RETHINKING DIVERSITY IN RLVR: FROM POLICY ENTROPY TO SEMANTIC COVERAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has become a standard paradigm for improving the reasoning capabilities of Large Language Models. However, we find that successful responses become increasingly concentrated in fewer semantic reasoning modes as training proceeds. Under a finite rollout budget, this concentration causes on-policy sampling to repeatedly revisit dominant modes, reducing coverage of alternative successful regions and limiting further policy improvement. Theoretically, we attribute this depressing phenomenon to the finite-sample policy gradient estimation, which theoretically leads the policy model collapse on a single response. Motivated by our analysis, to preserve the diverse response modes, we propose SCARS, a framework that organizes discovered responses and intermediate reasoning states in a hierarchical semantic tree and redirects rollout budget toward under-covered successful regions through prefix-conditioned generation. Experiments across multiple reasoning benchmarks show that our method improves overall reasoning performance while maintaining substantially higher successful-response diversity than other baseline methods.

1 INTRODUCTION

RLVR (Shao et al., 2024; Li et al., 2026e) has become a standard post-training paradigm for improving the reasoning capabilities of LLMs in verifiable domains such as mathematical reasoning, code generation, multi-agent system (OpenAI, 2026; Zeng et al., 2026; Xu et al., 2026; Zhang et al., 2026a;b; Li et al., 2026d). However, the outcome verifiers commonly used in RLVR assign the same reward to different successful reasoning paths. The resulting objective therefore optimizes the overall probability of success without distinguishing among different successful reasoning modes. A growing body of work has observed that, as post-training proceeds, the policy becomes increasingly deterministic, the resulting rollout data become highly homogeneous, and performance improvements eventually plateau (Chen et al., 2026b; Cui et al., 2025).

Existing studies commonly attribute this bottleneck to policy-entropy collapse and introduce entropy regularization or other reward-shaping strategies to maintain exploration (Cui et al., 2025; Jin et al., 2026; Hao et al., 2026; Li et al., 2026b;c). Token-level policy entropy, however, is only an indirect proxy for reasoning diversity. It primarily measures the stochasticity of model outputs at the token level, while higher token-level entropy does not necessarily imply a more semantic diversity (Wang et al., 2026b). For reasoning-oriented post-training, a more relevant question is:

Whether the model can preserve multiple distinct ways of arriving at the correct answer?

To explore this phenomenon, we first examine how successful semantic coverage evolves across different stages of RLVR training empirically shown in Figure 2. Specifically, for multiple responses to the same problem, we quantify the diversity using their semantic representations and t-SNE (Van der Maaten & Hinton, 2008) for complementary visualization. The empirical observations show that, as training proceeds, successful responses become increasingly concentrated in fewer semantic classes with the other successful pattern dropped. This trend indicates that reward improvement during post-training does not necessarily expand the set of successful reasoning strategies. Instead, it may come at the expense of low-frequency but effective reasoning paths.

To further understand this phenomenon, we theoretically analyze the practical training process. Under finite-budget on-policy sampling to estimate policy gradient, we show that both REINFORCE-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

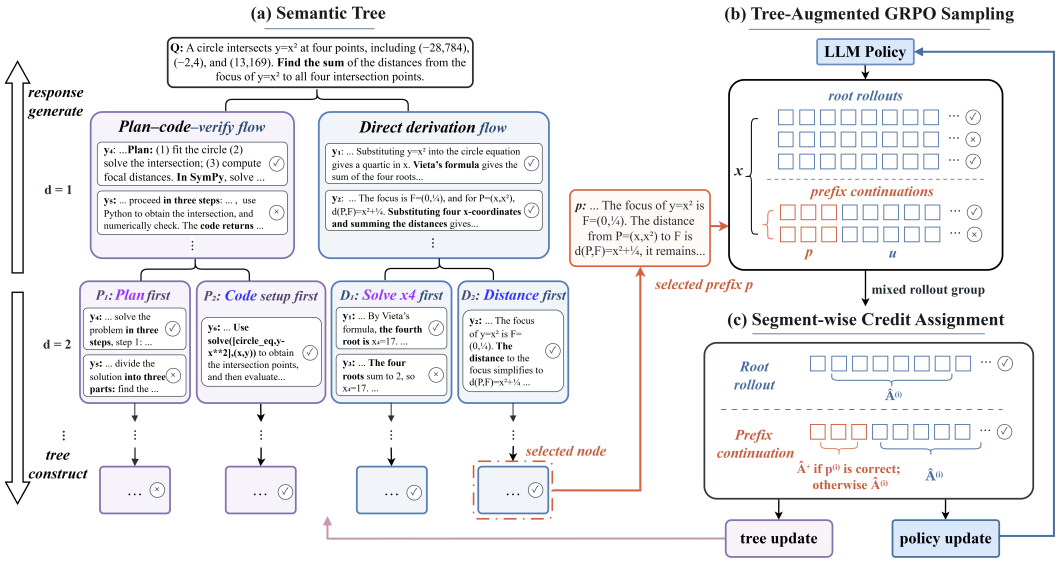


Figure 1: Overview of our framework. The method constructs and maintains a hierarchical semantic tree, allocates rollout budget toward under-covered successful reasoning regions, and optimizes the policy using prefix-conditioned rollouts.

style updates based directly on rewards and GRPO-style updates based on group-relative advantages, will drive the policy model concentrates to a single point mass on a successful response. Thus, its induced distribution over successful semantic classes collapses accordingly. This result identifies an intrinsic failure mode that standard updates can not protect low-frequency successful modes.

These findings suggest that preventing semantic collapse requires explicitly maintaining sampling coverage over distinct successful reasoning regions throughout training. Inspired by our analysis, we propose **S**emantic **C**overage-**A**ware **R**ollout **S**ampling (**SCARS**), a framework that hierarchically organizes semantic representations of historical responses and their intermediate states. During training, SCARS identifies under-covered successful regions and re-enters them by generating new continuations from selected prefixes. This strategy reallocates rollout budget toward reasoning regions that standard on-policy sampling is unlikely to revisit. The resulting prefix-continuation mechanism promotes semantic exploration within each prompt and increases the probability of revisiting low-frequency successful reasoning paths.

Experiments on eight reasoning benchmarks show that SCARS improves the average benchmark score from 46.08 to 48.34, yielding a gain of $+2.26$ over standard GRPO. More importantly, the diversity evaluation shows that the quantified semantic diversity (semantic entropy) of successful reasoning paths increases from 0.359 to 0.630. These results demonstrate that our method not only improves reasoning performance but also preserves a more diverse set of correct solution paths.

In summary, our contributions are:

- We characterize the finite-sample training dynamics of RLVR and prove that, the success-conditional response distribution will concentrate on a single successful response.
- We propose SCARS, a semantic coverage-aware sampling framework that maintains a hierarchical semantic tree and reallocates rollout budget toward under-covered successful regions through prefix-conditioned generation.
- The empirical results on SCARS show improved overall reasoning performance together with substantially higher semantic diversity among successful responses.

2 RELATED WORK

Exploration in RL of LLMs. Existing methods improve exploration in LLMs mainly at two stages of the training loop. One line directly guides the *online generation process*, e.g., by re-sampling from high-entropy positions (Li et al., 2025; Deng et al., 2026) or branching at uncertain

intermediate state in a tree structure (Yizhi et al., 2025). Another line intervenes in the *optimization process* by reshaping the training signal. Some methods use token-level entropy to identify critical or uncertain decisions and concentrate policy updates on these tokens (Wang et al., 2026a). Others directly incorporate entropy-based terms into rewards, advantages or other optimization signals to encourage exploration (Shen, 2026; Cheng et al., 2026; Yao et al., 2026). More recently, several methods instead use response-level semantic novelty or diversity to reshape these optimization signals (Li et al., 2026a; Hu et al., 2026b; Chen et al., 2026a; Hu et al., 2026a). Despite their different intervention points, these methods remain constrained by the coverage of the current policy. As the policy becomes increasingly concentrated, low-frequency reasoning modes become less likely to appear in online rollouts; our method instead maintains previously discovered modes across iterations and uses their semantic coverage to guide prefix-conditioned exploration of under-covered regions.

Tree-Structured Rollout Organization A line of work reorganizes the independent, chain-structured rollouts into tree-structured sampling processes. On the one hand, the tree structure allows rollouts to branch from intermediate states, enabling more efficient and targeted rollout generation (Hou et al., 2025; Yizhi et al., 2025; Xing et al., 2026; Zhao et al., 2026; Zou et al., 2026). On the other hand, the tree aggregates outcomes across rollouts sharing the same prefix, enabling fine-grained credit assignment to intermediate prefixes (Hou et al., 2025; Yizhi et al., 2025; Tran et al., 2026). These methods rely on the shared-prefix structure of rollouts to support efficient rollout expansion from intermediate states or fine-grained credit assignment. Unlike these methods, we use the tree structure to identify and track the policy's coverage over different semantic reasoning modes across training iterations. Accordingly, we first organize complete responses by their semantic modes and then associate each mode with its intermediate prefixes, which are used to guide subsequent exploration toward under-covered modes.

3 Why does Semantic diversity Collapse during RLVR

As mentioned in Section 1, the successful responses will become increasingly concentrated in semantic space during RLVR training. We will explore this phenomenon in this section. Concretely, we first verify this phenomenon empirically by tracking the semantic diversity of successful responses throughout training. We then theoretically analyze how finite-sample on-policy updates can lead to this collapse. Finally, we study the implications of this collapse and show that preserving broader successful semantic coverage mitigate distribution shift.

3.1 Empirical evidence of semantic diversity collapse

We begin by examining how the semantic diversity of successful responses evolves during standard GRPO training. Theoretically, for a given prompt x and reward $r(x, y) \in \{0, 1\}$, the KL regularized objective of RLVR has a closed-form solution (Su et al., 2025; Li et al., 2026b; Rafailov et al., 2023)

$$\pi^*(y \mid x) \propto \pi_{\text{ref}}(y \mid x) \exp\left(\frac{r(x, y)}{\lambda}\right), \tag{1}$$

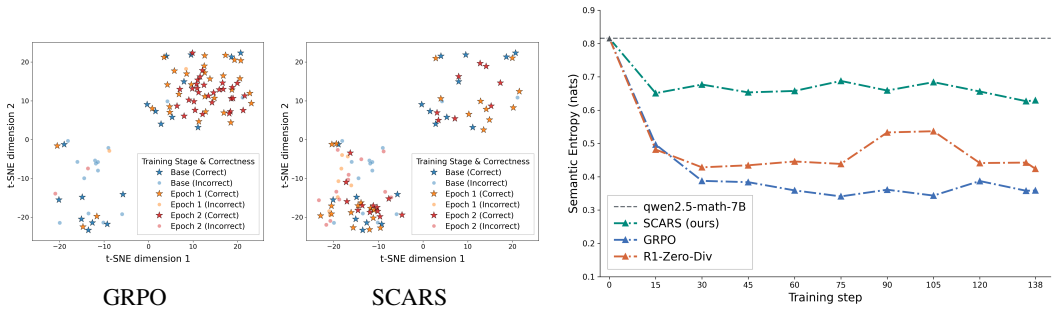
where π_{ref} is the given reference distribution and λ is a positive constant. The optimal policy model (1) naturally has semantic diversity over correct responses, since they are scaled by $\exp(r(x, y)/\lambda)$.

To check whether such diversity holds in practice, we evaluate policy models from different training steps on a fixed test set. For each prompt x and checkpoint of k-th training step, we sample multiple responses, encode their reasoning traces using an semantic embedding model (Zhang et al., 2025), and cluster them according to their similarity. Each resulting cluster represents a distinct reasoning mode. We then use verifier labels to examine how the successful responses are distributed across the resulting clusters, providing an empirical estimate of successful-response semantic diversity.

For a given prompt x and checkpoint of k-th training step, let $\hat{\nu}^+_{k,j}(x)$ denote the fraction of verifier-accepted responses assigned to the j-th cluster. We measure successful semantic diversity using

$$\hat{H}^+_k = -\sum_x \sum_j \hat{\nu}^+_{k,j}(x) \log \hat{\nu}^+_{k,j}(x). \tag{2}$$

Figure 2 provides qualitative and quantitative evidence of semantic diversity collapse during GRPO training. In the representative t-SNE case in Figure 2(a), successful responses become progressively



(a) t-SNE visualization across training stages (b) Successful semantic entropy during training

Figure 2: Empirical evidence of semantic diversity collapse and its mitigation by SCARS. **(a)** t-SNE visualizations compare the evolution of response semantics under GRPO and SCARS, different colors for different training stage and shape for correctness (five-pointed star for correct). **(b)** Successful semantic entropy throughout training further shows that SCARS maintains substantially higher semantic diversity than GRPO and the entropy-based baseline.

concentrated in fewer semantic regions across checkpoints, . Figure 2(b) shows the same trend at the dataset level, with successful-mode diversity consistently decreasing throughout training. These observations suggest that RLVR progressively concentrates probability mass among a smaller set of successful reasoning modes, motivating us to analyze how finite-sample policy updates reshape the relative distribution among successful responses.

3.2 Theoretical Analysis under Finite-Sample Updates

The empirical results above show that successful responses become increasingly concentrated in fewer semantic modes during RLVR training. Next, we make a theoretical explanation to this phenomenon. For a given prompt x, let $\mathcal{Y}(x) = \{y_1, \ldots, y_n\}$ denote a finite set of complete responses, where each response y_i receives a binary verifier reward $r(x, y_i) \in \{0, 1\}$. The policy is trained to maximize the expected reward:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{y \sim \pi_{\boldsymbol{\theta}}(\cdot|x)}[r(x, y)]. \tag{3}$$

At step k, we parameterize the response-level policy by tabular model (Razin et al., 2024; 2026)

$$\pi_{\boldsymbol{\theta}_k}(y_i \mid x) = \frac{\exp(\theta_{k,i})}{\sum_{j=1}^{n} \exp(\theta_{k,j})}.$$

Practically, due to the log-trick $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{y \sim \pi_{\boldsymbol{\theta}}(\cdot|x)}[r(x, y) \nabla \log \pi_{\boldsymbol{\theta}}(y \mid x)]$ (Williams, 1992), the gradient of (3) can be empirically estimated by sampling a group of responses $\{Y_k^{(i)}\}_{i=1}^{G}$ from $\pi_{\boldsymbol{\theta}_k}$. Then, the policy model can be updated by:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \frac{\eta_k}{G} \sum_{i=1}^{G} \widehat{A}_k^{(i)} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_k}(Y_k^{(i)} \mid x), \tag{4}$$

where η_k is the learning rate and $\widehat{A}_k^{(i)}$ denotes either the binary reward itself for a REINFORCE-style (Williams, 1992) update or the group-relative advantage for an idealized GRPO-style update (Shao et al., 2024) (see Appendix A.1 for more details). Next, we will show that the finite sample gradient estimation is exactly the root of mode collapse of trained policy model.

Under the parameter update rule in (4), we can characterize how the probabilities assigned to different responses evolve during training. To directly characterize how successful probability mass evolves across different reasoning modes, we further introduce a semantic mapping $\psi_x : \mathcal{Y}(x) \to \mathcal{C}_x$, where $\psi_x(y)$ denotes the semantic class of response y. Since our interest is in the relative coverage of successful reasoning modes, for each $c \in \mathcal{C}_x$ we define

$$\nu_{k,c}^{+}(x) := \Pr_{y \sim \pi_{\boldsymbol{\theta}_k}(\cdot|x)} (\psi_x(y) = c \mid r(x, y) = 1). \tag{5}$$

Next, we characterize the limiting behavior of $\boldsymbol{\nu}_k$ as $k \to \infty$ under the update rule defined in (4),

Theorem 1 (semantic diversity collapse)**.** *Under the Robbins-Monro conditions and the regularity conditions stated in Appendix A.3, there exists a random semantic class $C_\infty \in \mathcal{C}_x$ such that*

$$\nu_{k,c}^{+}(x) \longrightarrow \mathbf{1}\{c = C_\infty\}, \ \forall c \in \mathcal{C}_x, \quad \textit{almost surely} \tag{6}$$

The complete analysis is provided in Appendix A.3. Theorem 1 shows that finite-sample on-policy updates can asymptotically eliminate diversity within the successful set: even when multiple successful reasoning modes are initially available, their success-conditional probability mass eventually concentrates on a single semantic class. This provides a theoretical explanation for the progressive loss of successful semantic coverage observed in Section 3.1.

3.3 A ROBUSTNESS VIEW OF SEMANTIC COVERAGE

The previous analysis in section 3.2 shows that finite-sample policy updates can naturally lead to semantic diversity collapse among successful responses. We now ask what benefits can be gained by maintaining broad successful semantic coverage, even though such diversity does not directly affect expected reward. Prior work on out-of-distribution (OOD) generalization (Yi et al., 2021; Wang et al., 2022; Yi et al., 2023) suggests that reliance on a narrow subset of predictive features can make models brittle under distribution shift, whereas learning a richer and more diverse set of predictive features can improve robustness (Zhang et al., 2022; Springer et al., 2024). Next, we will analyze how the coverage of different semantic regions influences performance when their relative prevalence shifts across domains.

Let $C \in \{c_1, \ldots, c_K\}$ denote the all the semantic class ($\bigcup_x C_x$ in above). We consider a distribution shift from the source domain $\mathcal{D}_{\text{src}}$ to target domain $\mathcal{D}_{\text{tgt}}$ where the semantic class-conditional prompt distribution $\Pr(X|C)$ remains unchanged, while the semantic prior $\Pr(C)$ may vary. That says the shift of union distribution $\Pr(X, C) = \Pr(C)\Pr(X|C)$ of prompt X and the correct semantic C varied by semantic distribution $\Pr(C)$.

To isolate the effect of semantic coverage, we compare policies with the same performance on the source domain, i.e., every $\pi_{\boldsymbol{\theta}}$ satisfies $J_{\mathcal{D}_{\text{src}}}(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{x\sim\mathcal{D}_{\text{src}}}[E_{y\sim\pi_\theta(\cdot|x)}[r(x, y)]] = \bar{J} > 0$. Although these policies achieve the same overall source performance, they may differ substantially in their class-wise success rates $\Pr(r = 1 \mid c_i)$. Consequently, a target shift that places more mass on classes where a policy performs poorly can lead to a larger performance drop. We formalize this intuition in the following theorem.

Theorem 2 (Worst-case OOD robustness)**.** *Under the regularity conditions stated in Appendix A.4, consider two policies $\pi_{\boldsymbol{\theta}_1}$ and $\pi_{\boldsymbol{\theta}_2}$ with the same source reward. If $\pi_{\boldsymbol{\theta}_1}$ has more balanced successful semantic coverage than $\pi_{\boldsymbol{\theta}_2}$, as formalized in Appendix A.4, then*

$$\inf_{\substack{\boldsymbol{q}\in\Delta_K \\ q_i>0,\ \forall i}} J_{\mathcal{D}_{\text{tgt}}}(\pi_{\boldsymbol{\theta}_1}) \geq \inf_{\substack{\boldsymbol{q}\in\Delta_K \\ q_i>0,\ \forall i}} J_{\mathcal{D}_{\text{tgt}}}(\pi_{\boldsymbol{\theta}_2}) \tag{7}$$

where $\boldsymbol{q} = (q_1, \ldots, q_K) \in \Delta_K$, and $q_i = \Pr_{\mathcal{D}_{\text{tgt}}}(C = c_i)$ denote the target semantic prior.

