RAGEN-v2: Understanding Reasoning Collapse in LLM Agent Reinforcement Learning

Zihan Wang[†*,1], Chi Gui[†,2], Xing Jin[†,3], Qineng Wang[†,1], Licheng Liu[†,4],Kangrui Wang[1], Shiqi Chen[5], Linjie Li[6], Zhengyuan Yang[7], Pingyue Zhang[1], Yiping Lu[1], Jiajun Wu[8], Li Fei-Fei[8], Lijuan Wang[7], Yejin Choi[8], Manling Li[1]

[†]**Core contributors.** [*]**Project lead.**

[1]**Northwestern University** [2]**UIUC** [3]**University of British Columbia** [4]**Imperial College London**
[5]**Oxford University** [6]**University of Washington** [7]**Microsoft** [8]**Stanford University**

https://ragen-ai.github.io/v2/

RL training of multi-turn LLM agents is inherently unstable, and reasoning quality directly determines task performance. Entropy is widely used to track reasoning stability. However, entropy only measures diversity within the same input, and cannot tell whether reasoning actually responds to different inputs. We find that even with stable entropy, models can rely on fixed templates that look diverse but are input-agnostic. We call this **template collapse**, a failure mode invisible to entropy and all existing metrics. To diagnose this failure, we decompose reasoning quality into **within-input diversity** (Entropy) and **cross-input distinguishability** (Mutual Information, MI), and introduce a family of mutual information proxies for online diagnosis. Across diverse tasks, mutual information correlates with final performance much more strongly than entropy, making it a more reliable proxy for reasoning quality. We further explain template collapse with a *signal-to-noise ratio* (SNR) mechanism. Low reward variance weakens task gradients, letting regularization terms dominate and erase cross-input reasoning differences. To address this, we propose **SNR-Adaptive Filtering** to select high-signal prompts per iteration using reward variance as a lightweight proxy. Across planning, math reasoning, web navigation, and code execution, the method consistently improves both input dependence and task performance.

1. Introduction

Training multi-turn LLM agents with reinforcement learning (RL) is inherently challenging [34, 72, 68]. Researchers therefore monitor reward for **outcome stability** and entropy for **reasoning process stability** [42, 32, 62], treating both as stability indicators of RL training.

However, entropy can be an ambiguous signal to understand reasoning quality. When entropy decreases, it may simply reflect the model becoming more specialized and confident on the task, which is a natural outcome of RL optimization [68, 62]. When entropy remains high, reasoning can still drift toward fixed templates that appear diverse within any single input but are effectively the same across inputs (Figure 1). We call this **template collapse**, a failure mode

invisible to both metrics. This risk is especially acute in multi-turn settings: sparse rewards cannot distinguish input-driven reasoning from templated reasoning that merely happens to succeed [59, 60], and reasoning chains are hard to get directly supervised [44, 6]. As a result, template collapse can persist unnoticed during training, making agents unreliable and silently hurting their reasoning abilities.

To understand and mitigate template collapse, this paper addresses two questions. **(Q1) How to diagnose?** (§2) Entropy-based metrics [61, 63, 69] track within-input variability but miss input dependence across inputs, so they fail to detect template collapse. We propose a mutual information (MI) proxy [5] that scores each reasoning chain against all batch inputs to measure input dependence, without external models. **(Q2) Why does it happen?** (§3) We explain through a signal-to-noise ratio (SNR) lens. Task gradients draw signal from reward differences across within-input trajectories. Sampling noise and input-agnostic regularization (KL divergence and entropy regularization [42, 62]) dilute this signal. Low SNR lets noise dominate, erasing cross-input reasoning differences.

To address template collapse, based on the SNR view, we introduce **SNR-Adaptive Filtering**, which uses reward variance as a lightweight SNR proxy to select high-signal prompts each iteration, without additional supervision. Throughout training, the MI proxy monitors input dependence; across experiments, MI correlates with task performance significantly more strongly than entropy, validating it as a diagnostic for template collapse.

Together, they constitute a diagnostic framework for a systematic failure mode in multi-turn agent RL, validated across planning [39], mathematical reasoning [67, 16], web navigation, code execution, and tool use, under multiple RL algorithms, model scales, and modalities. SNR-Adaptive Filtering consistently improves input dependence and task performance, providing direct experimental support for the SNR mechanism.

Our contributions are summarized as follows:

1. **Identifying template collapse.** We find that template collapse occurs when reasoning appears diverse within inputs but becomes input-agnostic across inputs. We propose a mutual information proxy to detect it without external models.
2. **Explaining template collapse via SNR.** We show that low reward variance weakens task gradients while input-agnostic regularization remains constant, erasing input dependence. We provide gradient decomposition evidence across reward-variance buckets.
3. **SNR-Adaptive Filtering.** We propose filtering prompts by reward variance before each update. We demonstrate that this improves input dependence and performance across tasks, algorithms, scales, and modalities.

2. Template Collapse in Multi-turn Agent RL

2.1. Setup and Preliminaries

We study closed-loop multi-turn agent reinforcement learning [60], where a policy π_θ is trained by repeatedly rolling out trajectories under the current policy and environment and updating on the collected experience. At each time step t, the agent observes o_t, generates a response consisting of reasoning tokens z_t and an executable action a_t, and receives reward r_t, forming a trajectory $\tau = \{(o_t, z_t, a_t, r_t)\}_{t=1}^{T}$.

Two variables are central to our analysis. We use X to denote the full context available to the model immediately before generating reasoning at turn t: this comprises the system prompt, all

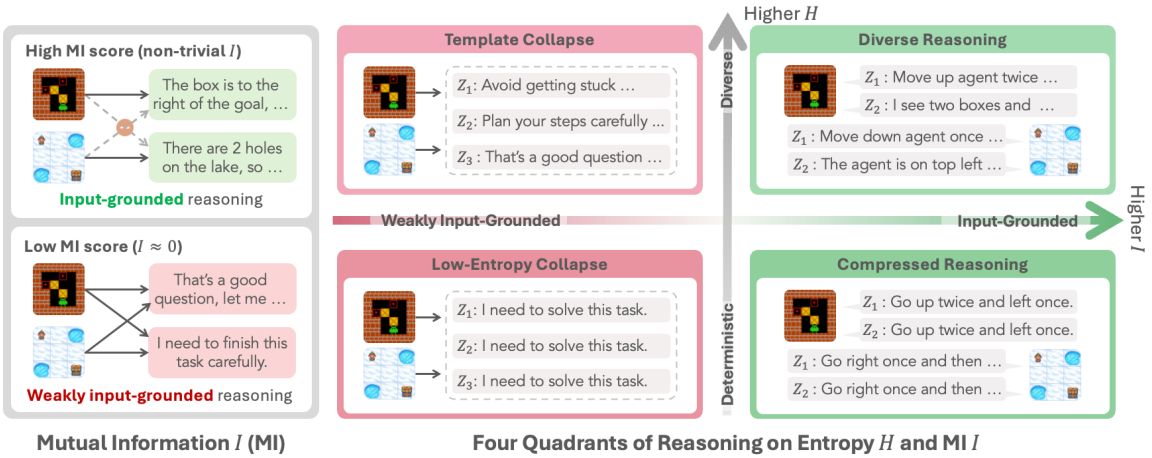

Figure 1 | Left: input-driven reasoning adapts to the current state; templated reasoning produces nearly identical responses across different inputs. Right: four reasoning regimes characterized along two axes: conditional entropy $H(Z \mid X)$ (within-input diversity) and mutual information $I(X;Z)$ (input dependence). Details in Section 2.

prior observations $o_{1:t}$, actions $a_{1:t-1}$, and reasoning tokens $z_{1:t-1}$. Each value of X is encountered during on-policy rollouts with the current policy interacting with the environment. We use Z to denote the reasoning token sequence the model generates for that turn, excluding action tokens and boundary markers (e.g., `</think>`).

The standard PPO/GRPO objective,

$$\mathcal{L}(\theta) = \mathbb{E}_{x,\tau}\big[A(\tau,x)\big] - \lambda_{\mathrm{KL}} D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}) + \lambda_H H(\pi_\theta),$$

where $A(\tau,x)$ is the advantage, contains regularization terms (KL divergence, entropy bonus) that act uniformly across all inputs regardless of their content. This input-agnostic character is central to the analysis that follows.

2.2. Rethinking Reasoning Collapse from an Information-Theoretic Lens

Why entropy is insufficient to measure reasoning quality? Researchers proxy process stability with entropy and outcome stability with reward, treating both as evidence of healthy training. Stable entropy, however, does not guarantee stable reasoning. Reasoning diversity (marginal entropy) $H(Z)$ decomposes via the standard identity [5]:

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

where $I(X;Z)$ is input dependence (**mutual information** between input X and reasoning Z), and $H(Z \mid X)$ is within-input diversity (**conditional entropy** of reasoning given input). Entropy metrics proxy $H(Z \mid X)$, but neither captures a decline in $I(X;Z)$: the policy can sustain high $H(Z \mid X)$ while $I(X;Z)$ drops to zero, producing diverse but input-agnostic boilerplate. We call this **template collapse**.

Reasoning regimes with a mutual information view. Figure 1 illustrates four reasoning states along these two axes: (i) *Diverse Reasoning* (high $H(Z \mid X)$, high $I(X;Z)$): the desired regime where reasoning is both varied within each input and systematically grounded across different inputs;

Table 1 | MI proxy family. All variants are derived from in-batch cross-scoring of reasoning traces against prompts, using matched (per-token log-prob under the true prompt) and marginal (per-token log-prob under the uniform prompt mixture) as base quantities. First-turn variants use only the first agent turn; trajectory variants sample across all turns.

