CURE: Critical-Token-Guided Re-Concatenation for Entropy-Collapse Prevention

Anonymous ACL submission

Abstract

Recent advances in Reinforcement Learning with Verified Reward (RLVR) have significantly bolstered the reasoning capabilities of Large Language Models (LLMs). However, conventional RLVR pipelines often rely on static initial-state sampling, leading to overly deterministic behavior, rapid entropy collapse, and plateaued performance during extended training. To mitigate this, we propose CURE (Critical-token-gUided Re-concatenation for Entropy-collapse prevention), a two-stage framework balancing exploration and exploitation. In Stage 1, CURE encourages exploration by re-generating branched trajectories at high-entropy critical tokens, jointly optimizing them with original paths to maintain diversity. Compared to vanilla DAPO, this stage yields superior reasoning performance while preserving high entropy. In Stage 2, we transition to static sampling using DAPO, placing the model in familiar states to consolidate exploitation. Extensive experiments on Qwen-2.5-Math-7B demonstrate that CURE outperforms existing RLVR methods by 5% across six math benchmarks, achieving state-of-the-art results in both reasoning accuracy and entropy maintenance.

1 Introduction

Recent advancements in Reinforcement Learning with Verification (RLVR) (Jaech et al., 2024; Zeng et al., 2025; Hu et al., 2025; Liu et al., 2025b) have driven significant progress in unlocking the reasoning capabilities of large language models. By replacing opaque reward surrogates with automatic verifiers that emit precise binary signals (DeepSeek-AI et al., 2025), RLVR enables scalable, self-improving training loops and has delivered strong gains on challenging reasoning benchmarks, from mathematical problem solving (Hendrycks et al., 2021; He et al., 2024) to scientific QA (Rein et al., 2024).

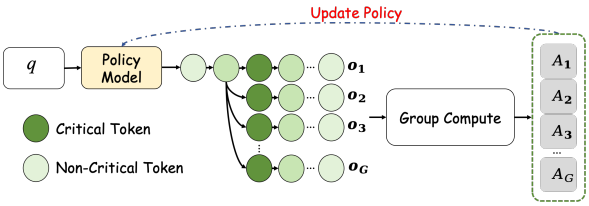

Figure 1: Overview of the CURE pipeline. In Stage 1, given an input query q, the policy model produces a pool of candidate responses. We compute token-level entropy to identify critical tokens (high entropy), extract the clauses immediately preceding those tokens, append them to q to form refined prompts, and query the model again. The newly generated responses are aggregated with the original ones and jointly optimized within a single group. In Stage 2, we continue training to translate the exploration bonus into realized performance.

Despite the impressive gains of RLVR, the community (Cui et al., 2025b; Tarvainen and Valpola, 2017) now recognizes policy entropy collapse as the key bottleneck blocking further progress. Once entropy collapses, probability mass concentrates on a few low-diversity responses, and performance plateaus early. Early—often ad-hoc—fixes include simply raising the sampling temperature (Zhang et al., 2025) and adding a small KL term (DeepSeek-AI et al., 2025; Zhou et al., 2025) to slow the drop. More principled attempts add entropy regularizers, redesign the loss, or reward shaping (Liu et al., 2025a; Cheng et al., 2025). Furthermore, ProRL (Liu et al., 2025a) applies reference-policy resets on a 1.5B-parameter model by periodically hard-resetting the reference policy to a recent snapshot of the online policy to curb entropy decay. However, it demands frequent reference-model updates and optimizer resets. Recently, Clip-Cov (Cui et al., 2025b) shows that clipping high-covariance tokens can also help prevent entropy collapse and prolong stable training, though it still requires fine-grained, task-specific

hyperparameter tuning, yields only limited performance improvements. Ultimately, the need for heavy hyperparameter tuning and manual model updates arises because prior work keeps adjusting the update rule while still relying on a fixed set of prompts and verifier signals. A static training-state distribution quickly narrows the exploration space, so entropy collapse is almost guaranteed.

We argue that an effective and conceptually simple strategy is to continually exploit uncertainty during response generation, thereby preventing entropy collapse (Ladosz et al., 2022; Ecoffet et al., 2019; Burda et al., 2018). We dynamically enrich the training signal by leveraging the model's own internal uncertainty—specifically, token-level policy entropy computed during autoregressive decoding. High-entropy tokens expose moments of genuine indecision. A naive tactic is to append a randomly sampled continuation to the original query so that subsequent rollouts start from a richer prompt. In practice, however, this inflates the context with irrelevant detail and still fails to probe the policy's true blind spots. Our remedy is to intervene exactly at the decision point where the model is most uncertain. For each generated answer, we locate the token with the highest policy entropy and truncate the sequence immediately before it. The retained prefix marks a high-stakes fork in the reasoning process. From this fork, the current policy produces multiple alternative continuations. Because these updates act precisely where the model was undecided, they simultaneously broaden the distribution of plausible next moves—maintaining healthy entropy and raise expected accuracy in future rollouts.

Building on these insights, we translate uncertainty-aware exploration into a concrete training routine that broadens the state distribution at critical decision points before consolidating gains. In this work, we propose **CURE** (**C**ritical-token-g**U**ided **R**e-concatenation for **E**ntropy-collapse prevention), a two-stage RLVR framework designed to balance exploration and exploitation by dynamically expanding the state distribution during training. Specifically, in the first stage, we first identify critical tokens in each response using a token-level entropy criterion; next, we re-concatenate only the clauses that precede these critical tokens with the original query to create a refined prompt; finally, we feed prompt back into the model, thereby encouraging exploration along trajectories that are conditioned on previously unseen yet semantically salient context. To improve training efficiency, we follow DAPO (Yu et al., 2025) by retaining only those prompts whose candidate responses include both verifier-accepted (correct) and verifier-rejected (incorrect) answers. In the second stage, CURE transitions from high-entropy exploration to low-entropy exploitation through a continue training process. Specifically, we revert to static initial-state sampling using the DAPO (Yu et al., 2025) strategy, deliberately placing the model back into familiar prompt contexts. This shift encourages the policy to consolidate knowledge gained during the exploratory phase and reinforce accurate behaviors in a more deterministic regime. Crucially, while exploration-focused re-concatenation is disabled in this stage, the model still benefits from the enriched policy distribution developed earlier. As training proceeds under a static query distribution, the reward naturally increases, which helps sharpen decision boundaries and stabilize convergence. This two-phase training regime ensures that the policy first explores broadly and then exploits precisely, maintaining strong generalization while achieving good performance on verifier-based reasoning tasks.

Our contributions can be summarized as follows:

- We present the first analysis of policy entropy collapse from the perspective of state distribution, showing that training on a fixed dataset depresses entropy; we further provide a principled remedy based on critical-token-guided re-concatenation.
- We propose CURE, a lightweight two-stage framework that balances exploration and exploitation by shaping the training-state distribution.
- We perform extensive experiments across several challenging math-reasoning benchmarks. CURE not only achieves state-of-the-art accuracy but also retains the highest entropy among similarly performing models, underscoring its capacity for continued improvements.

