - b(x).$$

A standard choice is the conditional-mean baseline $b(x) := \mathbb{E}[R(Z; x) \mid X = x]$. Then the advantage is zero-mean within each prompt:

$$\mathbb{E}[A(Z; x) \mid X = x] = \mathbb{E}[R(Z; x) \mid X = x] - b(x) = 0.$$

[Score function] Define the score function

$$s(z; x) := \nabla_\theta \log \pi_\theta(z \mid x).$$

It satisfies the normalization identity

$$\mathbb{E}_{z \sim \pi_\theta(\cdot|x)}[\, s(z; x)\,] = \nabla_\theta \int \pi_\theta(z \mid x)\, dz = 0.$$

[Within-prompt reward variance] We quantify within-prompt variation of observed rewards across rollouts by

$$\mathrm{RV}(x) := \mathrm{Var}(R \mid X = x), \qquad Z \sim \pi_\theta(\cdot \mid x).$$

Low $\mathrm{RV}(x)$ implies rewards are nearly constant within the prompt, so rollouts are weakly distinguishable by the reward signal. High $\mathrm{RV}(x)$ indicates large within-prompt variation of observed rewards which may arise from trajectory-dependent signal or evaluation noise.

D.2 Entropy and mutual information

[Conditional entropy] The within-input variability of reasoning is measured by

$$H(Z \mid X) := \mathbb{E}_{x \sim P(X)}\big[H(Z \mid X = x)\big] = -\mathbb{E}_{x \sim P(X),\, z \sim \pi_\theta(\cdot|x)}\big[\log \pi_\theta(z \mid x)\big].$$

The cross-input dependence of reasoning is measured by

$$I(X; Z) := \mathbb{E}_{x \sim P(X),\, z \sim \pi_\theta(\cdot|x)}\left[\log \frac{\pi_\theta(z \mid x)}{p_\theta(z)}\right], \qquad p_\theta(z) := \mathbb{E}_{x \sim P(X)}\big[\pi_\theta(z \mid x)\big].$$

Equivalently, $I(X; Z) = \mathbb{E}_{x \sim P(X)}\big[\mathrm{KL}(\pi_\theta(\cdot \mid x) \,\|\, p_\theta)\big]$.

Decomposition identity (Shannon quantities). For the true distribution induced by π_θ, the shannon identity

$$H(Z) \;=\; H(Z \mid X) \;+\; I(X; Z), \tag{1}$$

serves only as conceptual equation: it specifies the two components we aim to track (within-prompt variability and cross-prompt dependence). In practice we replace these Shannon quantities by scorer-defined proxies, e.g.,

$$\widehat{\mathcal{D}}_q := \widehat{\mathrm{NLL}}_q(Z \mid X) + \widehat{I}_q(X; Z),$$

which is in log-likelihood units under q and does not in general satisfy the Shannon identity unless q matches the evaluated distribution.

Interpretation for reasoning diversity. In our setting, Z is a proxy for a reasoning process (e.g., a chain-of-thought trajectory). A relative decrease in $H(Z \mid X)$ indicates within-prompt concentration of $\pi_\theta(\cdot \mid x)$ (entropy collapse). A relative decrease in $I(X; Z)$ indicates weakened input dependence, i.e., trajectories become less diagnostic of x. In our analysis, this can occur when reward-driven updates are weak (e.g., low $\mathrm{RV}(x)$) and the total update is dominated by *reward-agnostic* components (e.g., KL/entropy regularizers). We therefore track these two axes separately; in experiments we use scorer-defined proxies for $H(Z \mid X)$ and $I(X; Z)$.

E SCORER-BASED PROXIES FOR REASONING DIVERSITY

E.1 SETUP AND NOTATION

We define scorer-based proxies using a fixed collection of prompts and multiple rollouts per prompt. Throughout this appendix, the scorer q is fixed and used for evaluation.

[Prompts, rollouts, and prompt groups] Sample N prompts $\{x_i\}_{i=1}^{N} \sim P(X)$. For each prompt x_i, sample K trajectories

$$z_{i,k} \sim \pi_\theta(\cdot \mid x_i), \qquad k = 1, \ldots, K.$$

We refer to the set $\{z_{i,k}\}_{k=1}^{K}$ as a *prompt group*.

[Teacher-forced scorer and matched-pair score] Let q be a fixed language model used to score how compatible a trajectory z is with a prompt x. Define the matched-pair score

$$\ell_i(z) \;:=\; \log q(z \mid x_i).$$

All proxies in this appendix are built from $\ell_i(z)$ and therefore are measured in log-likelihood units under q.