Here, we consider maximizing the worst case reward over all OOD distributions as in standard OOD generalization literature (Wang et al., 2022; Arjovsky et al., 2019; Rojas-Carulla et al., 2018). Our result shows that, among policies with the same performance on source target, the policy model with diverse semantic output has larger worst-case reward on OOD data. Specifically, the policy model with uniform coverage on semantic achieves the optimal worst-case performance, showing the benefits brought by the semantic diversity. Our complete analysis is provided in Appendix A.4.

4 METHOD

Our analysis in Section 3 shows that successful responses become progressively concentrated in fewer semantic reasoning modes during RLVR, limiting continued exploration and potentially weakening robustness under distribution shift. These findings suggest that mitigating semantic collapse requires explicitly tracking the coverage of different successful reasoning modes and directing more rollout budget toward those that become under-covered. This is the main idea of our method.

According to our observations in Section 3.1 and the existing literature (Chen et al., 2026b), the initialized pretrained has sufficient semantic diversity. Therefore, our goal is preserving such diversity during training. To this end, we propose Semantic Coverage-Aware Rollout Sampling (SCARS), a sampling framework that realizes coverage-aware exploration through a hierarchical gradually updated semantic tree built upon policy model without post-training.

Framework overview. Concretely, our method consists of three steps. First, for each prompt, we generate rollouts from initial model and organize them into a hierarchical semantic tree, which records different reasoning modes together with their associated intermediate prefixes, these trees will be updated accordingly during training. Second, during the rollout stage of training, we allocate more budgets to under-covered successful modes based on the estimated semantic coverage and select a corresponding prefix for continuation to maintain semantic diversity. Finally, we generate a new continuation from the selected prefix and optimize the resulting prefix-conditioned rollouts with segment-wise credit assignment. Next, we present these steps in details.

4.1 STEP 1. SEMANTIC-TREE CONSTRUCTION

At the beginning, for each prompt, we generate rollouts by the pretrained model, and organize them into a hierarchical semantic tree. As shown in Figure 7, unlike a conventional forward trajectory tree, we organize the hierarchy backward from complete responses to progressively earlier prefixes, so deeper nodes correspond to shorter prefixes. Complete responses are clustered by semantic similarity to form distinct high-level reasoning modes. Within each mode, semantically similar prefixes are grouped into shared nodes representing intermediate reasoning states. This hierarchical organization reduces redundancy in the rollout data while preserving meaningful semantic distinctions owing to the diversity adopted from pretrained model. We next formally introduce how the semantic tree is constructed and maintained throughout training.

Hierarchical organization. Given a set of responses $\{y_i\}$, we segment each y_i into D ordered reasoning units $y_{i,1}, \ldots, y_{i,D}$, and define the cumulative prefix as

$$p_{i,j} = y_{i,1} \oplus \cdots \oplus y_{i,j}, \quad j = 1, \ldots, D. \tag{8}$$

Then, we encode each complete response and its prefixes at different steps using a frozen embedding model, i.e., Qwen3-Embedding-8B (Zhang et al., 2025), and organize them into the tree structure level by level according to semantic similarity, where complete responses are first clustered to form high-level reasoning modes, and at each deeper level, prefixes at the same granularity are clustered separately within each parent node.

Tree initialization. Since the pretrained model preserves the semantic diversity, before RLVR training, we initialize a semantic tree $\mathcal{T}_x$ with N_{init} responses sampled from the model. For each node $v \in \mathcal{T}_x$, we store its associated prefixes together with the number of observed and successful rollouts, denoted by n_v and s_v, respectively. These statistics characterize the empirical coverage of the corresponding semantic region and are used for subsequent rollout allocation. Overall, this initialization provides broad initial coverage of reasoning modes and reusable prefix states across these modes for subsequent exploration.

Online maintenance. During RLVR training, newly generated rollouts may reveal previously unseen reasoning modes and change the empirical coverage of existing ones. We therefore update the semantic tree online as new rollouts are generated. Each new rollout is segmented and encoded using the same procedure, and then inserted into the existing tree level by level according to semantic similarity. At each level, it is assigned to an existing node when sufficiently similar, and otherwise forms a new node. The corresponding node statistics are then updated along the insertion path, allowing the tree to continuously update the empirical estimate of the policy's semantic coverage.

4.2 STEP 2. COVERAGE-AWARE ROLLOUT ALLOCATION

Given the semantic tree constructed above, we next determine how to allocate the rollout budget across different semantic regions to preserve the diversity. We make this decision hierarchically: we first select an under-covered successful reasoning mode at the response level, and then choose an intermediate prefix within that mode as the starting point for continuation. In both selections, we favor candidates that are less covered relative to others at the same semantic granularity to make them been explored.

High-level reasoning mode selection. We first determine how to allocate the rollout budget across high-level reasoning modes. Let $\mathcal{A}_x^{\text{full}}$ denote the set of reasoning-mode nodes that contain at least

one successful response and are associated with at least one valid prefix for continuation. For each $v \in \mathcal{A}_x^{\text{full}}$, we quantify its relative coverage by the fraction of successful responses that follow the corresponding reasoning mode,

$$\hat{\nu}_v^+ = \frac{s_v}{\sum_{u \in \mathcal{A}_x^{\text{full}}} s_u},$$

where s_v, as defined in Section 4.1, denotes the number of successful samples associated with node v. We then allocate rollout budget across reasoning modes according to their relative coverage:

$$q_{\text{full}}(v \mid \mathcal{T}_x) \propto \exp\big(-\log \hat{\nu}_v^+ / \kappa_{\text{full}}\big),$$

where κ_{full} controls the strength of the coverage-aware allocation. Among reasoning modes with successful samples, those with fewer successful samples are selected with higher probabilities.

Conditional prefix selection. After selecting a high-level reasoning mode $v^{(1)}$, we restrict the subsequent prefix selection to the subtree rooted at $v^{(1)}$. We first sample a target depth $d \in \{2, \ldots, D\}$ from a predefined distribution $q_{\text{depth}}(d \mid v^{(1)})$, and let $\mathcal{A}_x(v^{(1)}, d)$ denote the set of its descendant nodes at depth d that contain at least one valid prefix. We then apply the same coverage-aware allocation rule within this candidate set. Unlike reasoning-mode selection, we also allow nodes without successful samples to be selected. Since their relative coverage is zero, we assign these nodes a fixed exploration score instead of using $-\log \hat{\nu}^+$. The detailed procedure is provided in Appendix B.2.

4.3 STEP 3. SEGMENT-WISE CREDIT ASSIGNMENT AND POLICY OPTIMIZATION.

After selecting a prefix using the procedures above, the remaining questions are how to construct a complete rollout from the prefix and how to optimize the resulting sample.

Prefix-conditioned exploration. Rather than replaying the historical response associated with the selected prefix, we use the prefix as the starting point and let the current policy generate a new continuation to avoid off-policy tokens (Schulman et al., 2015; Li et al., 2026c). For a prefix-conditioned rollout y_i, we write

$$y_i = p_{i,j} \oplus u_i,$$

where $p_{i,j}$ is the retrieved historical prefix and u_i is newly generated by the current policy. This allows the policy to revisit the selected reasoning mode while exploring new continuations conditioned on the retrieved prefix. The completed rollouts are then evaluated by the verifier and combined with standard root rollouts to compute group-relative advantages in GRPO (Shao et al., 2024).

Segment-wise credit assignment. A prefix-conditioned rollout combines a retrieved historical prefix with a continuation newly generated by the current policy. For a prefix with verified successful history, assigning the advantage of the current complete response to all its tokens may assign negative credit to the prefix whenever the new continuation fails, despite its previous successful outcome. We therefore assign credit to the two segments separately: such prefixes receive positive credit, while the continuation follows the advantage of the current complete response.

Formally, for an effective GRPO group containing both successful and unsuccessful responses, let $\mathcal{I}_{\text{pre}}$ denote the indices of prefix-conditioned rollouts, $h_i \in \{0, 1\}$ indicate whether $p_{i,j}$ was retrieved from a previously verified successful rollout and let $\widehat{A}^+$ denote the positive group-relative advantage assigned to successful responses in the group. We define the token-level advantage at position t as,

$$\widetilde{A}_{i,t} = \begin{cases} \widehat{A}^+, & i \in \mathcal{I}_{\text{pre}},\ t \le |p_{i,j}|,\ h_i = 1, \\ \widehat{A}_i, & \text{otherwise}, \end{cases}$$

Policy Optimization. Using the segment-wise advantage $\widetilde{A}_{i,t}$ defined above, we optimize both root and prefix-conditioned rollouts with the standard clipped policy objective

$$J(\boldsymbol{\theta}) = \frac{1}{G} \sum_{i=1}^{G} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min \left\{ \rho_{i,t}(\boldsymbol{\theta}) \widetilde{A}_{i,t}, \text{clip}\left(\rho_{i,t}(\boldsymbol{\theta}), 1-\delta, 1+\delta\right) \widetilde{A}_{i,t} \right\}, \tag{9}$$

where $\rho_{i,t}(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}\left(y_t^{(i)} | x, y_{<t}^{(i)}\right)}{\pi_{\boldsymbol{\theta}_{\text{sam}}}\left(y_t^{(i)} | x, y_{<t}^{(i)}\right)}$ is the token-level importance ratio, $\pi_{\boldsymbol{\theta}_{\text{sam}}}$ denotes the sampling policy that generated the corresponding token: the historical policy for retrieved prefix tokens and $\pi_{\boldsymbol{\theta}_{\text{old}}}$ for newly generated tokens.

Table 1: Main results on mathematical reasoning and code-generation benchmarks (OOD). Bold and underline indicate the best and second-best results among post-trained methods, respectively.

Methods	Math						Code			
	AIME24	AIME25	AMC23	CMIMC25	HMMT25	Avg.	LCB	HumanEval+	MBPP+	Avg.
Qwen2.5-Math-7B	48.35	32.49	91.23	30.41	15.23	43.54	10.44	47.26	50.07	35.92
SetPO	56.52	36.13	91.1	29.81	18.39	46.39	9.20	47.56	50.79	35.85
R1-Zero-Div	**60.33**	33.71	**94.38**	28.34	19.34	47.22	10.30	47.56	50.26	36.04
GRPO	56.49	36.62	93.54	28.69	15.04	46.08	10.17	46.95	50.40	35.84
DAPO	<u>59.16</u>	37.09	91.94	31.03	16.25	47.09	10.03	47.26	49.74	35.68
GRPO + SCARS	56.53	<u>38.68</u>	<u>94.05</u>	<u>31.81</u>	**20.63**	<u>48.34</u>	<u>10.71</u>	<u>48.02</u>	<u>50.86</u>	<u>36.53</u>
Δ vs. GRPO	(+0.04)	(+2.06)	(+0.51)	(+3.12)	(+5.59)	(+2.26)	(+0.54)	(+1.07)	(+0.46)	(+0.69)
DAPO + SCARS	58.06	**40.66**	93.86	**32.02**	<u>20.04</u>	**48.93**	**11.95**	**48.63**	**51.85**	**37.48**
Δ vs. DAPO	(-1.1)	(+3.57)	(+1.92)	(+0.99)	(+3.79)	(+1.84)	(+1.37)	(+1.37)	(+2.11)	(+1.80)

5 EXPERIMENTS

As clarified in Section 4, our SCARS is proposed to preserve the semantic diversity during RLVR, so that improve the performance of model. Next, we verify whether this argument is hold in practice.

5.1 MAIN RESULTS

Set up. We use Qwen2.5-Math-7B (Yang et al., 2024) as the primary backbone and train on Math12K (Lightman et al., 2024) using VeRL (Sheng et al., 2025), reporting the final checkpoint for all methods. Additional experiments on Qwen3-4B-Base (Yang et al., 2025) are provided in Appendix C.1 to evaluate generality across model families. The details of our experiments are referred to Appendix D.

Baselines. We use standard GRPO as the primary baseline. We additionally compare with two exploration-oriented RLVR methods: R1-Zero-Div (Yao et al., 2026) encourages exploration through entropy regularization over successful responses, while SetPO (Li et al., 2026a) promotes response-level semantic diversity. Since our SCARS is a plugged-in technique, we also compare DAPO (Yu et al., 2026) with DAPO+SCARS.

Evaluation. We evaluate mathematical reasoning on five challenging competition benchmarks: AIME24, AIME25, AMC23, CMIMC25, and HMMT February 2025 (Balunovic et al., 2026). For each problem, we sample 128 responses and compute pass@32 with the unbiased estimator (Chen et al., 2021). Moreover, since we argue that semantic diversity brought improved OOD generalization in Section 3, we further evaluate cross-domain generalization to code generation using LiveCodeBench (LCB) (Jain et al., 2025), HumanEval+, and MBPP+ (Liu et al., 2023). All three code benchmarks are evaluated with execution-based correctness and reported using pass@1.

5.1.1 REASONING PERFORMANCE

Mathematical reasoning. As shown in Table 1, SCARS consistently improves mathematical reasoning performance when applied to both GRPO and DAPO. With GRPO, SCARS improves the average pass@32 from 46.08 to 48.34, with gains across all five benchmarks, including +5.59 on HMMT25 and +3.12 on CMIMC25. The improvement remains when SCARS is applied to DAPO, increasing the average score from 47.09 to 48.93 and achieving the best overall mathematical reasoning performance. These results show that SCARS can be integrated with different policy-optimization methods and consistently improves their performance.

OOD generalization. We further evaluate SCARS on code-generation tasks that are not included in RLVR training. As shown in Table 1, standard GRPO and DAPO do not improve the average code performance over the base model, whereas incorporating SCARS increases the Code Avg. from 35.84 to 36.53 with GRPO and from 35.68 to 37.48 with DAPO. Notably, DAPO + SCARS achieves the best performance on all three code benchmarks. These results support the view that preserving broader semantic diversity among successful responses may improve OOD generalization.

5.1.2 SEMANTIC DIVERSITY

We further evaluate successful-response semantic diversity on AIME24, AIME25, HMMT25, and AMC23 using the semantic entropy defined in (2) as shown in Figure 3. We exclude CMIMC25 from this diversity evaluation because we observe different degrees of output contamination on this benchmark, which is particularly severe for R1-Zero-Div and substantially distorts the semantic-diversity estimates; see Appendix C.2 for details.

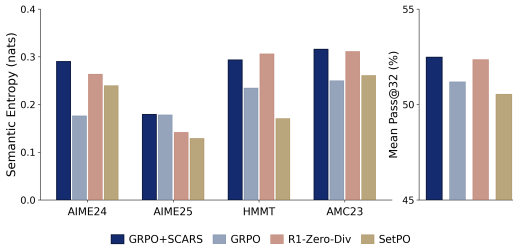


Figure 3: Semantic diversity and reasoning performance across benchmarks. Left: successful-response semantic entropy $\hat{H}^+$ on AIME24, AIME25, HMMT25, and AMC23. Right: mean Pass@32 averaged over the four benchmarks.

Compared with the baselines, SCARS achieves higher average reasoning performance while maintaining stronger successful-response semantic diversity. On most benchmarks, its semantic entropy is substantially higher than that of the other methods. These results suggest that the semantic diversity preserved during training translates into a broader range of reasoning paths on more challenging benchmarks, thereby improving the model's ability to solve difficult problems.

5.1.3 ABLATION STUDY

We ablate three key components of SCARS, with the result shown in Figure 4. **(1) w/o Prefix Exploration** replaces prefix-conditioned continuation with replay of the complete corresponding historical response, which still preserves part of the diversity benefit, but results in lower overall performance than prefix-conditioned exploration. **(2) w/o Node Selection** randomly selects candidate nodes and prefixes instead of prioritizing under-covered semantic regions, reducing semantic diversity on several benchmarks and highlighting the importance of explicitly directing rollouts toward under-covered regions. **(3) w/o Credit Reassignment** assigns retrieved prefix tokens the advantage of the current complete response which substantially reduces diversity, indicating that revisiting a successful prefix alone does not guarantee its corresponding reasoning mode is preserved during subsequent optimization. These results indicate that maintaining semantic diversity requires both revisiting under-covered successful modes during rollout generation and preserving their validated prefixes during policy optimization. Detailed results are provided in Appendix C.3.

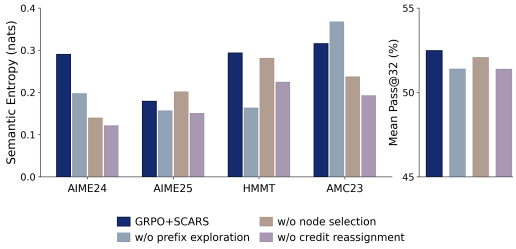


Figure 4: Component ablations of SCARS. Left: successful-response semantic entropy on AIME24, AIME25, HMMT25, and AMC23. Right: mean Pass@32 averaged over the same benchmarks.

6 CONCLUSION

In this paper, we identify a progressive collapse of successful-response semantic diversity during RLVR post-training and further empirically show that entropy-based exploration does not effectively prevent it. We further theoretically shows that finite-sample on-policy updates naturally leads to reasoning mode collapse, while more balanced successful semantic coverage leads to more robust performance under distribution shift. Motivated by these findings, we propose SCARS, a sampling framework that maintains historical rollouts in a tree structure to track semantic coverage across reasoning modes and redirects online rollout budget toward under-covered successful regions through prefix-conditioned exploration. Experiments show that SCARS consistently preserves broader successful semantic coverage while improving reasoning performance. Overall, our work provides an effective exploration mechanism that leverages historical experience to identify and revisit under-covered successful reasoning modes, thereby enabling more targeted exploration in RLVR.

AI USE STATEMENT

In this work, we used generative AI tools to assist with checking selected theoretical derivations and arguments for consistency and correctness, conducting literature search and identifying relevant prior work, and polishing the language of parts of the appendix to improve clarity and readability. All identified references, AI-assisted edits, and theoretical checks were independently reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. Preprint arXiv preprint arXiv:1907.02893, 2019.

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *Advances in Neural Information Processing Systems*, 2026.

Michel Benaïm. Dynamics of stochastic approximation algorithms. In *Seminaire de probabilites XXXIII*, pp. 1–68. Springer, 2006.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. Preprint arXiv:2107.03374, 2021.

Xiwen Chen, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Hao Wang, Haiyu Wu, Huayu Li, Aris Sotiras, Yalin Wang, and Abolfazl Razi. Dra-grpo: Your grpo needs to know diverse reasoning paths for mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026a.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 2026b.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. Preprint arXiv:2505.22617, 2025.

Huilin Deng, Hongchen Luo, Yue Zhu, Long Li, Zhuoyue Chen, Xinghao Zhao, Ming Li, Chuyang Zhao, Jihai Zhang, MengChang Wang, et al. I^2b-lpo: Latent policy optimization via iterative information bottleneck. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 2004.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. Rethinking entropy interventions in rlvr: An entropy change perspective. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. Treerl: Llm reinforcement learning with on-policy tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12355–12369, 2025.

Zhiyuan Hu, Yucheng Wang, Yufei He, Jiaying Wu, Yilun Zhao, See Kiong Ng, Cynthia Breazeal, Luu Anh Tuan, Hae Won Park, and Bryan Hooi. Rewarding the rare: Uniqueness-aware rl for creative problem solving in llms. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026a.

Zican Hu, Shilin Zhang, Yafu Li, Jianhao Elliott Yan, Xuyang Hu, Leyang Cui, Xiaoye Qu, Chunlin Chen, Yu Cheng, and Zhi Wang. Diversity-incentivized exploration for versatile reasoning. In *International Conference on Learning Representations*, 2026b.

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025.

Renren Jin, Pengzhi Gao, Yuqi Ren, Zhuowen Han, Tongxuan Zhang, Wuwei Huang, Wei Liu, Jian Luan, and Deyi Xiong. Revisiting entropy in reinforcement learning for large reasoning models. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026.