2 Related Work

Reinforcement Learning for LLM Reasoning. Early approaches to reinforcement learning for large language models leveraged reward-model-based fine-tuning and direct preference optimization (Rafailov et al., 2023; Christiano et al., 2017;

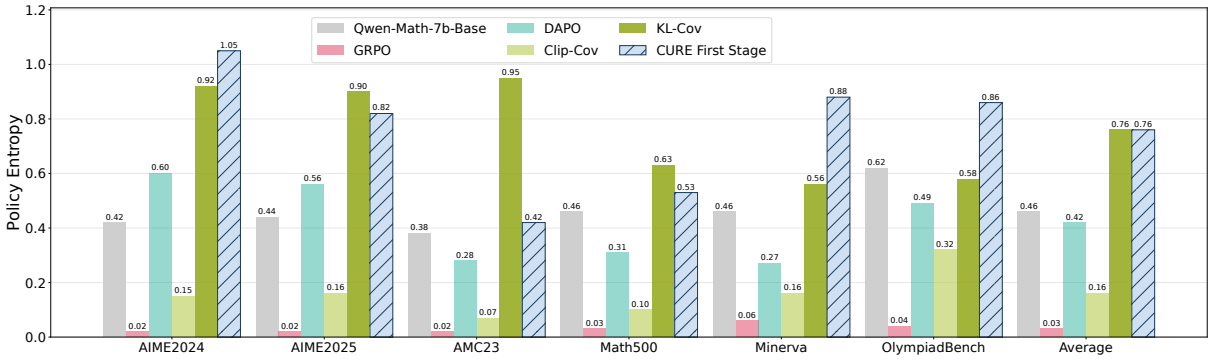

Figure 2: Entropy comparison of CURE first stage and other methods at temperature 1.0

Ouyang et al., 2022; Cui et al., 2024), simulating human feedback to guide model behavior. However, these methods depended heavily on manually collected human preferences or synthetic preference models (Yuan et al., 2024; Wei et al., 2024; Liu et al., 2024). More recently, the community has shifted toward pure RL at scale for reasoning. Recently, the research community has pivoted toward large-scale reinforcement learning (RL) for enhancing reasoning capabilities. Empirical evidence from several studies (DeepSeek-AI et al., 2025; Team et al., 2025; Cui et al., 2025a; Zeng et al., 2025; Hu et al., 2025; Liu et al., 2025b; Yan et al., 2025; Chen et al., 2025) suggests that a streamlined RL paradigm—relying solely on outcome-based signals without auxiliary preference models—scales effectively and yields significant performance gains.

Entropy Control. Policy entropy collapse constitutes a critical obstacle in reinforcement learning (RL) for large language models (LLMs), as diminishing exploration often leads to rapid performance stagnation. Regularization-based approaches (He et al., 2025; Liu et al., 2025a) add an entropy loss or KL loss whose weight often needs careful or adaptive tuning. Reward shaping methods (Cheng et al., 2025) similarly adds an entropy bonus to the reward or advantage to balance exploration and exploitation. Complementary interventions like loss reweighting (Wang et al., 2025; Cui et al., 2025b) and clip-higher (Yu et al., 2025) can also help to prevent entropy collapse.

In contrast to objective-shaping and static-sampling regularizers, CURE is a two-stage data-level framework: it leverages high-entropy critical tokens to drive exploration, then continues training with DAPO for exploitation, improving the exploration-exploitation trade-off.

3 Method

3.1 Background

3.1.1 MDP Formulation of Language Generation.

We formalize token-level language generation as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, R, d_0, \omega)$ with fully observable, deterministic dynamics. For a prompt $\mathbf{q} \sim d_0$ with $\mathbf{q} = (q_0, \ldots, q_m)$, we sample G rollouts from the behavior policy $\pi_{\theta_{\text{old}}}$. Rollout $i \in \{1, \ldots, G\}$ is the token sequence $\mathbf{o}_i = (o_{i,1}, \ldots, o_{i,T_i}) \in \mathcal{V}^{T_i}$ of possibly variable length T_i. At token position $t \in \{1, \ldots, T_i\}$, the state is $s_{i,t} = (\mathbf{q}, \mathbf{o}_{i,<t})$ and the action is $a_{i,t} = o_{i,t}$, where $\mathbf{o}_{i,<t} = (o_{i,1}, \ldots, o_{i,t-1})$, transitions satisfy $\mathbb{P}(s_{i,t+1} \mid s_{i,t}, a_{i,t}) = 1$ with $s_{i,t+1} = (\mathbf{q}, \mathbf{o}_{i,<t}, o_{i,t})$, the process initializes at $s_{i,1} = \mathbf{q}$, and generation terminates when the end-of-sequence symbol ω is emitted. Each action receives a scalar reward $R(s_{i,t}, a_{i,t})$ from an automatic verifier, a learned human-preference model, or task-specific rules. Under this formulation, learning amounts to optimizing a stochastic policy $\pi_\theta(a \mid s)$ to maximize the expected cumulative reward, and the deterministic, fully observable dynamics enable fine-grained analysis and explicit control of exploration metrics such as policy entropy.

3.1.2 GRPO.

GRPO replaces the PPO value function with an average of sampled rewards to calculate advantage.

Specifically, given a prompt $\mathbf{q} \sim P(Q)$, we sample G rollouts $\{\mathbf{o}_i\}_{i=1}^{G}$ from the current policy $\pi_{\theta_{\text{old}}}$. At each token position t in rollout i, the likelihood ratio is defined in Eq. 1.

$$r_{i,t}(\theta) = \frac{\pi_\theta\left(o_{i,t} \mid \mathbf{q}, \mathbf{o}_{i,<t}\right)}{\pi_{\theta_{\text{old}}}\left(o_{i,t} \mid \mathbf{q}, \mathbf{o}_{i,<t}\right)} \tag{1}$$

The group-relative advantage $\hat{A}_{i,t}$ is then obtained by standardizing each return R_i within the group, defined in Eq. 2.

$$\hat{A}_{i,t} = \frac{R_i \; - \; \mathrm{Mean}\big(\{R_j\}_{j=1}^{G}\big)}{\mathrm{Std}\big(\{R_j\}_{j=1}^{G}\big)}. \tag{2}$$

And then maximizes the clipped surrogate objective by

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{\mathbf{q},\{\mathbf{o}_i\}}\Big[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathbf{o}_i|}\sum_{t=1}^{|\mathbf{o}_i|}\Big(\min\big[r_{i,t}(\theta)\hat{A}_{i,t}, \; \mathrm{clip}(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\big] - \beta\, \mathrm{D}_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})\Big)\Big]. \tag{3}$$

Notably, $D_{\mathrm{KL}}[\pi_\theta \| \pi_{\mathrm{ref}}]$ acts as a regularization term that constrains the updated policy π_θ to remain close to the reference policy π_{ref}, representing one of the earliest efforts to prevent entropy collapse and some studies claimed to be the key parameter enabling GRPO to sustain prolonged training.

3.2 CURE

CURE employs a two-stage procedure. As shown in Fig. 1, in Stage 1, exploration is injected by dynamically reshaping the prompt distribution based on token-level uncertainty. In Stage 2, exploitation is applied by continuing training with DAPO under static initial-state sampling on a fixed corpus, consolidating Stage 1 gains into higher accuracy and overall performance. The corresponding pseudocode is presented in Appendix A.

3.2.1 CURE First Stage.

By sampling high-entropy ended prefixes and re-prompting, CURE's first stage explicitly injects novel yet coherent initial states, delaying premature entropy collapse and improving exploration efficiency.