[Mixture score across prompts] We evaluate each trajectory z under all prompts $\{x_j\}_{j=1}^{N}$ and define the mixture score

$$\ell_{\mathrm{mix}}(z) \;:=\; \log\left(\frac{1}{N}\sum_{j=1}^{N}\exp(\ell_j(z))\right) = \log\left(\frac{1}{N}\sum_{j=1}^{N} q(z \mid x_j)\right).$$

This is the log-likelihood of z under the uniform mixture over prompts induced by q. Equivalently, $\ell_{\mathrm{mix}}(z) = \log\big(\frac{1}{N}\sum_{j=1}^{N} q(z \mid x_j)\big)$ is the log-probability of z under the empirical prompt mixture.

The quantities defined above depend on the sampled prompt set $\{x_i\}_{i=1}^{N}$ and on the fixed scorer q. They are proxies for within-prompt variability and input dependence of trajectories, and should not be interpreted as exact Shannon entropies or mutual information unless q matches the evaluated conditional distribution.

F FORMAL DEFINITION OF THE FILTERING OPERATOR

[Filtering operator] Let $\mathcal{B}$ be a minibatch of samples. A *filtering operator* is specified by:

(i) Grouping key. A grouping function $g : \mathcal{B} \to \mathcal{G}$ that assigns each sample $\xi \in \mathcal{B}$ a group label

$$u = g(\xi).$$

For $u \in \mathcal{G}$, define the induced group subset

$$\mathcal{B}_u := \{\xi \in \mathcal{B} : g(\xi) = u\}.$$

(ii) Group statistic. A statistic $\phi : \mathcal{G} \to \mathbb{R}$ such that $\phi(u)$ depends only on the samples in $\mathcal{B}_u$.

(iii) Selection rule (mask). Given a threshold $\tau \in \mathbb{R}$, the binary mask is

$$m(u) := \mathbf{1}\{\phi(u) \geq \tau\}.$$

(iv) Filtered objective. For a per-sample RL loss $L_\theta(\xi)$, the filtered objective is

$$\mathcal{L}_{\mathrm{filt}}(\theta) \;=\; \frac{1}{|\mathcal{B}|}\sum_{\xi \in \mathcal{B}} m\big(g(\xi)\big)\, L_\theta(\xi).$$

Remark (post-sampling). Filtering is applied after sampling and only masks gradients; it does not change the rollout distribution.

Remark (normalization). In practice one may normalize by the number of kept samples or kept groups (instead of $|\mathcal{B}|$), which rescales the gradient but does not change which samples contribute nonzero gradients.

G RV Controls Task-Signal Magnitude and SNR

G.1 Setup

We use the policy/score/baseline/advantage notation from Appendix D.

In particular, for a fixed prompt x we write $z \sim \pi_\theta(\cdot \mid x)$, $s(z;x) = \nabla_\theta \log \pi_\theta(z \mid x)$, $A(z;x) = R(z;x) - b(x)$ with $b(x) = \mathbb{E}[R \mid X = x]$, and $\mathrm{RV}(x) = \mathrm{Var}(R \mid X = x) = \mathbb{E}[A^2 \mid X = x]$.

G.2 Assumption

The observed reward admits a decomposition

$$R(z;x) = \mu(x,z) + \varepsilon, \qquad \mu(x,z) := \mathbb{E}[R(z;x) \mid x, z],$$

where $\mu(x,z)$ is the trajectory-dependent mean reward and ε is a zero-mean noise term satisfying

$$\mathbb{E}[\varepsilon \mid x, z] = 0, \qquad \mathrm{Var}(\varepsilon \mid x, z) = \sigma^2(x) \geq 0.$$

Moreover, the score $s(z;x) = \nabla_\theta \log \pi_\theta(z \mid x)$ is a deterministic (measurable) function of (x,z).

G.3 Task-gradient magnitude is RV-controlled

[Task gradient magnitude is RV-controlled] Assume the baseline is the conditional mean $b(x) = \mathbb{E}[R \mid X = x]$. and $g_{\mathrm{task}}(x) := \mathbb{E}[A(z;x)\, s(z;x) \mid X = x]$. Then

$$\|g_{\mathrm{task}}(x)\| \;\leq\; \sqrt{\mathrm{RV}(x)}\, \sqrt{\mathbb{E}[\|s(z;x)\|^2 \mid X = x]}.$$

Proof. Fix a prompt x and take randomness over $z \sim \pi_\theta(\cdot \mid x)$. For brevity write $A := A(z;x)$ and $s := s(z;x)$. Then

$$g_{\mathrm{task}}(x) = \mathbb{E}[A\, s \mid X = x].$$

For any unit vector $u \in \mathbb{R}^d$ with $\|u\| = 1$,

$$\big|\langle u, g_{\mathrm{task}}(x)\rangle\big| = \big|\mathbb{E}[A\, \langle u, s\rangle \mid X = x]\big| \leq \sqrt{\mathbb{E}[A^2 \mid X = x]}\, \sqrt{\mathbb{E}[\langle u, s\rangle^2 \mid X = x]},$$

where the inequality is Cauchy-Schwarz. Moreover, $\langle u, s\rangle^2 \leq \|u\|^2 \|s\|^2 = \|s\|^2$, hence

$$\big|\langle u, g_{\mathrm{task}}(x)\rangle\big| \leq \sqrt{\mathbb{E}[A^2 \mid X = x]}\, \sqrt{\mathbb{E}[\|s\|^2 \mid X = x]}.$$