Chenyi Li, Yuan Zhang, Bo Wang, Guoqing Ma, Wei Tang, Haoyang Huang, and Nan Duan. Setpo: Set-level policy optimization for diversity-preserving llm reasoning. Preprint arXiv:2602.01062, 2026a.

Qingbin Li, Rongkun Xue, Jie Wang, Ming Zhou, Zhi Li, Xiaofeng Ji, Yongqi Wang, Miao Liu, Zheming Yang, Minghui Qiu, et al. Cure: Critical-token-guided re-concatenation for entropy-collapse prevention. Preprint arXiv:2508.11016, 2025.

Xiuyu Li, Jinkai Zhang, Mingyang Yi, Yu Li, Longqiang Wang, Yue Wang, and Ju Fan. Ets: Energy-guided test-time scaling for training-free rl alignment. In *Proceedings of the Conference on Machine Learning*, 2026b.

Yu Li, Xiuyu Li, Mingyang Yi, Jiaxing Wang, Liangxu Zhang, Zhaolong Xing, and Zhen Chen. Turning off-policy tokens on-policy: A plug-in approach for improving llm alignment. Preprint arXiv:2607.04728, 2026c.

Yu Li, Mingyang Yi, Xiuyu Li, Ju Fan, Fuxin Jiang, Binbin Chen, Peng Li, Jie Song, and Tieying Zhang. Reasoning and tool-use compete in agentic rl: From quantifying interference to disentangled tuning. Preprint arXiv:2602.00994, 2026d.

Zhaochun Li, Mingyang Yi, Yue Wang, Shisheng Cui, and Yong Liu. Towards a theoretical understanding to the generalization of rlhf. Preprint arXiv:2601.16403, 2026e.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 2023.

OpenAI. Gpt-5.6 system card. Technical report, OpenAI, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 2023.

Olivier Raimond and Pierre Tarres. Non-convergence to unstable equilibriums for continuous-time and discrete-time stochastic processes. Preprint arXiv:2311.02978, 2023.

Noam Razin, Hattie Zhou, Omid Saremi, Vimal Thilak, Arwen Bradley, Preetum Nakkiran, Joshua Susskind, and Etai Littwin. Vanishing gradients in reinforcement finetuning of language models. In *International Conference on Learning Representations*, 2024.

Noam Razin, Zixuan Wang, Hubert Strauss, Stanley Wei, Jason Lee, and Sanjeev Arora. What makes a reward model a good teacher? an optimization perspective. *Advances in neural information processing systems*, 2026.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *Journal of Machine Learning Research*, 19(36):1–34, 2018.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, 2015.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. Preprint arXiv:2402.03300, 2024.

Han Shen. On entropy control in llm-rl algorithms. In *International Conference on Learning Representations*, 2026.

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. In *Proceedings of the Twentieth European Conference on Computer Systems*, 2025.

Jacob Mitchell Springer, Vaishnavh Nagarajan, and Aditi Raghunathan. Sharpness-aware minimization enhances feature quality via balanced learning. Preprint arXiv:2405.20439, 2024.

Xuerui Su, Yue Wang, Jinhua Zhu, Mingyang Yi, Feng Xu, Zhiming Ma, and Yuting Liu. Reveal the mystery of dpo: The connection between dpo and rl algorithms. Preprint arXiv:2502.03095, 2025.

Hieu Tran, Zonghai Yao, and Hong Yu. Exploiting tree structure for credit assignment in reinforcement learning with large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

Ruoyu Wang, Mingyang Yi, Zhitang Chen, and Shengyu Zhu. Out-of-distribution generalization with causal invariant transformations. In *Conference on Computer Vision and Pattern Recognition*, 2022.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 2026a.

Zihan Wang, Chi Gui, Xing Jin, Qineng Wang, Licheng Liu, Kangrui Wang, Shiqi Chen, Linjie Li, Zhengyuan Yang, Pingyue Zhang, et al. Ragen-2: Reasoning collapse in agentic rl. Preprint arXiv:2604.06268, 2026b.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Shangyu Xing, Siyuan Wang, Chenyuan Yang, Xin-Yu Dai, and Xiang Ren. Lookahead tree-based rollouts for enhanced trajectory-level exploration in reinforcement learning with verifiable rewards. In *International Conference on Learning Representations*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. (arXiv:2606.19348), 2026.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. Preprint arXiv:2409.12122, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. Preprint arXiv:2505.09388, 2025.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning. *Advances in Neural Information Processing Systems*, 2026.

Mingyang Yi, Lu Hou, Jiacheng Sun, Lifeng Shang, Xin Jiang, Qun Liu, and Zhiming Ma. Improved ood generalization via adversarial training and pretraing. In *International Conference on Machine Learning*, 2021.

Mingyang Yi, Ruoyu Wang, Jiachen Sun, Zhenguo Li, and Zhi-Ming Ma. Breaking correlation shift via conditional invariant regularizer. In *International Conference on Learning Representations*, 2023.

LI Yizhi, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Ruibin Yuan, Tianshun Xing, Shuyue Guo, Tuney Zheng, Xin Zhou, Xingwei Qu, et al. Treepo: Enhancing policy efficacy and inference efficiency with tree modeling. In *Forty-third International Conference on Machine Learning*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 2026.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. Preprint arXiv:2602.15763, 2026.

Jianyu Zhang, David Lopez-Paz, and Léon Bottou. Rich feature construction for the optimization-generalization dilemma. In *International Conference on Machine Learning*, 2022.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. Preprint arXiv:2506.05176, 2025.

Yining Zhang, Mingyang Yi, Chen Wang, Xuwen Xiang, Tianhe Jia, Zedong Dan, Chengqing Zong, and Yue Wang. Htam: Hierarchical transition-attended memory for operator optimization. Preprint arXiv:2605.29734, 2026a.

Yuxin Zhang, Meihao Fan, Ju Fan, Mingyang Yi, Yuyu Luo, Guoliang Li, Bin Wu, and Wenchao Zhou. Reward-sql: Boosting text-to-sql via stepwise execution-aware reasoning and process-supervised rewards. *Proceedings of the ACM on Management of Data*, 2026b.

Ziqi Zhao, Zhaochun Ren, Jiahong Zou, Liu Yang, Zhiwei Xu, Xuri Ge, Zhumin Chen, Xinyu Ma, Daiting Shi, Shuaiqiang Wang, et al. Reinforced efficient reasoning via semantically diverse exploration. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026.

Heming Zou, Qi Wang, Yun Qu, Yuhang Jiang, Lizhou Cai, Yixiu Mao, Ru Peng, Xin Xu, Weijie Liu, Kai Yang, et al. Trace: A unified rollout budget allocation framework for efficient agentic reinforcement learning. Preprint arXiv:2606.11119, 2026.

Appendix

A Complete Theoretical Analysis **15**

A.1 Notation and Analysis Framework 15

A.2 Policy Updates under Expected Policy Gradients 17

A.3 Policy Updates under stochastic policy gradients 21

A.4 Semantic Coverage under Prior Shift 31

B Algorithmic and Implementation Details **33**

B.1 Semantic-Tree Construction and Online Maintenance 33

B.2 Coverage-Aware Rollout Allocation 34

B.3 Prefix-conditioned Generation and Segment-Wise Credit Assignment 35

C Additional Experimental Results **35**

C.1 Results on Qwen3-4B-Base 36

C.2 Additional Semantic-Diversity Results 36

C.3 Full Ablation Results 37

C.4 Optimization-Signal Dynamics 37

C.5 Additional t-SNE Case Study 38

C.6 A Semantic-Tree Case Study 38

D Experimental Details **43**

D.1 Models, Data, and Baselines 43

D.2 Training Hyperparameters 43

D.3 Benchmark Evaluation Protocol 43

D.4 Semantic-Diversity Evaluation Protocol 44

D.5 Compute and Reproducibility Details 44

A Complete Theoretical Analysis

This section provides the complete theoretical analysis for Section 3. In Appendices A.1-A.3, we analyze how finite-sample policy updates change the response distribution during post-training under REINFORCE and GRPO. We divide the analysis into three steps. First, Appendix A.1 formalizes the finite-group on-policy training process and derives the update rules induced by the two algorithms. Second, Appendix A.2 studies the expected update dynamics. We show that, under the expected policy-gradient update, REINFORCE and GRPO exhibit similar distributional dynamics: the policy probability mass among successful responses becomes increasingly concentrated toward a single response. Finally, Appendix A.3 further accounts for finite-sample noise and shows that this concentration occurs almost surely under the sampled training process, which directly proves the Theorem 1.

Appendix A.4 further studies why maintaining successful semantic coverage can benefit domain transfer. We consider a semantic-prior shift where the class-conditional prompt distribution remains unchanged while the relative frequencies of semantic classes vary across domains. Among policies with the same source-domain performance, we show that more balanced successful semantic coverage leads to no worse worst-case target-domain performance, with uniform coverage achieving the optimum whenever feasible. This proves Theorem 2.

A.1 Notation and Analysis Framework

We first formalize the policy-gradient framework used in our analysis. For a fixed prompt x, let

$$\mathcal{Y}(x) = \{y_1, \dots, y_n\}$$

denote a finite set of possible responses, where each response y_i receives a binary verifier reward

$$r_i = r(x, y_i) \in \{0, 1\}.$$

The policy is trained to maximize the expected reward

$$J(\boldsymbol{\theta}) = \mathbb{E}_{y \sim \pi_{\boldsymbol{\theta}}(\cdot|x)}\big[r(x, y)\big] = \sum_{i=1}^{n} \pi_{\boldsymbol{\theta}}(y_i \mid x) r_i. \tag{10}$$

At update k, we parameterize the response-level policy using a tabular softmax model:

$$\pi_{k,i} = \pi_{\boldsymbol{\theta}_t}(y_i \mid x) = \frac{\exp(\theta_{k,i})}{\sum_{j=1}^{n} \exp(\theta_{k,j})}. \tag{11}$$

Then according to the log-derivative trick, the policy gradient can be written as

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k) = \mathbb{E}_{y \sim \pi_{\boldsymbol{\theta}_k}(\cdot|x)} \left[r(x, y) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_k}(y \mid x) \right]. \tag{12}$$

where, under the tabular-softmax parameterization, it holds that

$$\frac{\partial}{\partial \theta_j} \log \pi_{\boldsymbol{\theta}}(y_i \mid x) = \mathbb{I}\{i = j\} - \pi_{\boldsymbol{\theta}}(y_j \mid x). \tag{13}$$

Therefore, the i-th coordinate for $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k)$ in (12) is

$$\frac{\partial J(\boldsymbol{\theta}_k)}{\partial \theta_{k,i}} = \pi_{k,i} \left(r_i - J(\boldsymbol{\theta}_k) \right). \tag{14}$$

In practical post-training, the policy gradient is estimated from a finite group of G responses sampled from the current policy, for any $g \in \{1, 2, \dots, G\}$, we sample a rollout $Y_k^{(g)}$ independently from $\pi_{\boldsymbol{\theta}_k}$ and receive a corresponding binary verifier reward $r_k^{(g)} = r(x, Y_k^{(g)})$.

We consider two policy-gradient estimators corresponding to the learning signals studied in the main text.

REINFORCE-style update. Based on (12), a direct Monte Carlo estimate of the policy gradient is given by the reward-weighted REINFORCE estimator:

$$\widehat{\boldsymbol{g}}_k^{\mathrm{R}} = \frac{1}{G}\sum_{g=1}^{G} r_k^{(g)} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_k}\big(Y_k^{(g)} \mid x\big). \tag{15}$$

The corresponding policy update is

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta_k \widehat{\boldsymbol{g}}_k^{\mathrm{R}}, \tag{16}$$

where η_k denotes the learning rate.

A standard variance-reduction technique is to subtract a baseline from the reward. For any baseline b_k that independent with the sampled response, we have

$$\mathbb{E}_{y\sim\pi_{\boldsymbol{\theta}_k}(\cdot|x)}\left[b_k \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_k}(y \mid x)\right] = b_k \nabla_{\boldsymbol{\theta}} \sum_{y\in\mathcal{Y}(x)} \pi_{\boldsymbol{\theta}_k}(y \mid x) = 0. \tag{17}$$

Therefore, subtracting any response-independent baseline b_k preserves the unbiasedness of the policy-gradient estimator, since

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k) = \mathbb{E}_{y\sim\pi_{\boldsymbol{\theta}_k}(\cdot|x)}\left[\big(r(x,y) - b_k\big)\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_k}(y \mid x)\right]. \tag{18}$$

Moreover, as shown in prior work on variance reduction for policy-gradient estimation (Greensmith et al., 2004), the choice of baseline affects the estimator variance, and an appropriately chosen baseline can reduce this variance while preserving unbiasedness.

GRPO-style update. Building on this idea, GRPO constructs a group-relative learning signal by centering and normalizing the rewards within each sampled group. Let

$$\bar{r}_k = \frac{1}{G}\sum_{g=1}^{G} r_k^{(g)}, \quad \sigma_k = \sqrt{\frac{1}{G}\sum_{g=1}^{G}\Big(r_k^{(g)} - \bar{r}_k\Big)^2}. \tag{19}$$

For a group with nonzero reward variance, the group-relative advantage is defined as

$$\widehat{A}_k^{(g)} = \frac{r_k^{(g)} - \bar{r}_k}{\sigma_k}. \tag{20}$$

The corresponding GRPO-style policy-gradient estimator is

$$\widehat{\boldsymbol{g}}_k^{\mathrm{G}} = \frac{1}{G}\sum_{g=1}^{G} \widehat{A}_k^{(g)} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_k}\big(Y_k^{(g)} \mid x\big), \tag{21}$$

with the update rule

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta_k \widehat{\boldsymbol{g}}_k^{\mathrm{G}}. \tag{22}$$

We regard a group as ineffective if it contains only successful responses or only unsuccessful responses, since the group-relative advantages are zero and makes no contribution to the policy update.

The update rules above determine how the policy changes during post-training. To study semantic diversity, however, we are specifically interested in how the probability mass is distributed among successful responses. We therefore separate the total success probability from the relative distribution within the successful set. We define the index set for all successful responses as

$$\mathcal{S}_+ = \{i : r_i = 1\}.$$

We define the total success probability as

$$R_k = \sum_{i\in\mathcal{S}_+} \pi_{k,i}, \tag{23}$$

and, for each $i \in \mathcal{S}_+$, define the response distribution conditional on success as

$$p_{k,i} = \frac{\pi_{k,i}}{R_k} = \pi_{\boldsymbol{\theta}_k}(Y = y_i \mid r(x, Y) = 1). \tag{24}$$

Here, R_k measures the accuracy of policy $\pi_{\boldsymbol{\theta}_k}$ on prompt x, whereas $\boldsymbol{p}_k$ characterizes the relative probabilities assigned to different successful responses. The semantic distribution $\boldsymbol{\nu}_k^+$ defined in the main text is obtained by aggregating $\boldsymbol{p}_k$ over responses belonging to the same semantic class. Therefore, the evolution of $\boldsymbol{p}_k$ is the key object in our subsequent analysis.

Next, in appendixes A.2 and A.3, we will analyze how $\boldsymbol{p}_k$ evolves when the policy is updated using the expected policy gradient or its finite-sample stochastic approximation, respectively.

A.2 Policy Updates under Expected Policy Gradients

We first derive the expected policy updates induced by the REINFORCE-style and GRPO-style estimators defined above. We consider the REINFORCE-style and GRPO-style updates separately, and then show that they induce the same relative update dynamics among successful responses.

REINFORCE-style update. As shown in Appendix A.1, the REINFORCE-style estimator is an unbiased estimate of the standard policy gradient. Under the tabular-softmax parameterization, for any successful response $i \in \mathcal{S}_+$, we have

$$\mathbb{E}\left[\widehat{g}^{\mathrm{R}}_{k,i}\right] = \frac{\partial J(\boldsymbol{\theta}_k)}{\partial \theta_{k,i}} = \pi_{k,i}\left(1 - J(\boldsymbol{\theta}_k)\right) = R_k(1 - R_k)p_{k,i}. \tag{25}$$

GRPO-style update. We next derive the expected GRPO-style gradient. The analysis is slightly more involved because the group-relative advantage depends on the number of successful responses within the sampled group. Let M_k denote the number of successful responses in a sampled group of size G. Since each response is independently sampled from the current policy and is successful with probability R_k, we have

$$\Pr(M_k = m) = \binom{G}{m} R_k^m (1 - R_k)^{G-m}, \quad m = 0, \ldots, G. \tag{26}$$

When $M_k = m$ with $1 \le m \le G - 1$, the group contains both successful and unsuccessful responses and is therefore effective. Under the binary reward setting, the group-relative advantages of a successful and an unsuccessful response are respectively

$$\widehat{A}^+ = \sqrt{\frac{G-m}{m}}, \qquad \widehat{A}^- = -\sqrt{\frac{m}{G-m}}. \tag{27}$$

According to (21), for any successful response $i \in \mathcal{S}_+$, conditional on $M_k = m$,

$$\begin{aligned}
\mathbb{E}\left[\widehat{g}^{\mathrm{G}}_{k,i} \mid M_k = m\right] &= \mathbb{E}\left[\frac{1}{G}\sum_{g=1}^{G} \widehat{A}_k^{(g)}\left(\mathbb{I}\{Y_k^{(g)} = y_i\} - \pi_{k,i}\right) \,\middle|\, M_k = m\right] \\
&= \frac{\widehat{A}^+}{G}\,\mathbb{E}\left[\sum_{g=1}^{G} \mathbb{I}\{Y_k^{(g)} = y_i\} \,\middle|\, M_k = m\right] \\
&= \frac{\widehat{A}^+}{G}\, m p_{k,i} \\
&= \frac{\sqrt{m(G-m)}}{G} p_{k,i}.
\end{aligned} \tag{28}$$

Ineffective groups contain either only successful or only unsuccessful responses and make no contribution to the policy update. Averaging (28) over M_k therefore gives

$$\mathbb{E}\left[\widehat{g}^{\mathrm{G}}_{k,i}\right] = c_k^{\mathrm{G}} p_{k,i}, \tag{29}$$

where

$$c_k^{\mathrm{G}} = \sum_{m=1}^{G-1} \binom{G}{m} R_k^m (1-R_k)^{G-m} \frac{\sqrt{m(G-m)}}{G} > 0. \tag{30}$$

(25) and (29) show that the two update rules have the same relative expected gradient among successful responses. For both policy-gradient estimators, the expected gradient assigned to each successful response is proportional to its conditional probability $p_{k,i}$. These two estimator only differ in the magnitude of their expected updates. Therefore we focus on the REINFORCE-style update in the following analysis, while the corresponding results for the GRPO-style update can be obtained similarly.

Under the expected REINFORCE-style gradient, combining (16) and (25), then for each successful response index $i \in \mathcal{S}_+$, the corresponding logit is updated as:

$$\theta_{k+1,i} = \theta_{k,i} + \eta_k R_k (1-R_k) p_{k,i}. \tag{31}$$

Recall that according to the softmax parameterization defined in (11), the conditional distribution over successful responses can be equivalently written as

$$p_{k,i} = \frac{\exp(\theta_{k,i})}{\sum_{j\in\mathcal{S}_+} \exp(\theta_{k,j})}. \tag{32}$$

Therefore, (31) induces the following exact update of the successful-response distribution:

$$p_{k+1,i} = \frac{p_{k,i} \exp(\eta_k R_k (1-R_k) p_{k,i})}{\sum_{j\in\mathcal{S}_+} p_{k,j} \exp(\eta_k R_k (1-R_k) p_{k,j})}. \tag{33}$$

Note that for any two successful responses $i, j \in \mathcal{S}_+$, (33) gives

$$\log \frac{p_{k+1,i}}{p_{k+1,j}} = \log \frac{p_{k,i}}{p_{k,j}} + \eta_k R_k (1-R_k) \left(p_{k,i} - p_{k,j}\right). \tag{34}$$

Hence, whenever $p_{k,i} > p_{k,j}$, their probability ratio further increases after the update, which means that, under expected policy-gradient updates, successful responses with higher current probabilities receive larger positive logit updates and are therefore reinforced more strongly. As this effect accumulates over training, these responses can gradually dominate the successful-response distribution, leading to increasing concentration among successful responses.

