OPTIMALITY-PRESERVING EXPLORATION ENHANCEMENT VIA CORRECTNESS-CONDITIONED ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) relies on exploration to discover high-quality solutions, yet the policy diversity of large language models (LLMs) often collapses early in training. Entropy regularization can alleviate this issue but is widely observed to reduce accuracy. Our geometric analysis attributes this tension to conflicting objectives: RLVR minimizes the projection distance to the optimal policy space, whereas entropy regularization pulls the policy toward a uniform distribution. Since the uniform distribution is generally not optimal, these two distances cannot be minimized simultaneously, implying an inherent conflict. We therefore propose Correctness-Conditioned Entropy (CCE) regularization, which regularizes the policy's projection onto the optimal policy space rather than the policy itself. Because the projection point and the projection distance are independent and can be optimized simultaneously, CCE regularization is optimality-preserving. We further show that RLVR with CCE is equivalent to variationally minimizing the Chernoff α-divergence to the maximum-entropy optimal policy, providing a principled interpretation of CCE. Empirically, integrating CCE into DAPO and MaxRL yields consistent improvements across most mathematical reasoning benchmarks. When added to DAPO, CCE improves average accuracy by 1.3–2.0 points while enabling more targeted exploration, achieving simultaneous gains in exploration and optimality.

1 INTRODUCTION

Reinforcement Learning with Verifiable Rewards (RLVR) has become one of the primary paradigms for improving the reasoning capabilities of large language models (LLMs) (Ouyang et al., 2022; Team et al., 2025; Guo et al., 2025; Comanici et al., 2025). LLMs repeatedly explore different reasoning trajectories to solve problems and update their reasoning policies through reinforcement learning (RL) according to the correctness of generated responses, thereby improving accuracy. Optimality is crucial for RLVR, as it determines whether the model can achieve the highest possible accuracy. Exploration and policy diversity are also important for RL, as they expand the action space explored during training and therefore increase the likelihood of discovering better solutions. Moreover, diverse reasoning strategies may improve generalization to unseen problems (Haarnoja et al., 2017). However, recent studies on RLVR have observed a rapid decline in policy diversity during training. Policy entropy can collapse within only a few optimization iterations (Yu et al., 2026), causing sampled trajectories to become increasingly concentrated and homogeneous. Therefore, maintaining or enhancing policy diversity and exploration during RLVR while preserving optimality has become an important objective.

However, existing approaches for improving exploration in RLVR are generally incompatible with optimality. Most methods primarily rely on entropy regularization (Haarnoja et al., 2017; 2018; Gai et al., 2026) or KL constraints (Ouyang et al., 2022) to alleviate the loss of exploration capability, yet they often observe degradation in the performance of the resulting models (Cui et al., 2025). Our theoretical analysis reveals that these regularizers effectively pull the policy toward a high-entropy but generally suboptimal policy, thereby driving the optimized policy away from the RLVR optimal policy space. Consequently, maximizing these regularization objectives conflicts with maximizing reward, forcing a trade-off between exploration and optimality. Although methods that mask entropy gradients from negative samples can alleviate performance degradation (Yao et al., 2026), we show later that they still fail to preserve optimality.

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.

Table 1: Comparison of different exploration-enhancement algorithms. Here, Opt.-Com. denotes optimality compatibility, and Div.-Pro. denotes explicit diversity promotion.

Regularization	Expression	Gradient	Opt.-Com.	Div.-Pro.
Clip-Cov (Cui et al., 2025)	-	-	✓	✗
KL-Cov (Cui et al., 2025)	-	-	✓	✗
Entropy (Haarnoja et al., 2017)	$-\mathbb{E}\left[\log \pi\right]$	$-\mathbb{E}\left[(1+\log \pi)\nabla \log \pi\right]$	✗	✓
DivPO (Yao et al., 2026)	$-\mathbb{E}\left[\log \pi \mid A^*\right]$	$-\mathbb{E}\left[(1+\log \pi)\,\nabla \log \pi \vert A^*\right]$	✗	✓
CCE (ours)	$-\mathbb{E}\left[\log \pi(\cdot\vert A^*)\vert A^*\right]$	$\mathbb{E}\left[(\mathbb{E}\left[\log \pi \vert A^*\right]-\log \pi)\,\nabla \log \pi \vert A^*\right]$	✓	✓

Our analysis of the conflict between entropy regularization and optimality suggests that preserving optimality requires a diversity measure that can be optimized jointly with the RLVR objective. To identify such a measure, we first show that the geometric essence of RLVR is to minimize the projection distance from the current policy to the optimal policy space. This suggests that a diversity measure compatible with RLVR should be defined on the projection point rather than on the policy itself, thereby making it independent of the projection distance. We further prove that this projected policy is exactly the posterior policy conditioned on correctness, which is both computable and sampleable. Motivated by this observation, we propose Correctness-Conditioned Entropy (CCE) regularization, which uses the entropy of the projected policy as the regularization term. We then prove that the optimal solution of CCE-regularized RLVR is exactly the maximum-entropy policy within the optimal policy space, establishing the compatibility of CCE with RLVR optimality. Furthermore, we show that optimizing RLVR with CCE regularization is equivalent to minimizing the Chernoff α-divergence to the maximum-entropy optimal policy, implying that even partial improvement of the regularized objective moves the policy closer to this target and is therefore meaningful.

Based on the above analysis, we integrate CCE regularization into mainstream policy optimization algorithms. We first derive an estimator for the CCE policy gradient and then integrate it into DAPO (Yu et al., 2026) and MaxRL (Tajwar et al., 2026). We compare the resulting methods with standard RLVR algorithms and various regularization approaches on mathematical reasoning tasks. Experimental results show that CCE regularization consistently improves both accuracy and policy diversity across most benchmarks for both DAPO and MaxRL, demonstrating that optimality and diversity can be improved simultaneously. The observed relationships among different diversity metrics during training further suggest that the design and study of diversity measures for RL remain an important direction for future research.

The main contributions of this work are summarized as follows:

- We provide a geometric interpretation of RLVR as minimizing the projection distance to the optimal policy space, and explain why entropy regularization conflicts with optimality.
- Based on this geometric analysis, we propose imposing diversity regularization on the projected policy and derive Correctness-Conditioned Entropy (CCE) regularization, while further establishing its optimality-preserving property and regression interpretation.
- We integrate CCE regularization into DAPO and MaxRL, achieving substantial improvements in both accuracy and policy diversity on mathematical reasoning tasks.

2 Related Work

Reinforcement learning (RL) post-training has become a primary approach for improving the reasoning capabilities of large language models (LLMs) (Comanici et al., 2025; Guo et al., 2025; Team et al., 2025; Achiam et al., 2023). By exploring diverse solution trajectories, RL increases the probability of correct reasoning paths and improves overall accuracy. Therefore, maintaining sufficient exploration capability during training is crucial, which has been extensively studied in traditional deep reinforcement learning (Haarnoja et al., 2017; 2018).

However, effective exploration enhancement remains challenging in LLM RL. In mainstream algorithms such as GRPO (Shao et al., 2024), policy entropy often rapidly collapses during early training, leading to insufficient exploration (Chen et al., 2026). Existing approaches mainly alleviate premature entropy reduction by modifying mechanisms such as clipping (Cui et al., 2025; Yu et al., 2026), but they only mitigate entropy collapse rather than actively promoting exploration. Entropy regularization can explicitly increase policy entropy but is widely observed to degrade training performance (Cui et al., 2025). DivPO attempts to address this issue by masking entropy gradients from negative samples (Yao et al., 2026); however, its uniform penalty on all tokens can disrupt deterministic decisions at critical tokens such as EOS, causing issues like non-terminating responses. In contrast, CCE regularization adaptively adjusts the regularization effect for each response, enabling exploration enhancement while preserving optimality. We summarize existing exploration-enhancement approaches in Table 1, which further supports our analysis.

3 PRELIMINARY

Reinforcement Learning with Verifiable Rewards. Reinforcement Learning with Verifiable Rewards (RLVR) improves LLMs on tasks with automatically verifiable correctness, such as mathematical reasoning and code generation. At each step t, the autoregressive policy π_θ takes the prompt s and preceding tokens $a_{0:t} = a_{<t}$ as input and predicts the next token a_t. Each response a receives a reward r based on correctness; rewards may vary across error types, while all fully correct responses receive the maximum reward $r^* = \max r$. RLVR optimizes π_θ by maximizing the expected reward:

$$\mathcal{L}(\pi_\theta) = \mathbb{E}_{p(a,s|\pi_\theta)}[r] = \mathbb{E}_{p(s)}\left[\sum_a r \prod_t \pi_\theta(a_t|a_{0:t}, s)\right]. \tag{1}$$

Since the sampling distribution depends on the policy being optimized, Equation 1 is generally intractable to optimize directly. In practice, policy-gradient estimators and auxiliary quantities such as advantage functions are commonly used instead (Williams, 1992).

Shannon Entropy. Shannon entropy (Shannon, 1948) $\mathcal{H}$ is a classical information-theoretic measure of distributional uncertainty and is widely used in RLVR to characterize policy diversity and exploration (Shao et al., 2024; Yu et al., 2026). For a distribution p over a random variable X, it is defined as

$$\mathcal{H}(p) = -\mathbb{E}_p[\log p] = -\sum_X p(X) \log p(X). \tag{2}$$

Entropy regularization is commonly used in RLVR to promote policy diversity; however, as we show later, it can compromise policy optimality.

4 ENHANCING EXPLORATION WHILE PRESERVING RLVR OPTIMALITY

4.1 OPTIMALITY AND EXPLORATION IN RLVR

Optimality. An optimal policy is one that maximizes the expected reward in Equation 1. Under the RLVR setting in Section 3, however, the optimal policy is generally not unique. Multiple reasoning trajectories may lead to a correct solution, and any policy that assigns all probability mass to the optimal action set $A_s^* = \{a \mid r(a, s) = r^*\}$ is optimal. These policies form the set

$$\Pi^* = \{\pi \mid \pi(a|s) = 0,\ \forall s, \forall a \notin A_s^*\}. \tag{3}$$

which generally contains multiple policies. In Appendix A, we prove that Π^* is exactly the solution space of minimizing Equation 1.

Maximum Exploration under Optimality. Since the optimal policy is generally not unique, additional criteria can be introduced to further select among optimal policies. Inspired by maximum-entropy RL (Haarnoja et al., 2017), we define the *maximum-entropy optimal policy* in RLVR as the policy with the highest entropy among all reward-optimal policies:

$$\pi^* = \arg\max_{\pi \in \Pi^*} \mathcal{H}(\pi) = \mathcal{U}_{A_s^*}. \tag{4}$$

Although Equation 4 admits the closed-form solution $\pi^*(a \mid s) = \mathcal{U}_{A_s^*}(a) = \frac{\mathbb{I}\{a \in A_s^*\}}{|A_s^*|}$, this does not mean that we can directly regress toward π^*. Minimizing $\mathbb{D}_{\mathrm{KL}}(\pi^* \| \pi)$ requires sampling from π^*,

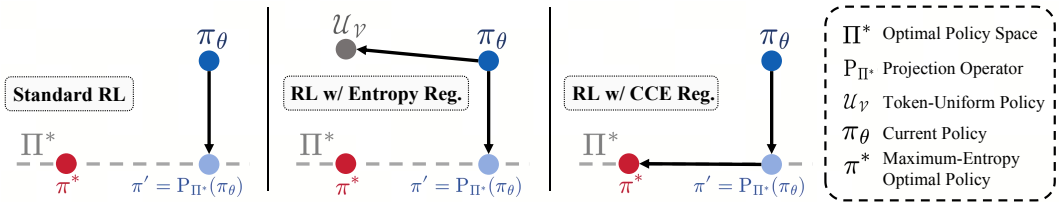

Figure 1: Standard RL minimizes the distance between π_θ and its projection π' onto Π^*. Entropy regularization (Reg.) additionally attempts to minimize the distance between π_θ and the non-optimal uniform distribution $\mathcal{U}_\mathcal{V}$, which cannot be minimized simultaneously with the projection distance. In contrast, CCE measures the distance between π' and the maximum-entropy optimal policy π^*, which is independent of the projection distance and can therefore be minimized simultaneously.

whereas $\mathbb{D}_{\mathrm{KL}}(\pi \| \pi^*) = +\infty$ whenever $\pi \notin \Pi^*$. Neither is generally practical in RLVR, motivating a regularization term that drives the policy toward π^*.

