CLIPO: CONTRASTIVE LEARNING IN POLICY OPTIMIZATION GENERALIZES RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has significantly advanced the reasoning capacity of Large Language Models (LLMs). However, RLVR solely relies on final answers as outcome rewards, neglecting the correctness of intermediate reasoning steps. Training on these process-wrong but outcome-correct rollouts can lead to hallucination and answer-copying, severely undermining the model's generalization and robustness. To address this, we incorporate a **C**ontrastive **L**earning mechanism **i**nto the **P**olicy **O**ptimization (CLIPO) to generalize the RLVR process. By optimizing a contrastive loss over successful rollouts, CLIPO steers the LLM to capture the invariant structure shared across correct reasoning paths. This provides a more robust cross-trajectory regularization than the original single-path supervision in RLVR, effectively mitigating reasoning inconsistencies and suppressing hallucinatory artifacts. In experiments, CLIPO consistently improves multiple RLVR baselines across diverse reasoning benchmarks, demonstrating improvements in generalization and robustness.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has recently emerged as a mainstream paradigm for improving the reasoning capabilities of Large Language Models (LLMs) (Jaech et al., 2024; Guo et al., 2025; Team et al., 2025). Group-based policy optimization methods such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) estimate relative advantages within a set of sampled responses, and have shown strong performance on mathematics, coding, and agentic tasks (Yu et al., 2026; Zheng et al., 2025a; Dai et al., 2026; Dong et al., 2026). However, RLVR faces ongoing doubts about its generalization capabilities (Alam & Rastogi, 2025; Chen et al., 2026), for it relies on outcome-based rewards without verifying intermediate reasoning steps, in which models are prone to overfitting by memorizing ground-truth answers (Wu et al., 2026; Ruan et al., 2025; Yan et al., 2026). Such a coarse and binary rewarding scheme fails to distinguish between logically sound and spurious reasoning, as it collapses diverse reasoning paths sharing the same outcome into identical reward signals. Existing remedies either rely on process-level supervision, which requires costly human step annotations, an extra learned reward model, or additional inference passes (Lightman et al., 2023; Cui et al., 2025a; Zhang et al., 2026), or on token-level entropy signals, which reflect distributional uncertainty rather than the logical validity of the reasoning process (Chen et al., 2025b; Cui et al., 2025b).

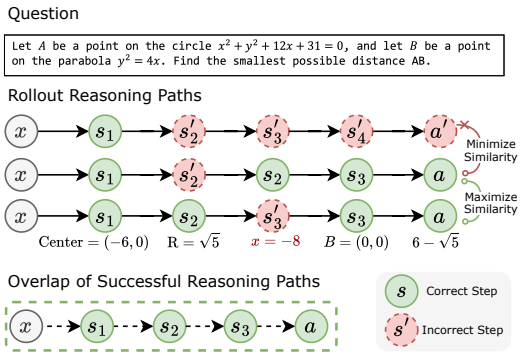


Figure 1: Intuition of CLIPO. Standard RLVR method only relies on outcomes, neglecting the quality of intermediate reasoning steps. CLIPO addresses this limitation by maximizing similarity between successful reasoning trajectories. By aligning multiple positive rollouts, CLIPO identifies the invariant reasoning structure, *i.e.*, the "overlap" of successful paths, implicitly suppressing sporadic incorrect and hallucinative reasoning patterns.

On the other hand, contrastive learning (Chen et al., 2020; He et al., 2020; Radford et al., 2021) has been widely recognized as an unsupervised scheme with strong generalization performance (Saunshi

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.

et al., 2019; Krishna et al., 2022; Castricato et al., 2022), in which models learn informative representations by maximizing similarity between positives and minimizing it between negatives. Since RLVR methods intrinsically utilize binary outcome rewards and target at enlarging the reward gap between positive (successful) and negative (failed) rollouts, an interesting question naturally arises: ***"Can contrastive learning enhance the generalization of RLVR?"*** By designating successful and failed trajectories as positive and negative instances, contrastive learning can be seamlessly integrated into RLVR. Our insight, illustrated in Figure 1, is that successful reasoning paths share a consistent underlying logic, whereas intermediate errors/hallucinations manifest as sporadic, uncorrelated noise. By enforcing proximity among successful trajectories in the embedding space, contrastive learning acts as a denoising mechanism: it amplifies the invariant logical flow while suppressing non-systematic reasoning failures and hallucinations, ultimately leading to more robust generalization.

Based on this analysis, we propose **C**ontrastive **L**earning **i**n **P**olicy **O**ptimization (CLIPO), which contrasts correct and incorrect trajectories within each rollout group and converts this signal into an auxiliary reward for group-based policy optimization. Our contributions are: (1) we introduce contrastive learning as a lightweight, annotation-free regularizer for RLVR, compatible with GRPO, GSPO, DAPO and GMPO; (2) across two tracks and 14 benchmarks, CLIPO consistently improves these baselines, with statistically significant gains under distribution shift; (3) a step-level evaluation shows that CLIPO reduces flawed-but-correct reasoning from 31.7% to 25.2%.

2 Preliminary

2.1 Reinforcement Learning with Verifiable Rewards

Unlike traditional reinforcement learning from human feedback (RLHF) relying on a learned reward model, RLVR (Jaech et al., 2024; Guo et al., 2025; Team et al., 2025) directly leverages deterministic environment feedback (e.g., unit tests for coding, or equivalence checkers for mathematics) to provide binary reward outcomes. Formally, given a prompt $\boldsymbol{x} \in \mathcal{X}$, the policy model $\pi_\theta(\cdot|\boldsymbol{x})$ generates a response $\boldsymbol{y} \in \mathcal{Y}$. The objective of RLVR is to maximize the expected verifiable reward:

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X},\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\big[r(\boldsymbol{x},\boldsymbol{y})\big] - \beta\mathrm{KL}[\pi_\theta\|\pi_{\text{ref}}], \tag{1}$$

where π_{ref} is a reference model and usually set as the initial checkpoint of π_θ, $\mathrm{KL}[\pi_\theta\|\pi_{\text{ref}}]$ is the KL divergence between the policy and reference model to maintain the training stability, and $\beta > 0$ is the coefficient for the KL penalty. Without loss of generality, we simplify the objective by omitting the KL penalty term in the following sections. The input $\boldsymbol{x}$ is associated with a ground-truth answer $\boldsymbol{a}^*$, and the verifiable reward $r(\boldsymbol{x},\boldsymbol{y})$ directly check if $\boldsymbol{a}^*$ and the predicted answer $\boldsymbol{a} = \mathcal{A}(\boldsymbol{y})$ is equivalent: $r(\boldsymbol{x},\boldsymbol{y}) = \mathbf{1}\{\mathcal{A}(\boldsymbol{y}) = \boldsymbol{a}^*\}$, where $\mathbf{1}\{\cdot\}$ is a binary indicator, and $\mathcal{A}(\cdot)$ extracts the predicted final answer from the response $\boldsymbol{y}$ (*e.g.*, the content enclosed in `\boxed{}`).