Taking the supremum over all unit vectors u yields

$$\|g_{\mathrm{task}}(x)\| \leq \sqrt{\mathbb{E}[A^2 \mid X = x]}\, \sqrt{\mathbb{E}[\|s\|^2 \mid X = x]}.$$

Finally, with $b(x) = \mathbb{E}[R \mid X = x]$ we have $\mathbb{E}[A \mid X = x] = 0$ and thus

$$\mathbb{E}[A^2 \mid X = x] = \mathrm{Var}(R \mid X = x) = \mathrm{RV}(x).$$

Substituting completes the proof. □

G.4 SNR is upper bounded by RV and reward noise

[SNR upper bound by RV and noise] Let $\widehat{g}_{\mathrm{task}}(x)$ be the K-sample Monte Carlo estimator

$$\widehat{g}_{\mathrm{task}}(x) := \frac{1}{K} \sum_{k=1}^{K} A_k\, s_k, \qquad A_k := A(z_k;x),\; s_k := s(z_k;x),$$

with $z_1, \ldots, z_K \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot \mid x)$. Define

$$\mathrm{SNR}(x) := \frac{\|g_{\mathrm{task}}(x)\|}{\sqrt{\mathbb{E}\big[\|\widehat{g}_{\mathrm{task}}(x) - g_{\mathrm{task}}(x)\|^2 \mid X = x\big]}}.$$

Under Assumption G.2,

$$\mathrm{SNR}(x) \;\le\; \sqrt{K}\cdot\frac{\sqrt{\mathrm{RV}(x)}}{\sigma(x)}.$$

If $\sigma(x)=0$, the bound is vacuous.

Proof. Fix a prompt x. Let $z_1,\dots,z_K \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot\mid x)$ and write

$$\widehat{g} \;=\; \frac{1}{K}\sum_{k=1}^{K} A_k s_k, \qquad g \;=\; \mathbb{E}[As\mid x],$$

where $(A_k,s_k)=(A(z_k;x),s(z_k;x))$ and $(A,s)=(A(z;x),s(z;x))$ for $z\sim\pi_\theta(\cdot\mid x)$.

Let $Y_k := A_k s_k$. Then $\widehat{g}=\frac{1}{K}\sum_{k=1}^{K} Y_k$ and $g=\mathbb{E}[Y_1\mid x]$, hence

$$\widehat{g}-g=\frac{1}{K}\sum_{k=1}^{K}(Y_k-g).$$

Using i.i.d. conditional on x,

$$\begin{aligned}
\mathbb{E}\big[\|\widehat{g}-g\|^2\mid x\big] &= \frac{1}{K^2}\mathbb{E}\left[\left\|\sum_{k=1}^{K}(Y_k-g)\right\|^2 \;\middle|\; x\right]\\
&= \frac{1}{K^2}\sum_{k=1}^{K}\mathbb{E}\big[\|Y_k-g\|^2\mid x\big] + \frac{1}{K^2}\sum_{k\neq\ell}\mathbb{E}[\langle Y_k-g,\;Y_\ell-g\rangle\mid x]\\
&= \frac{1}{K^2}\sum_{k=1}^{K}\mathbb{E}\big[\|Y_k-g\|^2\mid x\big]\\
&= \frac{1}{K}\,\mathbb{E}\big[\|As-g\|^2\mid x\big]\,.
\end{aligned}$$

Under Assumption G.2, write $R=\mu+\varepsilon$ with $\mu(x,z)=\mathbb{E}[R\mid x,z]$. Since $b(x)=\mathbb{E}[R\mid x]=\mathbb{E}[\mu\mid x]$,

$$A \;=\; R-b(x) \;=\; (\mu-\mathbb{E}[\mu\mid x])+\varepsilon \;=:\; A_\mu+\varepsilon.$$

Using $A=A_\mu+\varepsilon$,

$$As-g=(A_\mu s-g)+\varepsilon s,$$

so

$$\|As-g\|^2=\|A_\mu s-g\|^2+\|\varepsilon s\|^2+2\langle A_\mu s-g,\;\varepsilon s\rangle.$$

Moreover,

$$\mathbb{E}[\langle A_\mu s-g,\;\varepsilon s\rangle\mid x]=\mathbb{E}\Big[\mathbb{E}[\langle A_\mu s-g,\;\varepsilon s\rangle\mid x,z]\;\Big|\;x\Big]=\mathbb{E}\Big[\langle A_\mu s-g,\;s\rangle\,\mathbb{E}[\varepsilon\mid x,z]\;\Big|\;x\Big]=0,$$

hence

$$\mathbb{E}\big[\|As-g\|^2\mid x\big] \;\ge\; \mathbb{E}\big[\|\varepsilon s\|^2\mid x\big]\,.$$