1. **Initial Rollouts.** For each query $\mathbf{q}$ drawn directly from the dataset, we first sample N_1 trajectories from the old policy to estimate token-level uncertainty.

$$\mathcal{G}(\mathbf{q}, N_1) = \{\mathbf{o}_i\}_{i=1}^{N_1} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{q}). \tag{4}$$

2. **Token-Level Entropy.** We compute the policy entropy at every position in each trajectory to detect where the model is most uncertain, which will guide us to unexplored-but-coherent regions of the state space.

$$H_{i,t} = -\sum_{v \in \mathcal{V}} \pi_{\theta_{\text{old}}}\big(v \mid \mathbf{q}, \mathbf{o}_{i,<t}\big) \log \pi_{\theta_{\text{old}}}\big(v \mid \mathbf{q}, \mathbf{o}_{i,<t}\big), \quad t = 1, \ldots, T_i. \tag{5}$$

3. **Top-K Selection with Stochastic Choice.** We rank positions by entropy and uniformly select one index from the top-K most uncertain positions to avoid a deterministic bias toward the single highest-entropy token.

$$\mathcal{T}_K^{(i)} = \mathrm{TopK}_t\big(H_{i,t}, K\big), \qquad t_i^\star \sim \mathrm{Uniform}\big(\mathcal{T}_K^{(i)}\big). \tag{6}$$

4. **Frontier Prefix and Refined Prompt.** We take the prefix $\mathbf{p}_i$ up to (but not including) the sampled position $t_i^\star$ and prepend it to the original query, creating a refined prompt $\mathbf{q}_i'$ that stays semantically consistent yet was unseen during prior training.

$$\mathbf{p}_i = \mathbf{o}_{i,1:t_i^\star - 1}, \qquad \mathbf{q}_i' = \mathbf{q} \,\|\, \mathbf{p}_i. \tag{7}$$

5. **Re-Prompting Rollouts.** Each refined prompt q_i' is then fed back to the policy to produce N_2 additional trajectories, yielding a total of $N_1 * N_2$ re-prompted samples.

$$\mathcal{G}(\mathbf{q}_i', N_2) = \{\mathbf{o}_j\}_{j=1}^{N_2} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{q}_i'). \tag{8}$$

6. **Group Construction.** For each dataset-derived query $\mathbf{q}$, we merge the original trajectory with its re-prompted counterparts to form the group $\mathcal{G}(\mathbf{q})$. This formulation underpins the computation of our GRPO-like objective across all trajectories in $\mathcal{G}(\mathbf{q})$.

$$\mathcal{G}(\mathbf{q}) = \mathcal{G}(\mathbf{q}, N_1) \; \cup \; \Big(\bigcup_{i=1}^{N_1} \mathcal{G}(\mathbf{q}_i', N_2)\Big) \qquad \big|\mathcal{G}(\mathbf{q})\big| = N_1 + N_1 * N_2. \tag{9}$$

7. **Batch-Level Dynamic Sampling Construction.** To improve training efficiency, we follow DAPO: at each sampling round, we discard and resample any prompts whose group of G rollouts is entirely correct or entirely incorrect, as such groups provide minimal gradient information and accelerate premature determinization.

8. **Objective Function.** We jointly optimize all trajectories in $\mathcal{G}(\mathbf{q})$ by Eq. 10. Here **gt** is the ground truth. The importance weight $r_{i,t}(\theta)$ is computed by Eq. 1, and the group-relative

advantage $\hat{A}_{i,t}^{\text{grp}}$ is computed by Eq. 2.

$$\mathcal{J}_{\text{CURE}}(\theta) = \mathbb{E}_{\mathbf{q}\sim P(Q)}\Bigg[\frac{1}{\sum_{o_i\in\mathcal{G}(\mathbf{q})}|\mathbf{o_i}|}\sum_{\mathbf{o}_i\in\mathcal{G}(\mathbf{q})}\sum_{t=1}^{|\mathbf{o_i}|}$$
$$\Big(\min\big[r_{i,t}(\theta)\,\hat{A}_{i,t}^{\text{grp}},\ \text{clip}\big(r_{i,t}(\theta),\,1-\varepsilon,\,1+\varepsilon\big)\,\hat{A}_{i,t}^{\text{grp}}\big]\Big)\Bigg]$$
$$\text{s.t.}\quad 0 < \Big|\{\mathbf{o}\mid \texttt{is\_eq}(\mathbf{gt},\mathbf{o}),\mathbf{o}\in\mathcal{G}(\mathbf{q})\}\Big| < |\mathcal{G}(\mathbf{q})|. \tag{10}$$

Empirically, this objective sustains higher policy entropy and improves rewards, with the KL-divergence regularization term omitted.

3.2.2 CURE Second Stage.

In the second stage, we perform annealing utilizing the DAPO algorithm. Leveraging DAPO's inherently low-entropy training dynamics, we neither decay the learning rate nor introduce an explicit KL regularizer. Instead, we train directly on all dataset-derived queries in their original form, without any re-concatenation. In practice, we perform a 10-step warmup and then continue training to step 100 with a fixed learning rate of 1×10^{-6}. As shown in Sec. 4, unlike DAPO baseline—where entropy continuously decreases at all temperatures under extended training—when sampling at a temperature of 1.0 during training, our average policy entropy does not decrease. This demonstrates that our model can sustain diverse exploration at high temperatures even after continuing training. At the optimal evaluation temperature of 0.6, however, this procedure yields a 29.2% reduction in policy entropy and a 7.6% improvement in evaluation performance. Consequently, learning becomes biased toward the high-reward behaviors discovered in Stage 1, effectively converting exploration into stable accuracy gains.

3.2.3 Advantage.

Plug-and-Play with RLVR. CURE can be dropped into existing pipelines with minimal engineering: replace the sampler with our critical-token-guided re-concatenation routine and switch to a simple two-phase entropy schedule. Empirically, this decoupled explore-then-exploit design arrests entropy collapse, maintains policy diversity when it matters, and then selectively compresses it to harvest performance.

Sustained, High-Entropy Exploration. By dynamically re-prompting on high-uncertainty prefixes and normalizing advantages across groups, CURE maintains a persistently elevated policy entropy even in late training, thereby continuously steering the model into novel but coherent regions of the state space and avoiding premature convergence to deterministic behaviors.

Efficient Conversion of Exploration into Performance. The strong positive coupling we observe between entropy and reward demonstrates that CURE's exploration is not random noise, but directly fuels learning,yielding higher rewards per unit of entropy than DAPO, GRPO, KL-CoV, or Clip-CoV. Moreover, the two-phase entropy schedule ensures that the diversity injected in the first stage is rapidly consolidated into accuracy gains in the second stage.

Further evidence of CURE's effectiveness is provided in the Appendix B.

4 Experiments

We design our empirical study to answer the following research questions:

- **Q1: Overall Performance.** Does CURE improve mathematical reasoning compared to baseline methods?
- **Q2: Entropy Preservation.** Can CURE sustain exploration throughout training and benefit from preventing policy-entropy collapse?
- **Q3: Second-Stage Research.** Given that CURE generates first-stage models with significantly higher policy entropy, how can we leverage these high-entropy checkpoints to further improve reasoning performance?
- **Q4: Ablation on Critical-Token Strategy.** How do alternative ways of selecting or handling critical tokens affect performance and entropy?