4.2 A Geometric View of Standard RLVR

Building on the notions of optimality and exploration, we provide a geometric perspective that explains why standard RL pursues optimality while leaving exploration unconstrained. Under binary rewards, maximizing the expected reward is equivalent to maximizing the probability of generating a correct response, in which case $\mathbb{E}_{\pi_\theta}[r|s] = \pi_\theta(a|s)$, and MaxRL maximizes the logarithm of this probability (Tajwar et al., 2026). For each environment, these two objectives are equivalent, and maximizing the log-probability of correctness is exactly equivalent to minimizing the projection distance from the current policy π_θ to the optimal policy space Π^*:

$$\log \pi_\theta(a \in A_s^*) = -\mathbb{D}_{\mathrm{KL}}\left(\mathrm{P}_{\Pi^*}(\pi_\theta) \,\|\, \pi_\theta\right) = -\min_{\pi' \in \Pi^*} \mathbb{D}_{\mathrm{KL}}\left(\pi' \,\|\, \pi_\theta\right), \tag{5}$$

where P_{Π^*} denotes the projection operator onto Π^*. It is easy to proof that this projection is exactly the posterior policy obtained by conditioning π_θ on correctness. As shown in Figure 1, Equation 5 indicates that standard RL only minimizes the distance from π_θ to the optimal policy space Π^*, without imposing any preference on the properties of the resulting projection point. In this sense, standard RL explicitly optimizes for optimality but does not explicitly control exploration. Following the analytical framework of (Cui et al., 2025; Chen et al., 2026), we show in Appendix A that tabular softmax policies indeed converge to such a projection point, with the entropy of the converged policy determined by the initial policy. Since policy optimization in LLMs has been shown to exhibit behavior similar to that of tabular softmax policy optimization, analogous dynamics are likely to arise in LLMs and have also been empirically observed (Cui et al., 2025). Consequently, standard RL without explicit diversity control is inherently susceptible to limited policy diversity.

4.3 Why Entropy Regularization Compromises Optimality

Geometric Interpretation. The most widely used approach for enhancing exploration in RLVR is to explicitly add a token-level policy entropy regularization term to the training objective (Gai et al., 2026). From a geometric perspective, however, maximizing entropy is equivalent to minimizing the distance to the uniform distribution:

$$\mathcal{H}\big(\pi_\theta(\cdot \mid s, a_{0:t})\big) = \log|\mathcal{V}| - \mathbb{D}_{\mathrm{KL}}\left(\pi_\theta \,\|\, \mathcal{U}_\mathcal{V}\right), \tag{6}$$

where $\mathcal{U}_\mathcal{V}$ denotes the token-level uniform distribution over the vocabulary $\mathcal{V}$. Equation 6 therefore shows that entropy regularization introduces an additional force that pulls the policy toward $\mathcal{U}_\mathcal{V}$. Since the uniform distribution generally does not lie in the optimal policy space Π^*, this additional force may conflict with the movement toward Π^*, thereby compromising RLVR optimality. Note that if sequence-level entropy regularization is used, a similar conclusion can be obtained by simply replacing $\mathcal{U}_\mathcal{V}$ in Equation 6 with the uniform distribution over responses.

Optimality Analysis. The geometric behavior of entropy regularization raises concerns about compromised optimality, and an analysis of its optimal solution further confirms this issue. When β is finite, the policy π_e that maximizes $\mathcal{L}(\pi_\theta) + \beta\,\mathbb{E}[\mathcal{H}(\pi_\theta)]$ takes the form

$$\pi_e(a_t \mid a_{0:t}, s) \propto \exp\left(\beta^{-1} Q^{\pi_e}(a_{0:t+1}, s)\right), \tag{7}$$

where $Q^{\pi_e}(a_{0:t+1}, s) = \mathbb{E}_{\pi_e}[r|a_{0:t+1}, s]$ denotes the action-value function under π_e, and the derivation is provided in Appendix A. When the reward is bounded, every token and sequence has nonzero probability under π_e. Consequently, π_e assigns nonzero probability to negative samples, implying that $\pi_e \notin \Pi^*$. This shows that standard entropy regularization compromises RLVR optimality, which may explain the widely observed degradation in accuracy. Although some approaches remove gradients from negative samples (Yao et al., 2026), each token in a positive sample is still subject to a penalty proportional to $1 + \log \pi_\theta$. This perturbs the policy even at inherently deterministic decision points, such as the EOS token, and can therefore still conflict with RLVR optimality. These limitations motivate the need for a regularization scheme that enhances exploration without compromising the original RLVR optimum.

4.4 Preserving Optimality via Correctness-Conditioned Entropy

Geometric Interpretation. Since standard RL minimizes the projection distance to the optimal policy space Π^*, the key to enhancing exploration without sacrificing optimality is to construct a measure that is independent of this projection distance. A natural choice is the distance between the projection point and the maximum-entropy optimal policy π^*. Simultaneously minimizing the projection distance and the distance from the projection point to π^* encourages π_θ to reach both Π^* and π^*, thereby achieving both optimality and exploration. Although the distance from the projection point to π^* is difficult to estimate directly, our derivation shows that minimizing this distance is equivalent to maximizing the entropy of the policy posterior conditioned on correctness:

$$\mathcal{H}_c(\pi_\theta) = \mathcal{H}\left(\pi_\theta(\cdot \mid \cdot \in A_s^*)\right) = \mathcal{H}\left(\mathrm{P}_{\Pi^*}(\pi_\theta)\right) = \log|A_s^*| - \mathbb{D}_{\mathrm{KL}}\left(\mathrm{P}_{\Pi^*}(\pi_\theta) \,\|\, \pi^*\right), \tag{8}$$

where we refer to the conditional entropy $\mathcal{H}_c$ as *Correctness-Conditioned Entropy* (CCE). We further introduce the policy optimization objective with CCE regularization:

$$\mathcal{L}_c(\pi_\theta) = \mathcal{L}(\pi_\theta) + \beta\, \mathbb{E}_{p(s)}\left[\mathcal{H}_c(\pi_\theta)\right]. \tag{9}$$

Since CCE maximizes the entropy of the projection point independently of the projection distance, it does not interfere with standard RL's projection-distance minimization.

Optimality Analysis. The geometric analysis suggests that CCE regularization is compatible with RLVR optimality, and the following analysis formally shows that this optimality is preserved for any positive value of β. Since CCE first projects π_θ onto the optimal policy space before evaluating its entropy, it is independent of the expected reward. We formalize this property in the following theorem:

Theorem 1 (Independence between CCE maximization and reward maximization)**.** *For any policy π and $\beta > 0$, under the RLVR formulation described in Section 3, if $\pi(a \in A_s^*) > 0$, the following properties hold: (1) $\max_{\mathbb{E}_\pi[r]=r_0} \mathcal{H}_c(\pi) = \log|A_s^*|$, $\forall \min r \le r_0 \le r^*$, (2) $\max_{\mathcal{H}_c(\pi)=h_0} \mathbb{E}_\pi[r] = r^*$, $\forall 0 \le h_0 \le \log|A_s^*|$.*

This theorem establishes that maximizing CCE is independent of maximizing the expected reward, since the value of CCE does not affect the reward-maximizing solution, and vice versa. Building on this property, we further show that, for any $\beta > 0$, CCE regularization uniquely induces π^*, which simultaneously maximizes both the expected reward and CCE. Therefore, CCE regularization preserves RLVR optimality while simultaneously maximizing exploration.

5 CCE as Maximum-Entropy Optimal Policy Alignment

Although $\arg\max_\pi \mathcal{L}_c(\pi) = \pi^* \in \Pi^*$ guarantees that CCE preserves the optimal solution, in practice, optimization algorithms may fail to reach the global optimum exactly. This raises an important question: *what does the CCE-regularized objective actually optimize?* If optimizing $\mathcal{L}_c$ indeed corresponds to minimizing some notion of distance to π^*, then the entire optimization trajectory remains meaningful, since every iteration attempts to move the policy closer to π^*.

Section 4.1 shows that CCE regularization minimizes the KL divergence from the projected policy $\mathrm{P}_{\Pi^*}(\pi_\theta)$ to π^*, whereas standard RL minimizes the distance from π_θ to $\mathrm{P}_{\Pi^*}(\pi_\theta)$. Therefore, optimizing Equation 9 can be interpreted as solving

$$\min_{\pi'=\mathrm{P}_{\Pi^*}(\pi_\theta),\pi_\theta} \mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') = \min_{\pi'=\mathrm{P}_{\Pi^*}(\pi_\theta),\pi_\theta} \left\{\alpha\, \mathbb{D}_{\mathrm{KL}}(\pi' \| \pi^*) + (1-\alpha)\mathbb{D}_{\mathrm{KL}}(\pi' \| \pi_\theta)\right\}, \tag{10}$$

where $\alpha = 1/(1+\beta)$ is a constant under MaxRL, whereas under standard RL it becomes a dynamic quantity determined by the accuracy of the behavior policy. Since $\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi')$ measures a weighted sum of the distances from an intermediate policy π' to π_θ and π^*, respectively, it is natural to ask whether it upper-bounds a direct divergence between π_θ and π^*. We show that such a divergence indeed exists and is precisely the Chernoff α-divergence C_α:

$$\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') \geq C_\alpha(\pi_\theta, \pi^*) = \min_{\pi} \mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi). \tag{11}$$

Equation 11 provides two important insights. First, Equation 9 minimizes an upper bound on the divergence between π_θ and π^*, and can therefore be interpreted as a form of variational regression toward π^*. Second, the equality on the right-hand side shows that, if π' is also treated as an optimization variable, minimizing $\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi')$ is equivalent to minimizing the Chernoff α-divergence:

$$\min_{\pi_\theta} \min_{\pi' \in \Pi^*} \mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') = \min_{\pi_\theta} C_\alpha(\pi_\theta, \pi^*), \tag{12}$$

where π' is restricted to Π^* because otherwise $\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') \to +\infty$. Although Equation 12 is theoretically valid, jointly optimizing π' is computationally expensive and practically infeasible, as it would require explicitly constructing a parameterized policy constrained to lie in Π^*. Fortunately, the search space can be reduced without changing the optimum. In particular, $\pi' = \mathrm{P}_{\Pi^*}(\pi_\theta)$ satisfies $\pi' \in \Pi^*$ and also holds at the global optimum. Therefore, imposing this additional constraint leaves the optimal solution unchanged:

$$\min_{\pi_\theta} \min_{\pi' = \mathrm{P}_{\Pi^*}(\pi_\theta)} \mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') = \min_{\pi_\theta} \min_{\pi' \in \Pi^*} \mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') = \min_{\pi_\theta} C_\alpha(\pi_\theta, \pi^*). \tag{13}$$

The left-hand side is exactly equivalent to the CCE-regularized objective in Equation 9. Therefore, CCE-regularized policy optimization not only minimizes an upper bound on the divergence between π_θ and π^*, but is also equivalent to exactly minimizing their Chernoff α-divergence through a reparameterization. This also sheds light on the role of the CCE regularization coefficient β: although it does not affect optimality, it determines the divergence along which the policy is regressed toward π^*, which may in turn influence the convergence rate and overall training performance.