We now characterize this concentration tendency formally. Define

$$Q(\boldsymbol{p}) = \sum_{i\in\mathcal{S}_+} p_i^2. \tag{35}$$

Then we can calculate the first-order expansion of (33) with the definition in (35) as

$$p_{k+1,i} - p_{k,i} = \eta_k R_k (1-R_k) p_{k,i} \left(p_{k,i} - Q(\boldsymbol{p}_k)\right) + O(\eta_k^2). \tag{36}$$

Since $R_k(1-R_k) > 0$ at every finite update, this term only rescales the update magnitude and does not change its direction. Therefore, based on (36), we consider the following continuous-time approximation of the update dynamics in (33):

$$\dot{p}_i = p_i \left(p_i - Q(\boldsymbol{p})\right), \quad i \in \mathcal{S}_+. \tag{37}$$

It holds that the dynamics in (37) preserve the probability simplex. Indeed, note that

$$\sum_{i\in\mathcal{S}_+} \dot{p}_i = \sum_{i\in\mathcal{S}_+} p_i \left(p_i - Q(\boldsymbol{p})\right) = Q(\boldsymbol{p}) - Q(\boldsymbol{p}) \sum_{i\in\mathcal{S}_+} p_i = 0, \tag{38}$$

for $\sum_{i\in\mathcal{S}_+} p_i = 1$, and $\dot{p}_i = 0$ whenever $p_i = 0$.

We can therefore characterize the evolution of the successful-response distribution directly through (37). To characterize this concentration tendency quantitatively, we first define the Gini-Simpson diversity of the successful-response distribution:

$$D_{\mathrm{GS}}(\boldsymbol{p}) = 1 - \sum_{i\in\mathcal{S}_+} p_i^2. \tag{39}$$

Using the Gini-Simpson diversity defined in (39), we have the following results:

Proposition 1 (Monotonic decrease of diversity)**.** *Along the continuous-time policy update dynamics in* (37)*, the Gini-Simpson diversity is non-increasing:*

$$\frac{\mathrm{d}}{\mathrm{d}\tau} D_{\mathrm{GS}}(\boldsymbol{p}) = -\sum_{i,j\in\mathcal{S}_+} p_i p_j (p_i - p_j)^2 \le 0. \tag{40}$$

Proof. By (37), it is straightforward to calculate that

$$\frac{\mathrm{d}}{\mathrm{d}\tau} D_{\mathrm{GS}}(\boldsymbol{p}) = -2\sum_{i\in\mathcal{S}_+} p_i \dot{p}_i = -2\sum_{i\in\mathcal{S}_+} p_i^2 \left(p_i - Q(\boldsymbol{p})\right) = -\sum_{i,j\in\mathcal{S}_+} p_i p_j (p_i - p_j)^2 \le 0. \tag{41}$$

□

Proposition 1 shows that, under the policy update dynamics in (37), the Gini-Simpson diversity defined in (39), which measures the diversity of the successful-response distribution, is non-increasing. Moreover, the diversity remains non-decreasing if and only if all successful responses with positive probability have equal probability, as formalized in the following corollary.

Corollary 1 (condition for non-decreasing diversity)**.** *Under the policy update dynamics in* (37)*, the Gini-Simpson diversity remains unchanged if and only if the successful-response distribution is uniform over its nonzero support. Specifically, there exists a nonempty subset $\mathcal{S} \subseteq \mathcal{S}_+$ such that*

$$p_i = \begin{cases} \frac{1}{|\mathcal{S}|}, & i \in \mathcal{S}, \\ 0, & i \notin \mathcal{S}. \end{cases} \tag{42}$$

Proof. From (40), the diversity remains non-decreasing if and only if

$$p_i p_j (p_i - p_j)^2 = 0, \quad \forall i, j \in \mathcal{S}_+. \tag{43}$$

Therefore, all responses with positive probability must have equal probability. Since their probabilities sum to one, $\boldsymbol{p}$ must be uniform over its nonzero support, which gives (42). □

For any distribution characterized in Corollary 1, let $\mathcal{S}$ denote its nonzero support and $s = |\mathcal{S}|$. Then

$$Q(\boldsymbol{p}) = \sum_{i\in\mathcal{S}} \frac{1}{s^2} = \frac{1}{s}.$$

Hence, for every $i \in \mathcal{S}$,

$$\dot{p}_i = p_i \left(p_i - Q(\boldsymbol{p})\right) = 0,$$

while $\dot{p}_i = 0$ also holds for $i \notin \mathcal{S}$ since $p_i = 0$. Therefore, all distributions with unchanged diversity are stationary under the policy update dynamics.

We next examine whether these stationary distributions are stable. As shown below, any stationary distribution supported on more than one successful response is unstable, whereas a distribution concentrated on a single successful response is locally attracting.

Proposition 2 (Stability of stationary distributions)**.** *A stationary distribution concentrated on a single successful response is locally attracting. In contrast, every stationary distribution supported on two or more successful responses is unstable.*

Proof. Consider a stationary distribution that is uniform over a support $\mathcal{S}$ of size s. Let

$$f_i(\boldsymbol{p}) = p_i \left(p_i - Q(\boldsymbol{p})\right)$$

denote the right-hand side of (37).

For a perturbation $\boldsymbol{v}$ that only redistributes probability mass within $\mathcal{S}$, i.e.,

$$\sum_{i\in\mathcal{S}} v_i = 0, \quad \text{and} \quad v_i = 0 \text{ for } i \notin \mathcal{S}$$

The Jacobian of f satisfies

$$\frac{\partial f_i(\boldsymbol{p})}{\partial p_j} = \frac{\partial}{\partial p_j}\left[p_i^2 - p_i \sum_\ell p_\ell^2\right] = \mathbb{I}\{i=j\}\,(2p_i - Q(\boldsymbol{p})) - 2p_i p_j. \tag{44}$$

We define the stationary distribution as $\boldsymbol{p}^\star$, where for $i, j \in \mathcal{S}$, it holds that $p_i^\star = p_j^\star = 1/s$ and

$$Q(\boldsymbol{p}^\star) = \sum_{i\in\mathcal{S}} \frac{1}{s^2} = \frac{1}{s}. \tag{45}$$

Therefore, we have

$$\left.\frac{\partial f_i(\boldsymbol{p})}{\partial p_j}\right|_{\boldsymbol{p}=\boldsymbol{p}^\star} = \frac{1}{s}\mathbb{I}\{i=j\} - \frac{2}{s^2}, \quad i,j \in \mathcal{S}. \tag{46}$$

For a perturbation direction $\boldsymbol{v}$, applying the Jacobian of the vector field f evaluated at $\boldsymbol{p}^\star$ to $\boldsymbol{v}$ gives

$$[Df(\boldsymbol{p}^\star)\cdot\boldsymbol{v}]_i = \sum_{j\in\mathcal{S}}\left(\frac{1}{s}\mathbb{I}\{i=j\} - \frac{2}{s^2}\right)v_j = \frac{1}{s}v_i - \frac{2}{s^2}\sum_{j\in\mathcal{S}} v_j = \frac{1}{s}v_i. \tag{47}$$

Hence, the linearization around the stationary distribution satisfies

$$Df(\boldsymbol{p}^\star)\boldsymbol{v} = \frac{1}{s}\boldsymbol{v}. \tag{48}$$

For a small perturbation around the stationary distribution, write

$$\boldsymbol{p} = \boldsymbol{p}^\star + \epsilon\boldsymbol{v},$$

where $\boldsymbol{v}$ redistributes probability mass within the support. Since $f(\boldsymbol{p}^\star) = 0$, a first-order expansion gives

$$f(\boldsymbol{p}^\star + \epsilon\boldsymbol{v}) = \epsilon Df(\boldsymbol{p}^\star)\boldsymbol{v} + O(\epsilon^2) = \frac{\epsilon}{s}\boldsymbol{v} + O(\epsilon^2). \tag{49}$$

Therefore, to the first order, the perturbation magnitude satisfies

$$\dot{\epsilon} = \frac{1}{s}\epsilon,$$

and hence grows as

$$\epsilon(\tau) = \epsilon(0)e^{\tau/s}.$$

Since $1/s > 0$, any nonzero perturbation that redistributes probability mass within the support is amplified rather than attenuated. Thus, whenever $s \geq 2$, the stationary distribution is unstable.

When $s = 1$, the distribution is concentrated on a single successful response. There is no nonzero perturbation that redistributes probability mass within its support, while perturbations toward other responses are locally contracted. Hence such a stationary distribution is locally attracting. □

Together, Propositions 1, 2, and Corollary 1, show that when the policy is updated using the exact expected policy gradient according to (25) and (31), the Gini-Simpson diversity ((39)) of the successful-response distribution is non-increasing throughout the update process. Moreover, distributions concentrated on a single successful response are the only stable stationary distributions.

As discussed above, the expected GRPO-style and REINFORCE-style gradients differ only by a positive response-independent scaling coefficient. Therefore, the same concentration and stability properties also hold for the expected GRPO-style update.

However, the analysis above considers policy updates using the exact expected policy gradient and is not sufficient to characterize the behavior of the practical training process. In practice, the policy gradient at each iteration is estimated from a finite number of sampled responses. In the next section, we further analyze whether policy updates based on such finite-sample gradient estimates exhibit similar concentration behavior.

A.3 Policy Updates under stochastic policy gradients

We now consider the actual policy update based on a finite group of sampled responses. Unlike the deterministic updates analyzed in Appendix A.2, the gradient estimate at each iteration deviates from its expectation due to finite-sample randomness. We study whether the resulting stochastic update process exhibits the same concentration behavior.

For either the REINFORCE-style or GRPO-style update defined in Appendix A.1, let $\widehat{\boldsymbol{g}}_k$ denote the corresponding finite-sample policy-gradient estimator. The parameter update takes the form

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta_k \widehat{\boldsymbol{g}}_k. \tag{50}$$

Let $\mathcal{F}_k$ denote the information available before the response group at iteration k is sampled. We can decompose the gradient estimator as

$$\widehat{\boldsymbol{g}}_k = \mathbb{E}[\widehat{\boldsymbol{g}}_k \mid \mathcal{F}_k] + \boldsymbol{\delta}_{k+1}, \tag{51}$$

where

$$\mathbb{E}[\boldsymbol{\delta}_{k+1} \mid \mathcal{F}_k] = \mathbf{0}. \tag{52}$$

Thus, the finite-sample update differs from the expected-gradient update studied in Appendix A.2 by a zero-mean sampling error.

Analogously to (33), replacing the expected policy gradient with its finite-sample estimate gives

$$p_{k+1,i} = \frac{p_{k,i} \exp(\eta_k \widehat{g}_{k,i})}{\sum_{j \in \mathcal{S}_+} p_{k,j} \exp(\eta_k \widehat{g}_{k,j})}, \quad i \in \mathcal{S}_+. \tag{53}$$

A first-order expansion gives

$$p_{k+1,i} - p_{k,i} = \eta_k p_{k,i} \left(\widehat{g}_{k,i} - \sum_{j \in \mathcal{S}_+} p_{k,j} \widehat{g}_{k,j} \right) + O(\eta_k^2). \tag{54}$$

Substituting (51) into (54) gives

$$\begin{aligned} p_{k+1,i} - p_{k,i} = {} & \eta_k p_{k,i} \left(\mathbb{E}[\widehat{g}_{k,i} \mid \mathcal{F}_k] - \sum_{j \in \mathcal{S}_+} p_{k,j} \mathbb{E}[\widehat{g}_{k,j} \mid \mathcal{F}_k] \right) \\ & + \eta_k p_{k,i} \left(\delta_{k+1,i} - \sum_{j \in \mathcal{S}_+} p_{k,j} \delta_{k+1,j} \right) + O(\eta_k^2). \end{aligned} \tag{55}$$

Using the expected-gradient form derived in Appendix A.2, we further obtain that

$$\begin{aligned} p_{k+1,i} - p_{k,i} = {} & \eta_k c_k p_{k,i} \left(p_{k,i} - Q(\boldsymbol{p}_k) \right) \\ & + \eta_k p_{k,i} \left(\delta_{k+1,i} - \sum_{j \in \mathcal{S}_+} p_{k,j} \delta_{k+1,j} \right) + O(\eta_k^2). \end{aligned} \tag{56}$$

where $c_k > 0$ denotes the corresponding response-independent scaling coefficient of the expected gradient, for example, for REINFORCE-style update rule detailed in (31), $c_k = R_k(1 - R_k)$.

For simplicity, denote the noise term in the update of $p_{k,i}$ above by

$$\xi_{k+1,i} = p_{k,i} \left(\delta_{k+1,i} - \sum_{j \in \mathcal{S}_+} p_{k,j} \delta_{k+1,j} \right). \tag{57}$$

Since $\boldsymbol{p}_k$ is fixed conditional on $\mathcal{F}_k$ and $\mathbb{E}[\boldsymbol{\delta}_{k+1} \mid \mathcal{F}_k] = \mathbf{0}$, it is straightforward that

$$\mathbb{E}[\boldsymbol{\xi}_{k+1} \mid \mathcal{F}_k] = \mathbf{0}. \tag{58}$$

Using the function f defined in Appendix A.2, the update expanded in (54) can therefore be written compactly as

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \eta_k c_k f(\boldsymbol{p}_k) + \eta_k \boldsymbol{\xi}_{k+1} + \boldsymbol{\rho}_{k+1}, \tag{59}$$

where $\boldsymbol{\rho}_{k+1}$ collects the second-order terms and satisfies

$$\|\boldsymbol{\rho}_{k+1}\| = O(\eta_k^2). \tag{60}$$

Thus, the finite-sample update consists of the deterministic update term $(f(\boldsymbol{p}_k))$ analyzed in Appendix A.2, together with a zero-mean finite-sample perturbation $(\boldsymbol{\xi}_{k+1})$ and a second-order remainder $(\boldsymbol{\rho}_{k+1})$.

(59) separates the finite-sample update into the deterministic update analyzed in Appendix A.2 and the perturbation introduced by finite-sample gradient estimation. We first characterize the effect of this stochastic perturbation over the update process. In particular, we show that under diminishing step sizes, the accumulated sampling noise and the second-order remainder remain asymptotically controlled, so that the long-term evolution of $\boldsymbol{p}_k$ continues to follow the deterministic update trend identified in Appendix A.2. We then examine whether this deterministic effect accumulates sufficiently over training to drive the policy toward concentration on a single successful response.

Under the standard analysis framework introduced in Appendix A.1, the stochastic analysis below requires an additional condition on the learning-rate schedule.

Assumption 1 (Robbins-Monro conditions)**.** *The learning rates satisfy the Robbins–Monro conditions*

$$\sum_{k=0}^{\infty} \eta_k = \infty, \quad \sum_{k=0}^{\infty} \eta_k^2 < \infty. \tag{61}$$

These conditions are standard in stochastic approximation (Robbins & Monro, 1951). The first condition ensures that the cumulative update does not vanish too quickly, while the second controls the accumulated effect of stochastic gradient noise.

We first examine the accumulated effect of the finite-sample perturbation in (59). At each iteration, the sampling noise perturbs the policy update away from its expected direction. To relate the resulting stochastic process to the deterministic dynamics analyzed in Appendix A.2, we need to control how these perturbations accumulate over training. Under the Robbins-Monro learning-rate schedule, we show that both the accumulated sampling noise and the second-order remainder converge, so that the finite-sample noise does not continuously push the training trajectory away from the deterministic update dynamics analyzed in Appendix A.2.

We first examine the accumulated effect of the finite-sample perturbation in (59). Under the Robbins-Monro learning-rate schedule, both the sampling noise and the second-order remainder have a finite cumulative effect.

Proposition 3. *Under Assumption 1, for the REINFORCE-style finite-sample update,*

$$\sum_{k=0}^{\infty} \eta_k \boldsymbol{\xi}_{k+1} \tag{62}$$

converges almost surely, and

$$\sum_{k=0}^{\infty} \|\boldsymbol{\rho}_{k+1}\| < \infty. \tag{63}$$

Proof. We first consider the sampling-noise term. According to (13), we have

$$\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(y_i \mid x) = \boldsymbol{e}_i - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot \mid x), \tag{64}$$

which is uniformly bounded over all responses and policy parameters. Since the verifier reward is binary and the group size is finite, the REINFORCE-style gradient estimator $\widehat{\boldsymbol{g}}_k$ is therefore uniformly bounded. Consequently, the centered gradient noise

$$\boldsymbol{\delta}_{k+1} = \widehat{\boldsymbol{g}}_k - \mathbb{E}[\widehat{\boldsymbol{g}}_k \mid \mathcal{F}_k]$$

is also uniformly bounded. Recall that

$$\xi_{k+1,i} = p_{k,i}\left(\delta_{k+1,i} - \sum_{j\in\mathcal{S}_+} p_{k,j}\delta_{k+1,j}\right). \tag{65}$$

Since $\boldsymbol{p}_k$ lies in the probability simplex, there exists a constant $C > 0$ such that

$$\mathbb{E}\big[\|\boldsymbol{\xi}_{k+1}\|^2 \mid \mathcal{F}_k\big] \le C. \tag{66}$$

Moreover, we know that

$$\mathbb{E}[\boldsymbol{\xi}_{k+1} \mid \mathcal{F}_k] = \mathbf{0}. \tag{67}$$

Define the partial sums

$$\boldsymbol{M}_n = \sum_{k=0}^{n-1} \eta_k \boldsymbol{\xi}_{k+1}. \tag{68}$$

Then $\{\boldsymbol{M}_n, \mathcal{F}_n\}$ is a martingale. Using (66), it holds that

$$\mathbb{E}\|\boldsymbol{M}_n\|^2 = \sum_{k=0}^{n-1} \eta_k^2 \mathbb{E}\|\boldsymbol{\xi}_{k+1}\|^2 \le C\sum_{k=0}^{\infty} \eta_k^2 < \infty, \tag{69}$$

where the cross terms vanish because $\boldsymbol{\xi}_{k+1}$ is a martingale difference. Hence $\{\boldsymbol{M}_n\}$ is bounded in L^2. Then by the martingale convergence theorem, we have

$$\boldsymbol{M}_n \to \boldsymbol{M}_\infty \qquad \text{almost surely.} \tag{70}$$

Equivalently, the tail contribution of the sampling noise vanishes:

$$\lim_{K\to\infty} \sup_{n\ge K} \left\| \sum_{k=K}^{n} \eta_k \boldsymbol{\xi}_{k+1} \right\| = 0 \qquad \text{almost surely.} \tag{71}$$

For the second-order remainder, (59) gives

$$\|\boldsymbol{\rho}_{k+1}\| \le C_\rho \eta_k^2$$

for some constant $C_\rho > 0$. Therefore, according to Assumption 1, it holds that

$$\sum_{k=0}^{\infty} \|\boldsymbol{\rho}_{k+1}\| \le C_\rho \sum_{k=0}^{\infty} \eta_k^2 < \infty \tag{72}$$

□

Proposition 3 shows that the cumulative effect of the finite-sample gradient noise becomes negligible in the later stage of training. We next show that the expected policy gradient component continues to accumulate over training and therefore asymptotically dominates the policy update dynamics.