Combining with Step 1,

$$\mathbb{E}\big[\|\widehat{g}-g\|^2\mid x\big] \;\ge\; \frac{1}{K}\,\mathbb{E}\big[\|\varepsilon s\|^2\mid x\big]\,.$$

Since $\|\varepsilon s\|^2=\varepsilon^2\|s\|^2$ and s is measurable given (x,z),

$$\begin{aligned}
\mathbb{E}\big[\|\varepsilon s\|^2\mid x\big] &= \mathbb{E}\Big[\mathbb{E}\big[\varepsilon^2\|s\|^2\mid x,z\big]\;\Big|\;x\Big]\\
&= \mathbb{E}\Big[\|s\|^2\,\mathbb{E}\big[\varepsilon^2\mid x,z\big]\;\Big|\;x\Big]\\
&= \mathbb{E}\Big[\|s\|^2\,\sigma^2(x)\;\Big|\;x\Big]=\sigma^2(x)\,\mathbb{E}\big[\|s\|^2\mid x\big]\,,
\end{aligned}$$

where $\mathbb{E}[\varepsilon^2 \mid x, z] = \mathrm{Var}(\varepsilon \mid x, z) = \sigma^2(x)$ by Assumption G.2. Therefore

$$\mathbb{E}\big[\|\widehat{g} - g\|^2 \mid x\big] \;\geq\; \frac{1}{K}\,\sigma^2(x)\,\mathbb{E}\big[\|s\|^2 \mid x\big]\,.$$

By Theorem G.3,

$$\|g\| = \|\mathbb{E}[As \mid x]\| \leq \sqrt{\mathrm{RV}(x)}\,\sqrt{\mathbb{E}[\|s\|^2 \mid x]}.$$

Thus, with $\mathrm{SNR}(x) := \frac{\|g\|}{\sqrt{\mathbb{E}[\|\widehat{g}-g\|^2|x]}}$,

$$\mathrm{SNR}(x) \leq \frac{\sqrt{\mathrm{RV}(x)}\sqrt{\mathbb{E}[\|s\|^2 \mid x]}}{\sqrt{\frac{1}{K}\sigma^2(x)\mathbb{E}[\|s\|^2 \mid x]}} = \sqrt{K}\cdot\frac{\sqrt{\mathrm{RV}(x)}}{\sigma(x)}.$$ □

G.5 Low-SNR updates induce parameter drift

[Illustrative random-walk drift under zero-mean noise]

Consider SGD-style updates

$$\theta_{t+1} = \theta_t + \eta\,\xi_t,$$

where $\{\xi_t\}_{t\geq 0}$ are independent, $\mathbb{E}[\xi_t] = 0$, and $\mathbb{E}[\|\xi_t\|^2] = v < \infty$ for all t. Then for any $T \geq 1$,

$$\mathbb{E}\big[\|\theta_T - \theta_0\|^2\big] = \eta^2\,T\,v.$$

Proof. Unrolling the recursion yields

$$\theta_T - \theta_0 = \eta \sum_{t=0}^{T-1} \xi_t.$$

Therefore,

$$\|\theta_T - \theta_0\|^2 = \eta^2 \left\|\sum_{t=0}^{T-1} \xi_t\right\|^2 = \eta^2 \left(\sum_{t=0}^{T-1} \|\xi_t\|^2 + 2 \sum_{0\leq i<j\leq T-1} \langle \xi_i, \xi_j\rangle\right).$$

Taking expectation and using independence with $\mathbb{E}[\xi_t] = 0$,

$$\mathbb{E}\langle \xi_i, \xi_j\rangle = \langle \mathbb{E}[\xi_i],\, \mathbb{E}[\xi_j]\rangle = 0, \qquad i \neq j.$$

Hence the cross terms vanish and

$$\mathbb{E}\big[\|\theta_T - \theta_0\|^2\big] = \eta^2 \sum_{t=0}^{T-1} \mathbb{E}[\|\xi_t\|^2] = \eta^2\,T\,v,$$

where we used $\mathbb{E}[\|\xi_t\|^2] = v$ for all t. □

H Template Mixing Reduces Input Dependence

[Template mixing contracts mutual information] Let $X \sim P(X)$ and $Z \mid X = x \sim p(z \mid x)$ with marginal $p(z) = \mathbb{E}_{x\sim P}[p(z \mid x)]$. Fix any prompt-independent distribution $q(z)$. For $\alpha \in [0, 1]$, define the mixed conditional and marginal

$$p_\alpha(z \mid x) := (1-\alpha)p(z \mid x) + \alpha q(z), \qquad p_\alpha(z) := (1-\alpha)p(z) + \alpha q(z).$$