6 Practical Optimization of CCE

Similarly to the policy gradient, $\mathcal{H}_c$ is not directly tractable, and we therefore estimate its gradient using Monte Carlo samples for policy updates. Specifically, the gradient of CCE with respect to the model parameters is given by

$$\nabla \mathcal{H}_c(\pi_\theta(\cdot|s)) = -\mathbb{E}_{\pi_\theta(a|a \in A_s^*)} \left[\sum_t \frac{\nabla \pi_\theta(a_t|a_{0:t}, s)}{\pi_\theta(a_t|a_{0:t}, s)} \left(\log \pi_\theta(a|s) - \mathbb{E}_{\pi_\theta(\cdot|\cdot \in A_s^*)} \left[\log \pi_\theta \right] \right) \right]. \tag{14}$$

Although Equation 14 can be estimated directly using online Monte Carlo sampling, practical RL frameworks often use a behavior policy that differs from the target policy (Sheng et al., 2025; Liang et al., 2025; von Werra et al., 2020). This design improves hardware utilization (Hu et al., 2026; Tian et al., 2026), but makes Equation 14 difficult to estimate accurately. Therefore, following classical approaches such as PPO (Schulman et al., 2017), we construct an offline estimator of the CCE gradient and employ clipping to constrain policy shifts, thereby mitigating the resulting approximation error (Schulman et al., 2015):

$$L_{\mathrm{cce}} = \tfrac{1}{N} \textstyle\sum_{i,t} \min\{\omega_{i,t} \hat{A}_i,\ \mathrm{clip}\left(\omega_{i,t}, \tau_l, \tau_h\right) \hat{A}_i\}, \tag{15}$$

$$\hat{A}_i = A_i - \beta \tfrac{N}{K_i} \mathbb{I}_{a_i \in A^*_{s_i}, K_i \geq 2} \left(\log \pi_{\mathrm{old}}(a_i|s_i) - \tfrac{1}{K_i - 1} \textstyle\sum_{a_j \in A^*_{s_i},\, j \neq i,\, s_i = s_j} \log \pi_{\mathrm{old}}(a_j|s_i) \right), \tag{16}$$

where N denotes the total number of tokens, and K_i denotes the number of positive responses associated with the prompt of the i-th response. A_i denotes the advantage of the i-th response, which can be estimated using either a standard RL algorithm or MaxRL. The importance ratio $\omega_{i,t} = \frac{\pi_\theta(a_{i,t}|s_i, a_{i,0:t})}{\pi_{\mathrm{old}}(a_{i,t}|s_i, a_{i,0:t})}$ represents the policy ratio for the t-th token of the i-th response. Equation 16 uses π_{old} to construct offline estimates of both the policy gradient and the CCE gradient, while the standard clipping mechanism controls the policy shift and, consequently, the associated estimation error. Following the proof techniques of prior work (Schulman et al., 2015; Han et al., 2026), we derive a TRPO-style bound on the offline estimation error in Appendix A, which is upper-bounded by the token-level policy shift. This result shows that the offline estimation error of CCE regularization can likewise be controlled by constraining the policy shift, which is precisely the primary role of the clipping mechanism (Schulman et al., 2017).

7 EXPERIMENTS

7.1 EXPERIMENTAL SETUP

Training. We primarily conduct experiments on mathematical reasoning tasks, with results in other settings provided in Appendix D. Specifically, we train Qwen2.5-7B, Qwen3-4B, and Llama3.2-1B-Instruct using 32K samples from DeepMath-103k (He et al., 2025), DAPO-17k (Yu et al., 2026), and the Competition Math dataset (Hendrycks et al., 2021), with maximum response lengths of 7K, 9K, and 5K tokens, respectively. All experiments are implemented with Verl v0.8.0 (Sheng et al., 2025) and use binary rewards. For Qwen2.5-7B, we set the group size to 16 and perform 8 parameter updates per generation batch, while the remaining experiments use a group size of 8 with 4 updates per batch. The learning rate is set to 1×10^{-6} and the weight decay to 0.1 for all experiments.

Baselines. To comprehensively evaluate the effectiveness of CCE regularization, we compare it with popular RLVR algorithms, including REINFORCE++ (RF++) (Hu et al., 2025), GRPO (Shao et al., 2024), MaxRL (Tajwar et al., 2026), and DAPO (Yu et al., 2026). We also compare with methods that mitigate policy collapse through training techniques, including Clip-Cov and KL-Cov (Cui et al., 2025), as well as standard entropy regularization (Entropy) (Gai et al., 2026), which explicitly promotes policy entropy, and DivPO (Yao et al., 2026), which masks entropy gradients from negative samples. For all regularization-based methods, we train separate runs with regularization coefficients β selected from $\{1 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-2}\}$ and report the best result. For all DAPO-based methods, we disable dynamic sampling, as we find that it approximately doubles the training time without yielding significant performance improvements. All other methods use their default hyperparameters.

Benchmarks. We evaluate training performance on GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), AMC23, Olympiad (He et al., 2024), Minerva (Lewkowycz et al., 2022), AIME24, AIME25, and three HMMT benchmarks (2402, 2502, and 2511) (Dekoninck et al., 2026). We average the three HMMT scores to obtain the final HMMT score and average all benchmark scores to obtain the overall score.

7.2 BENCHMARK PERFORMANCE

CCE leads to consistent performance improvements. The benchmark performance of all models before and after training is reported in Table 2. CCE-regularized methods achieve substantially higher average scores than other baselines across all settings and obtain the best results on most individual benchmarks. In contrast, standard entropy regularization causes noticeable performance degradation in nearly all experiments, except for the Llama 1B model, which already has the highest initial entropy and slightly outperforms DAPO after entropy regularization. This is consistent with our analysis in Section 4.1 and observations reported by the community (Cui et al., 2025). Although DivPO alleviates this degradation, it still underperforms standard DAPO in most settings, which we later show is related to the disruption of critical decision points. Clip-Cov and KL-Cov, based on the correlation between log-likelihood and advantage, are relatively stable but provide limited gains due to explicitly removing policy gradients from some critical tokens. In contrast, CCE achieves the highest overall accuracy and, as shown later, also maintains strong policy diversity.

CCE Regularization with MaxRL. Beyond standard RL, we also apply CCE regularization to MaxRL and report the results in Table 3. CCE consistently improves MaxRL across all settings, achieving higher average accuracy than standard MaxRL. However, its performance remains slightly below that of CCE-regularized DAPO in Table 2. We attribute this gap to the MaxRL objective, which optimizes the logarithm of accuracy. Since this objective is unbounded below, it introduces higher estimation variance than conventional objectives that directly optimize accuracy. Therefore, although Section 4.1 shows that MaxRL more directly minimizes the two distances, its higher variance outweighs this theoretical advantage in practice.

Table 3: Comparison of the average accuracy across all benchmarks for different models trained with MaxRL, with (w/) and without CCE regularization. The best results are highlighted in **bold**.

Method	Qwen7B	Qwen4B	Llama1B
MaxRL	38.11	60.08	14.72
w/ CCE	**39.37**	**60.90**	**15.89**

Table 2: Benchmark scores. Qwen2.5-7B, Qwen3-4B, and Llama3.2-1B-Instruct are trained for one epoch on Deepmath-103K, DAPO-17K, and Competition Math, respectively. We report accuracy on standard mathematical benchmarks, and the average accuracy across all benchmarks (AVG). "Base" refers to the base model, "Entropy" denotes DAPO with entropy regularization. **Bold** numbers indicate the best results for each metric.

Method	AIME24 ↑	AIME25 ↑	GSM8k ↑	MATH500 ↑	AMC23 ↑	Olympiad ↑	Minerva ↑	HMMT ↑	AVG ↑
Qwen2.5-7B									
Base	4.00	3.33	59.06	41.40	26.00	18.22	9.19	3.11	20.54
RF++	12.53	9.20	90.30	73.60	53.67	34.47	24.63	4.39	37.85
GRPO	15.20	10.13	89.46	72.80	56.67	35.66	**26.47**	4.61	38.88
MaxRL	**17.73**	8.80	91.21	71.00	54.50	34.18	24.26	3.22	38.11
DAPO	14.93	8.67	91.36	74.00	59.17	36.26	23.90	4.00	39.03
Clip-Cov	14.13	8.13	90.45	73.20	57.17	34.92	25.55	5.00	38.57
KL-Cov	13.60	12.13	89.84	73.60	54.83	36.40	24.26	**5.05**	38.72
Entropy	14.53	7.60	90.98	72.40	51.17	36.70	24.08	3.61	37.63
DivPO	15.73	6.00	90.52	70.40	60.17	36.11	24.08	3.67	38.33
CCE	15.33	**13.07**	**92.49**	**76.40**	**62.33**	**39.52**	25.00	4.00	**41.02**
Qwen3-4B									
Base	38.93	26.53	93.48	83.40	74.17	47.40	29.60	18.55	51.51
RF++	54.67	41.73	93.93	87.20	84.17	53.34	**31.99**	24.71	58.97
GRPO	55.73	44.13	92.95	**89.00**	88.67	56.46	**31.99**	27.44	60.80
MaxRL	54.93	42.80	**94.47**	87.20	87.33	54.53	31.80	27.55	60.08
DAPO	55.60	44.53	93.93	87.80	89.00	57.21	31.62	27.32	60.88
Clip-Cov	55.60	43.73	93.71	87.40	87.50	56.76	31.25	27.49	60.43
KL-Cov	54.13	43.07	**94.47**	87.80	88.00	**58.69**	31.43	27.77	60.67
Entropy	53.47	42.13	93.03	87.80	87.33	56.91	31.80	27.27	59.97
DivPO	55.20	44.67	94.16	87.80	86.17	54.68	31.25	27.49	60.18
CCE	**59.47**	**47.33**	**94.47**	88.40	**89.67**	57.21	**31.99**	**28.99**	**62.19**
Llama3.2-1B-Instruct									
Base	0.53	0.00	4.85	14.20	5.33	4.16	1.47	0.22	3.84
RF++	1.73	0.27	52.24	30.00	15.67	6.69	5.33	**1.22**	14.14
GRPO	1.47	0.40	53.45	31.60	17.33	6.69	**7.17**	0.28	14.80
MaxRL	3.07	0.13	55.95	30.20	14.33	7.28	6.07	0.72	14.72
DAPO	2.53	0.00	53.75	34.00	15.50	7.43	5.15	0.61	14.87
Clip-Cov	2.80	0.27	56.48	34.60	15.00	8.32	5.70	1.17	15.54
KL-Cov	3.20	**0.93**	56.79	33.40	18.00	6.24	4.60	0.72	15.48
Entropy	2.53	0.00	52.31	33.20	16.17	8.17	6.25	0.61	14.91
DivPO	3.20	0.40	54.59	33.00	17.50	7.88	5.70	0.61	15.36
CCE	**4.40**	0.13	**57.01**	**35.00**	**18.83**	**9.81**	4.96	0.50	**16.33**

Ablation Studies. To investigate the properties of CCE, we conduct an ablation study on the regularization coefficient β. The accuracy, entropy, and CCE of models trained with different β values are reported in Table 4. As β increases, both entropy and CCE consistently increase, validating the effectiveness of our offline CCE gradient estimator. Notably, although entropy also increases, the growth is much smaller than that induced by standard Entropy regularization or DivPO, which can raise entropy above 10 even with $\beta = 1 \times 10^{-5}$. CCE is also robust across a wide range of β: all tested values outperform training without regularization. The performance differences among them may result from the fact that different β values correspond to optimizing different divergences for regression toward π^*, which can have different convergence and variance properties, as discussed in Section 5.