4.1 Experiment Setups

Training. perform fine-tuning on Qwen2.5-Math-7B-Base (Yang et al., 2024) using the publicly available DAPO-Math-17K (Yu et al., 2025) dataset, which contains only math questions paired with integer ground-truth answers. Specifically, we omit both the KL-divergence and entropy loss terms. Rollouts are generated with a batch size of 512, using a temperature of 1.0 and 16 rollouts per prompt. During policy updates, we use an update batch size of 32. To ensure a fair comparison, we maintain

Model	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg.
Qwen-Math-7B-Base (Yang et al., 2024)	16.6	6.3	52.2	52.4	10.7	19.0	26.2
Qwen-Math-7B-Instruct (Yang et al., 2024)	13.3	10.0	57.1	84.2	41.5	44.4	41.8
Previous Classical RLVR methods							
Eurus-2-7B-PRIME-Zero (Cui et al., 2025a)	18.9	11.7	57.7	79.8	41.5	48.0	42.9
SimpleRL-Zero (Zeng et al., 2025)	26.7	9.3	60.8	77.4	32.0	41.5	41.3
OpenReasoner-Zero (Hu et al., 2025)	15.4	13.4	56.5	80.6	39.0	45.9	41.8
Oat-Zero (Liu et al., 2025b)	28.8	10.8	65.2	80.0	42.3	43.7	45.1
LUFFY (Yan et al., 2025)	25.8	22.3	71.7	87.0	44.9	55.9	51.3
NFT (Chen et al., 2025)	32.0	18.3	88.5	83.2	40.8	47.3	51.7
Previous Entropy Control methods							
KL-Cov (Cui et al., 2025b)	33.4	17.1	77.1	83.8	43.0	49.9	50.7
Clip-Cov (Cui et al., 2025b)	32.4	14.3	81.6	84.8	44.5	48.0	50.9
Our Methods							
CURE First Stage	33.4	15.3	82.7	82.4	48.2	50.5	52.1
CURE Second Stage	35.5	18.5	89.7	83.4	48.2	50.5	54.3

Table 1: Performance of CURE and prior RLVR methods: avg@32 on AIME24, AIME25, and AMC23; avg@1 on MATH500, Minerva, and Olympiad.

consistent hyperparameter settings across all baselines, aligning parameters such as the number of rollouts per prompt, temperature, and batch sizes wherever applicable. Finally, we adopt the same reward function as DAPO (Yu et al., 2025), without any additional formatting reward. Detailed descriptions of reproducibility-related parameters and special hyperparameter settings, as well as the experimental setup, comparison protocols are provided in Appendix D.

Evulation. For evaluation, we benchmark six widely used math-reasoning datasets: AIME24 (Li et al., 2024), AIME25 (Li et al., 2024), AMC23 (Li et al., 2024) , MATH500 (Hendrycks et al., 2021), OlympiadBench (He et al., 2024), and Minerva Math (Lewkowycz et al., 2022). All inference model setups and sampling parameters are drawn from their official reports. Our method employs a top-p of 0.7 and a temperature of 0.6. Following SimpleRL (Zeng et al., 2025) and NFT (Chen et al., 2025), we combine the Math-Verify (Kydlíček), MathRuler-verifier (hiyouga, 2025), and SimpleRL verifier (Zeng et al., 2025) for final evaluation. To ensure a fair and balanced evaluation across benchmarks of varying scales, we adopt different reporting metrics based on dataset size. More detailed results are provided in Appendix D.2.

Baselines. We compare our method against six RLVR baselines: Eurus-2-7B-PRIME-Zero (Cui et al., 2025a), SimpleRL-Zero (Zeng et al., 2025), Open-Reasoner-Zero (Hu et al., 2025), Oat-Zero (Liu et al., 2025b), LUFFY (Yan et al., 2025), and NFT (Chen et al., 2025), as well as two recent entropy-control algorithms: Clip-Cov (Cui et al., 2025b) and KL-Cov (Cui et al., 2025b). Since no results or checkpoints were released, we used the official codebase and scripts to train the models to convergence. Detailed descriptions of all methods are provided in Appendix C.

4.2 Overall Performance

As shown in Tab. 1, our two-stage CURE framework achieves SOTA performance on diverse mathematical benchmarks with high data efficiency, outperforming both RLVR and entropy-control methods. Starting from the Qwen-7B-Math-Base model and leveraging only 17 K samples, Stage 1 of CURE achieves an average score of 52.1%, surpassing all other methods such as CLIP-COV (50.9%) and NFT (51.7%). In the second, critic-bootstrapped stage, CURE further raises the average accuracy to 54.3%, with particularly marked gains on AIME24 (35.5% vs. 26.6%) and AMC23 (89.7% vs. 52.2%), representing a 107% improvement over the base model.

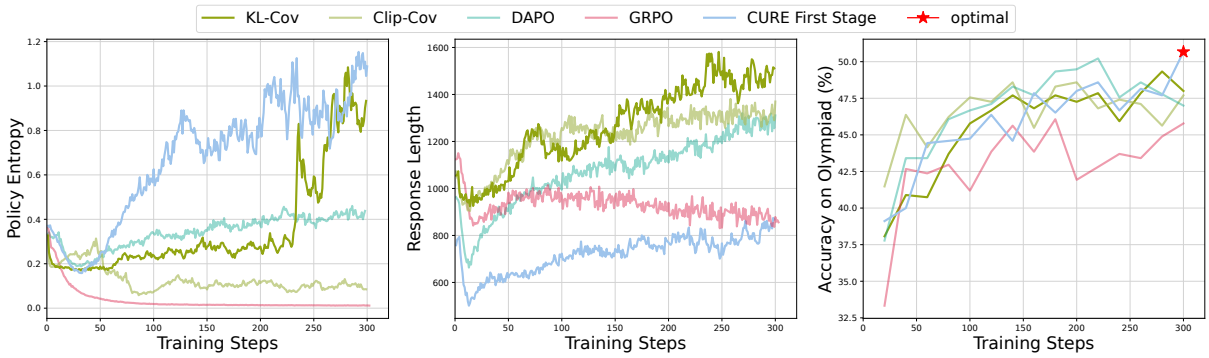

Figure 3: Depicts the evolution of policy entropy and sequence length on the training set, alongside validation accuracy on OlympiadBench, plotted over training steps for various methods.

4.3 Entropy Preservation

Fig. 3 plots, on the training set, (i) policy entropy vs. training steps and (ii) response length vs. training steps, together with (iii) OlympiadBench accuracy vs. training steps. From the left plot (policy entropy), the CURE First Stage exhibits the steepest entropy growth, indicating stronger exploration via a more diverse policy distribution. By contrast, GRPO's entropy collapses rapidly toward zero—signaling a loss of exploration—while DAPO and Clip-CoV plateau at moderate levels, reflecting only limited entropy regulation. As for KL-CoV, its entropy remains relatively low early on and then rises markedly in later training, approaching a higher-entropy regime. The middle plot (Response Length) shows that the CURE First Stage's responses grow longer throughout training; however, owing to its design—where queries splice responses before key tokens—the responses generated by CURE remain relatively shorter than those of other methods even as training advances. Meanwhile, the right plot (accuracy) indicates that the CURE First Stage achieves accuracy comparable to competing methods, striking a balance between exploration (driven by entropy) and exploitation (reflected in accuracy). Notably, at the stage-one high-entropy checkpoint, our first-stage model already delivers strong results.

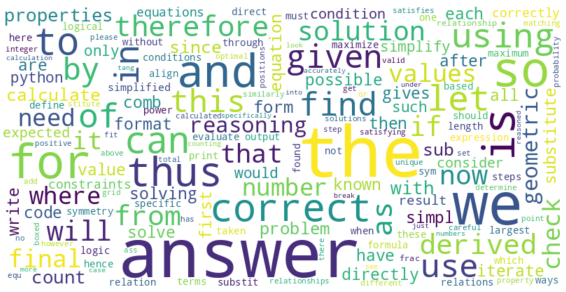

Figure 4: Word clouds generated by CURE First Stage.