Type	Proxy	Formula	Notes
Discrete	Retrieval-Acc	$\frac{1}{NK}\sum_{i,k}\mathbf{1}[\arg\max_j \mathbf{L}_{i,k,j}=i]$	Chance level $1/N$ under template collapse
	Recall@k	$\frac{1}{NK}\sum_{i,k}\mathbf{1}[i\in \text{top-}k_j(\mathbf{L}_{i,k,j})]$	$k\in\{2,4,8\}$
Continuous (raw)	MI-Est	$\frac{1}{NK}\sum_{i,k}(\text{matched}_{i,k}-\text{marginal}_{i,k})$	Per-token; approaches 0 under collapse
	MI-Seq-Est	$\frac{1}{NK}\sum_{i,k}\left(\mathbf{L}_{i,k,i}-\log\frac{1}{N}\sum_j e^{\mathbf{L}_{i,k,j}}\right)$	Per-sequence; no length normalization
Continuous (z-score)	MI-ZScore	$\frac{1}{NK}\sum_{i,k}\frac{\text{matched}_{i,k}-\text{marginal}_{i,k}}{\sigma_{\text{batch}}+\epsilon}$	Normalized by current-batch marginal std
	MI-ZScore-EMA	$\frac{1}{NK}\sum_{i,k}\frac{\text{matched}_{i,k}-\text{marginal}_{i,k}}{\sigma_{\text{EMA}}+\epsilon}$	$\sigma^{(t)}_{\text{EMA}}=\alpha\,\sigma^{(t-1)}_{\text{EMA}}+(1-\alpha)\,\sigma^{(t)}_{\text{batch}}$

(ii) *Template Collapse* (high $H(Z\mid X)$, low $I(X;Z)$): superficially diverse but input-agnostic—the systematic blind spot of existing stability metrics; (iii) *Compressed Reasoning* (low $H(Z\mid X)$, high $I(X;Z)$): input-faithful but overly deterministic; and (iv) *Low-Entropy Collapse* (low $H(Z\mid X)$, low $I(X;Z)$): fully degenerate with deterministic and input-agnostic outputs. Among these, Template Collapse is uniquely problematic because entropy-based metrics can remain high while input dependence collapses. Empirically, $I(X;Z)$ correlates significantly more strongly with task performance than entropy does (Figure 8).

2.3. Mutual Information Proxy Family

How do we estimate mutual information? True mutual information $I(X;Z)$ has no closed form for high-dimensional token sequences, so we propose an empirical proxy $\widehat{I}(X;Z)$ based on retrieval. The intuition: mutual information $I(X;Z)$ measures how much knowing the reasoning Z tells us about which input X produced it. When $I(X;Z)$ is high, different inputs yield distinguishable reasoning patterns—the model adapts its reasoning to the specific problem. When $I(X;Z)$ is low, reasoning becomes input-agnostic: observing Z gives little clue about which X it came from. This is the signature of template collapse. If reasoning truly collapses into templates, it should be easy to detect: a reasoning trace Z generated from input X_i will be equally likely under any other input X_j.

Method: In-Batch Cross-Scoring. Given N inputs and K reasoning samples per input from training rollouts, we compute teacher-forced log-likelihoods for every $(Z_{i,k}, X_j)$ pair, forming the scoring matrix $\mathbf{L}_{i,k,j}=\log p_\theta(Z_{i,k}\mid X_j)$. We extract two length-normalized quantities:

$$\text{matched}_{i,k}=\frac{\mathbf{L}_{i,k,i}}{|Z_{i,k}|},\qquad \text{marginal}_{i,k}=\frac{1}{|Z_{i,k}|}\log\frac{1}{N}\sum_j\exp(\mathbf{L}_{i,k,j}),\tag{1}$$

where $\text{matched}_{i,k}$ is the per-token log-likelihood of reasoning $Z_{i,k}$ under its true source input X_i, and $\text{marginal}_{i,k}$ approximates the marginal log-likelihood $\log p_\theta(Z_{i,k})$ via a uniform mixture over all prompts in the batch.

Two Primary Proxies. We use two complementary proxies derived from Eq. 1:

(1) Retrieval-Acc (discrete, interpretable): We define

$$\text{Acc}=\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{I}\Big[i=\arg\max_j\ \mathbf{L}_{i,k,j}\Big].$$

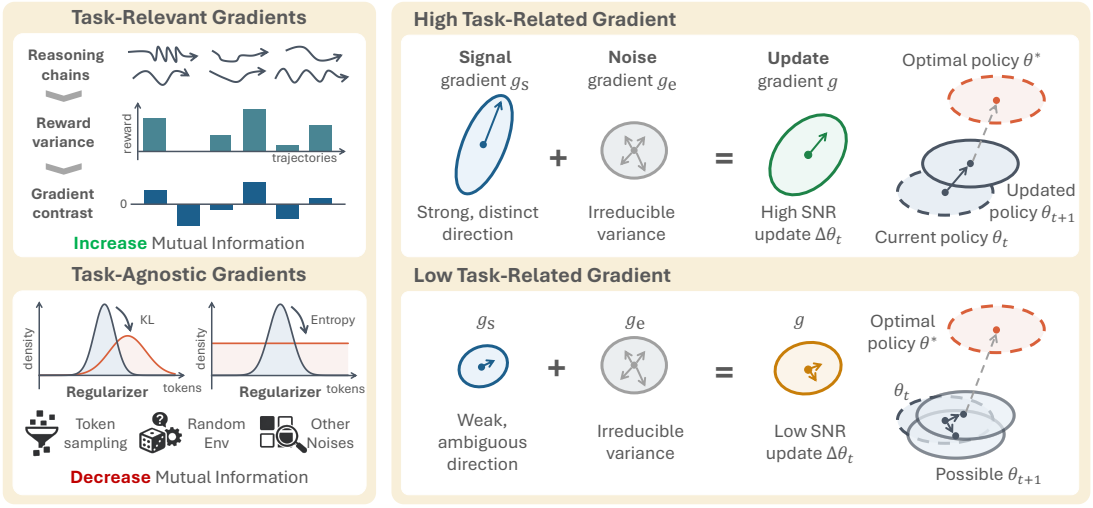

Figure 2 | Schematic Signal-to-Noise Ratio (SNR) view of RL updates. Left: total gradient decomposes into task gradient (sharpens with higher within-input reward variance) and regularization gradient. Right: high reward variance yields strong task gradient and better convergence (high SNR); low reward variance makes regularization gradient dominate, producing erratic updates and input-agnostic reasoning (low SNR).

Under collapse, Acc approaches chance level $1/N$ (1.56% at N=64), providing an absolute reference.

(2) MI-ZScore-EMA (continuous, robust): We estimate input dependence as

$$\widehat{I}(X;Z) = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\left(\text{matched}_{i,k} - \text{marginal}_{i,k}\right),$$

which increases when reasoning is much more compatible with its source input than with the batch mixture. In template-collapse regimes, $\text{matched}_{i,k} \approx \text{marginal}_{i,k}$ for many samples and thus $\widehat{I}(X;Z)$ approaches 0. We apply z-score normalization and exponential moving average (EMA) to stabilize training monitoring, yielding MI-ZScore-EMA.

Proxy Variants and Validation. Table 1 lists additional proxy variants, varying along three dimensions: (1) turn scope (first-turn only vs. trajectory-uniform sampling); (2) aggregation (discrete retrieval vs. continuous MI estimate); (3) length normalization (per-token vs. per-sequence). For comparison, conditional entropy $H(Z \mid X) = -\frac{1}{NK}\sum_{i,k}\text{matched}_{i,k}$ and marginal entropy $H(Z) = -\frac{1}{NK}\sum_{i,k}\text{marginal}_{i,k}$ are logged in parallel, satisfying $H(Z) = \widehat{I}(X;Z) + H(Z \mid X)$. We set $\epsilon = 10^{-3}$ and $\alpha = 0.9$ for z-score normalization and EMA, respectively.

Empirically, Retrieval-Acc and MI-ZScore-EMA achieve positive Spearman correlation with final task performance (+0.39 for Trajectory MI-ZScore), substantially above entropy metrics, which show negative correlations (−0.11 to −0.14), confirming entropy is misleading in direction (Figure 8). All proxies reuse $(X_i, Z_{i,k})$ pairs from the training rollout and require no additional model or inference pass; implementation details are in Appendix C.

3. The Mechanism of Template Collapse: A Signal-to-Noise Ratio (SNR) View

We have defined template collapse (low $I(X;Z)$, high $H(Z \mid X)$) and introduced an MI proxy to diagnose it. This section explains why RL training produces this failure mode and how to mitigate it. Our core finding: when policy gradient updates are dominated by input-agnostic

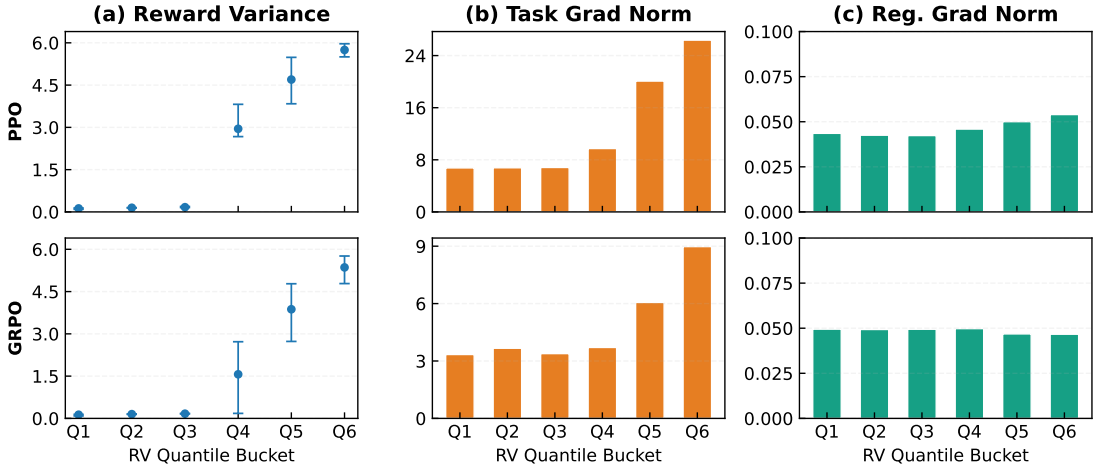

Figure 3 | Prompts sorted into six equal-sized reward-variance buckets Q1–Q6. We find: (a) Task gradient norm increases monotonically with bucket RV; (b) When RV near 0, substantial task gradients persist despite carrying almost no useful signal; (c) Regularizer gradient norm (KL + entropy) is flat across buckets. This directly supports the SNR mechanism under both algorithms.

noise rather than task-discriminative signal—low signal-to-noise ratio (SNR)—reasoning drifts toward templates that appear diverse within each input but ignore cross-input differences.