Table 4: Ablation study on the CCE regularization coefficient β. We report the average benchmark accuracy (Acc.), the CCE and entropy at the final checkpoint.

β	Acc.↑	CCE↑	Entropy↑
0	39.03	121.23	0.21
1×10^{-5}	39.37	136.52	0.21
1×10^{-4}	40.44	187.03	0.22
5×10^{-4}	41.02	511.88	0.47
1×10^{-3}	40.07	1809.69	1.02

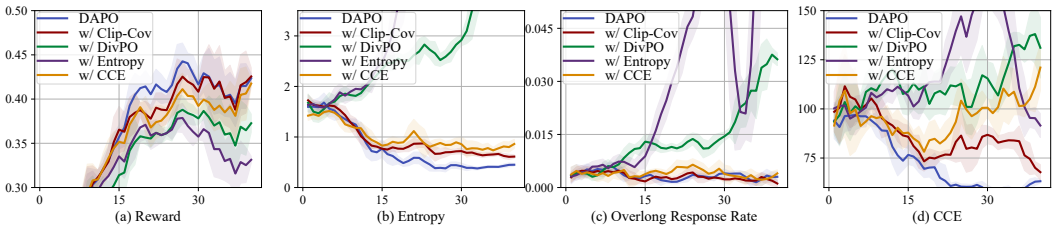

Figure 2: Training curves of reward, entropy, overlong-response rate, and CCE under different policy optimization algorithms in the Llama3.2-1B-Instruct experiments.

7.3 Effects of Regularization on Training Dynamics

Effects of Different Regularizers. To investigate the behavior of different regularization methods, we visualize the evolution of key training metrics under different entropy-enhancement approaches, as shown in Figure 2. Standard DAPO and Clip-Cov achieve the highest rewards in Figure 2(a), as they do not introduce regularization terms that interfere with reward maximization. Among regularized methods, CCE achieves the highest reward and remains close to the unregularized baselines, demonstrating its compatibility with RLVR optimality. Entropy regularization rapidly increases policy entropy in Figure 2(b), while DivPO still suffers from the risk of entropy explosion despite masking entropy gradients from negative samples. Figure 2(c) reveals that DivPO may perturb critical tokens such as EOS, causing generation failures and a steadily increasing overlong-response rate. Figure 2(d) further shows that although CCE only moderately increases overall entropy, it achieves one of the highest CCE values among all methods.

Relationships among Entropy, CCE, and Accuracy. To analyze the relationship between entropy and accuracy, we visualize entropy, CCE, and reward under different algorithms. As shown in Figure 3(a), DAPO, Clip-Cov, and CCE show a similar trend to (Cui et al., 2025), with reward strongly negatively correlated with log-entropy ($\rho < -0.9$). In contrast, this correlation becomes positive under Entropy regularization and DivPO. Figure 3(b) shows that, CCE and entropy are strongly positively correlated across almost all methods ($\rho > 0.8$), suggesting that these methods increase CCE mainly by increasing overall entropy. Under CCE regularization, however, the correlation is weak ($\rho = -0.22$), indicating a more targeted increase in CCE. This shows that CCE and conventional entropy are distinct diversity measures and need not be tightly coupled. Therefore, diversity optimization should not be limited to entropy maximization, as entropy may not be the most direct or effective diversity metric.

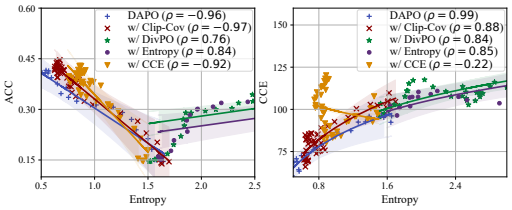

Figure 3: Relationships among entropy, CCE, and reward under different algorithms in the Llama3.2-1B-Instruct experiments, where ρ denotes the Pearson correlation coefficient between the two variables shown in each plot.

8 Conclusion

To address the loss of exploration capability in RLVR and the conflict between entropy regularization and optimality, we propose Correctness-Conditioned Entropy (CCE) regularization. Since standard RL can be interpreted as minimizing the projection distance from the policy to the optimal policy space, CCE first projects the policy onto this space before computing entropy. This separates exploration enhancement from optimality optimization, enabling improved exploration while preserving optimality. We establish the validity of CCE from both optimality and policy regression perspectives and demonstrate its effectiveness on mathematical reasoning tasks. Despite its strong performance, extending sequence-level CCE regularization to the token level and generalizing CCE beyond RLVR to broader classes of reward functions remain promising directions for future research.

AI USE STATEMENT

In this work, we used generative AI tools solely for grammatical polishing. They were not used for any core research activities, including theoretical modeling, mathematical formulation, experimental design, method implementation, data processing, or manuscript drafting. Tasks involving synthetic dataset generation or qualitative data analysis are not applicable to this work. We carefully reviewed all AI-assisted content for correctness and take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide sufficient details to facilitate reproduction of our theoretical and empirical results. All major theoretical results are presented in the main text following the derivation flow, with complete proofs provided in Appendix A. The main experimental results and training configurations are reported in the experimental section, while additional implementation details are provided in Appendix D. We use the open-source Verl reinforcement learning frameworks, and all models and datasets used in our experiments are publicly available.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Peter Chen, Xiaopeng Li, Ziniu Li, Wotao Yin, Xi Chen, and Tianyi Lin. Exploration vs exploitation: Rethinking RLVR through clipping, entropy, and spurious reward. In *International Conference on Learning Representations*, volume 2026, pp. 134506–134540, 2026.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. *arXiv preprint arXiv:2002.05867*, 2020.

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

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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. *arXiv preprint arXiv:2505.22617*, 2025.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs. 2026. URL https://arxiv.org/abs/2605.00674.

Jingchu Gai, Guanning Zeng, Huaqing Zhang, Han Zhong, Yige Hong, Andrej Risteski, and Aditi Raghunathan. Demystifying entropy control in llm rl training: Theoretical analysis and dynamic scheduling. In *Forty-third International Conference on Machine Learning*, 2026.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *International conference on machine learning*, pp. 2160–2169. PMLR, 2019.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Wei Han, Yuanxing Liu, Mingda Li, Ruiyu Xiao, Weinan Zhang, and Ting Liu. MRPO: Magnitude-regularized policy optimization via l1 constraints. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=e2xQL4BWrY.

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

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. REINFORCE++: Stabilizing critic-free policy optimization with global normalization. *arXiv preprint arXiv:2501.03262*, 2025.

Tianhao Hu, Xiangcheng Liu, Youshao Xiao, Yang Zheng, Xuan Huang, Jinrui Ding, Yufei Zhang, Tao Liang, Hongyu Zang, Quan Chen, et al. DORA: A scalable asynchronous reinforcement learning system for language model training. *arXiv preprint arXiv:2604.26256*, 2026.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind through question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5872–5877, 2019.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Jiaqing Liang, Jinyi Han, Xinyi Wang, Zishang Jiang, Chengyuan Xiong, Boyu Zhu, Jie Shi, Weijia Li, Tingyun Li, and Yanghua Xiao. KW-R1: A simple implementation of the GRPO algorithm. https://github.com/lsdefine/simple_GRPO, 2025.

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

Soumya Sanyal, Zeyi Liao, and Xiang Ren. RobustLR: A diagnostic benchmark for evaluating logical robustness of deductive reasoners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9614–9631, 2022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

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. *arXiv preprint arXiv:2402.03300*, 2024.

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*, pp. 1279–1297, 2025.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, 2021.

Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. *arXiv preprint arXiv:2602.02710*, 2026.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. KIMI K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Minghao Tian, Yunfei Xie, and Chen Wei. How off-policy can GRPO be? Mu-GRPO for efficient LLM reinforcement learning. *arXiv preprint arXiv:2605.17570*, 2026.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformer reinforcement learning. `https://github.com/huggingface/trl`, 2020.

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

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*, 38:94801–94826, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

A PROOFS

Lemma 1 (The optimal solution space in RLVR)**.** *Under the RLVR formulation introduced in Section 3, the solution set that maximizes Equation 1 is characterized by Equation 3.*

Proof. Let $r^* = \max_a r(a)$. It is straightforward to see that $\max \mathcal{L}(\pi) = r^*$. For any $\pi \in \Pi^*$, we have $\mathcal{L}(\pi) = r^*$ by definition, implying that Π^* belongs to the optimal solution set. In contrast, for any $\pi \notin \Pi^*$, there exists an action a' satisfying $r(a') < r^*$ and $\pi(a') > 0$. Therefore, we have:

$$\mathcal{L}(\pi) = \pi(a')\,(r(a') - r^*) + \mathbb{E}_\pi\left[r \mid r(a') = r^*\right] \leq \pi(a')\,(r(a') - r^*) + r^* < r^*. \tag{17}$$

This implies that π cannot be an optimal policy. Therefore, the set of optimal policies is contained in Π^*. Combined with the previous result, we conclude that the optimal policy space is exactly Π^*. □

Lemma 2 (The conflict between entropy regularization and RLVR optimality)**.** *The optimal policy for maximizing $\mathcal{L}(\pi_\theta) + \beta\,\mathbb{E}[\mathcal{H}(\pi(\cdot|a_{0:t}, s))]$ is expressed in Equation 7.*

Proof. According to the theory of Markov decision processes (MDPs) (Geist et al., 2019), maximizing $\mathcal{L}(\pi_\theta) + \beta\,\mathbb{E}[\mathcal{H}(\pi(\cdot|a_{0:t}, s))]$ is equivalent to maximizing:

$$\mathbb{E}_{\pi(a_t|a_{0:t},s)}\left[Q^{\pi_e}(a_{0:t+1}, s) - \beta \log \pi(a_t|a_{0:t}, s)\right]. \tag{18}$$

Considering $\sum_{a_t} \pi(a_t|\cdot) = 1$, the KKT conditions for maximizing Equation 18 are:

$$Q^{\pi_e}(a_{0:t+1}, s) - \beta - \beta \log \pi(a_t|a_{0:t}, s) + \lambda = 0. \tag{19}$$

Solving the KKT conditions in Equation 19 gives the optimal policy in Equation 7. □

Theorem 1 For any policy π and $\beta > 0$, under the RLVR formulation described in Section 3, if $\pi(a \in A_s^*) > 0$, the following properties hold: (1) $\max_{\mathbb{E}_\pi[r]=r_0} \mathcal{H}_c(\pi) = \log|A_s^*|$, $\forall \min r \le r_0 \le r^*$, (2) $\max_{\mathcal{H}_c(\pi)=h_0} \mathbb{E}_\pi[r] = r^*$, $\forall 0 \le h_0 \le \log|A_s^*|$.