Let $I_\alpha(X; Z)$ denote the mutual information under $p_\alpha(x, z) = P(x)p_\alpha(z \mid x)$. Then

$$I_\alpha(X; Z) \;\leq\; (1-\alpha)\,I(X; Z).$$

Proof. For any fixed x,

$$\begin{aligned}\mathrm{KL}\big(p_\alpha(\cdot \mid x) \,\|\, p_\alpha(\cdot)\big) &= \mathbb{E}_{z\sim p_\alpha(\cdot|x)}\left[\log \frac{p_\alpha(z \mid x)}{p_\alpha(z)}\right] \\ &= \mathbb{E}_{z\sim p_\alpha(\cdot|x)}\big[\log p_\alpha(z \mid x) - \log p_\alpha(z)\big].\end{aligned}$$

Taking expectation over $x \sim p(x)$ gives

$$I_\alpha(X;Z) = \mathbb{E}_x\Big[\mathrm{KL}\big(p_\alpha(\cdot \mid x) \,\|\, p_\alpha(\cdot)\big)\Big].$$

The same identity holds for $I(X;Z)$ with p_α replaced by p.

By joint convexity of $\mathrm{KL}(\cdot\|\cdot)$, for any distributions a, b, c, d and any $\alpha \in [0,1]$,

$$\mathrm{KL}\big((1-\alpha)a + \alpha b \,\|\, (1-\alpha)c + \alpha d\big) \le (1-\alpha)\mathrm{KL}(a\|c) + \alpha\,\mathrm{KL}(b\|d).$$

Let $a = p(\cdot \mid x)$, $b = q$, $c = p(\cdot)$, and $d = q$. Since

$$p_\alpha(\cdot \mid x) = (1-\alpha)p(\cdot \mid x) + \alpha q, \qquad p_\alpha(\cdot) = (1-\alpha)p(\cdot) + \alpha q,$$

we obtain

$$\begin{aligned}\mathrm{KL}\big(p_\alpha(\cdot \mid x) \,\|\, p_\alpha(\cdot)\big) &\le (1-\alpha)\mathrm{KL}\big(p(\cdot \mid x) \,\|\, p(\cdot)\big) + \alpha\,\mathrm{KL}(q\|q) \\ &= (1-\alpha)\mathrm{KL}\big(p(\cdot \mid x) \,\|\, p(\cdot)\big).\end{aligned}$$

Averaging over $x \sim p(x)$ yields

$$\mathbb{E}_x\Big[\mathrm{KL}\big(p_\alpha(\cdot \mid x) \,\|\, p_\alpha(\cdot)\big)\Big] \le (1-\alpha)\,\mathbb{E}_x\Big[\mathrm{KL}\big(p(\cdot \mid x) \,\|\, p(\cdot)\big)\Big].$$

Using the identity $I(X;Z) = \mathbb{E}_x\big[\mathrm{KL}(p(\cdot \mid x) \,\|\, p(\cdot))\big]$ (and the analogous one for I_α), we obtain

$$I_\alpha(X;Z) \le (1-\alpha)\, I(X;Z),$$

which proves the lemma. □

I Filtering Reduces Gradient-Estimation MSE

I.1 Setup

Consider N groups indexed by $i \in \{1, \dots, N\}$. Group i contains K rollouts, and $\widehat{g}_i \in \mathbb{R}^d$ denotes the *group-level* gradient estimator (already averaged over the K rollouts in the group). We model

$$\widehat{g}_i \;=\; g_i + \varepsilon_i, \qquad \mathbb{E}[\varepsilon_i] = 0, \qquad \mathbb{E}\|\varepsilon_i\|^2 = \sigma_i^2,$$

where $\{\varepsilon_i\}_{i=1}^N$ are independent across groups. For a kept set S of groups, we write $n := |S|$ for the number of kept groups.

I.2 Unfiltered vs. filtered estimators

Define the unfiltered batch estimator and its mean:

$$\widehat{G} := \frac{1}{N}\sum_{i=1}^N \widehat{g}_i, \qquad G := \frac{1}{N}\sum_{i=1}^N g_i.$$

Let $S \subseteq \{1, \dots, N\}$ be the set of kept groups with $|S| = n$. Define the filtered estimator and its mean:

$$\widehat{G}_S := \frac{1}{n}\sum_{i\in S} \widehat{g}_i, \qquad G_S := \frac{1}{n}\sum_{i\in S} g_i.$$

[MSE of the filtered estimator] $\widehat{G}_S$ is unbiased for G_S and satisfies

$$\mathbb{E}\big\|\widehat{G}_S - G_S\big\|^2 \;=\; \frac{1}{n^2}\sum_{i\in S}\sigma_i^2.$$