For the REINFORCE-style update, the scaling coefficient of the expected gradient is $c_k = R_k(1 - R_k)$ as analyzed in (25). We next show that its cumulative effect diverges:

$$\sum_{k=0}^{\infty} \eta_k R_k(1 - R_k) = \infty \qquad \text{almost surely.} \tag{73}$$

Proposition 4. *Under Assumption 1, consider the REINFORCE-style update based on a finite-sample policy-gradient estimate. Then,*

$$\sum_{k=0}^{\infty} \eta_k c_k = \sum_{k=0}^{\infty} \eta_k R_k(1 - R_k) = \infty \qquad \textit{almost surely.} \tag{74}$$

Moreover,

$$R_k \longrightarrow 1 \qquad \textit{almost surely.} \tag{75}$$

Proof. To track the evolution of the total successful probability, we additionally define the index set of unsuccessful responses as

$$\mathcal{S}_- = \{j : r_j = 0\},$$

and their conditional distribution as

$$q_{k,j} = \frac{\pi_{k,j}}{1 - R_k}, \quad j \in \mathcal{S}_-. \tag{76}$$

Since all logits are finite at every finite iteration, we have $0 < R_k < 1$. We therefore consider the log-odds of generating a successful response,

$$\ell_k = \log \frac{R_k}{1 - R_k} = \log \sum_{i \in \mathcal{S}_+} e^{\theta_{k,i}} - \log \sum_{j \in \mathcal{S}_-} e^{\theta_{k,j}}. \tag{77}$$

For the REINFORCE-style estimator, the expected gradient on a successful coordinate $i \in \mathcal{S}_+$ is

$$\mathbb{E}[\widehat{g}^{\mathrm{R}}_{k,i} \mid \mathcal{F}_k] = R_k(1 - R_k)p_{k,i}, \tag{78}$$

as derived in Appendix A.2. Similarly, for an unsuccessful coordinate $j \in \mathcal{S}_-$, we have

$$\mathbb{E}[\widehat{g}^{\mathrm{R}}_{k,j} \mid \mathcal{F}_k] = -R_k \cdot \pi_{k,j} = -R_k(1 - R_k)q_{k,j}. \tag{79}$$

Applying a first-order Taylor expansion to the two log-sum-exp terms in (77) gives

$$\ell_{k+1} - \ell_k = \eta_k \left(\sum_{i \in \mathcal{S}_+} p_{k,i}\widehat{g}^{\mathrm{R}}_{k,i} - \sum_{j \in \mathcal{S}_-} q_{k,j}\widehat{g}^{\mathrm{R}}_{k,j} \right) + r_{k+1}, \tag{80}$$

where the Taylor remainder satisfies

$$|r_{k+1}| \le C\eta_k^2 \tag{81}$$

for some constant $C > 0$. This follows because the softmax score and the finite-group REINFORCE gradient are uniformly bounded.

Taking the conditional expectation of the first-order term and using (78) and (79), we obtain

$$\mathbb{E}\left[\sum_{i \in \mathcal{S}_+} p_{k,i}\widehat{g}^{\mathrm{R}}_{k,i} - \sum_{j \in \mathcal{S}_-} q_{k,j}\widehat{g}^{\mathrm{R}}_{k,j} \,\middle|\, \mathcal{F}_k \right] = R_k(1 - R_k) \left(\sum_{i \in \mathcal{S}_+} p_{k,i}^2 + \sum_{j \in \mathcal{S}_-} q_{k,j}^2 \right). \tag{82}$$

Recall that $Q(\boldsymbol{p}_k) = \sum_{i \in \mathcal{S}_+} p_{k,i}^2$, and similarly define $Q(\boldsymbol{q}_k) = \sum_{j \in \mathcal{S}_-} q_{k,j}^2$. We denote the deviation of the first-order term from its conditional expectation by

$$\zeta_{k+1} = \sum_{i \in \mathcal{S}_+} p_{k,i}\widehat{g}^{\mathrm{R}}_{k,i} - \sum_{j \in \mathcal{S}_-} q_{k,j}\widehat{g}^{\mathrm{R}}_{k,j} - R_k(1 - R_k)\left(Q(\boldsymbol{p}_k) + Q(\boldsymbol{q}_k)\right). \tag{83}$$

By construction,

$$\mathbb{E}[\zeta_{k+1} \mid \mathcal{F}_k] = 0. \tag{84}$$

Moreover, ζ_{k+1} is uniformly bounded under the finite-response setting. Hence, by the same martingale-convergence argument used in Proposition 3,

$$\sum_{k=0}^{\infty} \eta_k \zeta_{k+1} \tag{85}$$

converges almost surely.

Combining the above results, (80) becomes

$$\ell_{k+1} - \ell_k = \eta_k R_k(1 - R_k)\left(Q(\boldsymbol{p}_k) + Q(\boldsymbol{q}_k)\right) + \eta_k \zeta_{k+1} + r_{k+1}. \tag{86}$$

We first prove (74), we prove this by contradiction. Suppose, to the contrary, that

$$\sum_{k=0}^{\infty} \eta_k R_k(1 - R_k) < \infty. \tag{87}$$

Since $Q(\boldsymbol{p}_k) \leq 1$ and $Q(\boldsymbol{q}_k) \leq 1$, the first term on the right-hand side of (86) would then have a finite cumulative sum. The sampling-noise term also has a convergent cumulative sum, and

$$\sum_{k=0}^{\infty} |r_{k+1}| \leq C \sum_{k=0}^{\infty} \eta_k^2 < \infty$$

by Assumption 1. Therefore, ℓ_k would converge to a finite value.

Since

$$R_k = \frac{e^{\ell_k}}{1 + e^{\ell_k}}, \tag{88}$$

a finite limit of ℓ_k implies that R_k converges to some value strictly between 0 and 1. Consequently, there exist constants $\varepsilon > 0$ and $K < \infty$ such that

$$R_k(1 - R_k) \geq \varepsilon, \quad k \geq K. \tag{89}$$

It follows that

$$\sum_{k=K}^{\infty} \eta_k R_k(1 - R_k) \geq \varepsilon \sum_{k=K}^{\infty} \eta_k = \infty, \tag{90}$$

where the last equality follows from the Robbins-Monro condition $\sum_k \eta_k = \infty$. This contradicts (87). Therefore,

$$\sum_{k=0}^{\infty} \eta_k R_k(1 - R_k) = \infty \qquad \text{almost surely.} \tag{91}$$

It remains to prove that $R_k \to 1$. Since $\boldsymbol{p}_k$ and $\boldsymbol{q}_k$ are probability distributions over finite sets, the Cauchy–Schwarz inequality gives

$$Q(\boldsymbol{p}_k) \geq \frac{1}{|\mathcal{S}_+|}, \quad Q(\boldsymbol{q}_k) \geq \frac{1}{|\mathcal{S}_-|}. \tag{92}$$

Hence

$$Q(\boldsymbol{p}_k) + Q(\boldsymbol{q}_k) \geq \frac{1}{|\mathcal{S}_+|} + \frac{1}{|\mathcal{S}_-|} > 0. \tag{93}$$

Summing (86) over k, the deterministic term therefore diverges to $+\infty$ by (74) and (93), whereas the accumulated sampling-noise and remainder terms converge to finite limits. Consequently,

$$\ell_k \longrightarrow +\infty \quad \text{almost surely.} \tag{94}$$

By (77), this is equivalent to

$$R_k \longrightarrow 1 \qquad \text{almost surely.} \tag{95}$$

□

Recall from (59) that the finite-sample update of $\boldsymbol{p}_k$ consists of the expected-gradient component $\eta_k c_k f(\boldsymbol{p}_k)$ and the finite-sample perturbation terms. Ignoring the perturbations gives

$$\boldsymbol{p}_{k+1} - \boldsymbol{p}_k = \eta_k c_k f(\boldsymbol{p}_k), \tag{96}$$

which corresponds to the continuous-time dynamics $\dot{\boldsymbol{p}} = f(\boldsymbol{p})$ analyzed in Appendix A.2.

We therefore measure the cumulative effect of this component by

$$\tau_n = \sum_{k=0}^{n-1} \eta_k c_k. \tag{97}$$

Proposition 4 shows that $\tau_n \to \infty$ almost surely, while Proposition 3 shows that the cumulative effect of the finite-sample perturbations becomes negligible as training proceeds. We next use these two results to show that $\boldsymbol{p}_k$ can only converge to the distributions characterized in Corollary 1, which are uniform over their nonzero support.

Proposition 5 (Concentration under finite-sample updates)**.** *Under Assumption 1, the successful-response distribution under the finite-sample REINFORCE-style update converges almost surely to a distribution that is uniform over its nonzero support. Specifically, there exists a nonempty subset $\mathcal{S} \subseteq \mathcal{S}_+$ such that*

$$\boldsymbol{p}_k \longrightarrow \boldsymbol{p}^{\mathcal{S}} \qquad \textit{almost surely}, \tag{98}$$

where

$$p_i^{\mathcal{S}} = \begin{cases} 1/|\mathcal{S}|, & i \in \mathcal{S}, \\ 0, & i \notin \mathcal{S}. \end{cases} \tag{99}$$

Proof. Recall from (59) that

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \alpha_k f(\boldsymbol{p}_k) + \boldsymbol{u}_{k+1}, \tag{100}$$

where $\alpha_k = \eta_k c_k$ and $\boldsymbol{u}_{k+1} = \eta_k \boldsymbol{\xi}_{k+1} + \boldsymbol{\rho}_{k+1}$. As we defined above,

$$\tau_n = \sum_{k=0}^{n-1} \alpha_k. \tag{101}$$

By Proposition 4, $\tau_n \to \infty$ almost surely, while $\alpha_k \to 0$. Moreover, Proposition 3 gives

$$\lim_{n\to\infty} \sup_{m \geq n} \left\| \sum_{k=n}^{m-1} \boldsymbol{u}_{k+1} \right\| = 0 \qquad \text{almost surely.} \tag{102}$$

Hence, over sufficiently late updates, the cumulative perturbation becomes arbitrarily small, whereas the deterministic component continues to evolve over an unbounded accumulated step size. It remains to show that the discrete deterministic increments $\alpha_k f(\boldsymbol{p}_k)$ approximate the continuous-time dynamics analyzed in Appendix A.2:

$$\dot{\boldsymbol{p}} = f(\boldsymbol{p}). \tag{103}$$

To make this comparison, let $\bar{\boldsymbol{p}}(\tau)$ be the piecewise-linear interpolation of the iterates on the time scale $\{\tau_k\}$, satisfying $\bar{\boldsymbol{p}}(\tau_k) = \boldsymbol{p}_k$. Let $\Phi_h(\boldsymbol{p})$ denote the solution of (103) after time h when initialized at $\boldsymbol{p}$.

Fix any $T > 0$. Starting from iteration n, consider all subsequent updates whose accumulated deterministic step size is at most T. Over this interval, the difference between the stochastic trajectory and the deterministic flow has two sources.

First, the cumulative finite-sample perturbation is bounded by

$$\epsilon_n(T) = \sup_{\substack{m \geq n \\ \tau_m - \tau_n \leq T}} \left\| \sum_{k=n}^{m-1} \boldsymbol{u}_{k+1} \right\|, \tag{104}$$

and (102) implies $\epsilon_n(T) \to 0$ almost surely.

Second, the deterministic sum $\sum_{k=n}^{m-1} \alpha_k f(\boldsymbol{p}_k)$ is the Euler approximation of the integral generated by (103). Since f is polynomial in $\boldsymbol{p}$, it is Lipschitz on the compact probability simplex. Moreover, $\max_{k \geq n} \alpha_k \to 0$, so the Euler discretization error over any fixed time interval $[0, T]$ also vanishes.

Therefore, for some constants $L, C > 0$ independent of n, the standard Gronwall bound gives

$$\sup_{0 \leq h \leq T} \|\bar{\boldsymbol{p}}(\tau_n + h) - \Phi_h(\boldsymbol{p}_n)\| \leq e^{LT} \left[\epsilon_n(T) + CT \max_{k \geq n} \alpha_k \right]. \tag{105}$$

Both terms on the right-hand side converge to zero almost surely. Hence, for every fixed $T > 0$,

$$\lim_{n\to\infty} \sup_{0 \leq h \leq T} \|\bar{\boldsymbol{p}}(\tau_n + h) - \Phi_h(\boldsymbol{p}_n)\| = 0 \qquad \text{almost surely.} \tag{106}$$

(106) states that, over every fixed future time interval, the finite-sample policy update trajectory becomes arbitrarily close to the deterministic trajectory when training is sufficiently advanced. In stochastic approximation terminology, $\bar{\boldsymbol{p}}$ is therefore an asymptotic pseudotrajectory of the flow generated by f.

We can now use the deterministic analysis in Appendix A.2. By the asymptotic-pseudotrajectory result of Benaïm (2006), the limit set of $\bar{\boldsymbol{p}}$ is an internally chain-transitive set of the deterministic flow. Intuitively, the stochastic process can therefore converge only to a set that is compatible with the long-term behavior of the deterministic dynamics.

Recall that the Gini-Simpson diversity define above

$$D_{\mathrm{GS}}(\boldsymbol{p}) = 1 - \sum_{i \in \mathcal{S}_+} p_i^2 \tag{107}$$

is a strict Lyapunov function for this flow: it strictly decreases whenever $\boldsymbol{p}$ is not uniform over its nonzero support. Hence the only stationary distributions are

$$\mathcal{E} = \left\{ \boldsymbol{p}^{\mathcal{S}} : \emptyset \neq \mathcal{S} \subseteq \mathcal{S}_+ \right\}, \tag{108}$$

where

$$p_i^{\mathcal{S}} = \begin{cases} 1/|\mathcal{S}|, & i \in \mathcal{S}, \\ 0, & i \notin \mathcal{S}. \end{cases} \tag{109}$$

Moreover,

$$D_{\mathrm{GS}}(\boldsymbol{p}^{\mathcal{S}}) = 1 - \frac{1}{|\mathcal{S}|}, \tag{110}$$

so D_{GS} takes only finitely many values on $\mathcal{E}$. The strict-Lyapunov criterion of Benaïm (2006) therefore implies that the limit set of the stochastic trajectory is contained in $\mathcal{E}$.

Since $\mathcal{E}$ is finite and consists entirely of stationary points, an internally chain-transitive subset of $\mathcal{E}$ can contain only one point. Consequently, there exists a possibly random nonempty subset $\mathcal{S}_\infty \subseteq \mathcal{S}_+$ such that

$$\boldsymbol{p}_k \longrightarrow \boldsymbol{p}^{\mathcal{S}_\infty} \qquad \text{almost surely.} \tag{111}$$

This proves the proposition. □

Proposition 5 restricts the possible limit of $\boldsymbol{p}_k$ to distributions that are uniform over their nonzero support. However, this set still includes distributions supported on multiple successful responses. We next show that the finite-sample update cannot converge to any such distribution with positive probability.

Recall that Appendix A.2 shows that every stationary distribution supported on more than one successful response is unstable: a perturbation that redistributes probability within its support is amplified by the expected policy-gradient update.

To exclude convergence to these unstable stationary distributions, it remains to verify that finite-sample sampling continues to introduce non-vanishing random perturbations along their unstable directions. Once this condition is established, a standard non-convergence result for stochastic approximation implies that the probability of converging to any such stationary distribution is zero.

Proposition 6. *For any nonempty subset $\mathcal{S} \subseteq \mathcal{S}_+$ with $|\mathcal{S}| \geq 2$,*

$$\Pr\left(\boldsymbol{p}_k \longrightarrow \boldsymbol{p}^{\mathcal{S}}\right) = 0. \tag{112}$$

Proof. Fix $\mathcal{S} \subseteq \mathcal{S}_+$ with $s = |\mathcal{S}| \geq 2$, and define the event

$$\Gamma_{\mathcal{S}} = \{\boldsymbol{p}_k \to \boldsymbol{p}^{\mathcal{S}}\}. \tag{113}$$

Next we prove that $\Pr(\Gamma_{\mathcal{S}}) = 0$.

If $\Gamma_{\mathcal{S}}$ occurs, then it holds that the probabilities of all responses in $\mathcal{S}$ become equal. According to (11), we have

$$\frac{p_{k,i}}{p_{k,j}} = \exp(\theta_{k,i} - \theta_{k,j}), \qquad i, j \in \mathcal{S}, \tag{114}$$

this requires all relative logits within $\mathcal{S}$ to converge to zero, i.e.,

$$\theta_{k,i} - \theta_{k,j} \longrightarrow 0, \qquad i, j \in \mathcal{S}. \tag{115}$$

To study these relative logits, let

$$P_{\mathcal{S}} = I_s - \frac{1}{s}\mathbf{1}\mathbf{1}^\top, \qquad \boldsymbol{x}_k = P_{\mathcal{S}}\boldsymbol{\theta}_{k,\mathcal{S}}. \tag{116}$$

where I_s denote the identity matrix of order s, and $\mathbf{1} \in \mathbb{R}^s$ denotes the all-ones vector.

Hence, when $\Gamma_{\mathcal{S}}$ occurs, we can further prove that

$$\boldsymbol{x}_k \to \mathbf{0}. \tag{117}$$

Recall that the REINFORCE-style policy update is defined as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta_k \widehat{\boldsymbol{g}}_k^{\mathrm{R}}. \tag{118}$$

Since $\boldsymbol{x}_k = P_{\mathcal{S}}\boldsymbol{\theta}_{k,\mathcal{S}}$, projecting the coordinates in $\mathcal{S}$ gives

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = \eta_k P_{\mathcal{S}} \widehat{\boldsymbol{g}}_{k,\mathcal{S}}^{\mathrm{R}}. \tag{119}$$

We decompose the finite-sample gradient into its conditional expectation and sampling error:

$$\widehat{\boldsymbol{g}}_{k,\mathcal{S}}^{\mathrm{R}} = \mathbb{E}[\widehat{\boldsymbol{g}}_{k,\mathcal{S}}^{\mathrm{R}} \mid \mathcal{F}_k] + \left(\widehat{\boldsymbol{g}}_{k,\mathcal{S}}^{\mathrm{R}} - \mathbb{E}[\widehat{\boldsymbol{g}}_{k,\mathcal{S}}^{\mathrm{R}} \mid \mathcal{F}_k]\right). \tag{120}$$

Using $\mathbb{E}[\widehat{\boldsymbol{g}}_{k,\mathcal{S}}^{\mathrm{R}} \mid \mathcal{F}_k] = c_k \boldsymbol{p}_{k,\mathcal{S}}$, (119) can be rewrote as

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = \eta_k \left(c_k P_{\mathcal{S}} \boldsymbol{p}_{k,\mathcal{S}} + \boldsymbol{\epsilon}_{k+1}\right), \tag{121}$$

where $\boldsymbol{\epsilon}_{k+1} = P_{\mathcal{S}}\left(\widehat{\boldsymbol{g}}_{k,\mathcal{S}}^{\mathrm{R}} - \mathbb{E}[\widehat{\boldsymbol{g}}_{k,\mathcal{S}}^{\mathrm{R}} \mid \mathcal{F}_k]\right)$ is the projected finite-sample gradient noise and satisfies

$$\mathbb{E}[\boldsymbol{\epsilon}_{k+1} \mid \mathcal{F}_k] = \mathbf{0}. \tag{122}$$

To show that $\boldsymbol{x}_k$ cannot converge to $\mathbf{0}$ with positive probability, we verify two properties: the expected-gradient component does not drive $\boldsymbol{x}_k$ toward zero, while the finite-sample gradient noise remains non-degenerate around zero.