3.1. Observing Signal-Noise Imbalance in RL Gradients

We begin with an empirical observation that motivates the mechanistic analysis. Sorting training prompts by their within-input reward variance $\widehat{\mathrm{Var}}(R \mid X)$ and grouping them into equal-sized buckets, we measure the gradient norms contributed by task objectives versus regularization terms (Figure 3). Three patterns emerge consistently across algorithms:

1. **Task gradient scales with reward variance**: $\|g_{\text{task}}\|$ increases monotonically with bucket RV. High-variance prompts yield strong task-discriminative gradients; low-variance prompts produce weak gradients even when non-zero.
2. **Regularization gradient is flat**: $\|g_{\text{reg}}\|$ (from KL and entropy terms) remains constant across all buckets, applying uniform contraction to every reasoning chain regardless of its source prompt or reward signal.
3. **Low-RV prompts produce gradient updates dominated by regularization**: In the lowest-variance buckets, task gradients nearly vanish while regularization gradients persist, meaning updates are driven almost entirely by input-agnostic noise.

This gradient imbalance suggests that low reward variance weakens the task-discriminative component of updates, allowing input-agnostic regularization to dominate. When many prompts fall into this regime, the model learns to produce reasoning that satisfies regularization constraints (diverse, fluent) but ignores input-specific requirements—exactly the signature of template collapse.

3.2. Formalizing the SNR Mechanism via Gradient Decomposition

The empirical pattern above can be formalized through a *signal-to-noise decomposition of policy gradients*. Low within-input reward variance collapses advantages toward zero, weakening the task gradient. Simultaneously, input-agnostic regularization terms apply uniform contraction to every reasoning chain regardless of its source prompt. When the task gradient is weak,

regularization dominates every update and pushes reasoning toward input-agnostic patterns, lowering $I(X;Z)$. This is the gradient-level mechanism behind template collapse (Figure 2; regularizer-dominance analysis in Appendix K).

For input x with G sampled trajectories, the advantage estimate is $A_g = R_g - \bar{R}(x)$ and the task gradient is

$$g_{\text{task}}(x) = \frac{1}{G} \sum_g A_g \nabla_\theta \log \pi_\theta(\tau_g \mid x).$$

The Cauchy-Schwarz inequality gives (Appendix H):

$$|g_{\text{task}}(x)| \leq \sqrt{\widehat{\text{Var}}(R \mid X = x)} \cdot C.$$

Low reward variance therefore weakens g_{task} while leaving g_{reg} unchanged, driving $I(X;Z) \to 0$. Critically, $H(Z \mid X)$ need not decline: entropy regularization can sustain within-input diversity while input dependence collapses.

We formalize this through a three-noise decomposition of the total gradient: $g_{\text{total}} = g_{\text{signal}} + g_{\text{task-noise}} + g_{\text{reg}}$.

Table 2 | Three-noise decomposition of the policy update gradient.

Component	Source	Level	Ctrl.	Mitigation
g_{signal}	Meaningful reward differences across same-prompt trajectories	Prompt	No	SNR-Adaptive Filtering
$g_{\text{task-noise}}$	Sampling and environment stochasticity	Prompt	No	Filter high-noise prompts
g_{reg}	Uniform contraction per chain, independent of input (KL, entropy)	Chain	Yes	Tune λ_{KL}, λ_{ent}

Signal and task noise both vary across prompts, but only the former carries task-discriminative information. Regularization noise acts uniformly at the chain level: every reasoning chain receives the same KL/entropy contraction regardless of its source prompt, making it inherently input-agnostic and the direct suppressive force on cross-input differences (Table 2).

In practice, $g_{\text{task}} = g_{\text{signal}} + g_{\text{task-noise}}$ merges the two prompt-level components. The SNR is $\text{SNR}(x) = \|g_{\text{signal}}(x)\| / (\|g_{\text{task-noise}}(x)\| + \|g_{\text{reg}}\|)$. Low SNR shifts updates toward input-agnostic directions, lowering $I(X;Z)$ even when $H(Z \mid X)$ remains high (Appendix K).

Low reward variance but non-vanishing gradient norm. When $\widehat{\text{Var}}(R \mid X) \approx 0$, advantages collapse to zero and $g_{\text{task}} \approx 0$, yet $\|g_{\text{total}}\| \approx \|g_{\text{reg}}\|$ because g_{reg} is independent of reward variance. Low-RV prompts therefore produce updates driven entirely by input-agnostic regularization noise, systematically lowering $I(X;Z)$. SNR-Adaptive Filtering removes these task-useless but regularization-active updates by filtering out low-RV prompts, the core mechanism by which the method restores input-conditioned reasoning.

3.3. SNR-Adaptive Filtering: Prioritizing High-Signal Updates

The gradient analysis above identifies the *mechanism behind template collapse*: low reward variance weakens task signal, allowing regularization noise to dominate and push reasoning toward input-agnostic patterns. This suggests a direct mitigation strategy: prioritize prompts with higher within-input reward variance, where advantage estimates carry stronger task-discriminative information and regularization is less likely to dominate the update.

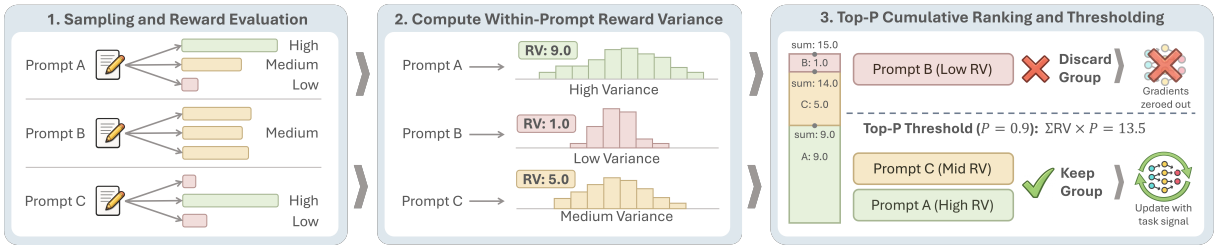

Figure 4 | SNR-Adaptive Filtering workflow. At each training iteration: (1) rollout generation collects trajectories; (2) within-prompt reward variance is computed as SNR proxy; (3) prompts are ranked by RV and top-p mass retained, policy update performed only on high-signal subset. This filtering loop can prevent updating on noisy rollouts and requires no additional models/rollouts beyond standard RL.

We propose **SNR-Adaptive Filtering**: at each training iteration, estimate $\widehat{\mathrm{Var}}(R \mid X)$ for each prompt and retain only the top fraction by variance before computing parameter updates (workflow in Figure 4). This concentrates gradient budget on high-SNR prompts and filters out low-variance updates that would be dominated by input-agnostic regularization.

Reward variance as SNR proxy. 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 sample variance of episode returns:

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

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

Higher $\widehat{\mathrm{Var}}(R \mid X)$ indicates trajectories can be meaningfully distinguished by reward, strengthening advantage estimates and increasing the likelihood that gradients align with task-relevant directions (Appendix H).

Top-p filtering by reward variance. We keep the top fraction of prompts by variance score with kept mass $\rho \in (0, 1]$, analogous to nucleus sampling [14] but ranking by per-prompt reward variance rather than token probability. Given N prompts indexed by $i = 1, \ldots, N$ with variance scores $\widehat{\mathrm{RV}}(x_i)$, we rank by descending variance:

$$\widehat{\mathrm{RV}}(x_{\sigma(1)}) \geq \widehat{\mathrm{RV}}(x_{\sigma(2)}) \geq \cdots \geq \widehat{\mathrm{RV}}(x_{\sigma(N)}),$$

where $\sigma : \{1, \ldots, N\} \to \{1, \ldots, N\}$ is a permutation. Define the selection threshold as

$$\tau = \rho \sum_{i=1}^{N} \widehat{\mathrm{RV}}(x_i),$$

and accumulate variance mass from the top until reaching τ:

$$S = \{\sigma(1), \ldots, \sigma(k^*)\}, \quad \text{where } k^* = \min \left\{ k : \sum_{j=1}^{k} \widehat{\mathrm{RV}}(x_{\sigma(j)}) \geq \tau \right\}.$$

The filtered objective becomes $\mathcal{L}_\rho(\theta) = \frac{1}{k^*} \sum_{i \in S} \sum_{j \in \mathcal{B}_i} L_\theta(\xi_j)$, where $\mathcal{B}_i$ is the set of samples in group i. This adaptive selection naturally concentrates updates on high-signal prompts while automatically adjusting the kept count based on the variance distribution. Other filtering strategies (top-k, min-p) and implementation details are in Appendix G.

Table 3 | 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
SearchQA	✗	✓	Text	Dense
WebShop	✗	✓	Text	Dense
DeepCoder	✗	✗	Text	Dense

4. Experiments

We first establish that template collapse occurs reliably across training configurations (Section 4.2), then evaluate SNR-Adaptive Filtering as an intervention across tasks, algorithms, model scales, and modalities (Section 4.3; Table 4).

4.1. Experimental Testbed

We adopt the RAGEN [60] 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). To further evaluate multi-turn reasoning and decision-making capabilities, we also include SearchQA [51], WebShop [64], and DeepCoder [25, 21, 15] (see Appendix B.1 for detailed descriptions).

Environments and tasks. Our testbed spans seven diverse environments with complementary characteristics (Table 3). **Sokoban** is a grid puzzle where the agent pushes boxes onto target cells; actions are effectively irreversible since boxes cannot be pulled [39]. **FrozenLake** is a navigation task with sparse rewards and stochastic transitions (slippery dynamics) [2]. **MetaMathQA** is a math QA task derived from MetaMathQA [67] where the agent may revise answers over multiple attempts, and we apply a diminishing reward across retries (halving each retry). **Countdown** is a single-turn numbers game [16] where the agent constructs an arithmetic expression to hit a target. **SearchQA** is a multi-turn question-answering task where the agent iteratively searches and synthesizes information to answer complex queries [51]. **WebShop** is an interactive web navigation task where the agent must search and purchase products matching user specifications [64]. **DeepCoder** is a code synthesis challenge where the agent generates program solutions to meet specified input-output requirements [25, 21, 15].