Proof. By the setup, $\widehat{g}_i = g_i + \varepsilon_i$ with $\mathbb{E}[\varepsilon_i] = 0$, hence

$$\mathbb{E}[\widehat{g}_i] = g_i.$$

Therefore,

$$\mathbb{E}[\widehat{G}_S] = \frac{1}{n}\sum_{i\in S}\mathbb{E}[\widehat{g}_i] = \frac{1}{n}\sum_{i\in S} g_i = G_S.$$

Moreover,

$$\widehat{G}_S - G_S = \frac{1}{n}\sum_{i\in S}(\widehat{g}_i - g_i) = \frac{1}{n}\sum_{i\in S}\varepsilon_i.$$

Therefore,

$$\begin{aligned}\mathbb{E}\big\|\widehat{G}_S - G_S\big\|^2 &= \frac{1}{n^2}\,\mathbb{E}\left\|\sum_{i\in S}\varepsilon_i\right\|^2 \\ &= \frac{1}{n^2}\left(\sum_{i\in S}\mathbb{E}\|\varepsilon_i\|^2 + \sum_{\substack{i,j\in S\\ i\neq j}}\mathbb{E}\langle\varepsilon_i,\varepsilon_j\rangle\right).\end{aligned}$$

By independence and $\mathbb{E}[\varepsilon_i] = 0$, for $i \neq j$ we have

$$\mathbb{E}\langle\varepsilon_i,\varepsilon_j\rangle = \langle\mathbb{E}[\varepsilon_i],\,\mathbb{E}[\varepsilon_j]\rangle = 0,$$

so the cross terms vanish. Hence

$$\mathbb{E}\big\|\widehat{G}_S - G_S\big\|^2 = \frac{1}{n^2}\sum_{i\in S}\mathbb{E}\|\varepsilon_i\|^2 = \frac{1}{n^2}\sum_{i\in S}\sigma_i^2.$$

□

Remark (bias relative to the original objective). While $\widehat{G}_S$ is unbiased for the *filtered* mean gradient G_S, it is generally biased for the *unfiltered* mean gradient G unless S is chosen independently of $\{g_i\}$ or g_i is constant across groups.

J Reward-Agnostic Regularizers and Update Dominance

J.1 Setup

Similarly, fix a prompt x and consider trajectories $z \sim \pi_\theta(\cdot \mid x)$ with reward $R(z;x)$ and baseline $b(x)$. Define the reward-driven (task) gradient

$$g_{\text{task}}(x) \;:=\; \mathbb{E}[(R(z;x) - b(x))\, s(z;x) \mid X = x]\,, \qquad s(z;x) := \nabla_\theta \log \pi_\theta(z \mid x).$$

Let $g_{\text{reg}}(x)$ denote an update component that is computed without multiplying the reward (or advantage), e.g.,

$$g_{\text{reg}}(x) := \lambda_{\text{KL}}\, g_{\text{KL}}(x) + \lambda_{\text{ent}}\, g_{\text{ent}}(x),$$

where $g_{\text{KL}}(x)$ and $g_{\text{ent}}(x)$ are gradients of prompt-level distributional regularizers. We write the total expected update as

$$g_{\text{total}}(x) = g_{\text{task}}(x) + g_{\text{reg}}(x).$$

To summarize relative influence, define the dominance ratio

$$\rho(x) := \frac{\|g_{\text{reg}}(x)\|}{\|g_{\text{task}}(x)\| + \|g_{\text{reg}}(x)\|} \in [0,1].$$

We refer to $g_{\text{reg}}(x)$ as *reward-agnostic* since it does not use within-prompt reward differences to weight trajectories.

J.2 Low-RV prompts amplify regularizer influence

By Theorem G.3, for any prompt x,

$$\|g_{\text{task}}(x)\| \le \sqrt{\mathrm{RV}(x)}\ \sqrt{\mathbb{E}[\|s\|^2 \mid X = x]}.$$

Therefore the dominance ratio

$$\rho(x) = \frac{\|g_{\text{reg}}(x)\|}{\|g_{\text{task}}(x)\| + \|g_{\text{reg}}(x)\|}$$

admits the lower bound

$$\rho(x) \ \ge\ \frac{\|g_{\text{reg}}(x)\|}{\|g_{\text{reg}}(x)\| + \sqrt{\mathrm{RV}(x)}\sqrt{\mathbb{E}[\|s\|^2 \mid X = x]}}.$$

In particular, if $\|g_{\text{reg}}(x)\|$ and $\mathbb{E}[\|s\|^2 \mid X = x]$ vary slowly across prompts compared to $\mathrm{RV}(x)$, then smaller $\mathrm{RV}(x)$ implies larger $\rho(x)$, i.e., the total update is more strongly shaped by reward-agnostic regularizers on low-RV prompts.