Proof. Property (1) is straightforward: $\mathcal{H}_c$ is maximized simply by distributing the probability mass assigned to correct actions uniformly across all correct actions.

For property (2), let π_0 satisfy $\mathcal{H}_c(\pi_0) = h_0$. It is straightforward that the conditional policy $\pi_0(\cdot|a \in A_s^*)$ has the same CCE value h_0 while achieving the maximum expected reward r^*. Since $\max \mathbb{E}_\pi[r] = r^*$, we have $\max_{\mathcal{H}_c(\pi)=h_0} \mathbb{E}_\pi[r] = r^*$. This complete the proof. □

Lemma 3. *Equation 11 holds, and when $\pi' \notin \Pi^*$, we have $\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') \to +\infty$.*

Proof. Since the following relation holds:

$$\begin{aligned}
\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') &= \alpha\,\mathbb{D}_{\mathrm{KL}}(\pi'||\pi^*) + (1-\alpha)\,\mathbb{D}_{\mathrm{KL}}(\pi'||\pi_\theta) = \mathbb{E}_{\pi'}\left[\log \frac{\pi'(a|s)^\alpha}{\pi^*(a|s)^\alpha} \frac{\pi'(a|s)^{1-\alpha}}{\pi_\theta(a|s)^{1-\alpha}}\right] \\
&= -\log Z_\alpha + \mathbb{E}_{\pi'}\left[\log \frac{\pi'(a|s) Z_\alpha}{\pi^*(a|s)^\alpha \pi_\theta(a|s)^{1-\alpha}}\right] \\
&= C_\alpha(\pi^*, \pi_\theta) + \mathbb{D}_{\mathrm{KL}}(\pi'||q_\alpha) \ge C_\alpha(\pi^*, \pi_\theta),
\end{aligned} \tag{20}$$

where $q_\alpha(a) = \frac{\pi^*(a|s)^\alpha \pi_\theta(a|s)^{1-\alpha}}{Z_\alpha}$, $C_\alpha(\pi^*, \pi_\theta) = -\log Z_\alpha$, and $Z_\alpha = \sum_a \pi^*(a \mid s)^\alpha \pi_\theta(a \mid s)^{1-\alpha}$. Clearly, when $\pi' = q_\alpha$, we have $\mathbb{D}_{\mathrm{KL}}(\pi'\|q_\alpha) = 0$, and the equality is attained. Moreover, since $q_\alpha \in \Pi^*$, whenever $\pi' \notin \Pi^*$, we have $\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') \ge \mathbb{D}_{\mathrm{KL}}(\pi'\|q_\alpha) = +\infty$. □

Theorem 2 (Deterministic Optimality of Tabular RLVR)**.** *Assume that the tabular softmax policy π is updated using natural policy gradient (Cui et al., 2025). Under the RLVR setting described in Section 3, the policy eventually converges to a fixed optimal policy π^*, where $\pi^*(a) = \pi_0(a|a \in A_s^*)$.*

Proof. Under the natural policy gradient update, the tabular softmax policy is updated according to (Cui et al., 2025):

$$\pi_{k+1}(a) = \frac{\pi_k(a) \exp\left(\eta_k r(a)\right)}{\mathbb{E}_{\pi_k}\left[\exp\left(\eta_k r\right)\right]}, \tag{21}$$

where π_k and π_{k+1} denote the policies at the k-th and $(k+1)$-th iterations, respectively, and η_k denotes the update step size at the k-th iteration.

Considering two consecutive iterations, we can obtain:

$$\begin{aligned}
\pi_{k+2}(a) &= \frac{\pi_{k+1}(a) \exp\left(\eta_{k+1} r(a)\right)}{\mathbb{E}_{\pi_{k+1}}\left[\exp\left(\eta_{k+1} r\right)\right]} \\
&= \frac{\frac{\pi_k(a) \exp(\eta_k r(a))}{\mathbb{E}_{\pi_k}[\exp(\eta_k r)]} \exp\left(\eta_{k+1} r(a)\right)}{\sum_{a'} \frac{\pi_k(a') \exp\left(\eta_k r(a')\right)}{\mathbb{E}_{\pi_k}[\exp(\eta_k r)]} \exp\left(\eta_k r(a')\right)} \\
&= \frac{\pi_k(a) \exp\left(\eta_k r(a)\right) \exp\left(\eta_{k+1} r(a)\right)}{\sum_{a'} \pi_k(a') \exp\left(\eta_k r(a')\right) \exp\left(\eta_k r(a')\right)} \\
&= \frac{\pi_k(a) \exp\left(\left(\eta_k + \eta_{k+1}\right) r(a)\right) \exp\left(\eta_{k+1} r(a)\right)}{\mathbb{E}_{\pi_k}\left[\exp\left(\left(\eta_k + \eta_{k+1}\right) r\right)\right]}.
\end{aligned} \tag{22}$$

By recursively applying Equation 22, the final policy π can be expressed as:

$$\pi(a) \propto \pi_0(a) \exp\left(r \cdot \textstyle\sum_k \eta_k\right), \tag{23}$$

where π_0 denotes the initial policy. Under the assumption that the policy converges to the optimal policy set Π^*, if the reward is bounded, the cumulative update step size $\sum_k \eta_k \to +\infty$ necessarily holds. In this case, the convergence point can be expressed as:

$$\pi(a) \propto \mathbb{I}_{a \in A^*} \pi_0(a) \propto \pi_0(a | a \in A_s^*). \tag{24}$$

This completes the proof. □

Lemma 4. *Using CCE regularization in MaxRL is equivalent to minimizing $\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi')$ when $\pi' = \pi_\theta(\cdot|\cdot \in A_s^*)$.*

Proof. Let $\beta = \frac{\alpha}{1-\alpha}$. Then, we have:

$$\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi') = (1-\alpha)\left(\mathbb{D}_{\mathrm{KL}}(\pi' || \pi_\theta) + \beta\, \mathbb{D}_{\mathrm{KL}}(\pi' || \pi^*)\right). \tag{25}$$

When $\pi' = \pi_\theta(\cdot|\cdot \in A_s^*)$, minimizing $\mathbb{D}_{\mathrm{KL}}(\pi' \| \pi_\theta)$ is equivalent to maximizing the MaxRL objective:

$$\begin{aligned} \mathbb{D}_{\mathrm{KL}}(\pi' || \pi_\theta) &= \mathbb{E}_{\pi'}\left[\log \frac{\pi'(a|s)}{\pi_\theta(a|s)}\right] \\ &= \mathbb{E}_{\pi'}\left[\log \frac{1}{\pi_\theta(a \in A_s^*)}\right] = -\log \pi_\theta(a \in A_s^*), \end{aligned} \tag{26}$$

where $\log \pi_\theta(a \in A_s^*)$ is the log-probability of generating a correct response and serves as the objective function of MaxRL (Tajwar et al., 2026).

Meanwhile, when $\pi' = \pi_\theta(\cdot \mid \cdot \in A_s^*)$, maximizing $\mathcal{H}_c(\pi_\theta)$ is equivalent to minimizing $\mathbb{D}_{\mathrm{KL}}(\pi' \| \pi^*)$:

$$\begin{aligned} \mathbb{D}_{\mathrm{KL}}(\pi' || \pi^*) = \mathbb{E}_{\pi'}\left[\log \frac{\pi'(a|s)}{\pi^*(a|s)}\right] &= \mathbb{E}_{\pi'}\left[\log \pi'(a|s) + \log |A_s^*|\right] \\ &= -\mathcal{H}(\pi') + \log |A_s^*| = -\mathcal{H}_c(\pi_\theta) + \log |A_s^*|, \end{aligned} \tag{27}$$

where $\log |A_s^*|$ is a constant and therefore does not need to be explicitly optimized. Substituting Equations 27 and 26 into Equation 25 completes the proof. □

Lemma 5. *Using CCE regularization in standard RL is equivalent to minimizing $\mathcal{M}_\alpha(\pi_\theta, \pi^*; \pi')$ when $\pi' = \pi_\theta(\cdot \mid \cdot \in A_s^*)$, with a dynamic α determined by the accuracy of the behavior policy.*

Proof. Since the actual policy gradient used in the MaxRL algorithm $\mathbb{E}_{\pi_{\text{old}}}\left[\frac{1}{\pi_{\text{old}}(a \in A_s^*)} A \frac{\nabla \pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}\right]$ is simply a positive scalar rescaling of the standard RL policy gradient:

$$\mathbb{E}_{\pi_{\text{old}}}\left[\frac{1}{\pi_{\text{old}}(a \in A_s^*)} A \frac{\nabla \pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}\right] = \mathbb{E}_{p(s)}\left[\frac{1}{\pi_{\text{old}}(a \in A_s^*)} \mathbb{E}_{\pi_{\text{old}}}\left[A \frac{\nabla \pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}\right]\right], \tag{28}$$

where A is the advantage function $\mathbb{E}_{\pi_{\text{old}}}\left[A \frac{\nabla \pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}\right]$ is the standard RL policy gradient, while setting $\beta(s) = \frac{\alpha}{1-\alpha} \propto \pi_{\text{old}}(a \in A_s^*)$ cancels the scaling factor $\frac{1}{\pi_{\text{old}}(a \in A_s^*)}$. □

Lemma 6. *Standard RL is equivalent to minimizing the distance from the current policy π_θ to its projection onto the optimal policy space Π^*.*

Proof. The proof follows directly from Equation 26. □

Lemma 7. *$\pi^* = \arg\max_\pi \mathcal{L}_c(\pi)$ holds.*

Proof. Since $\mathcal{L}_c(\pi^*)$ serves as both an upper and a lower bound on $\mathcal{L}_c$:

$$\mathcal{L}_c(\pi^*) = r^* + \beta \log |A_s^*| = \max_\pi \mathcal{L}(\pi) + \beta \max_\pi \mathcal{H}_c(\pi) \geq \max_\pi \mathcal{L}_c(\pi), \tag{29}$$

and

$$\mathcal{L}_c(\pi^*) \leq \max_\pi \mathcal{L}_c(\pi), \tag{30}$$

moreover, since it is independent of β, we have $\mathcal{L}_c(\pi^*) = \max_\pi \mathcal{L}_c(\pi)$. □

Lemma 8. *The gradient of CCE is given by Equation 14.*

Proof. The penultimate line takes the same form as Equation 14:

$$\begin{aligned}\nabla\mathcal{H}_c(\pi) &= -\nabla\left\{\mathbb{E}_{\pi(a|a\in A_s^*)}\left[\log\frac{\pi(a|s)}{\pi(a\in A_s^*)}\right]\right\}\\ &= -\nabla\left\{\frac{1}{\pi(a\in A_s^*)}\mathbb{E}_\pi\left[\mathbb{I}_{a\in A_s^*}\log\pi(a|s)\right]-\log\pi(a\in A_s^*)\right\}\\ &= \textstyle\sum_{a\in A_s^*}\left\{\frac{\mathbb{E}_\pi[\mathbb{I}_{a\in A_s^*}\log\pi(a|s)]}{\pi(a\in A_s^*)^2}-\frac{\mathbb{I}_{a\in A_s^*}(1+\log\pi(a|s))}{\pi(a\in A_s^*)}+\frac{1}{\pi(a\in A_s^*)}\right\}\nabla\pi(a|s)\\ &= \mathbb{E}_{\pi(a|a\in A_s^*)}\left[\left\{\mathbb{E}_{\pi(\cdot|\cdot\in A_s^*)}[\log\pi]-\log\pi(a|s)\right\}\frac{\nabla\pi(a|s)}{\pi(a|s)}\right]\end{aligned} \tag{31}$$