To optimize the above RLVR objective, policy gradient methods, such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024), dominate as the main solution. Generally, policy optimization methods estimate the gradient of equation 1 with the REINFORCE estimator (Williams, 1992): $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X},\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\Big[\sum_{t=1}^{|\boldsymbol{y}|}\nabla_\theta\log(\pi_\theta(y^t|\boldsymbol{y}^{<t},\boldsymbol{x})\cdot\hat{A}^t\Big]$, where $\boldsymbol{y}^{<t} = (y_1, y_2, \ldots, y_{t-1})$ is the length-t prefix of the token sequence $\boldsymbol{y}$, and $\hat{A}^t$ is estimated advantage of the t-th token y^t based on the outcome reward $r(\boldsymbol{x},\boldsymbol{y})$. PPO deploys a critic model to predict each token's value and calculate the advantage $\hat{A}^t$ with Generalized Advantage Estimation (GAE) (Schulman et al., 2015). GRPO eliminates the need for the additional critic model by estimating the baseline within a group of sampled responses $\mathcal{G} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_G\} \sim \pi_\theta(\cdot|\boldsymbol{x})$ and their corresponding rewards $\{r_i = r(\boldsymbol{x},\boldsymbol{y}_i)\}_{i=1}^G$. GRPO assumes all tokens within one rollout have the same advantage, which is computed by normalization over the rollout group:$\hat{A}_i^t = \frac{r_i - \text{Mean}(r_1,r_2,\ldots,r_G)}{\text{Std}(r_1,r_2,\ldots,r_G)}$. Therefore, the overall policy gradient for GRPO is:

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\Big[\frac{1}{G}\sum_{i=1}^{G}\Big[\sum_{t=1}^{|\boldsymbol{y}_i|}\nabla_\theta\log(\pi_\theta(y_i^t|\boldsymbol{y}_i^{<t},\boldsymbol{x})\cdot\hat{A}_i^t\Big]\Big]. \tag{2}$$

Recent work has improved the stability and efficiency of RLVR training (Zheng et al., 2025a; Yu et al., 2026; Zhao et al., 2026), but existing methods still rely on sparse binary feedback from verifiers, which lacks the resolution to distinguish the quality of intermediate reasoning steps.

2.2 CONTRASTIVE LEARNING

Contrastive learning (CL) has emerged as a powerful paradigm for self-supervised representation learning, aiming to map data points into a latent space where semantically similar samples are clustered together while dissimilar ones are pushed apart (Oord et al., 2018; Frosst et al., 2019; Khosla et al., 2020; Chen et al., 2020). A cornerstone of CL methods is the InfoNCE objective (Oord et al., 2018): Formally, given a group of pairwise instances $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_i^N$, CL seeks to optimize the embedding model as a representation function $f(\cdot)$ such that it minimizes the distance between each $\boldsymbol{x}_i$ and its positive counterpart $\boldsymbol{y}_i$, while simultaneously maximizing $\boldsymbol{x}_i$'s distance from a set of negative samples $\{\boldsymbol{y}_j\}_{j\neq i}$. A widely adopted approach to realize this objective is the InfoNCE loss (Oord et al., 2018), which frames the representation learning task as a multi-class categorical classification problem:

$$\mathcal{L}_{\text{InfoNCE}} := -\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\Big[\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(f(\boldsymbol{x}_i,\boldsymbol{y}_i))}{\sum_{j=1}^{N}\exp(f(\boldsymbol{x}_i,\boldsymbol{y}_j))}\Big], \tag{3}$$

i.e., a cross-entropy task identifying the positive sample among noise contrastive samples.

Theoretically, the negative InfoNCE loss is equivalent to a lower-bound estimator of the mutual information (MI) $\mathrm{I}(\boldsymbol{x};\boldsymbol{y})$ between the two variables $(\boldsymbol{x},\boldsymbol{y})$ with samples $\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^N$ (Poole et al., 2019):

$$\mathrm{I}(\boldsymbol{x};\boldsymbol{y}) := \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\Big[\log\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}\Big] \geq \log(N) - \mathcal{L}_{\text{InfoNCE}}, \tag{4}$$

where $p(\boldsymbol{x},\boldsymbol{y})$ is the joint distribution of $(\boldsymbol{x},\boldsymbol{y})$, and $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are the marginal distribution of $\boldsymbol{x}$ and $\boldsymbol{y}$. Hence, by minimizing the $\mathcal{L}_{\text{InfoNCE}}$, one can enlarge the information overlap between the two learnable features or embeddings, thereby distilling the underlying semantic invariants. Specifically, the loss encourages the representations of positive pairs to be close in the embedding space while pushing negative pairs apart, effectively capturing the shared features across similar instances.

3 METHODOLOGY

As illustrated in Figure 2, CLIPO augments group-based policy optimization with a contrastive reward: a lightweight contrastive head maps each rollout to a trajectory-level embedding, a contrastive loss is computed within each rollout group with correct responses as positives and incorrect ones as negatives, and this loss becomes an auxiliary reward added to the verifiable reward.

3.1 CLIPO OBJECTIVE

Suppose we have a binary reward function $r : \mathcal{X}\times\mathcal{Y}\to\{0,1\}$ where $r(\boldsymbol{x},\boldsymbol{y}) = 1$ if the model response $\boldsymbol{y}$ is correct, and $r(\boldsymbol{x},\boldsymbol{y}) = 0$ if the model response is incorrect. The mainstream reinforcement learning with verifiable rewards (RLVR) aims to optimize the policy optimization objective in equation 1. As discussed in Figure 1, we expect the policy model $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$ to identify the commonalities among the correct rollout samples during the RLVR process, *i.e.*, to maximize the mutual information (as in equation 4) among the positive rollouts:

$$\max_\theta \mathrm{I}\Big(\boldsymbol{y};\bar{\boldsymbol{y}}\Big|r(\boldsymbol{x},\boldsymbol{y})=1, r(\boldsymbol{x},\bar{\boldsymbol{y}})=1\Big), \quad \boldsymbol{x}\sim\mathcal{X},\ \boldsymbol{y},\bar{\boldsymbol{y}}\sim\pi_\theta(\cdot|\boldsymbol{x}). \tag{5}$$

Note that the mutual information calculation is conditioned on the event $\mathcal{P}_{\boldsymbol{y},\bar{\boldsymbol{y}}} := \{r(\boldsymbol{x},\boldsymbol{y}) = 1, r(\boldsymbol{x},\bar{\boldsymbol{y}}) = 1\}$, where $\boldsymbol{y}$ and $\bar{\boldsymbol{y}}$ correctly obtain positive rewards, with the prompt $\boldsymbol{x}$ marginalized. By maximizing equation 5, the policy model π_θ is required to seek the commonalities among positive rollouts, obtaining more robust experiences of successes for policy optimization. Therefore, the overall learning objective is:

$$\max_\theta \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X},\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\big[r(\boldsymbol{x},\boldsymbol{y})\big] + \lambda\cdot\mathrm{I}\big(\boldsymbol{y};\bar{\boldsymbol{y}}\big|\mathcal{P}_{\boldsymbol{y},\bar{\boldsymbol{y}}}\big). \tag{6}$$

By maximizing the mutual information among positive rollouts, we enforce semantic proximity among successful reasoning paths. This regularization provides a dense, informative training signal that guides the policy toward logically coherent reasoning, effectively compensating for the limitations of the sparse, binary outcome-based verifiers commonly used in standard RLVR.