First, consider the expected-gradient component. Since $\boldsymbol{x}_k = P_{\mathcal{S}}\boldsymbol{\theta}_{k,\mathcal{S}}$, we have $\mathbf{1}^\top \boldsymbol{x}_k = 0$. Let

$$m_{\mathcal{S},k} = \sum_{i \in \mathcal{S}} p_{k,i}.$$

Then for any $i \in \mathcal{S}$, the probabilities within $\mathcal{S}$ can be written as

$$p_{k,i} = m_{\mathcal{S},k} \frac{\exp(x_{k,i})}{\sum_{j \in \mathcal{S}} \exp(x_{k,j})}. \tag{123}$$

Therefore,

$$\langle \boldsymbol{x}_k, c_k P_{\mathcal{S}} \boldsymbol{p}_{k,\mathcal{S}} \rangle = c_k \langle \boldsymbol{x}_k, \boldsymbol{p}_{k,\mathcal{S}} \rangle = c_k m_{\mathcal{S},k} \frac{\sum_{i \in \mathcal{S}} x_{k,i} \exp(x_{k,i})}{\sum_{j \in \mathcal{S}} \exp(x_{k,j})} \geq 0. \tag{124}$$

The last inequality follows from $\sum_{i \in \mathcal{S}} x_{k,i} = 0$ and the convexity of the log-sum-exp function. (124) shows that the expected-gradient update does not drive the relative logits toward zero. Moreover, the inequality is strict whenever $\boldsymbol{x}_k \neq \mathbf{0}$, implying that any existing imbalance among the logits in $\mathcal{S}$ is amplified rather than corrected.

We next examine the finite-sample noise. Let $\boldsymbol{v} \in \mathbb{R}^s$ be any unit vector satisfying $\mathbf{1}^\top \boldsymbol{v} = 0$, i.e., any direction that changes the relative logits within $\mathcal{S}$. On the event $\Gamma_{\mathcal{S}}$,

$$\boldsymbol{p}_k \to \boldsymbol{p}^{\mathcal{S}}, \qquad R_k \to 1, \tag{125}$$

where the second convergence follows from Proposition 4. Hence, asymptotically, a sampled group consists of G draws from the uniform distribution over the s responses in $\mathcal{S}$.

When the event $\Gamma_{\mathcal{S}}$ occurs, we have $\boldsymbol{p}_k \to \boldsymbol{p}^{\mathcal{S}}$ and $R_k \to 1$. Hence, the conditional distribution of the response counts in the sampled group $\mathcal{S}$ converges to

$$(N_1, \ldots, N_s) \sim \text{Multinomial}\left(G, \left(\frac{1}{s}, \ldots, \frac{1}{s}\right)\right). \tag{126}$$

For any unit vector $\boldsymbol{v} \in \mathbb{R}^s$ satisfying $\mathbf{1}^\top \boldsymbol{v} = 0$, the multinomial covariance gives

$$\text{Var}\left(\frac{1}{G}\sum_{i=1}^{s} v_i N_i\right) = \frac{1}{G}\left[\frac{1}{s}\sum_{i=1}^{s} v_i^2 - \frac{1}{s^2}\left(\sum_{i=1}^{s} v_i\right)^2\right] = \frac{1}{Gs}, \tag{127}$$

where the last equality follows from $\|\boldsymbol{v}\|^2 = \sum_{i=1}^{s} v_i^2 = 1$ and $\mathbf{1}^\top \boldsymbol{v} = \sum_{i=1}^{s} v_i = 0$.

Since $\boldsymbol{v}^\top P_{\mathcal{S}} = \boldsymbol{v}^\top$, we have

$$\boldsymbol{v}^\top \boldsymbol{\epsilon}_{k+1} = \boldsymbol{v}^\top \widehat{\boldsymbol{g}}^R_{k,\mathcal{S}} - \mathbb{E}[\boldsymbol{v}^\top \widehat{\boldsymbol{g}}^R_{k,\mathcal{S}} \mid \mathcal{F}_k]. \tag{128}$$

By the definition of the REINFORCE-style estimator,

$$\boldsymbol{v}^\top \widehat{\boldsymbol{g}}^{\mathrm{R}}_{k,\mathcal{S}} = \frac{1}{G}\sum_{g=1}^{G} r_k^{(g)}\left[\sum_{i\in\mathcal{S}} v_i \mathbb{I}\{Y_k^{(g)} = y_i\} - \sum_{i\in\mathcal{S}} v_i \pi_{k,i}\right]. \tag{129}$$

According to the definition of event $\Gamma_{\mathcal{S}}$ and the result that $R_k \to 1$, we have $\pi_{k,i} \to \frac{1}{s}$ for any $i \in \mathcal{S}$, then combining with the definition of $\boldsymbol{v}$, it is straightforward to get

$$\boldsymbol{v}^\top \widehat{\boldsymbol{g}}^{\mathrm{R}}_{k,\mathcal{S}} \to \frac{1}{G}\sum_{i=1}^{s} v_i N_i$$

Therefore,

$$\lim_{k\to\infty} \mathbb{E}\big[(\boldsymbol{v}^\top \boldsymbol{\epsilon}_{k+1})^2 \mid \mathcal{F}_k\big] = \text{Var}\left(\frac{1}{G}\sum_{i=1}^{s} v_i N_i\right) = \frac{1}{Gs} > 0. \tag{130}$$

where the first equation holds for $\mathbb{E}[\boldsymbol{v}^\top \boldsymbol{\epsilon}_{k+1} \mid \mathcal{F}_k] = 0$

Thus, even when the policy approaches $\boldsymbol{p}^{\mathcal{S}}$, finite-sample gradient estimation continues to introduce non-vanishing random perturbations in every direction that changes the relative logits within $\mathcal{S}$.

Together, the two results analyzed in (124) and (130) above characterize the behavior of the update near $\boldsymbol{x} = \mathbf{0}$($\boldsymbol{x}$ defined in (116)). The expected policy gradient does not provide a driving force that reduces the differences among the logits in $\mathcal{S}$, while finite-sample sampling continues to introduce random perturbations along these directions even when the policy approaches the uniform distribution $\boldsymbol{p}^{\mathcal{S}}$. These are precisely the two key conditions required by the non-convergence result for unstable equilibria (Raimond & Tarres, 2023, Theorem 2.2.1).

Moreover, the finite response space and finite group size imply uniformly bounded noise moments, and the learning rates satisfy the Robbins–Monro conditions. The theorem therefore yields

$$\Pr\left(\boldsymbol{x}_k \to \mathbf{0}\right) = 0.$$

Since $\boldsymbol{p}_k \to \boldsymbol{p}^{\mathcal{S}}$ necessarily implies $\boldsymbol{x}_k \to \mathbf{0}$, we conclude that

$$\Pr\left(\boldsymbol{p}_k \to \boldsymbol{p}^{\mathcal{S}}\right) = 0.$$

This proves the proposition. □

Finally, based on all of the results above, we will introduce the main theorem that formalizes the tendency of diversity collapse during training.

Recall the definition of semantic class in Section 3, we have the following theorem holds.

Theorem 1 (semantic diversity collapse)**.** *Under the Robbins-Monro conditions and the regularity conditions stated in Appendix A.3, there exists a random semantic class $C_\infty \in \mathcal{C}_x$ such that*

$$\nu^+_{k,c}(x) \longrightarrow \mathbf{1}\{c = C_\infty\},\ \forall c \in \mathcal{C}_x, \quad \textit{almost surely} \tag{6}$$

Proof of Theorem 1. We first establish that the successful-response distribution almost surely concentrates on a single response. We then aggregate this response-level result according to the semantic mapping ψ_x to obtain the semantic collapse stated in Theorem 1.

Lemma 1 (Almost-sure response collapse)**.** *Under the conditions stated above, there exists a random successful-response index $I_\infty \in \mathcal{S}_+$ such that*

$$\boldsymbol{p}_k \longrightarrow \boldsymbol{e}_{I_\infty} \qquad \textit{almost surely}. \tag{131}$$

Proof. Proposition 5 shows that $\boldsymbol{p}_k$ converges almost surely to $\boldsymbol{p}^{\mathcal{S}_\infty}$ for some nonempty random subset $\mathcal{S}_\infty \subseteq \mathcal{S}_+$.

Proposition 6 further shows that, for every $\mathcal{S} \subseteq \mathcal{S}_+$ with $|\mathcal{S}| \geq 2$,

$$\Pr\big(\boldsymbol{p}_k \to \boldsymbol{p}^{\mathcal{S}}\big) = 0.$$

Since $\mathcal{S}_+$ is finite, there are only finitely many such subsets. Therefore,

$$\Pr(|\mathcal{S}_\infty| \geq 2) = 0.$$

Hence $\mathcal{S}_\infty = \{I_\infty\}$ almost surely for some random $I_\infty \in \mathcal{S}_+$, which gives (131). □

We now translate the response-level concentration in Lemma 1 to the semantic level.

Recall the semantic mapping

$$\psi_x : \mathcal{Y}(x) \to \mathcal{C}_x,$$

where $\psi_x(y)$ denotes the semantic class of response y.

Accordingly, the success-conditional probability assigned to semantic class $c \in \mathcal{C}_x$ can be written as

$$\nu^+_{k,c}(x) = \sum_{\substack{i \in \mathcal{S}_+ \\ \psi_x(y_i) = c}} p_{k,i}. \tag{132}$$

Let $C_\infty = \psi_x(y_{I_\infty}) \in \mathcal{C}_x$ be the semantic class of the limiting response. For every $c \in \mathcal{C}_x$, Lemma 1 gives $p_{k,i} \to \mathbb{I}\{i = I_\infty\}$ almost surely for each $i \in \mathcal{S}_+$. Since $\mathcal{S}_+$ is finite,

$$\lim_{k\to\infty} \nu^+_{k,c}(x) = \sum_{\substack{i \in \mathcal{S}_+ \\ \psi_x(y_i) = c}} \lim_{k\to\infty} p_{k,i} = \sum_{\substack{i \in \mathcal{S}_+ \\ \psi_x(y_i) = c}} \mathbb{I}\{i = I_\infty\} = \mathbb{I}\{c = C_\infty\}, \qquad \text{almost surely} \tag{133}$$

This proves Theorem 1.

Remark 1. *For the GRPO-style update, we only consider **effective** groups that contain both successful and unsuccessful responses, since a group containing only successful responses or only unsuccessful responses has zero group-relative advantages and therefore produces no policy update. Accordingly, throughout the stochastic analysis of GRPO, k indexes only these effective groups. Conditional on an effective group, the expected GRPO gradient differs from the REINFORCE-style expected gradient only by a positive response-independent scaling coefficient. Hence, the same expected update direction and stochastic analysis apply under this effective-group indexing.*

A.4 Semantic Coverage under Prior Shift

The collapse result above concerns the distribution among successful outputs. We now explain why this distribution can affect robustness even when two policies have the same source-domain reward. The argument has two steps. We first decompose reward into the prior of each semantic class and the policy's success rate within that class. We then use Bayes' rule to show that, under a uniform source prior, these class-wise success rates are proportional to the successful semantic profile. A shift in the semantic prior can therefore expose whichever successful class is least covered.

Let $C \in \{c_1, \dots, c_K\}$ denote the correct semantic class($\bigcup_x \mathcal{C}_x$ in above). For a policy $\pi_{\boldsymbol{\theta}}$, its expected reward on a domain $\mathcal{D}$ can be equivalently written as

$$J_{\mathcal{D}}(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{(x,c)\sim\mathcal{D},\, y\sim\pi_{\boldsymbol{\theta}}(\cdot|x)}\big[r(x,y)\big] = \sum_{i=1}^{K} \Pr_{\mathcal{D}}(c_i)\, \mathbb{E}_{\substack{x\sim\mathcal{D}(\cdot|c_i),\\ y\sim\pi_{\boldsymbol{\theta}}(\cdot|x)}}\big[r(x,y)\big] = \sum_{i=1}^{K} \Pr_{\mathcal{D}}(c_i) \Pr_{\mathcal{D},\pi_{\boldsymbol{\theta}}}\big(r=1 \mid c_i\big). \tag{134}$$

(134) separates the semantic composition of the data distribution from the policy's success on each semantic class. For a fixed domain, $\Pr_{\mathcal{D}}(C)$ is determined by the data and is therefore invariant to the policy being evaluated. Now, we consider a semantic prior shift,

$$\Pr_{\mathcal{D}}(X \mid C = c_i) = \Pr_{\mathcal{D}'}(X \mid C = c_i), \quad \forall i = 1, \dots, K \tag{135}$$

(135) indicates that the prompt distribution conditioned on each semantic class remains unchanged across domains, while the relative prevalence of the classes may vary.

As a regularity condition for Theorem 2, We further assume that the source domain has a uniform semantic prior,

$$\Pr_{\mathcal{D}_{\mathrm{src}}}(C = c_i) = \frac{1}{K}, \quad \forall i = 1, \dots, K. \tag{136}$$

Under this shift, the policy's success probability within each semantic class is unchanged; only how frequently different classes are encountered varies across domains.

To isolate the effect of semantic coverage, we compare policies that achieve the same performance on the source domain. For a given constant $\bar{J}$, consider every policy $\pi_{\boldsymbol{\theta}}$ satisfies

$$J_{\mathcal{D}_{\mathrm{src}}}(\pi_{\boldsymbol{\theta}}) = \bar{J} > 0. \tag{137}$$

By Bayes' rule, we have

$$\Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\boldsymbol{\theta}}}(C = c_i \mid r(x,y) = 1) = \frac{\Pr_{\mathcal{D}_{\mathrm{src}}}(c_i)\, \Pr_{\pi_{\boldsymbol{\theta}}}(r(x,y) = 1 \mid c_i)}{J_{\mathcal{D}_{\mathrm{src}}}(\pi_{\boldsymbol{\theta}})} = \frac{\Pr_{\pi_{\boldsymbol{\theta}}}(r(x,y) = 1 \mid c_i)}{K\bar{J}}. \tag{138}$$

Thus, among policies with equal source reward, a weakly covered successful class is exactly a class with a low success rate. Substituting this identity into (134) reduces the robustness question to a linear optimization over the target semantic prior.

Theorem 2 (Worst-case OOD robustness)**.** *Under the regularity conditions stated in Appendix A.4, consider two policies $\pi_{\boldsymbol{\theta}_1}$ and $\pi_{\boldsymbol{\theta}_2}$ with the same source reward. If $\pi_{\boldsymbol{\theta}_1}$ has more balanced successful semantic coverage than $\pi_{\boldsymbol{\theta}_2}$, as formalized in Appendix A.4, then*

$$\inf_{\substack{\boldsymbol{q}\in\Delta_K\\ q_i>0,\ \forall i}} J_{\mathcal{D}_{\mathrm{tgt}}}(\pi_{\boldsymbol{\theta}_1}) \geq \inf_{\substack{\boldsymbol{q}\in\Delta_K\\ q_i>0,\ \forall i}} J_{\mathcal{D}_{\mathrm{tgt}}}(\pi_{\boldsymbol{\theta}_2}) \tag{7}$$

where $\boldsymbol{q} = (q_1, \dots, q_K) \in \Delta_K$, and $q_i = \Pr_{\mathcal{D}_{\mathrm{tgt}}}(C = c_i)$ denote the target semantic prior.

Proof. Before proceeding to the proof, we first formalize what it means for one policy to have more balanced successful semantic coverage than another. For a policy $\pi_{\boldsymbol{\theta}}$, let

$$s_{[1]}(\pi_{\boldsymbol{\theta}}) \geq \cdots \geq s_{[K]}(\pi_{\boldsymbol{\theta}})$$

denote the values $\Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\boldsymbol{\theta}}}(C = c_i \mid r = 1)$ sorted in descending order, satisfies that the set $\{\Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\boldsymbol{\theta}}}(C = c_i \mid r = 1)\}_{i=1}^{K}$ is equal to the set $\{s_{[i]}(\pi_{\boldsymbol{\theta}})\}_{i=1}^{K}$.

Definition 1 (Balanced successful semantic coverage)**.** *We say that $\pi_{\boldsymbol{\theta}_1}$ has more balanced successful semantic coverage than $\pi_{\boldsymbol{\theta}_2}$ if*

$$\sum_{i=1}^{m} s_{[i]}(\pi_{\boldsymbol{\theta}_1}) \leq \sum_{i=1}^{m} s_{[i]}(\pi_{\boldsymbol{\theta}_2}), \qquad m = 1, \dots, K-1. \tag{139}$$

Equivalently, the successful semantic distribution of $\pi_{\boldsymbol{\theta}_1}$ is majorized by that of $\pi_{\boldsymbol{\theta}_2}$.

We next characterize the worst-case target reward under the semantic prior shift considered above.

Lemma 2 (Worst-case target reward)**.** *For any policy π_θ satisfying $J_{\mathcal{D}_{\mathrm{src}}}(\pi_{\boldsymbol{\theta}}) = \bar{J} > 0$,*

$$\inf_{\substack{\boldsymbol{q}\in\Delta_K \\ q_i>0,\ \forall i}} J_{\mathcal{D}_{\mathrm{tgt}}}(\pi_{\boldsymbol{\theta}}) = K\bar{J} \min_{i\in[K]} \Pr_{\mathcal{D}_{\mathrm{src}},\pi_\theta}(C = c_i \mid r = 1). \tag{140}$$

Proof. Substituting (138) into (134) gives

$$J_{\mathcal{D}_{\mathrm{tgt}}}(\pi_{\boldsymbol{\theta}}) = K\bar{J} \sum_{i=1}^{K} q_i \Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\boldsymbol{\theta}}}(C = c_i \mid r = 1). \tag{141}$$

Since $\boldsymbol{q} \in \Delta_K$,

$$J_{\mathcal{D}_{\mathrm{tgt}}}(\pi_{\boldsymbol{\theta}}) \geq K\bar{J} \min_{i\in[K]} \Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\boldsymbol{\theta}}}(C = c_i \mid r = 1). \tag{142}$$

To show that the lower bound is tight, consider the following sequence of target priors. Let $i^\star$ attain the minimum on the right-hand side. For any $0 < \epsilon < 1/K$, choose

$$q_{i^\star} = 1 - (K-1)\epsilon, \qquad q_i = \epsilon, \quad i \neq i^\star. \tag{143}$$

As $\epsilon \to 0$, the right-hand side of (141) converges to

$$K\bar{J} \Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\boldsymbol{\theta}}}(C = c_{i^\star} \mid r = 1). \tag{144}$$

Therefore, the lower bound in (142) is attainable as an infimum, which proves (140). □

We now return to the proof of Theorem 2. Suppose that $\pi_{\boldsymbol{\theta}_1}$ has more balanced successful semantic coverage than $\pi_{\boldsymbol{\theta}_2}$. By Definition 1, taking $m = K - 1$ in (139) gives

$$\sum_{i=1}^{K-1} s_{[i]}(\pi_{\boldsymbol{\theta}_1}) \leq \sum_{i=1}^{K-1} s_{[i]}(\pi_{\boldsymbol{\theta}_2}). \tag{145}$$

Since the successful semantic probabilities sum to one, it is straightforward to get

$$s_{[K]}(\pi_{\boldsymbol{\theta}_1}) \geq s_{[K]}(\pi_{\boldsymbol{\theta}_2}).$$

By construction,

$$s_{[K]}(\pi_{\boldsymbol{\theta}}) = \min_{i\in[K]} \Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\boldsymbol{\theta}}}(C = c_i \mid r = 1).$$

Applying Lemma 2 therefore yields

$$\inf_{\substack{\boldsymbol{q}\in\Delta_K \\ q_i>0,\ \forall i}} J_{\mathcal{D}_{\mathrm{tgt}}}(\pi_{\theta_1}) \geq \inf_{\substack{\boldsymbol{q}\in\Delta_K \\ q_i>0,\ \forall i}} J_{\mathcal{D}_{\mathrm{tgt}}}(\pi_{\theta_2}), \tag{146}$$

which proves Theorem 2. □

The exact expression in Lemma 2 also characterizes the optimal successful semantic coverage.