□

Lemma 9. *The offline estimator of the CCE gradient is linear in $\nabla\pi_\theta$ and is exact when $\pi_\theta=\pi_{\text{old}}$.*

Proof. Since:

$$\begin{aligned}&\mathbb{E}_{\pi_{\text{old}}}\left[\frac{1}{N}\sum_i\mathbb{I}_{a_i\in A^*}\frac{N}{K_i}\left\{\frac{\sum_{j\neq i,s_i=s_j,a_j\in A_s^*}\log\pi_{\text{old}}(a_j|s_i)}{K_i-1}-\log\pi_{\text{old}}(a_i|s_i)\right\}\frac{\nabla\pi_\theta(a_i|s_i)}{\pi_{\text{old}}(a_i|s_i)}\right]\\ =&\mathbb{E}_{p(s)}\left[\mathbb{E}_{\pi_{\text{old}}}\left[\frac{1}{K_i}\sum_{a_i\in A_s^*}\left\{\frac{\sum_{j\neq i,s_i=s_j,a_j\in A_s^*}\log\pi_{\text{old}}(a_j|s)}{K_i-1}-\log\pi_{\text{old}}(a_i|s)\right\}\frac{\nabla\pi_\theta(a_i|s)}{\pi_{\text{old}}(a_i|s)}\right]\right]\\ =&\mathbb{E}_{p(s)}\left[\mathbb{E}_{\pi_{\text{old}}(a_{i,j}|a_{i,j}\in A_s^*,s)}\left[\left\{\log\pi_{\text{old}}(a_j|s)-\log\pi_{\text{old}}(a_i|s)\right\}\frac{\nabla\pi_\theta(a_i|s)}{\pi_{\text{old}}(a_i|s)}\right]\right]\\ =&\mathbb{E}_{p(s),\pi_{\text{old}}(a|a\in A_s^*)}\left[\left\{\mathbb{E}_{\pi_{\text{old}}(\cdot|\cdot\in A_s^*)}[\log\pi_{\text{old}}]-\log\pi_{\text{old}}(a|s)\right\}\frac{\nabla\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}\right]\\ =&\mathbb{E}_{p(s),\pi_{\text{old}}(a|a\in A_s^*)}\left[\left\{\mathbb{E}_{\pi_{\text{old}}(\cdot|\cdot\in A_s^*)}[\log\pi_{\text{old}}]-\log\pi_{\text{old}}(a|s)\right\}\textstyle\sum_t\frac{\nabla\pi_\theta(a_t|a_{0:t},s)}{\pi_{\text{old}}(a_t|a_{0:t},s)}\right],\end{aligned} \tag{32}$$

substituting the condition into the above expression and applying Lemma 8 completes the proof. □

Theorem 3. *Assume that the expected accuracy of every prompt under the behavior policy π_{old} is nonzero. Then, the error of the CCE gradient estimated offline using π_{old} is upper bounded by $O\left(\mathbb{E}_{\pi_{\text{old}}}\left[\mathbb{D}_{\text{TV}}\left(\pi_\theta\,\|\,\pi_{\text{old}}\right)\right]\right)$, where $\mathbb{D}_{\text{TV}}$ denotes the token-level total variation distance.*

Proof. By Lemma 9, the offline estimator of the CCE gradient is equivalent to optimizing the first-order Taylor expansion $\hat{L}$ of CCE with respect to π_θ around π_{old}. Under the nonzero-accuracy assumption, the approximation error of this expansion is first-order:

$$\begin{aligned}\left|\mathbb{E}_{p(s)}\left[\hat{L}-\mathcal{H}_c\right]\right|\leq\mathbb{E}_{p(s)}\left|\hat{L}-\mathcal{H}_c\right|&\leq\mathbb{E}_{p(s)}\left[\frac{C}{2}\textstyle\sum_a|\pi_\theta(a|s)-\pi_{\text{old}}(a|s)|\right]\\ &= C\,\mathbb{E}_{p(s)}\left[\mathbb{D}_{\text{TV}}(\pi_\theta(\cdot|s)||\pi_{\text{old}}(\cdot|s))\right],\end{aligned} \tag{33}$$

where C is a constant. Note that the TV distance here is defined at the sequence level rather than the token level.

According to the results of MRPO (Han et al., 2026), the following inequality holds:

$$\left|\mathbb{E}_{\pi_{\text{old}}}[X]-\mathbb{E}_{\pi_\theta}[X]\right|\leq C_X\,\mathbb{E}_{\pi_{\text{old}}}\left[\mathbb{D}_{\text{TV}}(\pi_\theta||\pi_{\text{old}})\right], \tag{34}$$

where $\mathbb{D}_{\text{TV}}(\pi_\theta\|\pi_{\text{old}})$ denotes the token-level total variation (TV) distance between the two policies, and C_X is a finite constant.

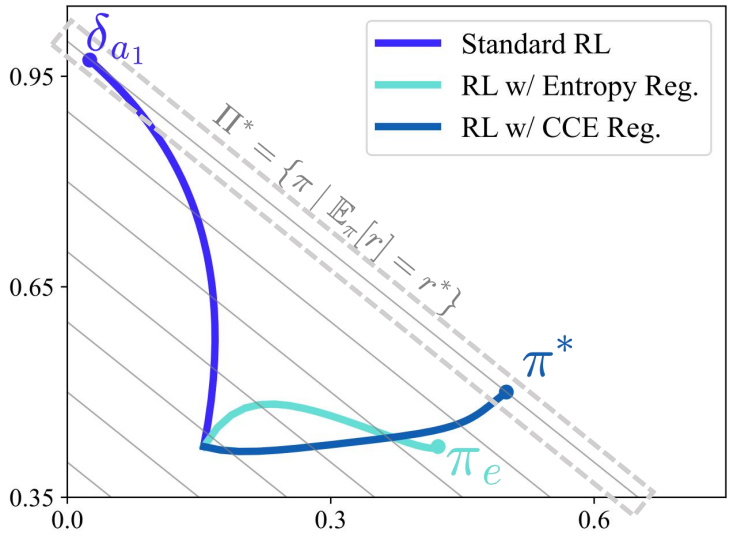

Figure 4: Policy convergence trajectories in a three-action environment. The axes denote the probabilities of the two correct actions. Standard RL rapidly collapses to the greedy policy δ_{a_1}, losing diversity. Entropy regularization preserves diversity but converges to $\pi_e \notin \Pi^*$, conflicting with optimality. In contrast, CCE converges to the maximum-entropy optimal policy π^*, achieving both optimality and exploration.

By defining $X = \text{sgn}[\pi_{\text{old}}(a|s) - \pi_\theta(a|s)]$ and applying Equation 34, we obtain:

$$\begin{aligned}\mathbb{D}_{\text{TV}}(\pi_\theta(\cdot|s)||\pi_{\text{old}}(\cdot|s)) &= \mathbb{E}_{\pi_{\text{old}}}\left[\frac{1}{\pi_{\text{old}}(a|s)}\left|\pi_\theta(a|s) - \pi_{\text{old}}(a|s)\right|\right] \\ &= \mathbb{E}_{\pi_{\text{old}}}\left[\frac{\pi_{\text{old}}(a|s) - \pi_\theta(a|s)}{\pi_{\text{old}}(a|s)} X\right] \\ &= \mathbb{E}_{\pi_\theta}[X] - \mathbb{E}_{\pi_{\text{old}}}[X] \leq C_X\, \mathbb{E}_{\pi_{\text{old}}}\left[\mathbb{D}_{\text{TV}}(\pi_\theta||\pi_{\text{old}})\right].\end{aligned} \tag{35}$$

By substituting Equation 35 into Equation 33, the proof is completed. □

B Simple Validation

We validate different algorithms in a simple environment with only three actions, where the first two actions are correct. The policy is modeled as a softmax over its parameters and updated using gradient descent. The convergence trajectories of different algorithms are shown in Figure 4. Standard RL quickly converges to a greedy policy δ_{a_1}, indicating a complete loss of diversity. Although entropy regularization assigns nonzero probability to both correct actions and thereby preserves diversity, its convergence point π_e does not lie in Π^*, resulting in a conflict with optimality. In contrast, CCE regularization converges to π^*, the maximum-entropy policy among all optimal policies.

C Detailed Experimental Settings

The main training parameters have already been reported in the main text. Here, we provide more detailed training parameters in Table 5.

D More Experimental Results

D.1 Effect of the Regularization Strength

The effects of different regularization terms and coefficients on model reward and CCE are shown in Figure 5. CCE regularization can readily increase CCE by using a larger regularization coefficient β,

Table 5: Common parameters used during training. Algorithm-specific parameters are either provided in the main text or set to their default values.

Parameter	Qwen7B	Qwen4B	Llama1B
kl coef	0	0	0
train batch size	256	256	256
group size	16	8	8
max prompt length	1k	1k	1k
max response length	7k	9k	5k
learning rate	1e-6	1e-6	1e-6
clip ratio low	0.20	0.20	0.20
clip ratio high	0.28	0.28	0.28
ppo mini batch size	32	64	64
lr warmup steps	5	5	5
weight decay	0.1	0.1	0.1
use kl loss	false	false	false
use dynamic bsz	true	true	true
grad clip	1.0	1.0	1.0
ulysses sequence parallel size	2	1	1
tensor model parallel size	2	1	1
gpu memory utilization	0.8	0.8	0.8
number of gpus	4	4	2
training time (hours)	~ 20	~ 65	~ 2

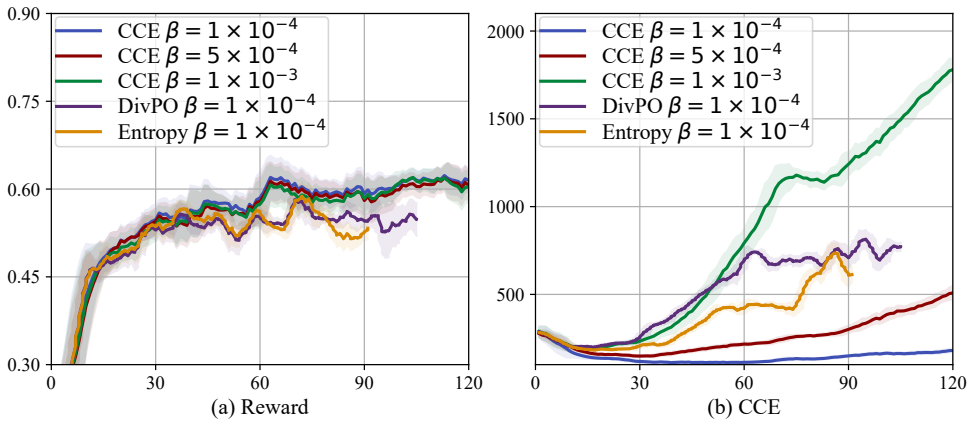

Figure 5: Reward and CCE trajectories of Qwen2.5-7B under different regularization terms and regularization coefficients β. As β increases, CCE regularization continues to improve CCE while having little effect on the reward. In contrast, the CCE of DivPO and entropy regularization fails to improve further due to entropy explosion, accompanied by a decline in accuracy.

whereas Entropy regularization and DivPO provide only limited improvements and adversely affect reward, as shown in Figure 5(a). Although Entropy regularization and DivPO can also increase CCE in Figure 5(b), their policy entropy has already exceeded 10 at this point, pushing training close to collapse and preventing further improvement in CCE.