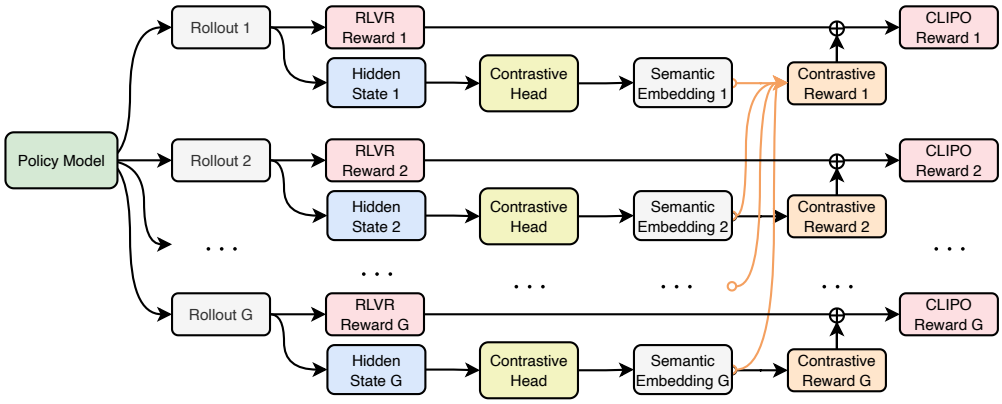


Figure 2: Framework of CLIPO. For each input prompt $\boldsymbol{x}$, policy optimization methods generate a group of rollouts $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_G\}$ then calculate corresponding RLVR rewards $\{r_1, r_2, \ldots, r_G\}$. CLIPO applies a contrastive head on the top of the last hidden states $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_G\}$ of the rollout group and outputs trajectory-level semantic embeddings $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_G\}$. The contrastive rewards $\{r_1^{\text{CL}}, r_2^{\text{CL}}, \ldots, r_G^{\text{CL}}\}$ are computed across the semantic embedding group to provide the similarity of successful and failed trajectories. The final CLIPO reward for the i-th rollout is $r_i' = r_i + r_i^{\text{CL}}$.

Practically, the exact MI value $\mathrm{I}\big(\boldsymbol{y}; \bar{\boldsymbol{y}} \big| \mathcal{P}_{\boldsymbol{y},\bar{\boldsymbol{y}}}\big)$ is challenging to compute, for the joint distribution $p(\boldsymbol{y}, \bar{\boldsymbol{y}} | \mathcal{P}_{\boldsymbol{y},\bar{\boldsymbol{y}}})$ is intractable (Cheng et al., 2020). Fortunately, in each policy optimization step, we obtain a group of rollouts $\mathcal{G} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_G\} \sim \pi_\theta(\cdot|\boldsymbol{x})$, where we can use a sample-based MI estimator to approximate the ground-truth value of $\mathrm{I}\big(\boldsymbol{y}; \bar{\boldsymbol{y}} \big| \mathcal{P}_{\boldsymbol{y},\bar{\boldsymbol{y}}}\big)$. More specifically, we select the InfoNCE (Oord et al., 2018; Khosla et al., 2020) as a per-prompt surrogate for MI maximization, averaged over $\boldsymbol{x}$:

$$\log(G-1) + \frac{1}{|\mathcal{P}|} \sum_{\boldsymbol{y}, \bar{\boldsymbol{y}} \in \mathcal{P}} \log \left(\frac{\exp(f(\boldsymbol{y}, \bar{\boldsymbol{y}}))}{\sum_{\boldsymbol{y}_i \in \mathcal{G} \setminus \{\boldsymbol{y}\}} \exp(f(\boldsymbol{y}, \boldsymbol{y}_i))} \right) = \log(G-1) - \mathcal{L}_{\text{CL}}(\boldsymbol{x}), \tag{7}$$

where $\mathcal{P} = \{\boldsymbol{y}_i \in \mathcal{G} : r(\boldsymbol{x}, \boldsymbol{y}_i) = 1\}$ denotes the rollout subset with successful outcome. For each positive rollout $\boldsymbol{y}$, we uniformly sample another positive rollout $\bar{\boldsymbol{y}} \in \mathcal{P}$ to make up the *positive* pair $(\boldsymbol{y}, \bar{\boldsymbol{y}})$, and use the remaining rollouts in group as the *negatives* $\{(\boldsymbol{y}, \boldsymbol{y}_i) \mid \boldsymbol{y}_i \in \mathcal{G}, \boldsymbol{y}_i \neq \boldsymbol{y}, \boldsymbol{y}_i \neq \bar{\boldsymbol{y}}\}$. The score function $f(\boldsymbol{y}, \bar{\boldsymbol{y}})$ measures the semantic similarity between $\boldsymbol{y}$ and $\bar{\boldsymbol{y}}$, which we implement as the inner product of embeddings, following the design from SimCLR (Chen et al., 2020):

$$f(\boldsymbol{y}, \bar{\boldsymbol{y}}) = g_\phi(\boldsymbol{h}_\theta(\boldsymbol{y}))^{\mathrm{T}} g_\phi(\boldsymbol{h}_\theta(\bar{\boldsymbol{y}}))/\tau, \tag{8}$$

where $\boldsymbol{h}_\theta(\boldsymbol{y})$ is the last hidden state output by the policy model $\pi_\theta(\cdot|\boldsymbol{x})$'s transformer backbone, and $g_\phi(\cdot)$ is a learnable contrastive head to extract core semantic information from the last hidden state of the transformer. $\tau > 0$ is the temperature hyper-parameter. The contrastive loss in equation 7 is inapplicable when $|\mathcal{P}| \leq 1$ (no positive pair) or $|\mathcal{P}| = G$ (no negative rollouts), so we apply it only to groups satisfying $1 < |\mathcal{P}| < G$, while all other groups receive no contrastive reward.