K KL-Closeness to the Base Implies MI-Closeness

To avoid measure-theoretic issues, assume X is supported on a finite set $\mathcal{X}$ and Z takes values in a finite set $\mathcal{Z}$. Let $P(X)$ be the prompt distribution and define

$$P_\theta(x, z) := P(x)\pi_\theta(z \mid x), \qquad P_0(x, z) := P(x)\pi_0(z \mid x).$$

If

$$\sup_{x \in \mathcal{X}} \mathrm{KL}(\pi_\theta(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x)) \le \varepsilon,$$

then there exists $f(\varepsilon) \to 0$ as $\varepsilon \to 0$ such that

$$\big|I_\theta(X; Z) - I_0(X; Z)\big| \le f(\varepsilon).$$

Proof. By the chain rule for KL divergence,

$$\mathrm{KL}(P_\theta(X, Z) \,\|\, P_0(X, Z)) = \mathbb{E}_{x \sim P}[\mathrm{KL}(\pi_\theta(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x))]\,.$$

Under the assumption $\sup_{x \in \mathcal{X}} \mathrm{KL}(\pi_\theta(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x)) \le \varepsilon$, we obtain

$$\mathrm{KL}(P_\theta(X, Z) \,\|\, P_0(X, Z)) \le \varepsilon.$$

By Pinsker's inequality,

$$\|P_\theta(X, Z) - P_0(X, Z)\|_{\mathrm{TV}} \le \sqrt{\tfrac{1}{2}\,\mathrm{KL}(P_\theta(X, Z) \,\|\, P_0(X, Z))} \le \sqrt{\tfrac{\varepsilon}{2}} =: \delta.$$

Since $\|P_\theta(X, Z) - P_0(X, Z)\|_{\mathrm{TV}} \le \delta$ and (X, Z) takes values in a finite alphabet $\mathcal{X} \times \mathcal{Z}$, the Fannes-Audenaert inequality implies

$$\big|H_\theta(X, Z) - H_0(X, Z)\big| \le \delta \log(|\mathcal{X}||\mathcal{Z}| - 1) + h_2(\delta),$$

where $H_\theta(\cdot)$ denotes entropy under P_θ, and $h_2(\cdot)$ is the binary entropy. Moreover, total variation does not increase under marginalization, so

$$\|P_\theta(Z) - P_0(Z)\|_{\mathrm{TV}} \le \delta,$$

and applying Fannes-Audenaert on the alphabet $\mathcal{Z}$ yields

$$\big|H_\theta(Z) - H_0(Z)\big| \le \delta \log(|\mathcal{Z}| - 1) + h_2(\delta) \le \delta \log(|\mathcal{X}||\mathcal{Z}| - 1) + h_2(\delta).$$

Finally, using $I(X; Z) = H(X) + H(Z) - H(X, Z)$ and noting that $P_\theta(X) = P_0(X) = P(X)$ (hence $H_\theta(X) = H_0(X)$),

$$\begin{aligned}
\big|I_\theta(X; Z) - I_0(X; Z)\big| &= \big|(H_\theta(Z) - H_0(Z)) - (H_\theta(X, Z) - H_0(X, Z))\big| \\
&\le \big|H_\theta(Z) - H_0(Z)\big| + \big|H_\theta(X, Z) - H_0(X, Z)\big| \\
&\le 2\Big(\delta \log(|\mathcal{X}||\mathcal{Z}| - 1) + h_2(\delta)\Big).
\end{aligned}$$

Thus we may take

$$f(\varepsilon) := 2\Big(\delta \log(|\mathcal{X}||\mathcal{Z}| - 1) + h_2(\delta)\Big), \qquad \delta := \sqrt{\tfrac{\varepsilon}{2}},$$

which satisfies $f(\varepsilon) \to 0$ as $\varepsilon \to 0$. □

L Decomposing Changes in Input Dependence

[Entropy changes] Let X be prompts and let $Z \sim \pi_\theta(\cdot \mid X)$ under the current policy, with reference policy π_0. Define the conditional-entropy and marginal-entropy changes

$$\Delta_{\text{in}} := H_\theta(Z \mid X) - H_0(Z \mid X), \qquad \Delta_{\text{marg}} := H_\theta(Z) - H_0(Z).$$

With Δ_{in} and Δ_{marg} defined above,

$$I_\theta(X;Z) - I_0(X;Z) = \Delta_{\text{marg}} - \Delta_{\text{in}}.$$

In particular, if $\Delta_{\text{in}} \geq \Delta_{\text{marg}} + \gamma$ for some $\gamma > 0$, then

$$I_\theta(X;Z) \leq I_0(X;Z) - \gamma,$$

and especially $I_\theta(X;Z) < I_0(X;Z)$ whenever $\Delta_{\text{in}} > \Delta_{\text{marg}}$.