Training and evaluation setup. We train Qwen2.5-3B [35] with the veRL/HybridFlow stack [45], following RAGEN [60] defaults unless otherwise stated. We compare PPO [42], DAPO [68], GRPO [44], and Dr. GRPO [23] for up to 400 rollout–update iterations. Each iteration collects $K = 128$ trajectories per environment, grouped as $P = 8$ prompts with $G = 16$ parallel samples per prompt. When applying SNR-Adaptive filtering with kept mass ρ, we reduce the effective minibatch size accordingly and scale the per-step loss by ρ, so the optimization step size remains comparable.

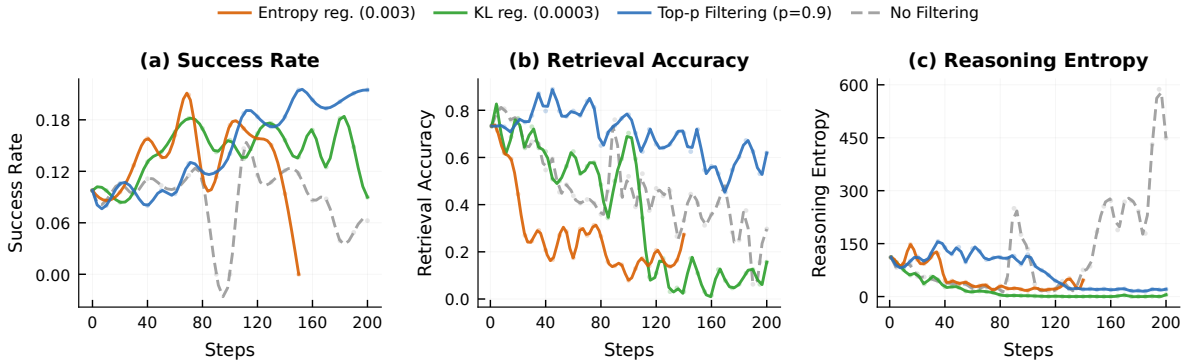

Figure 5 | Training dynamics under different intervention strategies. (a) Task success rate, (b) MI proxy (retrieval accuracy), and (c) reasoning entropy. Without filtering, MI degrades early while entropy spikes, signaling template collapse. Filtering effectively mitigates the decline in retrieval accuracy, with top-p SNR-Adaptive filtering best preserving both task performance and reasoning diversity.

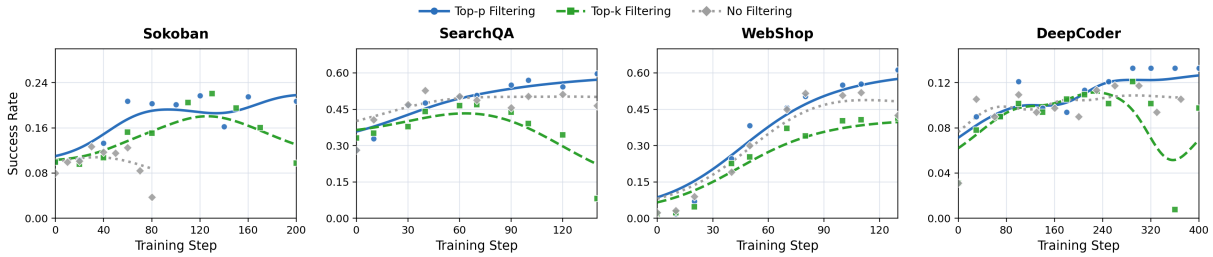

Figure 6 | Comparison of filtering strategies showing Top-p consistently outperforming Top-k and no-filter baselines across four environments.

4.2. Template Collapse as a Consistent Failure Mode

Across all training configurations, RL-trained agents reliably develop reasoning that is fluent but input-agnostic: $I(X;Z)$ declines while $H(Z \mid X)$ remains high, and this drift is invisible to entropy-based monitoring.

Observing template collapse through MI dynamics. We track three key metrics during training: task success rate, our MI proxy $\widehat{I}(X;Z)$ (Retrieval-Acc), and conditional entropy $H(Z \mid X)$ (Figure 5). The trajectory reveals a critical pattern: mutual information declines significantly before task performance degrades, while conditional entropy remains elevated throughout. This divergence is the hallmark of template collapse. Reasoning appears diverse within each input (high $H(Z \mid X)$) but becomes increasingly input-agnostic across inputs (low $I(X;Z)$).

The early decline of $\widehat{I}(X;Z)$ demonstrates that our MI proxy serves as an early warning signal, detecting reasoning degradation that entropy-based metrics miss entirely. This finding motivates using MI as a primary diagnostic alongside task performance, rather than relying solely on entropy for process monitoring.

Behavioral manifestation of template collapse. Beyond diagnostic metrics, template collapse manifests behaviorally as systematic reasoning compression. We do a more broadly evaluation by reproducing several experiments from existing evaluations spanning spatial agents [66], logic puzzle agents [3], visual agents [58], and math agents [22]. Figure 7 shows that reasoning length declines monotonically across all eight environments. As agents converge toward reusable templates, they produce shorter, more formulaic outputs—a behavioral signature of template collapse that complements MI-based diagnostics.

Table 4 | SNR-Adaptive filtering results (%) across algorithms, model scales, types, and modalities. Each cell reports baseline peak with filter delta in parentheses; Qwen2.5-VL-3B includes text (T) and image (V) inputs. Filtering improves average score across all variants. MetaMathQA and Countdown are pure-text and not feasible for visual input.

Experiment Variants	Sokoban	FrozenLake	MetaMathQA	Countdown	Average
Baseline					
PPO [42], Qwen2.5-3B [35]	12.9 (+16.0)	67.0 (+10.9)	92.6 (+0.6)	97.9 (+0.0)	67.6 (+6.9)
Algorithm					
DAPO [68]	16.2 (+5.1)	66.8 (+2.1)	90.8 (+2.8)	95.7 (+1.6)	67.4 (+2.9)
GRPO [44]	12.1 (+9.0)	70.9 (+2.3)	91.2 (+1.2)	95.7 (+2.2)	67.5 (+3.7)
Dr. GRPO [23]	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 [35]	3.3 (+22.9)	19.5 (+0.0)	10.0 (-0.2)	23.0 (-0.7)	14.0 (+5.5)
Qwen2.5-1.5B [35]	17.0 (+6.2)	36.5 (+1.6)	80.3 (+7.0)	56.6 (+1.6)	47.6 (+4.1)
Qwen2.5-7B [35]	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 [35]	22.5 (+14.2)	83.6 (+2.3)	91.2 (+0.4)	96.3 (-0.6)	73.4 (+4.1)
Llama3.2-3B [26]	24.4 (+18.8)	84.6 (-0.2)	86.1 (+3.7)	99.2 (-1.2)	73.6 (+5.3)
Modality (Input Type)					
Qwen2.5-VL-3B (T) [1]	53.0 (+6.0)	16.0 (+53.5)	-	-	34.5 (+29.8)
Qwen2.5-VL-3B (V) [1]	65.0 (+12.0)	19.5 (+59.5)	-	-	42.3 (+35.8)

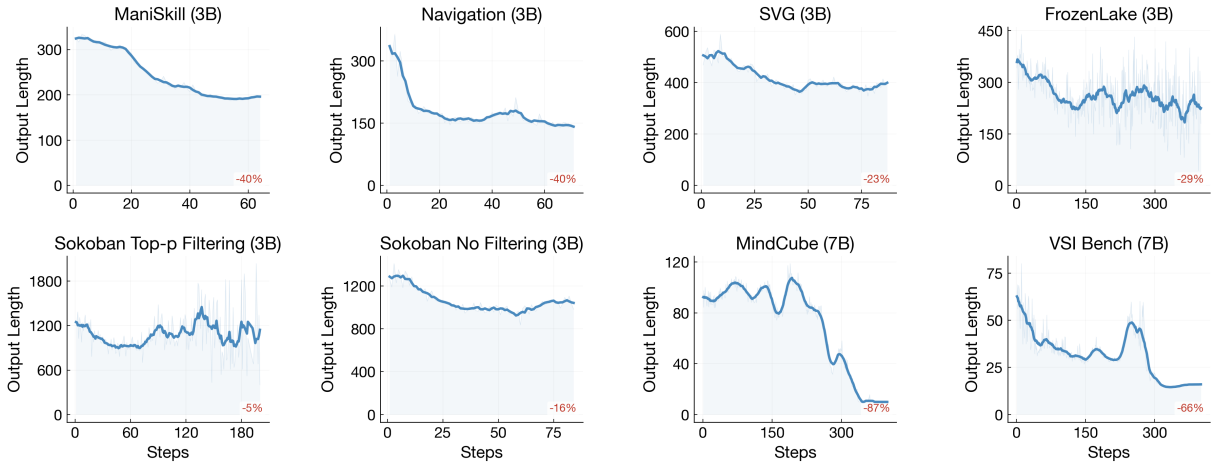

Figure 7 | Reasoning length decline across eight environments, showing systematic compression as a behavioral signature of template collapse.

4.3. SNR-Adaptive filtering Consistently Improves Performance

Comparing filtering strategies across environments. To evaluate the effectiveness of different filtering approaches, we compare three strategies: Top-p (nucleus-style) filtering, Top-k (fixed-count) filtering, and no filtering baseline. Figure 6 shows task success rates across four representative environments (Sokoban, FrozenLake, MetaMathQA, Countdown).

Top-p filtering consistently achieves higher success rates throughout training compared to both alternatives. The advantage over Top-k filtering is particularly noteworthy: while both methods prioritize high-variance prompts, Top-p's adaptive selection naturally adjusts to the variance

distribution, rejecting entire batches when most prompts carry weak signal. In contrast, Top-k always retains a fixed fraction regardless of signal quality, potentially including low-quality updates that dilute the training signal.