As illustrated in Fig. 2 and Tab. 1, CURE consistently outperforms other RLVR approaches—particularly entropy-control algorithms—achieving the highest entropy across all six test sets.

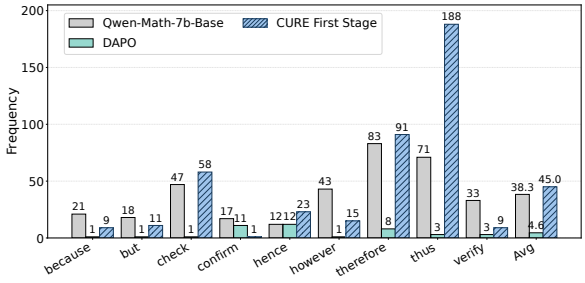

Figure 5: Probability of semantically meaningful high-entropy tokens across training paradigms in AIME24.

To further probe our high-entropy models, Fig. 4 shows the word cloud for CURE's first stage in AIME24, while Fig. 5 compares the frequencies of several key high-entropy connector tokens under DAPO and CURE. Consistent with Beyond the 80/20 Rule (Wang et al., 2025), which highlights these tokens as markers of new reasoning pathways, we observe that CURE not only restores their baseline usage but amplifies it, whereas DAPO's strong entropy penalty significantly suppresses them. After first-stage CURE training, such as "check" increases from 47 to 58 occurrences, "verify" surges from 33 to 188, and "hence" nearly doubles. This marked boost in connector usage shows that our high-entropy objective effectively promotes exploration and reinforces semantic transitions, providing a solid foundation for the subsequent continuing training stage.

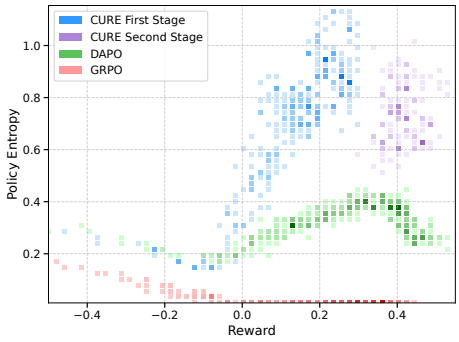

Figure 6: Scatter Plots of Policy Entropy vs. Reward for CURE , DAPO, and GRPO Methods at temperature 1.0

4.4 Second-Stage Research

Fig. 6 depicts the evolution of training entropy and reward. As the reward increases, the policy entropy of GRPO remains very low. DAPO exhibits an initial decrease before stabilizing within a relatively low-entropy regime. Compared with the first stage, CURE's second training phase exhibits markedly lower entropy overall, and the joint density of entropy and reward concentrates in the low-entropy, high-reward region—reflecting a gradual transition toward exploitation. Table 1 reports performance improvements from 52.1% to 54.3% across six benchmark test sets, further confirming this trend.

As shown in Fig. 7, compared to the DAPO, CURE achieves higher scores at low temperatures (0.6) by maintaining lower entropy, and at high temperatures (1.0) facilitates broader exploration through increased entropy. By making only slight adjustments to T, CURE achieves a controllable shift between exploration and exploitation, striking an optimal balance that delivers a consistent 5% accuracy improvement over DAPO on AIME24 across the entire temperature range.

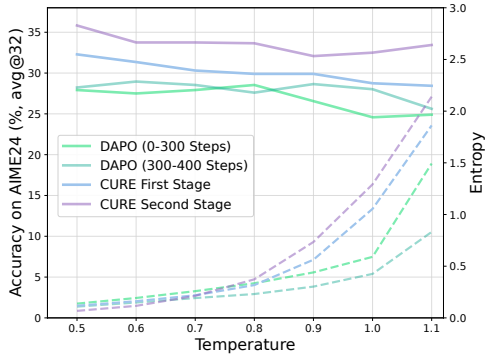

Figure 7: AIME24 accuracy and entropy vs. temperature for DAPO and CURE at different stages

4.5 Ablation on the Critical-Token Strategy

Model	AIME24	AMC	Minerva	Oly.	Avg.
Base	16.6	52.2	10.7	19.0	24.6
GRPO	31.0	80.8	40.1	44.9	49.2
DAPO	28.9	84.7	47.4	47.0	52.0
$\text{CURE}_{\text{Random}}$	35.7	80.3	41.5	48.2	51.4
$\text{CURE}_{\text{Entropy}}$	33.4	82.7	48.2	50.5	53.7

Table 2: Comparison of performance across various strategies on benchmark math-reasoning datasets. More details are provided in Sec. D.2.1

To isolate the effect of how we choose the truncation point $t^\star$ from Eq. 6, we compare two variants under the same training budget and hyperparameters, and include DAPO as a reference. $\text{CURE}_{\text{Random}}$ randomly selects a token for truncation and re-concatenation, preserving intervention frequency while ignoring token criticality. $\text{CURE}_{\text{Entropy}}$ selects a token from the top-20 highest-entropy positions, explicitly steering the model toward states of maximal distributional uncertainty. Finally, as Tab. 2, $\text{CURE}_{\text{Entropy}}$ achieves the best average performance, 53.7%, outperforming DAPO by +1.7% and $\text{CURE}_{\text{Random}}$ by +2.3%. In contrast, in terms of raw performance, $\text{CURE}_{\text{Random}}$ remains close to DAPO (51.4% vs. 52.0%, -0.6%). When we examine the training-set entropy, $\text{CURE}_{\text{Random}}$ shows only a 73% increase—far below the 137% gain of $\text{CURE}_{\text{Entropy}}$ both measured relative to the pre-training baseline. These results indicate that appending unguided random prefixes is ineffective, the intervention location is pivotal. Prioritizing high-entropy tokens offers a principled and effective criterion for selecting critical intervention points.

5 Conclusion

We identify static-sampling-induced entropy collapse as a major bottleneck in RLVR-based reasoning. To address this, we propose CURE, a two-stage framework: (1) injecting structured diversity via critical-token branching, and (2) consolidating gains through DAPO-based stable exploitation. CURE achieves SOTA performance on Qwen-2.5-Math-7B, outperforming existing RLVR methods by 5% on average. Our results demonstrate that this synergy between targeted exploration and controlled training is essential for sustaining policy entropy and improving accuracy.

6 Limitations

Our experimental scope is primarily constrained by limited computational resources. Consequently, we focus on medium-scale models in the Qwen family (up to 7B), and do not conduct extensive experiments on larger model sizes. With additional computational resources in the future, we aim to scale our approach to larger models and validate its effectiveness across more diverse architectures.

References

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. 2018. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*.

Huayu Chen, Kaiwen Zheng, Qinsheng Zhang, Ganqu Cui, Yin Cui, Haotian Ye, Tsung-Yi Lin, Ming-Yu Liu, Jun Zhu, and Haoxiang Wang. 2025. Bridging supervised learning and reinforcement learning in math reasoning. *arXiv preprint arXiv:2505.18116*.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. 2025. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. In *ICML*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, and 1 others. 2025a. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025b. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, and 1 others. 2025. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*.

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

hiyouga. 2025. Mathruler. https://github.com/hiyouga/MathRuler.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Hynek Kydlíček. Math-Verify: Math Verification Library.

Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. 2022. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q. Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. Hugging Face repository, 13:9.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025a. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Acemath: Advancing frontier math reasoning with post-training and reward modeling. *arXiv preprint arXiv:2412.15084*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2024. Magicoder: Empowering code generation with oss-instruct. In *Forty-first International Conference on Machine Learning*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. 2024. Advancing llm reasoning generalists with preference trees. *ArXiv*.

Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 2025. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. https://hkust-nlp.notion.site/simplerl-reason. Notion Blog.

Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, and 1 others. 2025. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*.

Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. 2025. R1-zero's" aha moment" in visual reasoning on a 2b non-sft model. *arXiv preprint arXiv:2503.05132*.

A Pseudocode

A.1 Pseudocode fot CURE Stage 1

Algorithm 1 CURE Stage 1

] **Input**: Training set loader, rollout counts N_1, N_2, top-K critical tokens, reward model $\mathcal{R}$, batch size B
Initialize: Parameter buffer $\mathcal{B} \leftarrow \emptyset$, step counter $g \leftarrow 0$

for each epoch $= 1$ to T **do**
  for each mini-batch $\mathbf{q}$ in loader **do**
    $\mathbf{o}_1 \leftarrow \textsc{Generate}(\mathbf{q}, n = N_1)$ // primary rollouts
    $\{t_i^\star\} \leftarrow \textsc{SelectCriticalTokens}(\mathbf{o}_1, \text{top-}K)$
    $\mathbf{q}' \leftarrow \textsc{ReconcatQuery}(\mathbf{q}, \mathbf{o}_1, \{t_i^\star\})$
    $\mathbf{o}_2 \leftarrow \textsc{Generate}(\mathbf{q}', n = N_2)$ // branch rollouts
    $\mathcal{T} \leftarrow \textsc{Merge}(\mathbf{o}_1, \mathbf{o}_2)$
    $\mathcal{T}' \leftarrow \textsc{Filter}(\mathcal{T})$ // dynamic sampling
    $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{T}'$
    if $|\mathcal{B}| < B$ **then**
      continue
    end if
    $\mathcal{D} \leftarrow$ First B samples in $\mathcal{B}$; $\mathcal{B} \leftarrow \emptyset$
    $\mathcal{L} \leftarrow \text{DAPO}\textsc{Optimize}(\mathcal{D})$
    $\textsc{UpdateParameters}(\mathcal{L})$
    $g \leftarrow g + 1$
  end for
end for

A.2 Pseudocode fot CURE Stage 2

Algorithm 2 CURE Stage 2 (DAPO Annealing)

Input: Training set loader, rollout counts $N = N_1 + N_1 * N_2$, top-K critical tokens, reward model $\mathcal{R}$, batch size B
Initialize: Parameter buffer $\mathcal{B} \leftarrow \emptyset$, step counter $g \leftarrow 0$

for each epoch $= 1$ to T **do**
  for each mini-batch $\mathbf{q}$ in loader **do**
    $\mathcal{T} \leftarrow \textsc{Generate}(\mathbf{q}, n = N)$ // rollouts
    $\mathcal{T}' \leftarrow \textsc{Filter}(\mathcal{T})$ // dynamic sampling
    $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{T}'$
    if $|\mathcal{B}| < B$ **then**
      continue
    end if
    $\mathcal{D} \leftarrow$ First B samples in $\mathcal{B}$; $\mathcal{B} \leftarrow \emptyset$
    $\mathcal{L} \leftarrow \text{DAPO}\textsc{Optimize}(\mathcal{D})$
    $\textsc{UpdateParameters}(\mathcal{L})$
    $g \leftarrow g + 1$
  end for
end for

B Why CURE can prevent entropy collapse?

From the perspective of traditional RL. A common line of work in exploration treats the (state, policy) uncertainty via the policy entropy $H(\pi(\cdot \mid s))$ and adds either an explicit entropy bonus or a state exploration bonus to the return:

$$J(\theta) \;=\; \mathbb{E}_{\pi_\theta}\Big[\sum_{t=0}^{\infty} \gamma^t \big(r(s_t, a_t) + \beta\, H(\pi_\theta(\cdot \mid s_t)) + b(s_t)\big)\Big], \tag{11}$$

where $b(s)$ increases visitation of uncertain or under-explored states (Ladosz et al., 2022; Ecoffet et al., 2019; Burda et al., 2018). Building on this insight, we propose to improve exploration by directly exposing an LLM-based agent to unfamiliar states, thereby steering data collection toward regions of high uncertainty and increasing exploration efficiency.

Operational recap. We first draw N_1 initial rollouts $\mathcal{G}(\mathbf{q}, N_1) = \{\mathbf{o}_i\}_{i=1}^{N_1} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{q})$. For each $\mathbf{o}_i$, we identify a high-entropy critical position $t_i^\star$ (Sec. 3.2.1, Eqs. 5–6), extract the frontier prefix $\mathbf{p}_i = \mathbf{o}_{i,1:t_i^\star-1}$, and construct the refined prompt $\mathbf{q}_i' = \mathbf{q} \,\|\, \mathbf{p}_i$ (Eq. 7). From each $\mathbf{q}_i'$ we sample N_2 additional trajectories $\mathcal{G}(\mathbf{q}_i', N_2)$ (Eq. 8), then group all rollouts as

$$\mathcal{G}(\mathbf{q}) = \mathcal{G}(\mathbf{q}, N_1) \;\cup\; \Big(\bigcup_{i=1}^{N_1} \mathcal{G}(\mathbf{q}_i', N_2)\Big), \qquad \big|\mathcal{G}(\mathbf{q})\big| = N_1 + N_1 * N_2 \tag{12}$$

(Eq. 9). We finally optimize the CURE objective in Eq. 10 with group-relative advantages, which empirically sustains higher policy entropy and improves rewards during exploration.

Why the re-prompted state is "unfamiliar": no gradient flows through the injected prefix. The policy is *not* trained to traverse the path $\mathbf{q} \to \mathbf{q}_i' = \mathbf{q} \,\|\, \mathbf{p}_i$. The prefix $\mathbf{p}_i$ is treated as an exogenous intervention on the initial state rather than as an action sequence to be reinforced. Formalizing this with a stop–gradient operator $\mathrm{sg}(\cdot)$,

$$\mathbf{q}_i' \;=\; \mathbf{q} \,\|\, \mathrm{sg}(\mathbf{p}_i), \qquad \nabla_\theta\, \mathbf{q}_i' \;=\; \mathbf{0},$$

so the policy gradient contains no term that explains or rewards how to "arrive" at $\mathbf{q}_i'$ from $\mathbf{q}$. Instead, for the $N_1 N_2$ rollouts that begin at the critical-token re-prompted state, the update includes likelihood factors only for tokens generated after the (re-)prompt is instantiated. By contrast, although the N_1 original rollouts initiated from $\mathbf{q}$ can carry gradients along trajectories that move from $\mathbf{q}$ toward $\mathbf{q}_i'$, they constitute only a small fraction of the batch and therefore contribute comparatively little to the overall update:

$$\nabla_\theta\, \mathcal{J}_{\text{CURE}}(\theta) = \mathbb{E}_{\mathbf{q}\sim P(Q)}\Bigg[\frac{1}{\sum_{\mathbf{o}_i \in \mathcal{G}(\mathbf{q})} |\mathbf{o}_i|} \sum_{\mathbf{o}_i \in \mathcal{G}(\mathbf{q})} \sum_{t=1}^{|\mathbf{o}_i|} \nabla_\theta\, r_{i,t}(\theta)\, \hat{A}_{i,t}^{\text{grp}}\Bigg], \tag{13}$$

where $s_{i,t} = \left(\tilde{\mathbf{q}}_{\mathbf{o}_i}, \mathbf{o}_{i,<t}\right)$ with $\tilde{\mathbf{q}}_{\mathbf{o}_i} \in \{\mathbf{q}\} \cup \{\mathbf{q}'_i\}_{i=1}^{N_1}$, $r_{i,t}(\theta)$ is the importance weight in Eq. 1, and $\hat{A}^{\text{grp}}_{i,t}$ is the group-relative advantage in Eq. 2. Crucially, there are no log-probability (hence no gradient) terms for tokens inside $\mathbf{p}_i$. The mapping $\mathbf{q} \mapsto \mathbf{q}'_i$ is therefore never reinforced, rendering $\mathbf{q}'_i$ a genuinely *novel* initial state that helps delay premature entropy collapse.