3.2 CLIPO Implementation

Sentence-Level Representation. For each response $\boldsymbol{y}$, the policy model $\pi_\theta(\cdot|\boldsymbol{x})$ produces token-level hidden states $\bar{\boldsymbol{h}}_\theta(\boldsymbol{y}) \in \mathbb{R}^{T \times D}$, where T is the length of the response sequence and D is the dimension of hidden states. To derive a sentence-level representation from the token-level hidden states, we apply a mean pooling operation following the findings of ENNADIR et al. (2026): $\boldsymbol{h}_\theta(\boldsymbol{y}) = \frac{1}{T} \sum_{t=1}^{T} \bar{\boldsymbol{h}}_\theta(\boldsymbol{y})_t$, obtaining $\boldsymbol{h}_\theta(\boldsymbol{y}) \in \mathbb{R}^D$ that summarizes the whole response. This pooled representation is passed through the contrastive head $g_\phi : \mathbb{R}^D \to \mathbb{R}^d$ to obtain the semantic embedding: $\boldsymbol{e}(\boldsymbol{y}) = g_\phi(\boldsymbol{h}_\theta(\boldsymbol{y}))$, where $\boldsymbol{e}(\boldsymbol{y}) \in \mathbb{R}^d$ is used for similarity calculation in equation 8.

Contrastive Head Optimization. The contrastive head g_ϕ typically is a simple linear layer $W \in \mathbb{R}^{d \times D}$, designed to project the pooled representation into a latent space suitable for contrastive

learning (Chen et al., 2020). Then the contrastive loss for each correct response $\boldsymbol{y}_i$ in the group $\mathcal{G}$ is:

$$\mathcal{L}_{\text{CL}}(\boldsymbol{x}, \boldsymbol{y}_i) = -\log\Big(\frac{\exp(f(\boldsymbol{y}_i, \bar{\boldsymbol{y}}_i))}{\sum_{j=1, j\neq i}^{G} \exp(f(\boldsymbol{y}_i, \boldsymbol{y}_j))}\Big), \tag{9}$$

where $\bar{\boldsymbol{y}}_i$ is a uniformly sampled positive sample from the set of other correct responses in the group $\mathcal{G}$. The contrastive head is updated jointly with the policy model during training, and its parameters ϕ are optimized to minimize the average contrastive loss across all valid anchors in the batch. The anchor with no positive pair in the group is considered invalid and excluded from loss computation.

Contrastive Reward Integration. The total reward r'_i for a given response y_i is formulated as the sum of the original verifiable reward r_i and the contrastive reward r_i^{CL}:

$$r'_i = r_i + r_i^{\text{CL}}, \tag{10}$$

To prevent the auxiliary signal from dominating the verifiable objective, we scale it by the re-weighting parameter $\lambda > 0$ in equation 6 and clip it with a lower threshold:

$$r_i^{\text{CL}} = \max(-\lambda \cdot \mathcal{L}_{\text{CL}}(\boldsymbol{x}, \boldsymbol{y}_i), -0.5), \tag{11}$$

where $\mathcal{L}_{\text{CL}}(\boldsymbol{x}, \boldsymbol{y}_i)$ is the contrastive loss in equation 9. Incorrect rollouts receive $r_i^{\text{CL}} = 0$. By incorporating r_i^{CL}, we move beyond the limitations of sparse, binary feedback of RLVR. This mechanism provides granular guidance even among multiple correct responses by prioritizing those that converge within the "consensus" region of the solution space. Consequently, the model is compelled to refine its reasoning strategies toward more consistent and robust reasoning trajectories, rather than merely satisfying the pass/fail threshold of the binary outcome verifier.

4 Experiments

4.1 Experimental Setup

Datasets. To rigorously evaluate the effectiveness of our proposed method, we construct two separate experimental tracks using training sets of varying difficulty levels.

- **Track I: Grade-School and General Reasoning.** In the first setup, we fine-tune our model on the GSM8K dataset (Cobbe et al., 2021). We evaluate it on the GSM8K test set, as well as GSM8K-Symbolic, GSM8K-P1, and GSM8K-P2 (Mirzadeh et al., 2025). These variants introduce increasing complexity and distribution shifts to assess generalization. Furthermore, we assess its broader reasoning and general knowledge capabilities across diverse benchmarks such as CommonsenseQA (Talmor et al., 2019), TruthfulQA (Lin et al., 2022), TheoremQA (Chen et al., 2023), and the MMLU (Hendrycks et al., 2021a) suite. We use SYM, P1, P2, cQA, tQA, thrmQA to denote GSM8K-Symbolic, GSM8K-P1, GSM8K-P2, three QA datasets respectively.
- **Track II: Competition-Level Mathematical Reasoning.** To further investigate the impact of task difficulty on model performance, we conduct an additional set of experiments by training on the MATH 7.5k dataset (Hendrycks et al., 2021b). The evaluation datasets include MATH500 (Hendrycks et al., 2021b), Math-Perturb (Huang et al., 2025) (covering simple and hard variants), which further test robustness to problem perturbations. Meanwhile, we also assess performance on competition-level challenges such as AMC23, AIME and AIME25, which feature problems of significantly higher complexity. We use Math, Math_S, Math_H to denote the MATH500, Math-Perturb Simple, and Math-Perturb Hard datasets, respectively.

Models and Baselines. For experiments on Track I, we utilize Qwen2.5-3B-Instruct as the base model. For the more challenging MATH7.5k dataset, we evaluate our method across Qwen2.5-7B-Instruct, Llama3.1-8B-Instruct, and reasoning DeepSeek-R1-Distill-Qwen-7B. We integrate and compare CLIPO with several state-of-the-art group-based RLVR baselines, including GRPO (Shao et al., 2024), GSPO (Zheng et al., 2025a), DAPO (Yu et al., 2026) and GMPO (Zhao et al., 2026). We report Pass@1 as our primary evaluation metric on all benchmarks in both tracks.

Contrastive Head Designs. The contrastive head is a linear layer (Chen et al., 2020; Khosla et al., 2020) producing L_2-normalized embeddings of dimension $d = 512$ (Track I) or 2048 (Track II), with the contrastive reward weight $\lambda = 0.2$ and temperature $\tau = 0.02$ by default. All methods are implemented with VERL (Sheng et al., 2024) using 16 rollouts per prompt and a maximum response length of 2048 (Track I) or 4096 (Track II) tokens, with all other settings shared across baselines for fair comparison. Full hyperparameters for both tracks are listed in Appendix C.1.

Table 1: Performance comparison of different RLVR methods on Track I (GSM8K and General Reasoning). 'Avg.' represents the average performance across all datasets, and 'M-Avg.' represents the average performance on math-related datasets. 'G-Avg.' represents the average performance on general and QA tasks. Small numbers denote the gain over the Base Model; light green marks the averaged scores of CLIPO-enhanced methods. The best results are highlighted in **bold**, and the second-best results are underlined.