The no-filter baseline shows the weakest performance. This confirms that indiscriminate updates on all prompts, including those with near-zero reward variance, systematically degrades learning. These results motivate our choice of Top-p filtering as the primary SNR-Adaptive mechanism, with more comprehensive cross-environment results reported in Table 4.

Table 4 summarizes our experimental matrix over four tasks, multiple RL algorithms, model scales/types, and input modalities. Across this grid, SNR-Adaptive Filtering yields two consistent effects. First, it improves peak task success rate in most settings (reported as the $+\Delta$ next to each peak), demonstrating that prioritizing high-signal updates strengthens learning efficiency. Second, the gains span multiple experimental axes, including (i) the RL optimizer (PPO / DAPO / GRPO / Dr. GRPO), (ii) the model family and scale (Qwen2.5 from 0.5B to 7B; Llama3.2-3B), and (iii) the input modality (text- and image-conditioned Qwen2.5-VL). Here, DAPO and Dr. GRPO are recent strong baselines that directly target stable training and mitigate collapse-like failure modes. In Table 4, DAPO "no-filter" results correspond to the original algorithms without our filtering applied. DAPO itself also includes a filtering/acceptance step; it can be interpreted as a special case of our framework where the selection is fixed (equivalently, a top-P filter with $P \to 1.0$), while our SNR-Adaptive Filtering provides an explicit, tunable SNR knob via the kept mass ρ. This breadth suggests SNR-Adaptive Filtering serves as a general-purpose SNR control knob and works alongside standard stabilization terms (e.g., KL and entropy regularization).

5. Analysis

5.1. MI Diagnoses Collapse Better Than Entropy Across All Interventions

MI separates high- and low-performance runs across all three intervention families; entropy conflates them.

At the same training budget, stronger RV-filtering moves runs toward higher MI and better performance; KL and entropy tuning shift entropy without moving MI. We sweep three families of interventions (entropy regularization strength, KL constraint strength, and SNR-Adaptive filtering kept mass) and compare their trajectories in both diagnostic spaces at fixed training steps (Figure 13). Entropy- and KL-based stabilizers induce larger changes in $H(Z \mid X)$ than in $\widehat{I}(X;Z)$, and rarely move the model into the high-$\widehat{I}(X;Z)$ regime with clearly improved performance. In contrast, SNR-Adaptive filtering traces a monotone improvement in both $\widehat{I}(X;Z)$ and task success; pushing entropy too high leads to instability and performance collapse, while KL constraint mainly anchors the policy near its reference distribution without boosting input dependence.

We compute Spearman correlation between task success rate and each candidate diagnostic across runs with varying entropy regularization strength, KL constraint strength, and Top-p filtering kept mass (Figure 8). MI-family metrics achieve positive correlations, with Trajectory MI-ZScore reaching +0.39. In contrast, Reasoning Entropy and Conditional Entropy metrics show near-zero or negative correlations (between -0.11 and -0.14). This confirms that MI predicts performance twice as reliably as entropy does, and entropy actually points in the wrong direction. These results validate MI as a superior training monitor compared to entropy-based diagnostics for multi-turn agent RL.

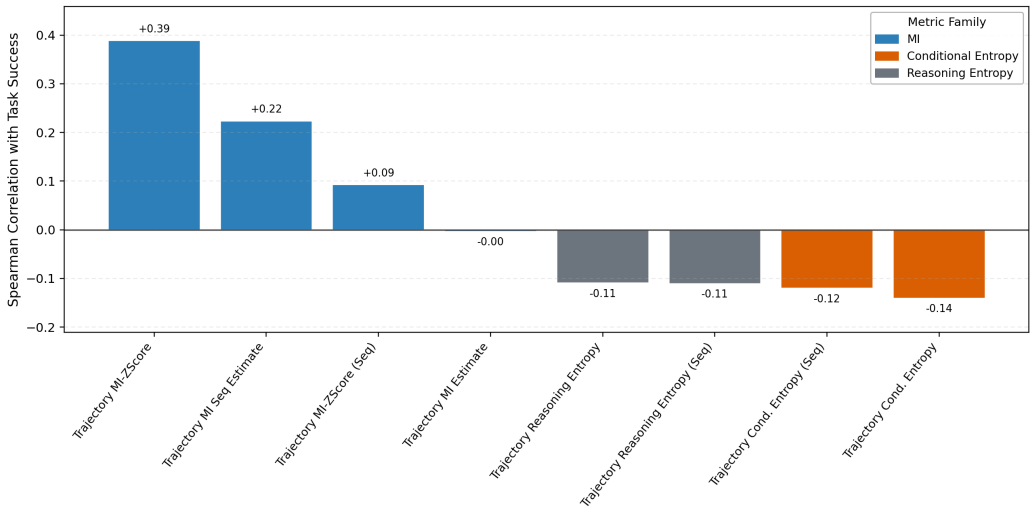

Figure 8 | Spearman correlations showing MI-family metrics positively predict performance while entropy metrics are near-zero or negative.

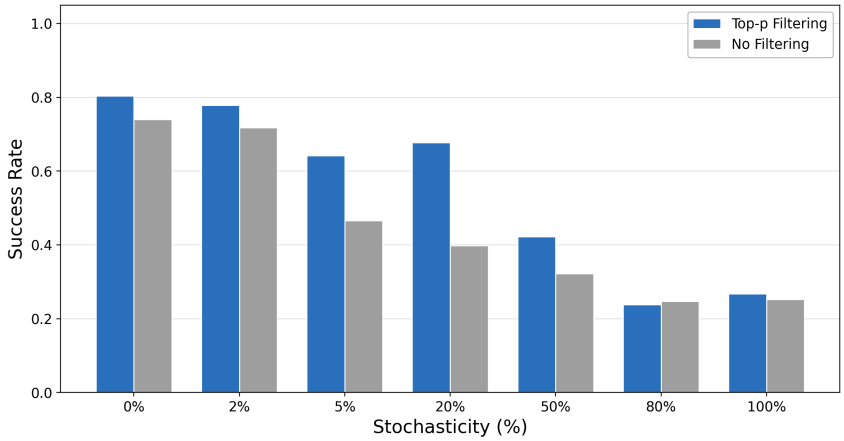

Figure 9 | In FrozenLake, median success rates for both SNR-Adaptive Filtering (orange) and no filtering (gray) decrease as environment stochasticity increases from 0% to 100%. RV filtering maintains a clear advantage from 0% to 50% stochasticity, but the gap closes at 80%–100%, where high transition noise weakens reward variance as an informative signal proxy.

5.2. Does SNR Mechanism Really Interpret Agent RL?

The SNR framing makes a concrete causal claim: template collapse is a gradient-level consequence of low reward variance, not a side effect of aggressive regularization or model capacity. We stress-test this claim with three questions: (1) Does gradient behavior follow the RV signal-to-noise structure? (2) Can we *manipulate* collapse by controlling signal quality? (3) Can SNR-Adaptive Filtering outperform existing stabilization methods by leading to higher SNR? A positive answer to all three would make the SNR account difficult to dismiss.

Controlled noise injection causally weakens MI. Showing that gradient norms correlate with RV is consistent with the theory, but correlation alone cannot rule out confounders. We therefore run a direct intervention: varying environmental stochasticity and asking whether MI declines *predictably* in response. As environment and policy randomness increases, task return drops, conditional entropy rises, and $\widehat{I}(X;Z)$ decreases monotonically (Figure 9). This is the expected consequence of the SNR chain. Additional noise inflates within-prompt return variance in a

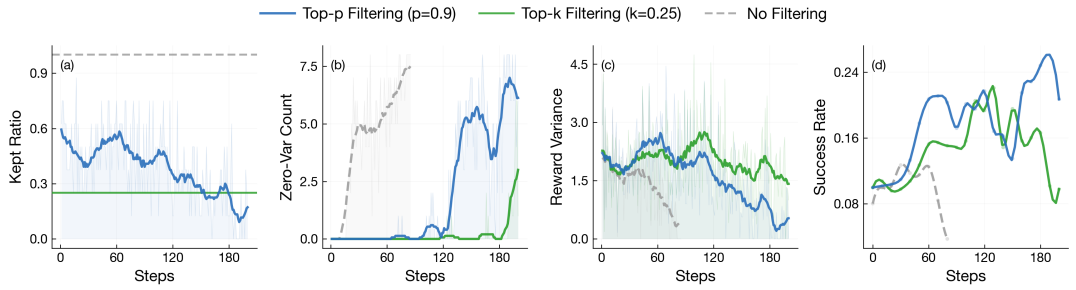

Figure 10 | Effective kept ratio and zero-variance prompt count, showing adaptive selection pressure as variance collapses during training.

signal-free way, diluting the advantage estimates that task gradients depend on. Importantly, the filter's advantage also attenuates at very high noise (80–100%), which is itself informative: when the environment is so stochastic that even high-effort prompts yield noisy rewards, RV loses its discriminative power. The mechanism predicts exactly this boundary condition.

How filtering adapts as training progresses? With the three predictions confirmed, we can now characterize how SNR-Adaptive Filtering behaves over the full training trajectory. Figure 10 tracks the effective kept ratio ρ_{eff} and zero-variance prompt count over training. Both move in the expected direction: as the policy improves and converges, more prompts yield near-identical rollout rewards (zero-variance count rises), and the filter responds by becoming more selective (kept ratio falls). This automatic tightening is precisely what a fixed strategy like Top-k with constant k cannot replicate. It would continue absorbing gradient budget from uninformative prompts even as signal quality deteriorates.

Reward collapse is visible at the distribution level. Figure 11 provides a complementary view of the same dynamics, tracking prompt-level reward distributions across early, mid, and late training in Sokoban. The shift is systematic: the hard portion shrinks as the policy improves, the mixed portion expands, and overall prompt-level variance collapses toward the late stages. This distribution-level signature mirrors the gradient-level story. Late training is not simply "easier" for the policy; it is a regime where reward variation has been compressed to the point that gradient updates carry progressively less task-discriminative information.

Format validity cannot substitute for content-sensitive diagnostics. One might hope that a coarser signal (whether the model's output follows the required format) could serve as a collapse indicator without the overhead of MI estimation. Figure 12 shows this does not hold: format validity is largely decoupled from collapse, with runs maintaining near-perfect validity while exhibiting low MI. Structural correctness and semantic input-dependence are separate dimensions. This reinforces the need for content-sensitive diagnostics, and explains why the MI proxy provides signal that format-based checks miss.