D.2 MaxRL with CCE Regularization

CCE regularization applied to MaxRL provides a more direct form of policy regression than applying CCE regularization to standard RL, as discussed in Section 5. In the main experimental results, we report the overall score of MaxRL with CCE regularization; here, we further provide the detailed scores on each benchmark, as shown in Table 6. Adding CCE regularization improves MaxRL on

Table 6: More RLVR results. Qwen2.5-7B, Qwen3-4B, and Llama3.2-1B-Instruct are trained for one epoch on a subset of Deepmath-103K, DAPO-17K, and Competition Math, respectively. We report accuracy on standard mathematical reasoning benchmarks, as well as the average accuracy across all benchmarks (AVG). "Base" refers to the metrics of the base model. **Bold** numbers indicate the best results for each metric.

Method	AIME24 ↑	AIME25 ↑	GSM8k ↑	MATH500 ↑	AMC23 ↑	Olympiad ↑	Minerva ↑	HMMT ↑	AVG ↑
Qwen2.5-7B									
Base	4.00	3.33	59.06	41.40	26.00	18.22	9.19	3.11	20.54
MaxRL	**17.73**	**8.80**	**91.21**	71.00	54.50	34.18	**24.26**	3.22	38.11
w/ CCE	14.93	8.27	**91.21**	**75.20**	**57.17**	**38.93**	23.90	**5.33**	**39.37**
Qwen3-4B									
Base	38.93	26.53	93.48	83.40	74.17	47.40	29.60	18.55	51.51
MaxRL	54.93	**42.80**	**94.47**	87.20	87.33	54.53	31.80	**27.55**	60.08
w/ CCE	**58.13**	41.60	94.39	**87.80**	**89.33**	**57.21**	**32.72**	26.05	**60.90**
Llama3.2-1B-Instruct									
Base	0.53	0.00	4.85	14.20	5.33	4.16	1.47	0.22	3.84
MaxRL	3.07	**0.13**	**55.95**	30.20	14.33	7.28	**6.07**	0.72	14.72
w/ CCE	**4.00**	**0.13**	54.44	**34.60**	**20.50**	**8.47**	4.04	**0.94**	**15.89**

Table 7: Diversity metrics of different algorithms in the Qwen2.5-7B experiments. We report the average Jaccard distance (Jaccard), average 4-gram distance (4-gram), coefficient of variation of response length (CV), and accuracy (Acc.) of positive samples on the AIME24 and AIME25 benchmarks. **Bold** indicates the highest value.

Method	AIME24				AIME25			
	Jaccard	4-Gram	CV	Acc.	Jaccard	4-Gram	CV	Acc.
Base	0.14	0.11	**0.14**	4.00	0.26	0.14	0.12	3.33
DAPO	0.13	0.11	0.05	14.93	0.26	0.12	0.09	8.67
Clip-Cov	0.23	0.17	0.09	14.13	0.27	**0.20**	0.11	8.13
Entropy	0.17	0.12	0.07	14.53	0.23	0.15	0.11	7.60
CCE	**0.33**	**0.18**	0.10	**15.33**	**0.32**	0.19	**0.13**	**13.07**

most benchmarks as well as in the overall score, further demonstrating the effectiveness of CCE regularization.

D.3 Diversity Evaluation

To investigate response diversity, we report diversity metrics on AIME24 and AIME25 for models trained with different algorithms in the Qwen2.5-7B experiments. As shown in Table 7, models trained with CCE regularization outperform the other methods on most diversity metrics. Although entropy regularization explicitly increases policy entropy, the diversity among correct samples remains lower than that achieved by Clip-Cov. In addition, both Clip-Cov and entropy regularization obtain lower scores than standard DAPO, suggesting a potential conflict with optimality.

D.4 DAPO with Dynamic Sampling

On Qwen2.5-7B, we retrain DAPO using exactly the same experimental settings as in the main text, except that dynamic sampling is enabled, and obtain an average benchmark score of 39.25. Despite requiring approximately 60 hours of training, DAPO with dynamic sampling still achieves lower accuracy than the CCE-regularized model without dynamic sampling reported in the main text. This further demonstrates the performance advantage of CCE. For Qwen3-4B, training DAPO with

Table 8: Benchmark scores of Qwen2.5-7B trained on the ProofWriter training set using different algorithms. AVG denotes the average score across all benchmarks, **bold** indicates the best performance, and "Base" refers to the base model. Policy optimization with CCE regularization achieves the highest overall average score.

Method	ProofWriter	RuleTaker	Tomi-NLI	RobustLR	AVG
Base	66.00	42.90	**21.80**	45.50	44.05
DAPO	85.40	47.70	16.90	66.30	54.07
Clip-Cov	86.00	46.50	19.40	66.20	54.52
Entropy	85.00	47.10	17.70	64.30	53.52
DivPO	86.80	**48.50**	16.80	63.50	53.90
CCE	**87.00**	47.10	18.90	**67.70**	**55.18**

dynamic sampling is estimated to take more than 10 days, and we therefore leave this experiment to future work.

D.5 Logical Reasoning Task

In the main text, we validate the effectiveness of CCE on mathematical reasoning tasks. Here, we further evaluate it on logical reasoning tasks to demonstrate the generality of CCE. We train Qwen2.5-7B on the ProofWriter (Tafjord et al., 2021) training set, using the same training configuration except that the maximum response length is reduced to 6K. The trained models are evaluated on four benchmarks: ProofWriter (Tafjord et al., 2021), RuleTaker (Clark et al., 2020), Tomi-NLI (Le et al., 2019), and RobustLR (Sanyal et al., 2022). For each benchmark, we randomly sample 1,000 examples from the validation set for evaluation. As shown in Table 8, CCE regularization substantially improves the overall score of the trained model. Moreover, compared with DAPO and other regularization methods, CCE exhibits a smaller performance degradation on Tomi-NLI, demonstrating its robustness.

D.6 Case Study

Using the first problem in AIME25 as a case study, we illustrate the training effects of different algorithms:

Prompt

```
Find the sum of all integer bases $b>9$ for which $17_b$ is a
    divisor of $97_b.$ Let's think step by step and output the final
     answer within \boxed{}.
```

Qwen2.5-7B

```
To solve this problem, we need to follow these steps:

1. Convert the numbers $17_b$ and $97_b$ from base $b$ to base 10.
2. Check if $17_b$ (converted to base 10) is a divisor of $97_b$ (
    converted to base 10).
3. Sum all the bases $b$ that satisfy the condition.

Let's start by converting $17_b$ and $97_b$ to base 10. The number
    $17_b$ in base 10 is $1 \cdot b + 7$, and the number $97_b$ in
    base 10 is $9 \cdot b + 7$. We need to find all bases $b > 9$
    such that $1 \cdot b + 7$ divides $9 \cdot b + 7$.
```

````
Let's denote $d = 1 \cdot b + 7$. Then we need $d$ to divide $9 \
    cdot b + 7$. This means that there exists an integer $k$ such
    that $9 \cdot b + 7 = k \cdot d$. Substituting $d = b + 7$, we
    get $9 \cdot b + 7 = k \cdot (b + 7)$. Rearranging terms, we get
     $9 \cdot b + 7 = k \cdot b + 7 \cdot k$, which simplifies to $9
     \cdot b - k \cdot b = 7 \cdot k - 7$, or $(9 - k) \cdot b = 7 \
    cdot (k - 1)$. This equation must hold for some integer $k$ and
    for all $b > 9$.

We can iterate over possible values of $k$ and check if $b = \frac
    {7 \cdot (k - 1)}{9 - k}$ is an integer and greater than 9. Let'
    s write the code to find all such bases $b$ and sum them up.
```python
# Initialize the sum of valid bases
sum_of_bases = 0

# Iterate over possible values of k
for k in range(2, 10):  # k must be at least 2 and less than 9
    if (7 * (k - 1)) % (9 - k) == 0:
        b = (7 * (k - 1)) // (9 - k)
        if b > 9:
            sum_of_bases += b

print(sum_of_bases)
```
```output
0
```
It seems that there are no valid bases $b > 9$ that satisfy the
    condition. Let's verify the logic and the calculations step by
    step to ensure there are no mistakes.

We need to find $b$ such that $b + 7$ divides $9b + 7$. This means
    that there exists an integer $k$ such that $9b + 7 = k(b + 7)$.
    Rearranging terms, we get $9b + 7 = kb + 7k$, which simplifies
    to $(9 - k)b = 7(k - 1)$. This equation must hold for some
    integer $k$ and for all $b > 9$.

Let's recheck the possible values of $k$ and the corresponding
    values of $b$:

1. For $k = 2$, we get $b = \frac{7 \cdot (2 - 1)}{9 - 2} = \frac
    {7}{7} = 1$, which is not greater than 9.
2. For $k = 3$, we get $b = \frac{7 \cdot (3 - 1)}{9 - 3} = \frac
    {14}{6} = \frac{7}{3}$, which is not an integer.
3. For $k = 4$, we get $b = \frac{7 \cdot (4 - 1)}{9 - 4} = \frac
    {21}{5}$, which is not an integer.
4. For $k = 5$, we get $b = \frac{7 \cdot (5 - 1)}{9 - 5} = \frac
    {28}{4} = 7$, which is not greater than 9.
5. For $k = 6$, we get $b = \frac{7 \cdot (6 - 1)}{9 - 6} = \frac
    {35}{3}$, which is not an integer.
6. For $k = 7$, we get $b = \frac{7 \cdot (7 - 1)}{9 - 7} = \frac
    {42}{2} = 21$, which is greater than 9.
7. For $k = 8$, we get $b = \frac{7 \cdot (8 - 1)}{9 - 8} = \frac
    {49}{1} = 49$, which is greater than 9.

So the valid bases are $b = 21$ and $b = 49$. The sum of these
    bases is $21 + 49 = 70$.

Let's update the code to reflect this and verify the result.
```python
# Initialize the sum of valid bases
sum_of_bases = 0
````
````

````
# Iterate over possible values of k
for k in range(2, 10):  # k must be at least 2 and less than 9
    if (7 * (k - 1)) % (9 - k) == 0:
        b = (7 * (k - 1)) // (9 - k)
        if b > 9:
            sum_of_bases += b

print(sum_of_bases)
```
```output
0
```
It seems there is still an issue with the code. Let's manually
    verify the calculations again and ensure that we are correctly
    identifying the valid bases.

We need to find $b$ such that $b + 7$ divides $9b + 7$. This means
    that there exists an integer $k$ such that $9b + 7 = k(b + 7)$.
    Rearranging terms, we get $9b + 7 = kb + 7k$, which simplifies
    to $(9 - k)b = 7(k - 1)$. This equation must hold for some
    integer $k$ and for all $b > 9$.