Methods	GSM8K	SYM	P1	P2	cQA	tQA	thrmQA	MMLU	M-Avg.	G-Avg.	Avg.
Base Model	86.05	82.40	70.90	47.96	73.63	29.99	24.90	68.09	71.83	49.15	60.49
GRPO	87.79	84.04	73.12	50.80	74.94	30.48	**27.71**	68.19	73.94 +2.11	50.33 +1.18	62.13 +1.64
GSPO	87.19	83.94	73.28	51.36	74.86	30.23	27.38	67.68	73.94 +2.11	50.04 +0.89	61.99 +1.50
DAPO	87.41	84.56	74.70	52.32	76.00	31.58	27.11	**69.00**	74.75 +2.92	50.92 +1.77	62.84 +2.35
GMPO	86.81	84.02	73.90	49.80	76.33	30.97	26.57	68.49	73.63 +1.80	50.59 +1.44	62.11 +1.62
Methods enhanced with our approach											
GRPO+CLIPO	**88.02**	84.62	74.60	**54.16**	76.90	31.52	27.58	68.64	**75.35** +3.52	**51.16** +2.01	**63.26** +2.77
GSPO+CLIPO	87.95	83.98	74.54	52.92	75.92	**32.19**	26.10	67.48	74.85 +3.02	50.42 +1.27	62.63 +2.14
DAPO+CLIPO	87.57	84.66	74.54	53.12	**77.07**	31.40	27.44	67.93	74.97 +3.14	50.96 +1.81	62.97 +2.48
GMPO+CLIPO	87.64	**85.52**	**75.52**	50.92	75.35	31.95	27.58	67.63	74.90 +3.07	50.63 +1.48	62.76 +2.27

Table 2: Performance on Track II (Competition-Level Reasoning). 'M-Avg.' averages Math, Math_S, and Math_H; 'C-Avg.' represents the average performance on AMC, AIME, and AIME25. Small numbers denote the gain over the Base Model.

Methods	Math	Math_S	Math_H	AMC	AIME	AIME25	M-Avg.	C-Avg.	Avg.
Base Model	75.35	59.24	35.31	44.13	10.63	6.25	56.63	20.33	38.48
GRPO	76.46	64.83	37.86	46.84	17.92	9.58	59.72 +3.09	24.78 +4.45	42.25 +3.77
GSPO	77.58	64.56	38.36	50.08	16.25	10.42	60.17 +3.54	25.58 +5.25	42.87 +4.39
DAPO	76.03	66.08	**39.94**	47.36	19.79	7.92	60.68 +4.05	25.02 +4.69	42.85 +4.37
GMPO	**78.47**	64.27	37.25	48.42	19.58	9.58	60.00 +3.37	25.86 +5.53	42.93 +4.45
Methods enhanced with our approach									
GRPO+CLIPO	77.49	**67.21**	38.97	49.17	18.13	10.63	61.22 +4.59	25.97 +5.64	43.60 +5.12
GSPO+CLIPO	77.16	66.69	37.66	**50.53**	**20.63**	9.38	60.50 +3.87	**26.84** +6.51	43.67 +5.19
DAPO+CLIPO	77.43	66.55	39.82	48.42	18.75	**13.33**	61.27 +4.64	26.83 +6.50	**44.05** +5.57
GMPO+CLIPO	77.87	67.14	38.83	47.67	19.38	11.67	**61.28** +4.65	26.24 +5.91	43.76 +5.28

4.2 Main Results

Tables 1 and 2 compare RLVR baselines with and without CLIPO (denoted 'method+CLIPO').

Results on GSM8K and General Reasoning. As shown in Table 1, GRPO+CLIPO achieves the best overall average (63.26) and M-Avg. (75.35), with notable gains on the perturbed variants GSM8K-P1 (+1.48) and GSM8K-P2 (+3.36), and CLIPO variants rank best or second-best on all four general reasoning benchmarks. The contrastive reward thus particularly benefits robustness under distribution shift and compositional reasoning, while preserving general capabilities.

Results on Competition-Level Reasoning. As shown in Table 2, CLIPO improves the average score of all four RLVR methods (+1.35, +0.80, +1.20 and +0.83 for GRPO, GSPO, DAPO and GMPO), with DAPO+CLIPO achieving the best average of 44.05 and GMPO+CLIPO ranking second with 43.76. Across both tracks, the gains are consistent and concentrate on perturbed and competition-level benchmarks. This supports our hypothesis that clustering successful trajectories while pushing away failed ones provides a more informative learning signal than binary outcome feedback alone, and that this signal is largely orthogonal to the underlying RLVR algorithm.

Statistical Significance. Figure 3(a) reports paired 95% confidence intervals of GRPO+CLIPO over GRPO across three training seeds. The improvement is positive on all 13 evaluated benchmarks (sign test $p = 2.4 \times 10^{-4}$) and significant on GSM8K-P2 (+3.98), Math_S (+2.73) and AMC (+3.63) but not in-distribution (GSM8K, Math), consistent with CLIPO mainly improving robustness.

Process-Level Analysis. To test whether CLIPO improves the reasoning process rather than only final answers, a method-blind step-level judge (DeepSeek-V4-Flash-0423) following Process-Bench (Zheng et al., 2025b) locates the earliest erroneous step in 10,573 outcome-correct Track II rollouts from 2,686 prompt-matched pairs (Appendix C.8). As shown in Figure 3(b), CLIPO reduces the flawed-but-correct rate from 31.7% to 25.2% ($\Delta = -6.52$ pp, 95% CI $[-8.21, -4.80]$), consistently across three training checkpoints. The reduction is smallest in-distribution (Math, -5.2 pp) and largest under distribution shift (Math_H -8.4, AIME25 -9.1, AMC -11.1 pp), mirroring

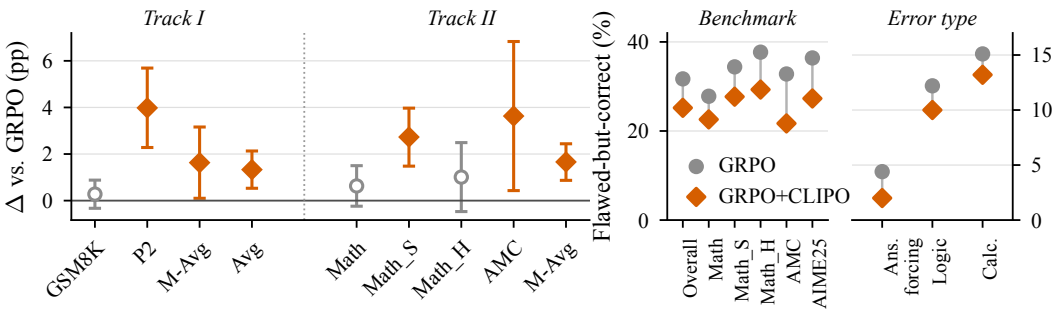


Figure 3: CLIPO's gains are significant under distribution shift, and it reduces flawed-but-correct reasoning. (a) Improvement of GRPO+CLIPO over GRPO with paired 95% confidence intervals over three training seeds; filled diamonds mark improvements whose interval excludes zero, open circles those that do not. (b) Rate of outcome-correct rollouts on Track II that contain at least one erroneous step, as judged by a method-blind step-level judge (lower is better), per benchmark; (c) the same rate broken down by error type. Exact numbers are listed in Tables 21 and 22 in the appendix.

the accuracy gains. Among error types (Figure 3(c)), answer forcing, i.e., asserting the known answer without deriving it, shows the largest relative reduction (4.4% $\rightarrow$ 2.0%), together with fewer logic and calculation errors. The effect is not explained by surface features: controlling for length, formatting and math density in a logistic regression still yields -5.50 pp ($p = 1.1 \times 10^{-10}$).