RV is largely orthogonal to entropy and response length, which explains why entropy-based stabilizers cannot prevent template collapse. Reward variance correlates weakly with conditional entropy (Spearman −0.14) and response length (0.12), while correlating strongly with task reward (0.63). RV therefore targets a distinct axis of update quality rather than surface statistics, making it a complementary control knob to KL and entropy regularization. Figure 10 further shows that the effective kept ratio adapts over training: as more prompts drift toward near-zero RV, the filter automatically concentrates gradient updates on the shrinking pool of still-informative prompts.

What is the relationship between SNR Filtering and KL/entropy tuning stabilization? When training RL agents, practitioners typically tune KL penalty and entropy regularization coeffi-

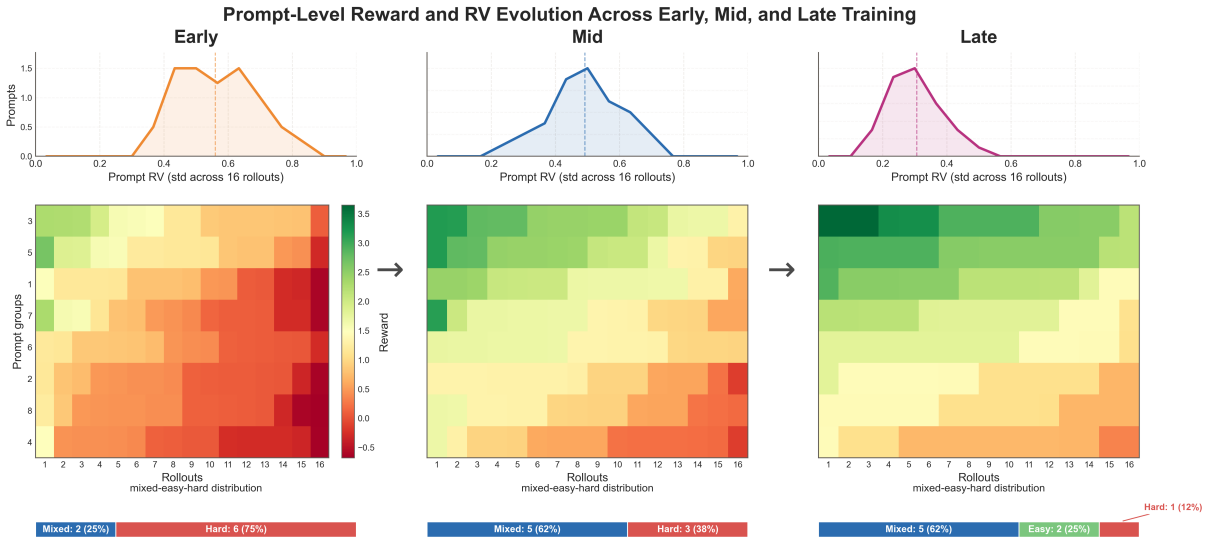

Figure 11 | Prompt-level reward distribution across training phases, showing RV collapse as prompts shift toward uniform reward structures.

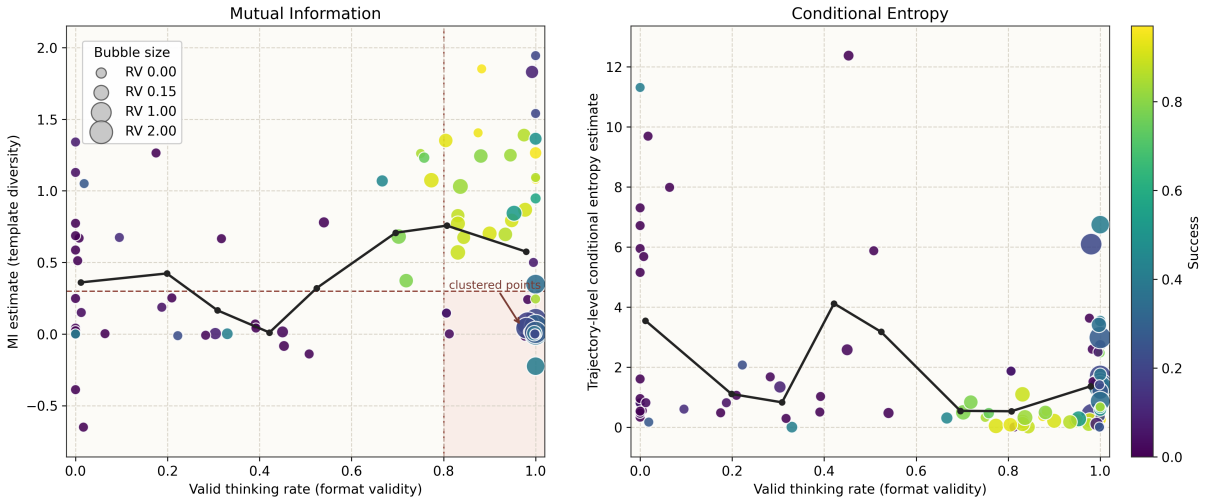

Figure 12 | Format validity versus MI and entropy diagnostics, showing that high validity does not guarantee high input dependence.

cients to maintain training stability and prevent mode collapse. However, these interventions primarily control within-input diversity ($H(Z \mid X)$) and cannot directly address the signal-to-noise imbalance that drives template collapse. Even with carefully tuned regularization, if most prompts have low reward variance, the task gradient remains weak and regularization forces still dominate the update direction.

SNR-Adaptive Filtering is complementary: it selects high-signal prompts at each iteration, directly boosting the fraction of task-discriminative gradient in each update. This acts as a signal-enhancement mechanism rather than a noise-control mechanism. We provide a detailed empirical comparison of KL tuning, entropy tuning, and SNR-Adaptive filtering in Section 5.1, showing that the three interventions move training dynamics along different axes (Figure 13).

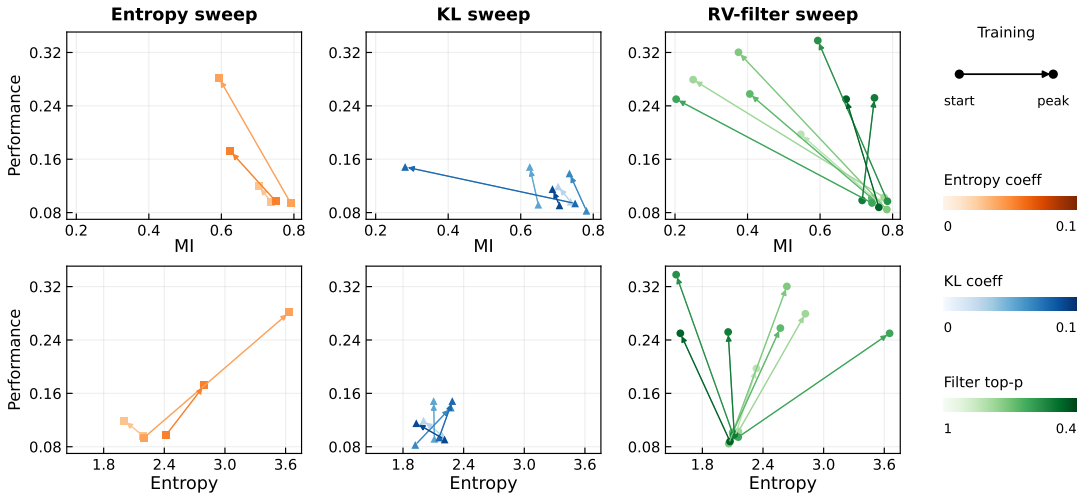

Figure 13 | Training dynamics under three interventions. For each setting, we choose two checkpoints (steps 10/400) and connect them into a trajectory (arrows point to later steps). Color intensity indicates weaker to stronger intervention.

6. Related Work

Reasoning collapse and policy degeneracy in closed-loop LM and Agent RL training.

LLM-agent RL reports various collapse phenomena [7, 61]: *reasoning collapse* (rationales becoming templated with weaker input correspondence) [61, 63, 69] and *policy-level degeneracy* (behavior concentrating on easy-to-reproduce patterns) [9, 59]. These echo model collapse in self-training, even when average metrics appear stable [12, 47].

Evaluating reasoning diversity, input dependence, and reasoning faithfulness.

Most diversity metrics do not test whether differences are *systematically driven by inputs* [53, 69]. Common measures include lexical overlap [20, 74], embedding dispersion [33, 53], and uncertainty analyses [27, 43], primarily capturing within-input variability and missing cross-input shifts [43, 53]. Recent work probes input dependence via behavioral tests [11, 37, 73] and retrieval-style matching [28, 10, 71, 19]. Work on *reasoning faithfulness* asks whether explanations reflect true decision bases [18, 54, 48, 70]. Our focus differs: reasoning may become less input-sensitive even when locally self-consistent.

Stabilizing multi-turn Agent RL under closed-loop sampling.

Stability work spans KL control, entropy regularization, clipping, reward shaping, curricula, and replay mixtures [40, 42, 41, 13, 49, 32, 36, 50, 9, 59, 57, 62, 63]. For multi-step agents, stepwise rewards and self-correction signals are common [4, 30, 55, 24, 46, 56, 65, 8, 61]. However, these methods do not prevent drift toward input-agnostic templates: if rollouts receive similar rewards regardless of reasoning quality, gradients carry little information [29, 31, 47, 69]. We adopt a SNR view, using reward variance filtering low-signal samples to maintain effective SNR.

7. Conclusions and Limitations

We find closed-loop multi-turn agent RL can fail silently: reasoning drifts toward fluent but input-agnostic boilerplate while conditional entropy remains stable. We define this as **template collapse**. Built on this, the paper makes three contributions. First, we introduce a mutual information (MI) proxy between inputs and reasoning tracks task performance substantially

better than conditional entropy, suggesting MI as a more reliable training diagnostic. Second, to explain why collapse occurs, we further show that low within-input reward variance suppresses task gradients and lets regularization forces dominate, pushing policy outputs toward input-agnostic templates. Finally, we introduce SNR-Adaptive Filtering to mitigate template collapse under the SNR view, which prioritizes prompts with higher reward variance before each parameter update, improving performance on average across tasks, model scales, and modalities while integrating transparently with existing training pipelines.