C Baselines

- **Eurus-2-7B-PRIME-Zero** (Cui et al., 2025a) is a reinforcement learning method for large language models that enables online updates of process reward models (PRMs) without requiring explicit process-level annotations. It leverages implicit process rewards derived from policy rollouts and outcome-level labels, removing the need for a separate reward model training phase. PRIME is compatible with various advantage functions and is designed to reduce reliance on costly supervision in multi-step reasoning tasks.

- **SimpleRL-Zero** (Zeng et al., 2025)is a rule-based reinforcement-learning recipe for math reasoning Zero RL Training. DeepSeek-R1 demonstrates that long chain-of-thought (CoT) reasoning can emerge from a simple reinforcement learning framework with rule-based rewards, starting directly from base models—a setting referred to as zero RL training. Recent work extends this paradigm to ten diverse base models, including LLama3-8B, Mistral-7B/24B, DeepSeek-Math-7B, and Qwen2.5 models from 0.5B to 32B. Key strategies include adjusting format rewards and controlling input difficulty to guide reasoning development during training.

- **Open-Reasoner-Zero** (Hu et al., 2025) is an open-source implementation of large-scale reasoning-oriented reinforcement learning directly on base models. ORZ adopts a minimalist approach using vanilla PPO with GAE and rule-based rewards, without KL regularization. It maintains scalability and training simplicity by eliminating auxiliary components. ORZ further incorporates a learned critic that penalizes repetitive patterns, enabling more stable advantage estimation.

- **Oat-Zero** (Liu et al., 2025b), analyzes two key components of the R1-Zero paradigm: base models and reinforcement learning algorithms. It examines how pretraining characteristics influence RL dynamics by comparing various base models, including DeepSeek-V3-Base and Qwen2.5 series. To address the optimization bias in Group Relative Policy Optimization (GRPO) that inflates response length, it introduces Dr. GRPO, an unbiased optimization method designed to improve token efficiency. Based on these analyses, it proposes a simplified R1-Zero training recipe.

- **LUFFY** (Yan et al., 2025) is an off-policy RLVR framework that augments on-policy learning with external reasoning traces, allowing models to acquire abilities beyond their own outputs. It mixes off-policy demonstrations with on-policy rollouts, combining Mixed-Policy GRPO whose convergence rate is theoretically guaranteed with policy shaping via regularized importance sampling to balance imitation and exploration while avoiding brittle, surface-level mimicry. It succeeds at training weak models where on-policy RLVR fails. The framework points toward scalable, data-efficient RLVR that leverages broad off-policy guidance while preserving exploration.

- **NFT** (Chen et al., 2025), is a supervised, verifier-driven training scheme that builds an implicit negative policy from an LLM's self-generated wrong answers while sharing parameters with the positive policy. By optimizing directly over all model generations using binary verifier feedback—without external teachers—NFT recycles failures into learning signals and removes the dependence on rejection sampling style supervision. Experiments on 7B and 32B math reasoning models show consistent gains over SL (Supervised Learning) baselines and parity or improvements versus leading RL methods such as GRPO and DAPO. Theoretically, NFT is equivalent to GRPO under strict on-policy training, helping bridge SL and RL in binary-feedback settings and opening a scalable path for self-improving LLMs.

- **Clip-CoV** (Cui et al., 2025b) is an entropy-aware reinforcement learning update that clips token-wise gradients when the covariance between action probability and logit change is

Name	Where in formulas	Role / Budget Impact
Initial Rollouts N_1	$\mathcal{G}(\mathbf{q}, N_1) = \{\mathbf{o}_i\}_{i=1}^{N_1} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{q})$ (Eq. 4)	Number of initial trajectories per query $\mathbf{q}_i$, adds N_1 samples.
Top-K Entropy K	$\mathcal{T}_K^{(i)} = \text{TopK}_t(H_{i,t}, K)$; $t_i^\star \sim \text{Uniform}(\mathcal{T}_K^{(i)})$ (Eq. 6)	Size of high-uncertainty candidate set for stochastic selection.
Re-Prompting Rollouts N_2	$\mathcal{G}(\mathbf{q}_i', N_2) = \{\mathbf{o}_j\}_{j=1}^{N_2} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{q}_i')$ (Eq. 8)	Number of trajectories per refined prompt $\mathbf{q}_i'$, adds $N_1 * N_2$ re-prompted samples.

Table 3: CURE hyperparameters and their occurrences in the corresponding formulas (Sec. 3.2.1).

high (an advantage-proportional signal). By limiting confidence amplification on dominant tokens, it preserves policy entropy, sustains exploration, and prevents early entropy collapse. Clip-Cov is plug-and-play with Policy Gradient variants, optimizer-agnostic, and incurs negligible overhead, prevents entropy collapse in multi-step reasoning.

- **KL-CoV** (Cui et al., 2025b) is a selective KL regularization that applies a KL penalty to tokens with high covariance, while leaving rare-but-promising actions less constrained. This targeted control curbs overconfident updates, maintains a healthy entropy trajectory, and encourages exploration. KL-CoV complements global KL control, works with common advantage functions, and prevents entropy collapse in reasoning tasks.

- **Beyond the 80/20 Rule** (Wang et al., 2025) presents an in-depth analysis of token-level entropy patterns in chain-of-thought reasoning, showing that a small subset of high-entropy "forking" tokens drives the model's multi-path exploration. By confining RLVR's policy-gradient updates to this critical 20 % of tokens, Qwen28 achieves performance on the Qwen3-8B base model that is indistinguishable from full-gradient training. **However, because the authors have not released the training code or any usable checkpoints, we did not include this work as a baseline.**

D Experiment Details

D.1 Hyperparameters and Statistics for the data

All experiments are conducted using 32 NVIDIA H20 GPUs, with each experiment completed within 72–120 h. We utilize the DAPO-Math-17K dataset, which consists of 17K high-quality mathematical problems for training. Tab. 4 presents the hyperparameters applied in both training stages. For the baseline model, we adopt the same DAPO hyperparameter configuration as used in the two-stage setup. Tab. 3 reports the CURE-specific hyperparameters.