Proof. Using $I(X;Z) = H(Z) - H(Z \mid X)$,

$$\begin{aligned} I_\theta(X;Z) - I_0(X;Z) &= \big(H_\theta(Z) - H_0(Z)\big) - \big(H_\theta(Z \mid X) - H_0(Z \mid X)\big) \\ &= \Delta_{\text{marg}} - \Delta_{\text{in}}. \end{aligned}$$

The sufficient-condition statements follow by rearranging the inequality. □

An entropy bonus acts directly on the per-prompt dispersion and increases $H_\theta(Z \mid X)$, but it does not explicitly encourage cross-prompt separation that would increase the marginal entropy $H_\theta(Z)$ by a comparable amount. Hence it is plausible that Δ_{in} exceeds Δ_{marg}, in which case Theorem L implies $I_\theta(X;Z)$ decreases.

Appendix J explains that when $\mathrm{RV}(x)$ is small, the task update can be weak, so reward-agnostic regularizers can have larger relative influence on the total update.

M GRPO Normalization Amplifies Noise at Low RV

GRPO-style normalization divides the advantage by $\sqrt{\mathrm{RV}(x)}$, which induces a $\mathrm{RV}(x)^{-1}$ noise amplification in the mean-squared error of the per-prompt gradient estimator.

For a fixed prompt x, define the normalized advantage

$$\widetilde{A}(z;x) := \frac{A(z;x)}{\sqrt{\mathrm{RV}(x)}}, \qquad A(z;x) := R(z;x) - b(x), \qquad b(x) := \mathbb{E}_{z\sim\pi_\theta(\cdot|x)}[R(z;x)].$$

Given K i.i.d. rollouts $z_1, \ldots, z_K \sim \pi_\theta(\cdot \mid x)$, define

$$\widehat{g}_{\text{GRPO}}(x) := \frac{1}{K}\sum_{k=1}^{K} \widetilde{A}_k\, s_k, \qquad g_{\text{GRPO}}(x) := \mathbb{E}[\widetilde{A}\, s \mid X = x],$$

where $s_k = \nabla_\theta \log \pi_\theta(z_k \mid x)$.

Under the reward decomposition in Appendix G (under Assumption G.2), the GRPO estimator satisfies

$$\mathbb{E}\Big[\big\|\widehat{g}_{\text{GRPO}}(x) - g_{\text{GRPO}}(x)\big\|^2 \mid X = x\Big] \;\geq\; \frac{1}{K}\cdot\frac{\sigma^2(x)}{\mathrm{RV}(x)}\,\mathbb{E}[\|s\|^2 \mid X = x].$$

If $\sigma(x) = 0$ (deterministic rewards given (x, z)), the lower bound is zero and thus vacuous. This bound makes explicit that smaller $\mathrm{RV}(x)$ yields a larger variance floor for the normalized estimator if all other factors are the same.

N Wallclock Performance

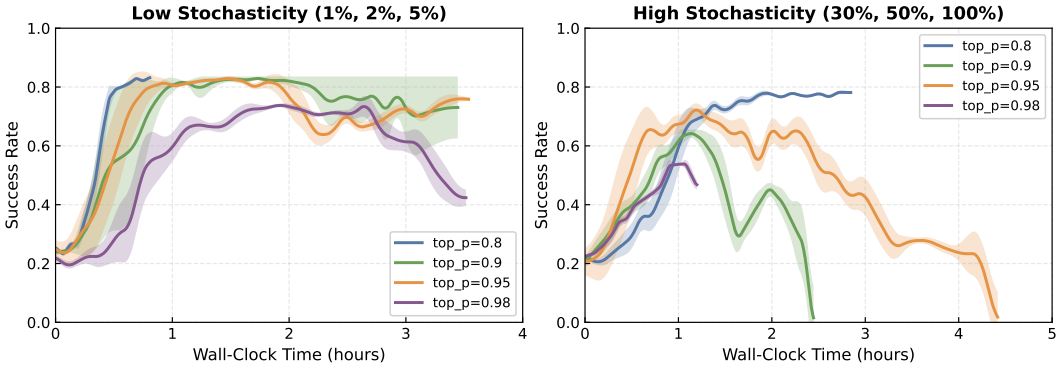

Figure 6: Wall-clock efficiency of RV filtering under low- vs. high-stochasticity environments (left: 1/2/5%; right: 30/50/100%). We fix the update budget to 200 training steps and report *training-only* wall-clock time (evaluation excluded), sweeping the filter keep rate (implemented via different top_p thresholds) while keeping all other settings unchanged. Despite discarding samples, filtering does not hurt sample efficiency; instead, it improves wall-clock efficiency by skipping noisy updates. In both regimes, stronger filtering reaches higher success rates earlier in time, indicating that removing low-quality updates both saves compute and reduces drift from noisy gradients. The best-performing keep rate shifts toward more aggressive filtering as the time budget increases, consistent with "filter more when you can train longer".