Limitations. The SNR decomposition assumes task-signal and regularization noise separate cleanly, though they may couple through gradient accumulation in practice. All experiments are single-agent; how template collapse propagates in multi-agent RL remains open. A capable model could game the filtering criterion by artificially inflating reward variance, a risk worth monitoring over long training horizons. The method requires reward variance to be a reliable signal proxy, which degrades in sparse or noisy reward environments. Aggressive filtering may narrow exploration coverage; the kept mass requires per-task tuning.

8. Acknowledgements

We thank Yuxiang Lin for help with RAGEN infrastructure and environments, and Kyunghyun Cho for insightful discussions on the manuscript.

References

[1] 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.

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[3] Shiqi Chen, Tongyao Zhu, Zian Wang, Jinghan Zhang, Kangrui Wang, Siyang Gao, Teng Xiao, Yee Whye Teh, Junxian He, and Manling Li. Internalizing world models via self-play finetuning for agentic rl, 2025.

[4] 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.

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

[6] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025.

[7] DeepSeek AI. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, September 2025.

[8] Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. Re-rest: Reflection-reinforced self-training for language agents, 2025.

[9] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025.

[10] Lirong Gao, Ru Peng, Yiming Zhang, and Junbo Zhao. Dory: Deliberative prompt recovery for llm, 2024.

[11] 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.

[12] 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.

[13] 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.

[14] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020.

[15] Naman Jain, Kimin Han, Alex Gu, Wen-Ding Li, Feng Yan, Tianjun Zhang, Yizhou Wang, Koushik Sen, Ion Stoica, and Joseph E. Gonzalez. Livecodebench: Holistic and contamination-free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[16] Michael Katz, Harsha Kokel, and Sarath Sreedharan. Benchmarking llms on the game of countdown, 2025.

[17] 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.

[18] 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.

[19] Hanqing Li and Diego Klabjan. Reverse prompt engineering, 2025.

[20] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models, 2016.

[21] Raymond Li, Loubna Ben Allal, Yijia Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Taco: Topics in algorithmic code generation. *arXiv preprint arXiv:2312.14852*, 2023.

[22] Licheng Liu, Zihan Wang, Linjie Li, Chenwei Xu, Yiping Lu, Han Liu, Avirup Sil, and Manling Li. Unary feedback as observation: Incentivizing self-reflection in large language models via multi-turn RL, 2026.

[23] 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.

[24] 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.

[25] Justus Mattern, Sami Jaghouar, Manveer Basra, Jannik Straube, Matthew Di Ferrante, Felix Gabriel, Jack Min Ong, Vincent Weisser, and Johannes Hagemann. Synthetic-1: Two million collaboratively generated reasoning traces from deepseek-r1. https://www.primeintellect.ai/blog/synthetic-1-release, 2025. Prime Intellect dataset release.

[26] Meta Llama. Llama 3.2 3b model card, 2024. Accessed 2026-01-28.

[27] Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. Jointly measuring diversity and quality in text generation models, 2019.

[28] John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. Language model inversion, 2023.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] Penghui Qi, Zichen Liu, Xiangxin Zhou, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Defeating the training-inference mismatch via fp16, 2025.

[35] Qwen Team. Qwen2.5 technical report, 2024.

[36] 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.

[37] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist, 2020.

[38] Joshua Romoff, Peter Henderson, Alexandre Piché, Vincent Francois-Lavet, and Joelle Pineau. Reward estimation for variance reduction in deep reinforcement learning, 2018.

[39] Max-Philipp B. Schrader. gym-sokoban, 2018. Accessed 2026-01-29.

[40] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017.

[41] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018.

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[43] Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. On accurate evaluation of gans for language generation, 2019.

[44] 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.

[45] 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.

[46] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] Sijun Tan, Michael Luo, Colin Cai, Tarun Venkat, Kyle Montgomery, Aaron Hao, Tianhao Wu, Arnav Balyan, Manan Roongta, Chenguang Wang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. rllm: A framework for post-training language agents. https://pretty-radio-b75.notion.site/rLLM-A-Framework-for-Post-Training-Language-Agents-21b81902c146819db63cd98a54ba5f31, 2025. Notion Blog.

[52] 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.

[53] Guy Tevet and Jonathan Berant. Evaluating the evaluation of diversity in natural language generation, 2021.

[54] 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.

[55] 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.

[56] 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.

[57] 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, 2025.

[58] Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Fei-Fei Li, Yejin Choi, and Manling Li. VAGEN: Reinforcing world model reasoning for multi-turn VLM agents. *arXiv preprint arXiv:2510.16907*, 2025.

[59] Ruiyi Wang and Prithviraj Ammanabrolu. A practitioner's guide to multi-turn agentic reinforcement learning, 2025.

[60] 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, 2025.

[61] 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.

[62] 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.

[63] Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning, 2025.

[64] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pages 20744–20757, 2022.

[65] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.

[66] Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Manling Li, Jiajun Wu, and Li Fei-Fei. Spatial mental modeling from limited views. *arXiv preprint arXiv:2506.21458*, 2025.

[67] 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.

[68] 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.

[69] Longfei Yun, Chenyang An, Zilong Wang, Letian Peng, and Jingbo Shang. The price of format: Diversity collapse in llms, 2025.

[70] Kerem Zaman and Shashank Srivastava. Is chain-of-thought really not explainability? chain-of-thought can be faithful without hint verbalization, 2025.

[71] Collin Zhang, John X. Morris, and Vitaly Shmatikov. Extracting prompts by inverting llm outputs, 2024.

[72] Yaxiang Zhang, Yingru Li, Jiacai Liu, Jiawei Xu, Ziniu Li, Qian Liu, and Haoyuan Li. Beyond precision: Training-inference mismatch is an optimization problem and simple lr scheduling fixes it, 2026.

[73] Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. Promptbench: A unified library for evaluation of large language models, 2024.

[74] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models, 2018.

Appendix Contents

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 [7, 61]. After the model is updated on self-sampled trajectories over time, it may gradually exhibit *reasoning collapse* and *policy-level degeneracy* [7, 61]. 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 [61, 63, 69]. 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 [9, 59].

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 [12, 47]. 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 [61, 62]. In other words, reasoning-level degeneration can decouple from policy-level degeneracy, and in some settings it may precede it [59]. 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 [57, 62].

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 [53, 69]. Concretely, common metrics range from lexical measures such as n-gram statistics and self-BLEU [20, 74], to embedding-based dispersion and distributional distances [33, 53], as well as token-level uncertainty proxies and multi-sample coverage or consistency analyses [27, 43]. 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 [43, 53]. 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 [17, 69]. 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 [59]. Recent work has started to probe input dependence via behavioral tests and local boundary checks [11, 37], prompt robustness benchmarks [73], and retrieval-style output–input matching or prompt reconstruction signals [28, 10, 71, 19]. 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 [9, 63].

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 [18, 54, 48, 70]. 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 [17]. 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 [40, 42, 41, 13, 49, 32, 36, 50, 9, 59, 57, 62, 63]. 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 [4, 30, 55, 24, 46, 56, 65, 8, 61].

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 [29, 31, 47, 69]. 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 [38, 44, 52, 9, 63].

B. Detailed Experimental Settings

B.1. Environments and Tasks

We construct a diverse seven-environment testbed to evaluate LLM agents across complementary axes of decision-making complexity, including planning under irreversible dynamics (Sokoban), long-horizon control with non-deterministic transitions (FrozenLake), multi-step symbolic reasoning in mathematics (MetaMathQA, Countdown), multi-turn search and information synthesis (SearchQA), goal-directed web navigation (WebShop), and program synthesis from input-output specifications (DeepCoder). 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 [39] 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 [2] 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 [67], 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" [16], 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.

DeepCoder. To evaluate agent capabilities in coding environments, we use DeepCoder, a coding benchmark consisting of competitive programming problems. It was used to train DeepSeek-R1-Distill-Qwen-14B with reinforcement learning. The benchmark draws from three resources: PrimeIntellect [25], TACO[21], and LiveCodeBench v5 (LCBv5) [15]. In this environment, agents are required to generate a Python function that solves the given programming problem and passes all hidden and public test cases. During training, rewards are assigned based on the number of test cases successfully passed.

SearchQA. To evaluate multi-turn search and question-answering capabilities, we include SearchQA from the RLLM framework [51], specifically the Search R1 variant. This environment requires the agent to perform iterative web search and reasoning to answer open-domain questions. The agent must formulate search queries, extract relevant information from retrieved documents, and synthesize answers across multiple interaction turns. Rewards are based on answer correctness and search efficiency, encouraging the agent to balance exploration breadth with reasoning depth.

WebShop. We use WebShop [64], an interactive e-commerce environment for evaluating goal-directed multi-turn decision-making. The agent is presented with a shopping instruction (e.g., "find a red shirt under $30") and must navigate a simulated online shopping website by issuing search queries, clicking on products, and selecting appropriate items. The environment features a large action space with realistic product catalogs and requires the agent to perform language understanding, attribute matching, and sequential decision-making. Rewards are assigned based on how well the purchased item matches the specified attributes and constraints.

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 5. 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 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

Table 5 | Ablation results for sampling strategies, filtering metrics, and keep strategies. Values in parentheses denote the change relative to the corresponding baseline in each block. A crossmark in the Stable column indicates training collapse.

EXPERIMENT SETUP	TASK PERF	MI PROXY	ENTROPY	STABLE
Sampling Strategies				
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	✗
Reward Variance	0.38 **(+0.20)**	0.84 **(+0.29)**	1.64 (-1.12)	✓
Reward Sum	0.24 **(+0.07)**	0.80 **(+0.26)**	4.18 **(+1.42)**	✗
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 Strategies				
Keep Largest (Baseline)	0.44	0.89	1.47	✓
Keep Smallest	0.29 (-0.15)	0.47 (-0.42)	5.31 **(+3.84)**	✓

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. Additional Experimental Visualizations

This section presents supplementary visualizations that provide deeper insights into the mechanisms and diagnostics discussed in the main paper. These figures complement the core experimental results with detailed breakdowns of gradient dynamics, diagnostic validity, and reward distribution patterns.