Let's recheck the possible values of $k$ and the corresponding
    values of $b$:

1. For $k = 2$, we get $b = \frac{7 \cdot (2 - 1)}{9 - 2} = \frac
    {7}{7} = 1$, which is not greater than 9.
2. For $k = 3$, we get $b = \frac{7 \cdot (3 - 1)}{9 - 3} = \frac
    {14}{6} = \frac{7}{3}$, which is not an integer.
3. For $k = 4$, we get $b = \frac{7 \cdot (4 - 1)}{9 - 4} = \frac
    {21}{5}$, which is not an integer.
4. For $k = 5$, we get $b = \frac{7 \cdot (5 - 1)}{9 - 5} = \frac
    {28}{4} = 7$, which is not greater than 9.
5. For $k = 6$, we get $b = \frac{7 \cdot (6 - 1)}{9 - 6} = \frac
    {35}{3}$, which is not an integer.
6. For $k = 7$, we get $b = \frac{7 \cdot (7 - 1)}{9 - 7} = \frac
    {42}{2} = 21$, which is greater than 9.
7. For $k = 8$, we get $b = \frac{7 \cdot (8 - 1)}{9 - 8} = \frac
    {49}{1} = 49$, which is greater than 9.

So the valid bases are $b = 21$ and $b = 49$. The sum of these
    bases is $21 + 49 = 70$.

Let's manually verify the calculations and ensure that the code is
    correctly identifying the valid bases. The correct sum of the
    valid bases is indeed $70$.

The final answer is \(\boxed{70}\).
````

DAPO w/ Clip-Cov

````
To solve the problem, we need to find the sum of all integer bases
    \( b > 9 \) for which \( 17_b \) is a divisor of \( 97_b \).

First, let's convert the numbers \( 17_b \) and \( 97_b \) from
    base \( b \) to base 10.

For \( 17_b \):
\[
````
````

```
17_b = 1 \cdot b + 7 = b + 7
\]

For \( 97_b \):
\[
97_b = 9 \cdot b + 7 = 9b + 7
\]

We need \( 17_b \) to be a divisor of \( 97_b \). This means:
\[
9b + 7 \equiv 0 \pmod{b + 7}
\]

Rewriting the congruence:
\[
9b + 7 = k(b + 7) \quad \text{for some integer } k
\]

Since \( 9b + 7 \) must be divisible by \( b + 7 \), let's perform
    polynomial division or use the Euclidean algorithm to simplify
    this condition. We start by expressing \( 9b + 7 \) in terms of
    \( b + 7 \):

\[
9b + 7 = 9(b + 7) - 56
\]

This simplifies to:
\[
9b + 7 \equiv -56 \pmod{b + 7}
\]

Thus, for \( 9b + 7 \) to be divisible by \( b + 7 \):
\[
-56 \equiv 0 \pmod{b + 7}
\]

This implies:
\[
b + 7 \text{ must be a divisor of } 56
\]

Next, we list the divisors of 56:
\[
\pm 1, \pm 2, \pm 4, \pm 7, \pm 8, \pm 14, \pm 28, \pm 56
\]

Since \( b > 9 \), we need \( b + 7 > 16 \). Therefore, we only
    consider the positive divisors greater than 16:
\[
28, 56
\]

Now, solve for \( b \):
\[
b + 7 = 28 \implies b = 21
\]
\[
b + 7 = 56 \implies b = 49
\]

Thus, the possible values for \( b \) are 21 and 49. We need to
    find the sum of these bases:
```

````
\[
21 + 49 = 70
\]

Therefore, the sum of all integer bases \( b > 9 \) for which \( 17
    _b \) is a divisor of \( 97_b \) is:
\[
\boxed{70}
\]
```
````

DAPO w/ Entropy Regularization

````
```text
To solve the problem, we start by converting the numbers $17_b$ and
     $97_b$ from base $b$ to base 10.

The number $17_b$ in base 10 is:
\[17_b = 1 \cdot b + 7 = b + 7.\]

The number $97_b$ in base 10 is:
\[97_b = 9 \cdot b + 7 = 9b + 7.\]

We need to find the integer bases $b > 9$ for which $17_b$ divides
    $97_b$. This means that $b + 7$ must divide $9b + 7$. Therefore,
     we need:
\[9b + 7 \equiv 0 \pmod{b+7}.\]
This simplifies to:
\[9b + 7 \equiv 9(b + 7) - 56 \equiv -56 \pmod{b+7}.\]
So, we need:
\[b + 7 \mid 56.\]
The divisors of 56 are: $1, 2, 4, 7, 8, 14, 28, 56$. Since $b > 9$,
     we discard the divisors that are less than or equal to 9. The
    possible values for $b+7$ are $14, 28, 56$. Therefore, the
    possible values for $b$ are:
\[b+7 = 14 \implies b = 7 \quad (\text{not valid since } b > 9),\]
\[b+7 = 28 \implies b = 21,\]
\[b+7 = 56 \implies b = 49.\]
So, the valid bases are $b = 21$ and $b = 49$. The sum of these
    bases is:
\[21 + 49 = 70.\]
Thus, the sum of all integer bases $b > 9$ for which $17_b$ is a
    divisor of $97_b$ is \(\boxed{70}\).
```
````

DAPO w/ CCE Regularization

````
To solve the problem of finding the sum of all integer bases \( b >
     9 \) for which \( 17_b \) is a divisor of \( 97_b \), let's
    proceed step-by-step:

1. **Convert the numbers \( 17_b \) and \( 97_b \) from base \( b
    \) to base 10**:

   - The number \( 17_b \) in base \( b \) can be expressed in base
     10 as:
     \[
     1 \cdot b + 7 = b + 7
````
````

```
     \]
   - The number \( 97_b \) in base \( b \) can be expressed in base
     10 as:
     \[
     9 \cdot b + 7 = 9b + 7
     \]

2. **Condition for divisibility**:

   We need \( 17_b \) to be a divisor of \( 97_b \). This
    translates to the condition that \( b + 7 \) must divide \( 9b +
     7 \). Mathematically, this means:
   \[
   9b + 7 \equiv 0 \pmod{b + 7}
   \]
   This implies:
   \[
   9b + 7 \equiv 0 \pmod{b + 7}
   \]
   To simplify this, let's perform the division of \( 9b + 7 \) by
    \( b + 7 \):

3. **Perform polynomial division**:

   We perform the division \( 9b + 7 \div (b + 7) \):

   - First, divide the leading term \( 9b \) by \( b \), which
    gives \( 9 \). So, multiply \( b + 7 \) by \( 9 \):
     \[
     9 \cdot (b + 7) = 9b + 63
     \]
   - Now, subtract \( 9b + 63 \) from \( 9b + 7 \):
     \[
     (9b + 7) - (9b + 63) = 9b + 7 - 9b - 63 = 7 - 63 = -56
     \]
   Hence, we have:
   \[
   9b + 7 = (b + 7) \cdot 9 - 56
   \]
   This implies:
   \[
   9b + 7 \equiv -56 \pmod{b + 7}
   \]
   Since \( 9b + 7 \equiv 0 \pmod{b + 7} \), it follows that:
   \[
   0 \equiv -56 \pmod{b + 7}
   \]
   Which simplifies to:
   \[
   b + 7 \mid 56
   \]
   This means \( b + 7 \) must be a divisor of \( 56 \).

4. **List the divisors of 56**:

   The divisors of \( 56 \) are \( \pm 1, \pm 2, \pm 4, \pm 7, \pm
    8, \pm 14, \pm 28, \pm 56 \). Since \( b \) is a base greater
    than \( 9 \), \( b \) must be positive and \( b > 9 \). Hence,
    \( b + 7 \) must be a positive divisor of \( 56 \) that is
    greater than \( 16 \) (since \( b > 9 \) implies \( b + 7 > 16
    \)). The relevant positive divisors of \( 56 \) that are greater
     than \( 16 \) are \( 28 \) and \( 56 \).
```

```
5. **Solve for \( b \)**:

   - If \( b + 7 = 28 \), then:
     \[
     b = 28 - 7 = 21
     \]
   - If \( b + 7 = 56 \), then:
     \[
     b = 56 - 7 = 49
     \]

6. **Verify the solutions**:

   We need to verify that \( b = 21 \) and \( b = 49 \) satisfy the
     original condition that \( 17_b \) divides \( 97_b \).

   - For \( b = 21 \):
     \[
     17_{21} = 1 \cdot 21 + 7 = 28
     \]
     \[
     97_{21} = 9 \cdot 21 + 7 = 189 + 7 = 196
     \]
     Check divisibility:
     \[
     196 \div 28 = 7
     \]
     Since \( 196 \) is divisible by \( 28 \), \( b = 21 \) is
    valid.

   - For \( b = 49 \):
     \[
     17_{49} = 1 \cdot 49 + 7 = 56
     \]
     \[
     97_{49} = 9 \cdot 49 + 7 = 441 + 7 = 448
     \]
     Check divisibility:
     \[
     448 \div 56 = 8
     \]
     Since \( 448 \) is divisible by \( 56 \), \( b = 49 \) is
    valid.

7. **Sum the valid bases**:

   The valid bases are \( b = 21 \) and \( b = 49 \). The sum of
    these bases is:
   \[
   21 + 49 = 70
   \]

8. **Final Answer**:

   The sum of all integer bases \( b > 9 \) for which \( 17_b \) is
     a divisor of \( 97_b \) is \(\boxed{70}\).
```

DivPO

```
    To solve the problem, we start by converting the numbers \(17_b
    \) and \(97_b\) from base \(b\) to base 10.

The number \(17_b\) in base 10 is:
\[
17_b = b + 7.
\]

The number \(97_b\) in base 10 is:
\[
97_b = 9b + 7.
\]

We need to find the integer bases \(b > 9\) for which \(17_b\)
    divides \(97_b\). This means that \(b + 7\) must divide \(9b +
    7\). Therefore:
\[
9b + 7 \equiv 0 \pmod{b+7}.
\]

This simplifies to:
\[
9b + 7 = k(b+7)
\]
for some integer \(k\).

Rewriting the congruence, we get:
\[
9b + 7 \equiv 0 \pmod{b+7}
\implies 9b + 7 = m(b+7)
\]
for some integer \(m\).

So,
\[
9b + 7 = mb + 7m
\]
\[
9b - mb = 7m - 7
\]
\[
b(9-m) = 7(m-1).
\]

Thus,
\[
b = \frac{7(m-1)}{9-m}.
\]

Since \(b\) is an integer, \(9-m\) must be a divisor of \(7(m-1)\).
     Let's list the divisors of 7, which are \(\pm 1, \pm 7\).

Checking these cases gives only:
\[
b = 49.
\]

To confirm:
\[
17_{49} = 56,\quad 97_{49}=448.
\]
```

```
Since \(448 \div 56 = 8\), it is divisible.

Therefore:
\[
\boxed{49}
\]
```

Although the base model (Qwen2.5-7B) can occasionally solve the problem correctly, its responses contain substantial mixtures of code and natural-language reasoning, with the code being only virtually executed and thus constituting ineffective reasoning. Clip-Cov produces more structured responses, but they still exhibit a strong code-oriented style and remain similar to the policy of the base model. Entropy regularization encourages the model to generate short, code-free reasoning that is rarely observed in the base model, but the resulting structure is disorganized and does not represent a highly generalizable strategy. DivPO, in contrast, exhibits severe logical repetition, suggesting that critical transition tokens may be disrupted. In contrast, CCE regularization leads to clean and standard solution procedures, suggesting that it enables the model to explore more generalizable reasoning strategies that are difficult for the base model to discover.