D.2 Evaluation Protocol

Eurus-2-7B-PRIME-Zero, SimpleRL-Zero, Open-Reasoner-Zero, Oat-Zero, and LUFFY were retrieved from their official repositories and evaluated using each author's recommended sampling parameters. NFT's results are reported directly from its original paper, as neither the model nor its training code were released. In contrast, Clip-Cov and KL-CoV were trained in-house using the official training scripts provided by their authors with the initial model replaced by Qwen2.5-math-7b, since no pretrained checkpoints were available. To ensure the highest evaluation fidelity—and following the protocols established for Eurus-2-7B-PRIME-Zero and NFT we employed a combined verification pipeline (math-verify plus math-grader) and then manually reviewed every correct and incorrect sample across multiple test sets for a truly precise assessment.

Specifically, we report avg@32 for smaller datasets, including AIME24, AIME25, and AMC23, to provide a more robust performance estimate. Conversely, for larger-scale benchmarks such as MATH-500, Minerva, and OlympiadBench, we report avg@1. This setting fully follows prior work (e.g., NFT, RePO, Steering-Reasoning, and CFT) under the same evaluation protocol.

D.2.1 Evaluation Protocol of Ablation Study

Due to space constraints, Tab 2 in the main paper reports only a subset of benchmarks. We select representative benchmarks such that the relative comparisons remain unaffected. The complete results are provided in Tab 5.

Settings	First Stage	Second Stage
Training Settings		
GPU	$4 \times 8 \times$ H20	$4 \times 8 \times$ H20
Optimizer	AdamW	AdamW
LR	$1e-6$	$1e-6$
Warmup Steps	10	10
Weight Decay	0.1	0.1
Entropy Coeff	0	0
Grad Clip	1.0	1.0
Loss Agg Mode	token-mean	token-mean
Clip Ratio Low	0.2	0.2
Clip Ratio High	0.28	0.28
KL in Reward	False	False
KL Coeff	0.0	0.0
KL Loss	False	False
KL Loss Coeff	0.0	0.0
Adv Estimator	GRPO	GRPO
N Responses / Prompt	16	16
Train BSZ	512	512
Mini BSZ	32	32
Gen BSZ	1024	1024
Filter Groups	True	True
Metric	acc	acc
Max Gen Batches	10	10
Overlong Buffer	Enabled	Enabled
Overlong Buffer Len	512	512
Penalty Factor	1.0	1.0
Top-p	1.0	1.0
Top-k	-1	-1
Temperature	1.0	1.0
Sampling	Enabled	Enabled
Offload	False	False
Use Dynamic BSZ	True	True
Max Prompt Len	1500	512
Max Resp Len	2596	3584
Initial Rollouts N_1	4	N/A
Re-Prompting Rollouts N_2	3	N/A
Top-K **Entropy**	20	N/A
Testing Settings		
Top-p	0.7	0.7
Top-k	-1	-1
Temperature	0.6	0.6
Max Prompt Len	512	512
Max Resp Len	3584	3584

Table 4: Hyperparameter settings for CURE (First & Second Stage) on Qwen-2.5-7B-Math

E Case Study

E.1 Template

System: Please reason step by step and enclose your final answer in \boxed{} .

User: { Problem }

Assistant: { Answer }

We used the simplest template with no special modifications.

E.2 Policy Entropy Visualization

Since our initial model is Qwen2.5-Math-7B, we present here the variation of token-level entropy with token position in AIME24 in Fig. 8 and Fig. 9. It can be observed that a large number of high-entropy tokens are either connectives or words that determine the direction of reasoning. These tokens exhibit relatively high entropy, which we attribute to the fact that their corresponding prefixes are

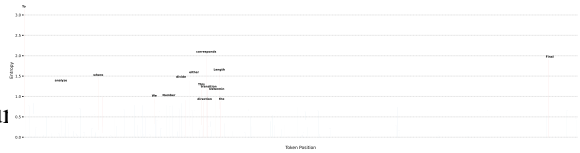

Figure 8: AIME24 Token-Level Entropy vs. Position (Qwen2.5-Math-7B) Case 1

Model	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg.
Base	16.6	6.3	52.2	52.4	10.7	19.0	26.2
GRPO	31.0	10.9	80.8	82.0	40.1	44.9	48.3
DAPO	28.9	18.2	84.7	82.4	47.4	47.0	51.4
$\text{CURE}_{\text{Random}}$	35.7	16.5	80.3	83.2	41.5	48.2	50.9
$\text{CURE}_{\text{Entropy}}$	33.4	15.3	82.7	82.4	48.2	50.5	52.1

Table 5: Comparison of performance across various strategies on math reasoning benchmarks.

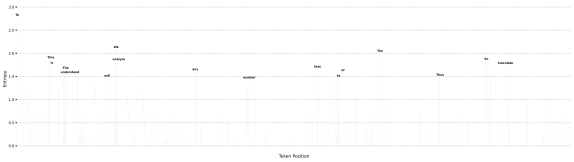

Figure 9: AIME24 Token-Level Entropy vs. Position (Qwen2.5-Math-7B) Case 2

more unfamiliar to the LLM. Therefore, we aim to treat the prefixes as new queries to enhance the model's exploratory capability.

E.3 Open Source

Our code builds upon the foundations of VERL. To enhance the reproducibility of our work and support community development, we will release the model, training code, and evaluation code in the near future.

E.4 Word Cloud

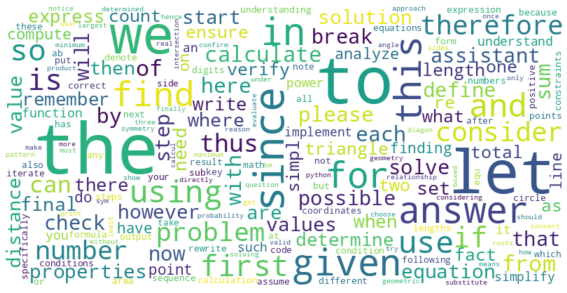

Figure 10: Word cloud generated by Qwen-2.5-Math-7B.

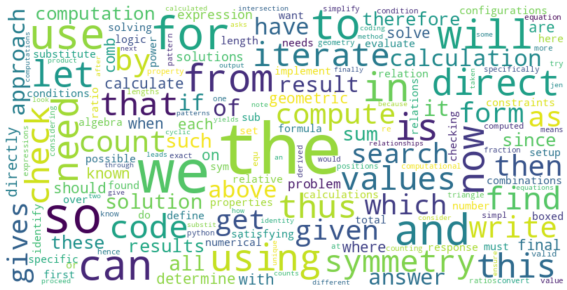

Figure 11: Word cloud generated by DAPO.

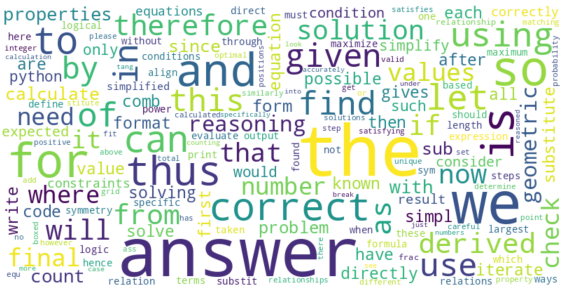

Figure 12: Word cloud generated by CURE First Stage.

F Future Work

Looking forward, CURE's data-generation intervention is orthogonal to reward shaping and KL-based regularization, suggesting broad applicability across models, tasks, and verification protocols. Future work includes adaptive scheduling of the two stages, principled detection of critical tokens beyond math (e.g., code and multimodal reasoning), theoretical analysis of entropy dynamics under branching, and systems-level optimizations for efficient batched regeneration. We believe this line of inquiry offers a practical recipe for sustained reasoning gains in RLVR: maintain exploration where it matters, then exploit deliberately.