D.1. Gradient Decomposition Across Reward-Variance Buckets

E. Notation and basic identities

E.1. Random variables and distributions

Definition E.1 (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.

Definition E.2 (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:

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

Definition E.3 (Score function)**.** Define the score function

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

It satisfies the normalization identity

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

Definition E.4 (Within-prompt reward variance)**.** We quantify within-prompt variation of observed rewards across rollouts by

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

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.

E.2. Entropy and mutual information

Definition E.5 (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

$$H(Z) \;=\; H(Z \mid X) \;+\; I(X;Z), \tag{2}$$

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 $I(X;Z)$.

F. Scorer-based Proxies for Reasoning Diversity

F.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.

Definition F.1 (Prompt groups)**.** Using the notation from Definition E.1, 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*.

Definition F.2 (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.

Definition F.3 (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.

G. Formal Definition of the Filtering Operator

Definition G.1 (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 $\phi : \in^{\mathcal{B}} \to \mathbb{R}$ that depends only on the samples in the group, and we write $\phi(\mathcal{B}_u)$ for the value computed from $\mathcal{B}_u$.

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

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

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

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

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.1. Filtering Strategy Variants

We compare multiple filtering strategies for selecting high-signal prompt groups. All variants share the same grouping structure (prompts with G rollouts each) and statistic (reward variance $\widehat{\mathrm{RV}}(x_i)$ for group i), but differ in the selection rule.

Top-p (nucleus-style) filtering. The main method used in this paper. Given keep rate $\rho \in (0, 1]$, rank prompts by descending reward variance and select the smallest prefix whose cumulative

variance mass reaches $\rho \sum_i \widehat{\mathrm{RV}}(x_i)$. Formally, let σ be the permutation such that $\widehat{\mathrm{RV}}(x_{\sigma(1)}) \geq \cdots \geq \widehat{\mathrm{RV}}(x_{\sigma(N)})$, and define

$$k^* = \min \left\{ k : \sum_{j=1}^{k} \widehat{\mathrm{RV}}(x_{\sigma(j)}) \geq \rho \sum_{i=1}^{N} \widehat{\mathrm{RV}}(x_i) \right\}.$$

The kept set is $S = \{\sigma(1), \ldots, \sigma(k^*)\}$. This adaptive selection concentrates updates on high-variance prompts while automatically adjusting the kept count based on the variance distribution. When the batch contains many near-zero-variance prompts, top-p can reject the entire batch if the threshold cannot be reached, providing a natural safeguard against degenerate updates.

Top-k (proportional) filtering. An alternative fixed-proportion baseline. Given $\rho \in (0, 1]$, compute $k = \lfloor \rho N \rfloor$ and select the top k prompts by reward variance:

$$S = \{\sigma(1), \ldots, \sigma(k)\}.$$

Unlike top-p, top-k always retains exactly k groups regardless of the variance distribution. This can be less adaptive: when most prompts have near-zero variance, top-k still keeps the highest-variance subset even if all retained prompts carry weak signal.

Min-p (threshold) filtering. Inspired by min-p sampling, this strategy keeps all prompts whose variance exceeds a fraction of the maximum variance. Given threshold parameter $p \in (0, 1]$, define

$$\tau = p \cdot \max_i \widehat{\mathrm{RV}}(x_i),$$

and keep all groups above the threshold:

$$S = \left\{ i : \widehat{\mathrm{RV}}(x_i) \geq \tau \right\}.$$

This directly enforces a minimum quality bar: only prompts within a factor of p of the best prompt are retained. The kept count varies with the variance distribution, making this method highly adaptive but potentially unstable when the maximum variance fluctuates.

Reverse top-p (low-variance) filtering. A diagnostic baseline that intentionally selects low-variance prompts. Rank prompts by *ascending* reward variance and select the smallest prefix whose cumulative variance mass reaches $\rho \sum_i \widehat{\mathrm{RV}}(x_i)$. This inverted strategy is used in ablation studies to confirm that high variance is essential for effective updates: training on low-variance prompts should degrade both MI and task performance, validating the SNR hypothesis.

Implementation notes. All strategies can be configured to exclude zero-variance groups (setting `include_zero=False`) before selection, which removes prompts where all rollouts received identical rewards. For top-p, we use a small epsilon $\varepsilon = 0.01$ to ensure numerical stability when checking whether the cumulative threshold is reached. Additional implementation details and hyperparameter sensitivity are in the codebase.

H. RV Controls Task-Signal Magnitude and SNR

H.1. Setup

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

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

H.2. Assumption

Assumption H.1 (Reward decomposition)**.** 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).

H.3. Task-gradient magnitude is RV-controlled

The next result shows that the task-gradient norm for a given prompt is at most proportional to the square root of its within-prompt reward variance $\mathrm{RV}(x)$. In particular, when $\mathrm{RV}(x)$ is small, the task gradient is provably weak.

Theorem H.2 (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. □

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

The following theorem shows that the signal-to-noise ratio of the K-sample Monte Carlo gradient estimator is upper-bounded by $\sqrt{K} \cdot \sqrt{\mathrm{RV}(x)}/\sigma(x)$. When reward variance is low relative to reward noise, the estimator is dominated by noise.

Theorem H.3 (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 $z_1, \ldots, z_K \overset{i.i.d.}{\sim} \pi_\theta(\cdot \mid x)$. Define

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

Under Assumption H.1 and with baseline $b(x) = \mathbb{E}[R \mid X = x]$,

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

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

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

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

Under Assumption H.1 and with baseline $b(x) = \mathbb{E}[R \mid X = x]$, 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,

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

hence

$$\mathbb{E}\left[\|As - g\|^2 \mid x\right] \ \geq\ \mathbb{E}\left[\|\varepsilon s\|^2 \mid x\right].$$

Combining with the variance decomposition above,

$$\mathbb{E}\left[\|\widehat{g} - g\|^2 \mid x\right] \ \geq\ \frac{1}{K}\,\mathbb{E}\left[\|\varepsilon s\|^2 \mid x\right].$$

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

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

where $\mathbb{E}[\varepsilon^2 \mid x, z] = \mathrm{Var}(\varepsilon \mid x, z) = \sigma^2(x)$ by Assumption H.1. Therefore

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

By Theorem H.2,

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

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

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

H.5. Low-SNR updates induce parameter drift

When updates carry no directional signal (zero mean), the parameter drifts away from initialization at a rate linear in the number of steps. This illustrates why sustained low-SNR updates are harmful even if they do not systematically push in a wrong direction.

Theorem H.4 (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* $T \geq 1$*,*

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

Proof. Unrolling the recursion yields

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

Therefore,

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

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

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

Hence the cross terms vanish and

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

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

I. Template Mixing Reduces Input Dependence

If the policy's conditional distribution is contaminated by a prompt-independent component $q(z)$ with mixing weight α, the resulting mutual information $I_\alpha(X;Z)$ contracts by at least a factor of $(1-\alpha)$. This formalizes the intuition that even partial drift toward a shared template erodes input dependence.

Lemma I.1 (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

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

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 $\mathrm{KL}(\cdot\|\cdot)$ [5, Theorem 2.7.2], 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. □

Remark I.2. The continuity bound $f(\varepsilon)$ depends on $\log(|\mathcal{X}||\mathcal{Z}|)$, which can be extremely large for LLM token spaces. Therefore, this result should be understood as a qualitative guarantee that KL-closeness implies MI-closeness in principle, rather than a tight quantitative bound in practice.

J. Filtering Reduces Gradient-Estimation MSE

J.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.

J.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:

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

By retaining only a subset of prompt groups, the filtered estimator's mean-squared error depends solely on the noise variances of the kept groups. Dropping high-noise (low-RV) groups directly lowers the estimation error.

Theorem J.1 (MSE of the filtered estimator)**.** *$\widehat{G}_S$ is unbiased for G_S and satisfies*

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

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,

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

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

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

so the cross terms vanish. Hence

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

□

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.

K. Reward-Agnostic Regularizers and Update Dominance

K.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

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

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

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

To summarize relative influence, define the dominance ratio

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

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

K.2. Low-RV prompts amplify regularizer influence

When reward variance is small, the task gradient weakens (by Theorem H.2) while regularizer gradients remain largely flat across prompts. Consequently, the regularizer's share of the total update grows on low-RV prompts, formalizing why these prompts are more prone to input-agnostic drift.

By Theorem H.2, for any prompt x,

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

Therefore the dominance ratio

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

admits the lower bound

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

In particular, if $\|g_{\mathrm{reg}}(x)\|$ 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.

L. KL-Closeness to the Base Implies MI-Closeness

If the current policy stays uniformly close to a reference policy in KL divergence, then the mutual information $I(X;Z)$ between inputs and reasoning also remains close. This means strong KL constraints preserve—but do not necessarily increase—input dependence.

Theorem L.1. *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

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

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 $\varepsilon \to 0$. □

M. Decomposing Changes in Input Dependence

Definition M.1 (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

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

The change in mutual information decomposes as $\Delta I = \Delta_{\mathrm{marg}} - \Delta_{\mathrm{in}}$. If an intervention (e.g., an entropy bonus) increases within-prompt variability $H(Z \mid X)$ more than it increases the marginal diversity $H(Z)$, input dependence necessarily decreases.

Theorem M.2. *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 M.2 implies $I_\theta(X;Z)$ decreases.

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

N. GRPO Normalization Amplifies Noise at Low RV

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

For a fixed prompt x, define the normalized advantage

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

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

Dividing the advantage by $\sqrt{\text{RV}(x)}$ causes the gradient estimator's variance floor to scale as $\text{RV}(x)^{-1}$, so prompts with small reward variance suffer disproportionately noisy updates under GRPO-style normalization.

Proposition N.1 (GRPO variance floor)**.** *Under Assumption H.1, the GRPO estimator satisfies*

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

If $\sigma(x) = 0$*, the lower bound is zero and thus vacuous.*

This bound makes explicit that smaller $\text{RV}(x)$ yields a larger variance floor for the normalized estimator if all other factors are the same.