Solution Diversity. Pulling successful trajectories together does not collapse solution diversity: on Track II, Pass@16 is retained (59.95% $\rightarrow$ 60.53%) while Pass@1 improves by $+1.35$, raising the share of solvable problems solved on the first attempt on 5 of 6 benchmarks. Diversity collapse would show the opposite signature, with Pass@16 falling as Pass@1 rises (Appendix C.9).

4.3 Ablation Studies and Analysis

Contrastive Head Analysis. CLIPO introduces a contrastive head that maps the LLM's hidden states into a new embedding space, thereby learning a novel geometric structure. To validate the effectiveness of the contrastive head, we conduct an ablation where the contrastive head is fixed and not updated during training (denoted as CLIPO-fixed). As shown in Table 3, freezing the head lowers all average scores in both tracks, with the largest drops on distribution-shifted benchmarks. These results highlight the importance of jointly optimizing the contrastive head with the base model. By allowing the head to adapt during training, the model can learn more effective representations that better capture the nuances of high-quality versus low-quality solutions, thereby enhancing the overall learning process. Note that CLIPO-fixed is not a fully clean control for the learned geometry, since reward clipping partially reduces it to GRPO; the complete results, this caveat, and a visualization of the embeddings are presented in Appendix C.2.

Table 3: Comparison of CLIPO with and without fixed head on Track I and II. Metric 1/2 denote M-Avg. and G-Avg. for Track I, and M-Avg. and C-Avg. for Track II. Here, Δ denotes the average performance drop of the fixed-head variant.

Track	Methods	Metric 1	Metric 2	Avg.
Track I	CLIPO	75.35	51.16	63.26
	CLIPO-fixed	74.73	50.24	62.48
	Δ	-0.62	-0.92	-0.77
Track II	CLIPO	61.22	25.97	43.60
	CLIPO-fixed	60.19	25.06	42.63
	Δ	-1.03	-0.91	-0.97

Contrastive Loss Variants. Except for the InfoNCE loss (Oord et al., 2018; Chen et al., 2020), we also experiment with two contrastive loss functions, SupCon loss (Khosla et al., 2020) and SoftNN loss (Salakhutdinov & Hinton, 2007; Frosst et al., 2019). The main differences among these loss functions are summarized in Table 4. As shown in Figure 4, all three contrastive loss variants yield notable performance improvements over the base GRPO method on Track I. These results suggest that while the specific choice of contrastive loss can influence performance, the overall benefit of incorporating contrastive rewards is robust across different loss formulations.

Group Size Analysis. The effectiveness of CLIPO relies on group-based policy optimization, where the group size (i.e., the number of rollouts) can significantly influence contrastive learning outcomes. A larger number of rollouts results in more non-trivial groups, providing denser reward signals. Additionally, more rollouts offer a richer and more diverse set of positive and negative

Table 4: Comparison among contrastive loss variants. Here, s_{ia} denotes the similarity between i and a, while P is the set of positive examples for i and p^* is a single sampled positive example.

Loss	Formulation	Positive Aggregation
InfoNCE	$-\log \frac{e^{s_{ip^*}}}{\sum_a e^{s_{ia}}}$	Single Positive
SupCon	$-\frac{1}{\lvert P\rvert}\sum_p \log \frac{e^{s_{ip}}}{\sum_a e^{s_{ia}}}$	Mean Positives
SoftNN	$-\log \frac{\sum_p e^{s_{ip}}}{\sum_a e^{s_{ia}}}$	Sum Positives

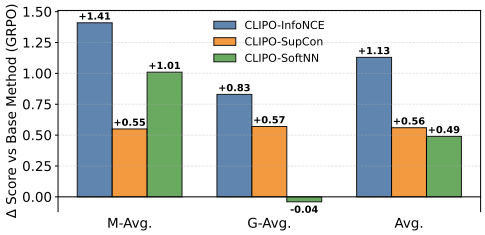


Figure 4: Gain Across Various Contrastive Losses.

Table 5: Effect of different group sizes on CLIPO performance. Larger group sizes lead to better performance improvements. The best results are highlighted in bold. It can be observed that larger group sizes generally lead to better performance improvements across Avg. metrics.

GroupSize	Math	Math_S	Math_H	AMC	AIME	AIME25	M-Avg.	C-Avg.	Avg.
$\lvert G\rvert = 8$	77.41	64.90	38.49	47.67	15.42	8.54	60.27	23.87	42.07
$\lvert G\rvert = 16$	77.49	67.21	38.97	49.17	18.13	10.63	**61.22**	25.97	43.60
$\lvert G\rvert = 32$	77.26	66.03	39.89	50.45	20.83	8.33	61.06	**26.54**	**43.80**

Table 6: Performance comparison of CLIPO with different pooling strategies. 'Mean', 'Max', and 'EOS' respectively denote Mean Pooling, Max Pooling, and EOS Pooling (ENNADIR et al., 2026).

Methods	Math	Math_S	Math_H	AMC23	AIME24	AIME25	M-Avg.	C-Avg.	Avg.
GRPO	76.46	64.83	37.86	46.84	17.92	9.58	59.72	24.78	42.25
CLIPO (Mean)	77.49	**67.21**	38.97	49.17	18.13	**10.63**	61.22 +1.50	**25.97** +1.19	**43.60** +1.35
CLIPO (Max)	**78.42**	66.94	**40.77**	47.15	**18.75**	8.96	**62.04** +2.32	24.95 +0.17	43.50 +1.25
CLIPO (EOS)	77.15	66.78	40.25	**49.55**	17.92	8.33	61.39 +1.67	25.27 +0.49	43.33 +1.08

samples for contrastive learning. We conduct experiments to evaluate the impact of varying the number of rollouts. The results are presented in Table 5 where we compare group sizes of 8, 16, and 32. Larger groups directly increase the fraction of groups where the contrastive objective is active (16.9%, 28.5% and 37.0% for $G = 4, 8, 16$; Appendix C.10). Since a larger G also improves the group-relative baseline itself, we do not attribute the gain at $G = 32$ to contrastive learning alone.

Pooling Strategies. In our implementation, we use the mean pooling to obtain sequence-level representations by averaging the hidden states of all tokens in the response. We further evaluate the impact of different pooling strategies, including mean pooling, max pooling, and last-token/EOS pooling. From Table 6, it is evident that mean pooling achieves the best performance, which is consistent with previous work (ENNADIR et al., 2026), suggesting that mean pooling can better capture the semantic information of the entire response. Meanwhile, all three pooling strategies yield performance improvements, indicating that the head can learn from different representation strategies to enhance the model's performance. This further demonstrates the robustness of CLIPO.