Corollary 2. *Among policies with source reward $\bar{J}$, the worst-case target reward satisfies*

$$\inf_{\substack{\boldsymbol{q}\in\Delta_K \\ q_i>0,\ \forall i}} J_{\mathcal{D}_{\mathrm{tgt}}}(\pi_{\boldsymbol{\theta}}) \leq \bar{J}.$$

Equality holds, whenever feasible, if and only if

$$\Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\boldsymbol{\theta}}}(C = c_i \mid r = 1) = \frac{1}{K}, \qquad \forall i \in [K].$$

Proof. Since the K successful semantic probabilities sum to one,

$$\min_{i\in[K]} \Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\boldsymbol{\theta}}}(C = c_i \mid r = 1) \leq \frac{1}{K},$$

with equality if and only if all K probabilities equal $1/K$. The result then follows directly from Lemma 2. □

Remark 2 (Relation to semantic entropy)**.** *The majorization order in Definition 1 is also consistent with the semantic entropy used to quantify successful-response diversity in Section 3.1. Since Shannon entropy is Schur-concave, if $\pi_{\boldsymbol{\theta}_1}$ has more balanced successful semantic coverage than $\pi_{\boldsymbol{\theta}_2}$ under Definition 1, then their Shannon entropies follow the same ordering, i.e.,*

$$-\sum_{i=1}^{K} \Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\theta_1}}(C = c_i \mid r = 1) \log \Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\theta_1}}(C = c_i \mid r = 1)$$
$$\geq -\sum_{i=1}^{K} \Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\theta_2}}(C = c_i \mid r = 1) \log \Pr_{\mathcal{D}_{\mathrm{src}},\pi_{\theta_2}}(C = c_i \mid r = 1).$$

In particular, the uniform successful semantic distribution simultaneously maximizes the semantic entropy and the worst-case target reward. Note that the converse does not generally hold: a higher semantic entropy alone does not imply a more balanced distribution in the majorization order.

B Algorithmic and Implementation Details

Section 4 presents the main design of SCARS. In this appendix, we provide the complete algorithmic details omitted from the main text. Following the three steps in Section 4, we first describe the construction, initialization, and online maintenance of the semantic tree. We then specify the coverage-aware rollout allocation procedure, including reasoning-mode selection, prefix-node selection, and prefix retrieval. Finally, we detail prefix-conditioned generation and segment-wise credit assignment, and summarize the complete training procedure. Numerical hyperparameters and experimental settings are provided in Appendix D.

B.1 Semantic-Tree Construction and Online Maintenance

For each training prompt x, SCARS maintains a semantic tree $\mathcal{T}_x$ that organizes previously observed responses and their intermediate prefixes. The tree is constructed backward from complete responses to progressively earlier reasoning states: complete responses form high-level semantic modes, while their prefixes are further organized within the corresponding parent modes. This construction follows the hierarchy introduced in Section 4 and is updated throughout training as new rollouts are generated.

Initialization and response processing. Before RLVR training, we sample N_{init} responses from the base model for each prompt. Each response is evaluated by the same verifier used during training and segmented into ordered reasoning units. We then extract cumulative prefixes at different granularities, as defined in Section 4.1. In our experiments, $N_{\mathrm{init}} = 32$. We discard only clearly corrupted responses, including role leakage, cross-problem contamination, malformed tool outputs, degenerate text, and exact duplicates, and resample either until the initialization pool again contains 32 valid responses, or the resample turns up to the rollout budget we set.

Hierarchical construction. Given the initialization responses, we first embed the complete responses using a frozen semantic embedding model and cluster them according to semantic similarity. Each resulting node at the first level represents a high-level reasoning mode. We then repeat the same procedure for prefixes: within each parent node, prefixes at the same granularity are clustered separately to form the next level of the tree. Therefore, responses belonging to different high-level reasoning modes are never merged at the prefix level.

The main text allows a general depth D. In the experiments, we use $D = 3$: d_1 contains complete responses, d_2 contains approximately $3/4$-length prefixes, and d_3 contains approximately $1/4$-length prefixes. Prefix endpoints are moved to the nearest semantic boundary so that paragraphs, sentences, displayed mathematics, and code blocks are not split whenever possible. Thus, as in Section 4.1, moving deeper in the tree corresponds to moving to an earlier reasoning state.

We encode response and prefix text with frozen Qwen3-Embedding-8B and use cosine distance between normalized embeddings. Initialization proceeds from complete responses to shorter prefixes: responses are clustered into d_1, long prefixes are clustered separately within each d_1 parent,

and short prefixes are then clustered within each d_2 parent. All three levels use average-linkage agglomerative clustering,

$$D(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\boldsymbol{z} \in C_i} \sum_{\boldsymbol{z}' \in C_j} d(\boldsymbol{z}, \boldsymbol{z}').$$

Starting from singleton clusters, we merge the closest pair until the nearest distance exceeds the threshold and the level-specific cluster cap is satisfied. If the cap is still exceeded, merging continues even when the next distance is above the threshold. A short prefix may attach directly to its d_1 parent when the corresponding long prefix is unavailable; no node is created below d_3. Exact thresholds and caps are reported in Table 8.

Each node v stores the responses or prefixes assigned to it together with its total number of observed rollouts n_v and successful rollouts s_v. These statistics provide the empirical semantic coverage used by the rollout-allocation procedure in Appendix B.2.

Online maintenance. The semantic tree is updated whenever a group of new rollouts are completed. We first segment and encode the new response using the same procedure as during initialization. At the complete-response level, the rollout is assigned to the nearest existing node if its distance is below the clustering threshold; otherwise, a new reasoning-mode node is created. Its prefixes are then inserted recursively within the selected parent nodes using the same rule.

After insertion, the corresponding node statistics are updated along the resulting path. Unlike initialization, online insertion is governed only by the semantic-distance threshold: the initialization cluster limits do not prevent a newly observed semantic mode from creating a new node during training. This allows $\mathcal{T}_x$ to expand as the policy discovers previously unseen reasoning patterns.

B.2 Coverage-Aware Rollout Allocation

Given the semantic tree $\mathcal{T}_x$, SCARS allocates prefix-conditioned rollouts hierarchically. We first select a successful high-level reasoning mode based on its empirical coverage, and then select an intermediate prefix within the corresponding subtree. Both selections favor semantic regions that are less covered relative to other candidates at the same level.

B.2.1 High-level reasoning-mode selection.

Let $\mathcal{A}_x^{\text{full}}$ denote the set of complete-response nodes that contain at least one successful response and have at least one valid descendant prefix for continuation. For each $v \in \mathcal{A}_x^{\text{full}}$, we estimate its successful semantic coverage by

$$\hat{\nu}_v^+ = \frac{s_v}{\sum_{u \in \mathcal{A}_x^{\text{full}}} s_u}.$$

We then sample a reasoning mode according to

$$q^{\text{full}}(v \mid T_x) = \frac{\exp(-\log \hat{\nu}_v^+ / \kappa_{\text{full}})}{\sum_{u \in \mathcal{A}_x^{\text{full}}} \exp\big(-\log \hat{\nu}_u^+ / \kappa_{\text{full}}\big)}.$$

Therefore, among reasoning modes that have already produced successful responses, modes with lower empirical coverage receive larger sampling probability. The temperature κ_{full} controls the strength of this coverage-aware reallocation.

B.2.2 Conditional Prefix Selection

As described in Section 4.2, after selecting a high-level reasoning mode $v^{(1)}$, SCARS further selects an intermediate prefix from its subtree. Here we specify the implementation details of this conditional prefix-selection stage.

Depth selection. In our experiments, the semantic tree contains two prefix levels, d_2 and d_3, corresponding to the long- and short-prefix states defined in Appendix B.1. We sample the target depth uniformly:

$$q_{\text{depth}}(d \mid v^{(1)}) = \frac{1}{2}, \qquad d \in \{2, 3\}.$$

Prefix-node selection is then performed only among the valid descendants of $v^{(1)}$ at the sampled depth, i.e., $\mathcal{A}_x(v^{(1)}, d)$.

Nodes without successful samples. For a node $v \in \mathcal{A}_x(v^{(1)}, d)$ with $s_v > 0$, we compute its relative successful coverage within the current candidate set as

$$\hat{\nu}_v^+ = \frac{s_v}{\sum_{u \in \mathcal{A}_x(v^{(1)}, d)} s_u},$$

and apply the same coverage-aware sampling rule as in the main text. A node with $s_v = 0$ may still contain a valid prefix and therefore remains explorable. Since $-\log \hat{\nu}_v^+$ is undefined for such a node, we assign it the fixed exploration score $\mathcal{N}_0$, so that

$$q_{\mathrm{pre}}(v \mid v^{(1)}, d) \propto \begin{cases} \exp(-\log \hat{\nu}_v^+ / \kappa_{\mathrm{pre}}), & s_v > 0, \\ \exp(\mathcal{N}_0 / \kappa_{\mathrm{pre}}), & s_v = 0. \end{cases}$$

We use $\kappa_{\mathrm{pre}} = 0.5$ and $\mathcal{N}_0 = 0$. If none of the candidate nodes at the selected depth has produced a successful response, all candidates receive the same default score and are sampled uniformly.

Valid-prefix filtering and retrieval. A stored prefix is considered valid for continuation only if it does not already contain a completed final answer, contains at most 768 tokens, and leaves sufficient sequence budget for generating a new continuation. Prefixes that already include the final $\texttt{\textbackslash boxed}\{\cdot\}$ answer are excluded because they no longer represent intermediate reasoning states. If no valid prefix remains at the sampled depth, we try the other prefix depth; if neither depth contains a valid prefix, the corresponding rollout slot falls back to a standard root rollout.

After a prefix node is selected, we preferentially sample prefixes extracted from previously verified successful responses. If no such prefix is available, all valid prefixes stored in the node remain eligible. Among the eligible prefixes, we favor more recently observed records using

$$\Pr(p_j \mid v) \propto \exp\left(-\frac{\upsilon_{\max} - \upsilon_j}{\tau_r}\right), \tag{147}$$

where υ_j is the policy version at which prefix p_j was observed, $\upsilon_{\max}$ is the most recent version represented in the node, and we set $\tau_r = 50$.

If no valid prefix exists at the sampled depth, we attempt the other prefix depth. If no valid prefix is available in the selected reasoning mode, the corresponding slot falls back to a standard root rollout. Within the same group, different prefix nodes are used whenever possible.

B.3 Prefix-conditioned Generation and Segment-Wise Credit Assignment

We use the prefix-conditioned generation and segment-wise credit assignment defined in Section 4.3, and provide here the implementation details omitted from the main text.

Each training group contains three standard root rollouts and two prefix-conditioned rollouts. The five completed responses are verified together and used to compute the group-relative advantages.

For a retrieved prefix, directly using the current rollout policy $\pi_{\boldsymbol{\theta}_{\mathrm{old}}}$ to compute its importance ratio would be incorrect, since the prefix was generated by an earlier policy. This would treat historical tokens as if they were sampled from the current policy and introduce bias into the importance-weighted policy-gradient estimate. We therefore store, for each reusable prefix, the token-level log probabilities under the policy that originally generated it. When the prefix is retrieved, these stored log probabilities are used to compute the importance ratio for its tokens, while the newly generated continuation uses $\pi_{\boldsymbol{\theta}_{\mathrm{old}}}$ as in standard PPO.

For policy optimization, root and prefix-conditioned rollouts use the same PPO clipping range and response-level sequence normalization. We use a numerical stabilizer of 10^{-6} when normalizing rewards, the corresponding hyperparameters are summarized in Table 8.

C Additional Experimental Results

This section provides additional experimental results that complement the main evaluations in Section 5. We first examine the generality of SCARS on a different model backbone in Appendix C.1.

Table 2: Results on Qwen3-4B-Base. Mathematical benchmarks are evaluated with pass@32, and code-generation benchmarks with pass@1. Bold and underline indicate the best and second-best results, respectively.

Methods	Math						Code			
	AIME24	AIME25	AMC23	CMIMC25	HMMT25	Avg.	LCB	HumanEval+	MBPP+	Avg.
Qwen3-4B-Base	35.49	33.34	89.19	24.35	13.60	39.20	14.97	50.46	56.28	40.57
GRPO	<u>49.38</u>	**50.97**	94.69	<u>41.15</u>	32.02	<u>53.64</u>	14.42	51.37	56.42	40.74
SetPO	48.52	46.60	93.92	39.17	28.98	51.44	14.97	<u>52.74</u>	56.28	<u>41.33</u>
R1-Zero-Div	46.89	50.38	<u>95.22</u>	38.01	**33.46**	52.79	<u>15.11</u>	51.98	<u>56.48</u>	41.19
GRPO + SCARS	**51.29**	<u>50.89</u>	**95.54**	**41.19**	<u>33.42</u>	**54.47**	**15.38**	**53.05**	**57.28**	**41.90**

We then provide detailed semantic-diversity and ablation results, including the Math12K test-set analysis and benchmark-level evaluations corresponding to the figures in the main text in Appendix C.2 and C.3. Finally, we analyze additional training logs to further characterize the different optimization dynamics across methods, and provide a case study to complement the method illustrations in the main text in Appendix C.4 and C.6.

C.1 Results on Qwen3-4B-Base

We further evaluate SCARS with Qwen3-4B-Base to examine whether its benefits persist on a different model backbone. Table 2 reports the complete mathematical-reasoning and code-generation results. All methods are evaluated at their final checkpoints after three training epochs.

As shown in Table 2, SCARS consistently improves the GRPO baseline across the mathematical and code-generation benchmarks, suggesting that semantic coverage-aware rollout sampling has the potential to be applied across different model backbones.

C.2 Additional Semantic-Diversity Results

We report the detailed successful-response semantic entropy on the Math12K test set and four held-out mathematical reasoning benchmarks in Table 3. For AIME24, AIME25, HMMT25, and AMC23, we directly reuse the 128 responses generated for benchmark evaluation.

We first detail the whole semantic diversity evaluation procedure. Taking the Math12K test set as an example, for each prompt we collect multiple responses generated by the evaluated policy. We encode the reasoning trace of each response using Qwen3-Embedding-8B and cluster the responses within each prompt according to their semantic similarity. Each resulting cluster represents a distinct reasoning mode. We then use verifier labels to examine how the successful responses are distributed across the resulting clusters, providing an empirical estimate of successful-response semantic diversity. If the successful responses of a prompt are partitioned into semantic clusters with empirical proportions $\hat{\boldsymbol{\nu}}^{+}(x)$, we compute its successful semantic entropy as

$$\hat{H}^{+}(x) = -\sum_{j} \hat{\nu}_j^{+}(x) \log \hat{\nu}_j^{+}(x).$$

The reported benchmark-level semantic entropy is then obtained by averaging $\hat{H}^{+}(x)$ over all prompts in the corresponding evaluation set. Prompts for which no successful response is sampled are assigned zero entropy.

Table 3 reports the exact successful semantic entropy across the Math12K test set and four held-out mathematical reasoning benchmarks. SCARS achieves the highest semantic entropy on four of the five evaluation sets, including Math12K test set, AIME24, AIME25, and AMC23, and ranks second on HMMT25. The improvement is particularly clear on Math12K test set and AIME24, where SCARS substantially increases semantic diversity over standard GRPO. On AIME25, SCARS and GRPO exhibit similar entropy, while on HMMT25 R1-Zero-Div achieves the highest value. Overall, these results show that the higher semantic diversity maintained by SCARS is not limited to a single evaluation set, but is consistently reflected across different reasoning benchmarks. Together with the performance results in the main text, this suggests that the semantic diversity preserved during

Table 3: Successful semantic entropy $\hat{H}^+$ across different evaluation sets. Bold and underline indicate the highest and second-highest values among post-trained methods, respectively.

Method	Math12K	AIME24	AIME25	HMMT25	AMC23
GRPO	0.359	0.176	<u>0.178</u>	0.234	0.250
R1-Zero-Div	<u>0.424</u>	<u>0.263</u>	0.141	**0.306**	<u>0.311</u>
SetPO	0.401	0.239	0.129	0.171	0.261
GRPO + SCARS	**0.630**	**0.290**	**0.179**	<u>0.293</u>	**0.316**

Table 4: Successful semantic entropy $\hat{H}^+$ of SCARS and its component ablations across different evaluation sets. Bold and underline indicate the highest and second-highest values, respectively.

Variant	Math12K	AIME24	AIME25	HMMT25	AMC23
w/o Prefix Exploration	0.563	<u>0.197</u>	0.156	0.163	**0.367**
w/o Node Selection	0.413	0.140	**0.201**	<u>0.281</u>	0.237
w/o Credit Reassignment	0.227	0.121	0.150	0.224	0.193
SCARS	**0.630**	**0.290**	<u>0.179</u>	**0.293**	<u>0.316</u>

training can be carried to more challenging evaluation problems, where the model retains a broader range of reasoning paths.

C.3 Full Ablation Results

We provide the detailed configurations and complete results for the three component ablations in Figure 4. All variants keep the training setting and rollout budget unchanged and modify only the specified component.

- **w/o Prefix Exploration** retains the coverage-aware node selection procedure, but replaces prefix-conditioned generation with replay of the complete historical response associated with the selected prefix.
- **w/o Node Selection** replaces coverage-aware allocation with uniform sampling over viable candidate nodes, while retaining prefix-conditioned generation and segment-wise credit assignment.
- **w/o Credit Reassignment** retains coverage-aware selection and prefix-conditioned generation, but assigns retrieved prefix tokens the group-relative advantage of the current complete response instead of the positive credit associated with their verified successful history.

The complete results show that removing any of the three components changes the diversity/performance behavior across benchmarks, consistent with the trends summarized in Figure 4.

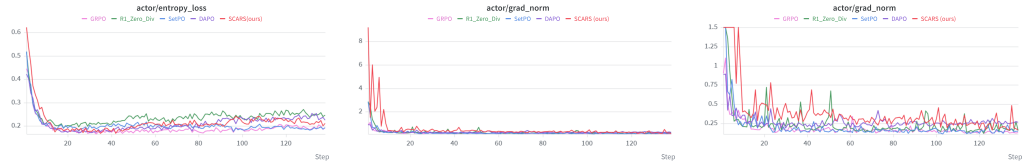
C.4 Optimization-Signal Dynamics

We complement the output-level diversity analysis with two signals recorded during policy optimization: the actor entropy statistic and the actor gradient norm. These quantities characterize different aspects of training. The entropy statistic measures token-level uncertainty under the current policy, whereas the gradient norm, $\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{actor}}\|_2$, measures the magnitude of the parameter-update signal produced by the current rollout batch. Neither quantity directly measures successful semantic coverage; we report them to examine how the different sampling strategies affect the optimization dynamics that accompany the semantic-diversity results.

Figure 5(a) shows that SCARS does not obtain its semantic-diversity improvement by simply maintaining the largest token-level entropy. In particular, R1-Zero-Div exhibits a higher entropy statistic during much of training, while SCARS achieves substantially stronger successful semantic diversity in Table 3. This distinction is consistent with our motivation that token-level uncertainty and coverage of semantically distinct successful reasoning modes capture different properties of the policy.

Table 5: Benchmark-level results for the component and threshold ablations. Mathematical benchmarks are evaluated with pass@32, and code benchmarks with pass@1.

Methods	Math						Code			
	AIME24	AIME25	AMC23	CMIMC25	HMMT25	Avg.	LiveCodeBench	HumanEval+	MBPP+	Avg.
w/o Prefix Exploration	55.53	39.14	93.62	25.69	17.24	46.24	11.40	47.56	51.52	36.83
w/o Coverage-Aware Selection	57.15	40.90	91.97	27.95	18.27	47.25	12.09	47.71	50.60	36.80
w/o Prefix Credit Reassignment	57.94	37.06	90.76	29.50	19.73	47.00	9.48	47.87	50.13	35.83
Threshold = 0.045	56.11	32.95	94.08	29.85	17.30	46.06	10.85	48.17	50.86	36.63
SCARS (threshold = 0.070)	56.53	38.68	94.05	31.81	20.63	48.34	10.71	48.02	50.86	36.53
Threshold = 0.100	55.78	39.73	91.67	31.96	14.65	46.76	10.03	46.95	51.32	36.10



Entropy loss Full range Magnified view

(a) Actor entropy **(b)** Actor gradient norm

Figure 5: Optimization-signal dynamics across training methods. **(a)** The actor entropy loss characterizes the evolution of token-level policy uncertainty. **(b)** The actor gradient norm is shown over the full numerical range (left) and under a magnified scale (right). The magnified view makes the differences among methods after the initial large-gradient phase easier to distinguish. SCARS generally retains a stronger gradient signal during the middle and later stages of training, while not maintaining the largest entropy signal.

Figures 5(b) report the actor gradient norm over the full training trajectory and under a magnified late-stage scale, respectively. After the initial transient phase, the gradient norms of GRPO and SetPO remain concentrated near the lower end of the observed range. SCARS generally retains a stronger update signal during the middle and later stages of training, this pattern is consistent with the interpretation that re-entering under-covered reasoning regions continues to expose the policy to non-redundant training signals after ordinary on-policy rollouts have become more concentrated.

C.5 Additional t-SNE Case Study

We provide additional t-SNE visualizations to complement the representative case in Figure 2. For each prompt in Math12K test set, we collect 32 responses generated by the base model and checkpoints at different training stages training with standard GRPO or SCARS, respectively. Then we encode their complete reasoning traces using the same semantic embedding model as in our diversity evaluation, and project the resulting embeddings into two dimensions using t-SNE. Colors distinguish training stages: blue, orange, and red denote the base model, Epoch 1, and Epoch 2, respectively. Marker shapes indicate verifier correctness, with stars denoting correct responses and circles denoting incorrect responses.

Across these additional cases, GRPO progressively concentrates correct responses into fewer semantic regions, whereas SCARS maintains broader semantic coverage throughout training, providing further qualitative evidence consistent with the dataset-level semantic-entropy results in Section 5.

C.6 A Semantic-Tree Case Study

We name a node as `d`ℓ`_n`j, where ℓ is its tree level and j is a local node identifier. Thus, `d1_n0` denotes a complete-response cluster, `d2_n0` a long-prefix cluster beneath it, and `d3_n0` a shorter-prefix cluster. Although the tree is constructed from complete responses toward shorter prefixes, a generated response is read in the opposite direction: a short state at d_3 is extended through d_2 into a complete response at d_1.

Figure 7 examines the tree initialized for one geometry problem from 32 base-policy rollouts. At the complete-response level, the largest split recovers the two modes illustrated in Figure 1. The

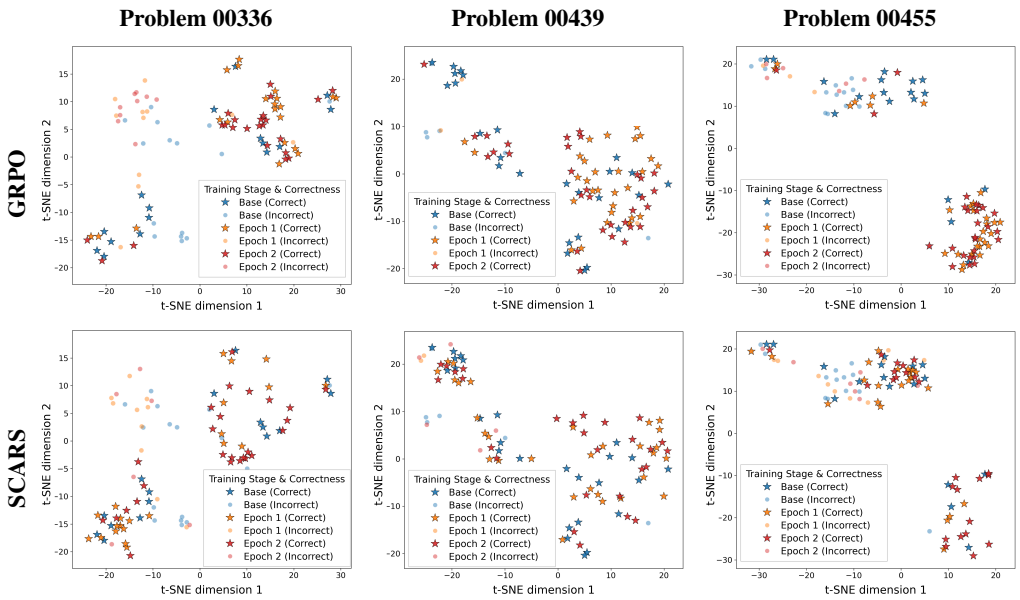


Figure 6: Additional prompt-level t-SNE visualizations of response semantics during RLVR training. Each column corresponds to one problem, with GRPO shown in the first row and SCARS in the second row. Blue, orange, and red denote the base model, Epoch 1, and Epoch 2, respectively; stars denote correct responses and circles denote incorrect responses.

direct-derivation branch works out the fourth intersection and focal distances in the response text, sometimes followed by a code check. The plan–code–verify branch instead states a solution plan and then uses Python/SymPy to execute it and report the resulting output. At the short-prefix level, the former branch further distinguishes whether the response begins with the fourth-intersection derivation, with the focus or focal-distance setup, or only with the given curves and points. The latter branch contains a step-by-step plan before code and one rare prefix that only restates the two required outputs. These descriptions are obtained from the records assigned to each node rather than inferred from the tree topology alone.

The node counts reinforce this interpretation: the direct-derivation branch contains 23 rollouts with 16 successes, whereas the plan–code–verify branch contains 9 rollouts with 3 successes; the short-prefix level retains both common and rare states. Because this snapshot records initialization only, it supports the interpretability of the tree rather than claims about later rollout allocation.

Tracing one complete response. Table 6 enumerates every node in the tree and gives a representative record for each non-root state. Figure 8 instead follows one successful response along the emphasized path in Figure 7. The three shaded regions are consecutive rather than alternative excerpts: the first region is the short prefix assigned to `d3_n0`; appending the second produces the long prefix assigned to `d2_n0`; appending the third produces the complete response assigned to `d1_n0`. This cumulative view makes explicit what information becomes available at each semantic state.

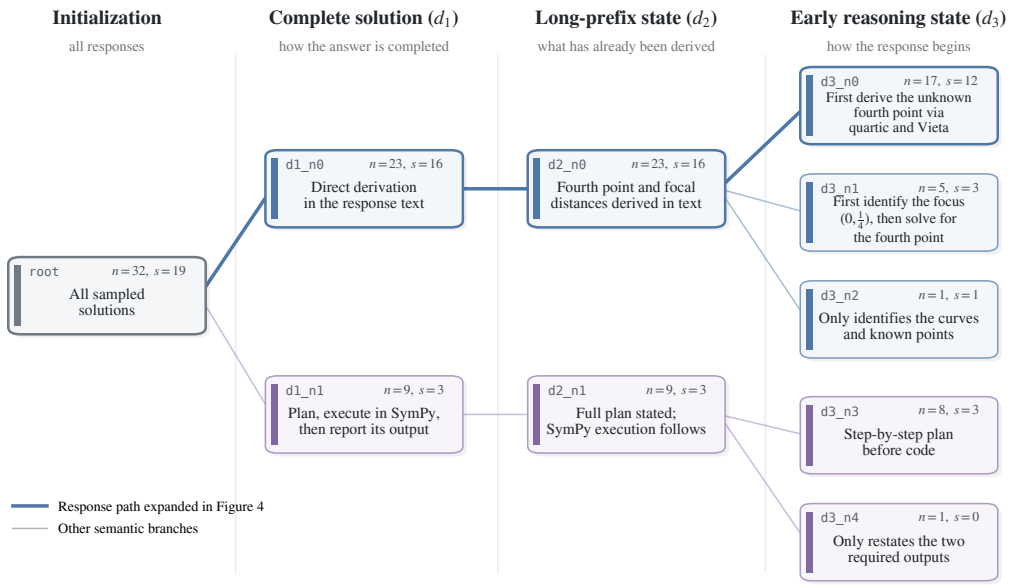


Figure 7: A semantic tree initialized from 32 rollouts for prompt 4995. The tree separates a direct-derivation mode from a plan–code–verify mode and then resolves their associated long-prefix and early reasoning states. Each node states what its assigned records express at that stage and reports the number of assigned and successful rollouts, n and s. Blue and purple fills distinguish the two principal solution branches; the emphasized blue path is expanded into a complete response in Figure 8.

Table 6: All nodes in the semantic tree of Figure 7. The semantic-state descriptions summarize the records assigned to each node; representative excerpts are shortened only for display.

Node	Level	n	s	Interpreted semantic state	Representative response or prefix excerpt
`root`	Initialization	32	19	All initialization rollouts.	No single representative prefix: this node contains all 32 base-policy responses before semantic partitioning.
`d1_n0`	Complete response	23	16	Directly derive the fourth intersection and focal distances in text, with an optional code check afterward.	"The sum of the roots is zero because there is no x^3 term. Therefore, $-28 - 2 + 13 + a = 0$, so $a = 17$."
`d1_n1`	Complete response	9	3	State a plan, execute the solution in Python/SymPy, and report the program output.	"We will use Python and SymPy to help us with the calculations. . . . Let's implement this in Python."
`d2_n0`	Long prefix	23	16	The response has carried out the direct derivation of the fourth point and focal distances in text.	"Vieta's formulas give $a = 17$. . . . The distance from $(0, \frac{1}{4})$ to (x, x^2) simplifies to $x^2 + \frac{1}{4}$."
`d2_n1`	Long prefix	9	3	A step-by-step plan has specified the circle fit, intersection solve, and distance calculation before code execution.	"Find the equation of the circle; find the fourth intersection point; calculate the distances; sum these distances."
`d3_n0`	Short prefix	17	12	Begin with the fourth-intersection derivation, typically by introducing (a, a^2) and the induced quartic.	"The fourth intersection point will be (a, a^2). . . . Substituting $y = x^2$ gives a quartic equation in x."
`d3_n1`	Short prefix	5	3	Begin with the focus or focal-distance setup before completing the intersection derivation.	"For $y = x^2$, the focus is at $(0, \frac{1}{4})$."
`d3_n2`	Short prefix	1	1	Identify only the two curves and the three known intersections.	"The parabola is given by $y = x^2$. The circle intersects it at four points, and we know three of them."
`d3_n3`	Short prefix	8	3	Give a step-by-step solution plan before the later Python/SymPy execution.	"We will use Python and SymPy to help us with the calculations. . . . Find the fourth intersection point; calculate the distances."
`d3_n4`	Short prefix	1	0	Only restate the two required outputs, without yet giving a plan or code setup.	"Find the fourth intersection point . . . Then, calculate the sum of the distances from the focus to all four intersection points."

To solve this problem, we need to find the fourth intersection point of the parabola $y = x^2$ and the circle $\mathcal{C}$. Three intersection points are $(-28, 784)$, $(-2, 4)$, and $(13, 169)$; denote the fourth by (a, a^2). The general equation of a circle is

$$(x - h)^2 + (y - k)^2 = r^2,$$

where (h, k) is its center and r its radius.

`d3_n0`: introduce the unknown fourth point and the circle model

Substituting $y = x^2$ into the circle equation gives

$$(x - h)^2 + (x^2 - k)^2 = r^2,$$

or, after expansion,

$$x^4 + (1 - 2k)x^2 - 2hx + h^2 + k^2 - r^2 = 0.$$

Its roots are the x-coordinates of the four intersection points. Because there is no cubic term, Vieta's formulas give

$$-28 - 2 + 13 + a = 0, \qquad \text{so} \qquad a = 17.$$

The fourth point is therefore $(17, 289)$. The focus of $y = x^2$ is $(0, \frac{1}{4})$.

`d3_n0` plus this region forms `d2_n0`: derive the quartic and recover the fourth root

For any point (x, x^2) on the parabola, its distance to the focus is

$$d = \sqrt{x^2 + \left(x^2 - \tfrac{1}{4}\right)^2} = \sqrt{x^4 + \tfrac{1}{2}x^2 + \tfrac{1}{16}}$$
$$= \sqrt{\left(x^2 + \tfrac{1}{4}\right)^2} = x^2 + \tfrac{1}{4}.$$

Hence the four distances are

$$784 + \tfrac{1}{4}, \qquad 4 + \tfrac{1}{4}, \qquad 169 + \tfrac{1}{4}, \qquad 289 + \tfrac{1}{4}.$$

Their sum is

$$784 + 4 + 169 + 289 + 1 = \boxed{1247}.$$

all three regions form `d1_n0`: complete the focus-distance calculation

Figure 8: One complete successful response (rollout 16 in the supplied case data) from the emphasized tree path. The response is typeset compactly, with its mathematical content and ordering preserved. Progressively lighter backgrounds mark the additional text introduced when moving from the short prefix `d3_n0`, through the long prefix `d2_n0`, to the complete response `d1_n0`. The braces state the semantic information contributed at each stage.

Table 7: Shared training hyperparameters for the Qwen2.5-Math-7B experiments.

Setting	Value	Setting	Value
Training epochs / steps	3 / 138	Learning rate	10^{-6}
Prompt rollout batch	256	Actor batch	64 prompts / 320 responses
Group size	5	PPO epochs	1
Prompt / response length	2,560 / 1,536	PPO clip range	$[0.8, 1.2]$
KL coefficient	0.02	Gradient clipping	1.0
Loss aggregation	response sequence mean	Rollout temperature	1.0
Top-p / top-k	1.0 / disabled	LR schedule / warmup	constant / none
AdamW (β_1, β_2)	$(0.9, 0.999)$	Weight decay	0.01
Fine-tuning / precision	full / BF16	Random seed	1

Table 8: SCARS-specific tree construction, sampling, and credit settings.

Setting	Value
Initialization responses per prompt	32
Tree levels	complete, $3/4$-prefix, $1/4$-prefix
Embedding model / maximum length	Qwen3-Embedding-8B / 3,072
Distance / linkage	cosine / average
Distance threshold	0.07 at all levels
Maximum initialization clusters	4 at d_1; 3 children at d_2, d_3
Root / prefix-conditioned slots	3 / 2
κ_{full} / κ_{pre}	0.5 / 0.5
Long / short prefix probability	0.5 / 0.5
Maximum prefix length	768 tokens
Prefix recency scale τ_r	50 policy versions
Default local novelty $\mathcal{N}_0$	0
Prefix credit coefficient	1.0

D Experimental Details

D.1 Models, Data, and Baselines

The primary experiments start from Qwen2.5-Math-7B and train on Math12K; Appendix C.1 repeats the comparison with Qwen3-4B-Base. Standard GRPO is the controlled baseline. SetPO and R1-Zero-Div provide exploration-oriented comparisons, while DAPO tests whether SCARS can be combined with another policy-optimization method. Unless a method definition requires otherwise, reimplemented methods use the same training prompts, binary verifier, rollout budget, maximum lengths, and final-checkpoint evaluation. This keeps differences in sampling or optimization from being confounded with additional data or evaluation-time sampling.

D.2 Training Hyperparameters

Tables 7 and 8 report the settings that determine the optimization and the proposed sampling procedure.

D.3 Benchmark Evaluation Protocol

For the five mathematical-reasoning benchmarks in Table 1, we generate $n = 128$ responses for each problem and report pass@32 using the unbiased estimator

$$\widehat{\text{pass@}}\, k = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}, \qquad n = 128, \quad k = 32, \tag{148}$$

where c is the number of correct responses among the n samples. The three code-generation benchmarks use execution-based correctness and report pass@1. Standard benchmark evaluation uses the `qwen-boxed` template, temperature $T = 0.1$, top-$p = 0.95$, tensor parallelism 2, a maximum gen-

eration length of 16,384 and with different seeds. Diversity evaluation follows the separate protocol in Appendix D.4.

D.4 Semantic-Diversity Evaluation Protocol

We evaluate a set $\mathcal{X}_{\text{eval}}$ of several test prompts, here we use the math12K test set as $\mathcal{X}_{\text{eval}}$ exactly. For each model checkpoint k and prompt x, we independently sample $N = 32$ responses with $T = 0.7$ and top-$p = 0.95$, retain responses accepted by the verifier, encode their reasoning traces with Qwen3-Embedding-8B, and cluster them within the prompt. Let $n_{k,j}(x)$ denote the number of successful responses assigned to j-th semantic cluster. Then, define the relative successful rate of j-th cluster as:

$$\widehat{\nu}^{+}_{k,j}(x) = \frac{n_{k,j}(x)}{\sum_i n_{k,i}(x)}. \tag{149}$$

We assign zero entropy to prompts with no successful response and otherwise compute the within-prompt successful semantic entropy:

$$\widehat{H}^{+}_{k}(x) = \begin{cases} -\sum_{c} \widehat{\nu}^{+}_{k,j}(x) \log \widehat{\nu}^{+}_{k,j}(x), & \sum_i n_{k,i}(x) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{150}$$

The dataset-level quantity reported in the main text is the mean across all evaluation prompts,

$$\widehat{H}^{+}_{k} = \frac{1}{|\mathcal{X}_{\text{eval}}|} \sum_{x \in \mathcal{X}_{\text{eval}}} \widehat{H}^{+}_{k}(x). \tag{151}$$

Consistent with 2, all entropy values use the natural logarithm.

What's more, we evaluate the semantic diversity on different reasoning benchmarks. For the benchmark-level semantic-diversity evaluation, we consider AIME24, AIME25, AMC23, and HMMT25. We exclude CMIMC25 because we observed several verifier-accepted responses that contain long stretches of irrelevant text but nevertheless end with the correct answer in the final `\boxed{·}` expression. For our diversity metric clusters the complete reasoning traces, such abnormal responses can be identified as distinct semantic modes and substantially distort the estimated diversity. What's more, CMIMC25 is substantially more challenging and successful responses are sparse on many of the remaining problems, these few affected problems disproportionately dominate the benchmark-level semantic-diversity estimate. This phenomenon is particularly pronounced for the entropy-based method, making the benchmark-level diversity comparison across methods unreliable. We therefore omit CMIMC25 from the diversity figures to ensure a fair comparison.

D.5 Compute and Reproducibility Details

Training uses four NVIDIA A100-80GB GPUs with FSDP and vLLM through verl. We use full-parameter BF16 fine-tuning and seed 1. Tree states are checkpointed separately from model parameters and restored from the snapshot associated with the same policy step; using a mismatched tree would change both node counts and subsequent rollout allocation. The software environment uses Python 3.10.20, CUDA 12.6, PyTorch 2.7.1, verl 0.3.3.dev0, vLLM 0.10.0, and Transformers 4.57.6.