Base Model Variations. To further validate the generalizability of our proposed method across different architectures, we conduct additional experiments using various base models, including DeepSeek-R1-Distill-Qwen-7B (DS-7B), and Llama3.1-8B (Llama-8B). As presented in Table 7, we observe that both DS-7B and Llama-8B exhibit similar performance improvements upon integrating CLIPO. Specifically, DS-7B shows an increase of +0.53 in average score, while Llama-8B achieves a notable gain of +1.31. These results underscore the versatility and effectiveness of our approach in enhancing reasoning capabilities across diverse language model architectures. There is a slight decrease of -0.11 in M-Avg. for DS-7B. We speculate that DS-7B has already undergone fine-tuning on mathematical tasks.

Table 7: Performance on diverse models. CLIPO can enhance the performance of different family of models, demonstrating its generalizability.

Model	Method	M-Avg.	C-Avg.	Avg.
DS-7B	GRPO	79.39	45.38	62.38
	+CLIPO	79.28	46.54	62.91
	Δ	-0.11	1.17	0.53
Llama-8B	GRPO	34.84	11.42	23.13
	+CLIPO	37.10	11.79	24.44
	Δ	2.25	0.37	1.31

Table 8: Effect of different contrastive temperature τ. It can be observed that lower temperatures lead to better improvements; however, excessively low temperatures may result in degraded performance.

τ	M-Avg.	C-Avg.	Avg.
0.2	59.90 +0.18	24.36 -0.42	42.13 -0.12
0.1	59.76 +0.04	**26.81** +2.03	<u>43.28</u> +1.03
0.05	**61.28** +1.56	24.85 +0.07	43.07 +0.82
0.02	<u>61.22</u> +1.50	<u>25.97</u> +1.19	**43.60** +1.35
0.015	60.64 +0.92	25.19 +0.41	42.91 +0.66

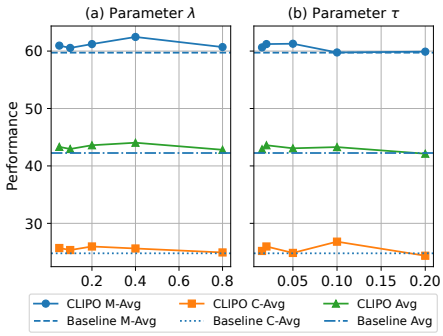


Figure 5: Sensitivity on parameter λ and τ.

Analysis on Parameter λ and τ. CLIPO introduces two additional hyperparameters: λ, which controls the contrastive reward's contribution; and temperature τ, which scales the similarity scores in the contrastive loss. At first, we conduct experiments to evaluate the sensitivity of these two parameters. We vary λ in the range of [0.05, 0.8] and τ in the range of [0.015, 0.2]. As shown in Figure 5, compared to the baseline (GRPO) method, CLIPO shows improvements across a wide range of parameter values, indicating a low sensitivity to λ and τ. Since temperature τ plays a more critical role in contrastive learning, we further analyze its impact on training effectiveness, presented in Table 8. From the table, it is evident that lower temperature values generally yield better performance improvements. This is attributed to the model's increased focus on hard negatives, which enhances its discriminative capabilities, aligning with findings from Wang & Isola (2020); Wang & Liu (2021). Meanwhile, higher temperature settings (e.g., 0.2) lead to performance degradation. Our findings suggest that lower temperatures are more effective for contrastive learning in our context. One possible explanation is that our setting involves a significant number of hard negatives, and a lower temperature enhances the model's discriminative power (Wang & Isola, 2020).

Overhead Analysis. We evaluate the computational overhead introduced by CLIPO from the perspectives of parameters, memory, computation, and wall-clock runtime. The additional contrastive head is lightweight, containing only 7.34M parameters on Qwen2.5-7B compared with the 7B backbone. In terms of memory, the baseline already stores rollouts with complexity $O(BTD)$, while CLIPO only adds an extra $O(BD)$ memory cost for contrastive representations, corresponding to approximately 0.024% additional memory when $T = 4096$. We further measure the practical runtime overhead, shown in Table 9. On Qwen2.5-7B training, the baseline GRPO requires 84.83s ($\pm$18.1s) per step, while CLIPO introduces only 1.03s ($\pm$0.1s) additional runtime, demonstrating that CLIPO incurs negligible overhead in practice.

Table 9: Runtime of each training step for GRPO and the additional CLIPO overhead.

GRPO Runtime	CLIPO Extra Runtime
84.83s ($\pm$18.1s)	1.03s ($\pm$0.1s)

5 Conclusions

We introduced CLIPO, a contrastive learning augmented framework that generalizes RLVR beyond coarse, outcome-based supervision. Specifically, we employ a lightweight contrastive head with an InfoNCE objective to align the representations of successful reasoning trajectories while enforcing separation from erroneous ones. This design effectively distills the latent semantic consistency among successful rollouts into a dense contrastive reward, which complements standard verifier feedback. By reshaping the reward landscape from sparse and binary to informative and relational, CLIPO facilitates more stable policy optimization without requiring external process annotations or additional supervision. Extensive experiments across diverse benchmarks validate the efficacy of our approach: notably, consistent gains on symbolic, perturbed, and out-of-distribution tasks demonstrate that contrastive reward shaping fosters more robust and generalizable reasoning. Ultimately, this work highlights a promising paradigm for advancing RLVR, that leveraging the inherent relational structure among successful reasoning solutions as a principled and scalable learning signal. Beyond mathematical reasoning, CLIPO is broadly applicable to other structured domains like code generation and agent planning, paving the way for more reliable and generalized reasoning intelligence.

AI USE STATEMENT

In this work, we used generative AI tools for translation. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims or proofs, propose or refine hypotheses, design research methodology or experiments, implement methods, clean or reformat datasets, perform qualitative or thematic data analysis, or interpret results. Additionally, we used generative AI tools to polish the writing of the paper to improve readability. We have reviewed all AI-assisted text and verified that it faithfully reflects our methods and results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we release our code, training data, and experimental results, including all diagnostics reported in this paper, in an anonymous repository: https://anonymous.4open.science/r/CLIPO. The CLIPO objective and its implementation are described in Section 3. All training and evaluation datasets are publicly available (Section 4.1). Implementation details of all baselines, the full training and contrastive-head hyperparameters (Tables 10 and 11), the prompt templates (Table 12), and the verifiable reward are provided in Appendix C.1. The multi-seed evaluation and the protocol of the step-level process evaluation are detailed in Appendices C.7 and C.8.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.04697.

Md Tanvirul Alam and Nidhi Rastogi. Limits of generalization in RLVR: Two case studies in mathematical reasoning. In *The 5th Workshop on Mathematical Reasoning and AI at NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=EwHz7cUHBE.

Louis Castricato, Alexander Havrilla, Shahbuland Matiana, Michael Pieler, Anbang Ye, Ian Yang, Spencer Frazier, and Mark Riedl. Robust preference learning for storytelling via contrastive reinforcement learning. *arXiv preprint arXiv:2210.07792*, 2022.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive llm agents. *arXiv preprint arXiv:2502.01600*, 2025a.

Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025b.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=sE8DCSJTzd.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. In *EMNLP*, pp. 7889–7901, 2023. URL https://doi.org/10.18653/v1/2023.emnlp-main.489.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pp. 1779–1788. PMLR, 2020.

Pengyu Cheng, Yifan Yang, Jian Li, Yong Dai, Tianhao Hu, Peixin Cao, Nan Du, and Xiaolong Li. Adversarial preference optimization: Enhancing your alignment via rm-llm game. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3705–3716, 2024.

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

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025a.

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*, 2025b.

Silong Dai, Haolun Wu, Huanran Zheng, Tao Ji, Yuanbin Wu, Junchi Yan, Dell Zhang, Xiaoling Wang, Xuelong Li, and Changzhi Sun. Group verification-based policy optimization for interactive coding agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RY47Tq0VsV.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=TX4k7BF6aO.

Sofiane ENNADIR, Levente Zólyomi, Oleg Smirnov, Tianze Wang, John Pertoft, Filip Cornell, and Lele Cao. Pool me wisely: On the effect of pooling in transformer-based models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=8uhXfdSJmA.

Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International conference on machine learning*, pp. 2012–2020. PMLR, 2019.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025a.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous RL. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025b. URL https://openreview.net/forum?id=JSYCLJcniH.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021b. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. MATH-perturb: Benchmarking LLMs' math reasoning abilities against hard perturbations. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=OZy70UggXr.

Siyuan Huang, Xiaoye Qu, Yafu Li, Yun Luo, Zefeng He, Daizong Liu, and Yu Cheng. Spotlight on token perception for multimodal reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=bRA4lVWJVQ.

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

Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*, 2025.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. Rankgen: Improving text generation with large ranking models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 199–232, 2022.

Zhuo Li, Pengyu Cheng, Zhechao Yu, Feifei Tong, Anningzhe Gao, Tsung-Hui Chang, Xiang Wan, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Eliminating inductive bias in reward models with information-theoretic guidance. In *International Conference on Learning Representations*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Hongliang Lu, Yuhang Wen, Pengyu Cheng, Ruijin Ding, Jiaqi Guo, Haotian Xu, Chutian Wang, Haonan Chen, Xiaoxi Jiang, and Guanjun Jiang. Search self-play: Pushing the frontier of agent capability without supervision. In *International Conference on Learning Representations*, 2026.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AjXkRZIvjB.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

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.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International conference on machine learning*, pp. 5171–5180. PMLR, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Zhiwen Ruan, Yun Chen, Yutao Hou, Peng Li, Yang Liu, and Guanhua Chen. Unveiling over-memorization in finetuning llms for reasoning tasks. *arXiv preprint arXiv:2508.04117*, 2025.

Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In Marina Meila and Xiaotong Shen (eds.), *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pp. 412–419, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. URL https://proceedings.mlr.press/v2/salakhutdinov07a.html.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *International conference on machine learning*, pp. 5628–5637. PMLR, 2019.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

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. *arXiv preprint arXiv: 2409.19256*, 2024.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Hongze Tan, Jianfei Pan, Jinghao Lin, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*, 2025.

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

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2021.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei Huang, and Feng Zhao. VRAG-RL: Empower vision-perception-based RAG for visually rich information understanding via iterative reasoning with reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=EeAHhNwXPV.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=yfcpdY4gMP.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pp. 9929–9939. PMLR, 2020.

Yining Wang, Jinman Zhao, Chuangxin Zhao, Shuhao Guan, Gerald Penn, and Shinan Liu. λ-grpo: Unifying the grpo frameworks with learnable token preferences. *arXiv preprint arXiv:2510.06870*, 2025b.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jGbRWwIidy.

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

Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Huijie Lv, Ming Zhang, et al. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33944–33952, 2026.

Lecheng Yan, Ruizhe Li, Guanhua Chen, Qing Li, Jiahui Geng, Wenxi Li, Vincent Wang, and Chris Lee. Spurious rewards paradox: Mechanistically understanding how rlvr activates memorization shortcuts in llms. *arXiv preprint arXiv:2601.11061*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=2a36EMSSTp.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*, 2025a.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.547. URL https://aclanthology.org/2025.findings-acl.547/.

Zizhuo Zhang, Jianing Zhu, Xinmu Ge, Zihua Zhao, Zhanke Zhou, Xuan Li, Xiao Feng, Jiangchao Yao, and Bo Han. Co-rewarding: Stable self-supervised RL for eliciting reasoning in large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fDk95XPsCU.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. Geometric-mean policy optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nCEs0tSwc2.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, 2025b.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431, 2025c.

Sining Zhoubian, Dan Zhang, and Jie Tang. Rest-rl: Achieving accurate code reasoning of llms with optimized self-training and decoding. *arXiv preprint arXiv:2508.19576*, 2025.

A Related Work

Reinforcement learning for LLM reasoning. Unlike reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Cheng et al., 2024; Li et al., 2026), which incurs substantial human annotation costs and suffers from inconsistent human judgment, RLVR leverages external environments (*e.g.*, mathematical verifiers or code compilers) to provide objective, explicit, and consistent feedback (Jaech et al., 2024; Guo et al., 2025; Team et al., 2025). Group-based policy optimization methods such as GRPO (Shao et al., 2024) have demonstrated strong empirical performance across a wide range of reasoning tasks, including mathematics (Yu et al., 2026; Zheng et al., 2025a; Zhao et al., 2026; Gao et al., 2025a), coding (Zhoubian et al., 2025; Dai et al., 2026), deep search (Jin et al., 2025; Lu et al., 2026; Wang et al., 2026; Sun et al., 2025; Zheng et al., 2025c) and agents (Chen et al., 2025a; Gao et al., 2025b; Dong et al., 2026).

Process-level supervision. Process reward models (PRMs) provide finer-grained supervision by identifying errors during the reasoning process (Lightman et al., 2023; Wang et al., 2024; Luo et al., 2024; Zheng et al., 2025b; Zhang et al., 2025b), but require substantial human annotation to collect high-quality process reward data, making them costly and difficult to scale (Zhang et al., 2025b). Annotation-free alternatives either learn implicit process rewards with an additional reward model (Cui et al., 2025a) or construct self-supervised rewards from agreement across multiple inference passes (Zhang et al., 2026), and outcome-only RLVR has itself been shown to implicitly improve reasoning correctness (Wen et al., 2026). In contrast, CLIPO extracts a trajectory-level signal within the existing rollout group, without an additional reward model, step annotations, or extra inference.

Entropy-based guidance. Another line of work leverages token-level entropy to provide fine-grained training guidance and enhance exploration (Chen et al., 2025b; Zhang et al., 2025a; Tan et al., 2025; Jiang et al., 2025; Wang et al., 2025b; Huang et al., 2026). Such signals primarily reflect distributional uncertainty rather than semantic logical importance, and their effectiveness is highly dependent on model capacity (Cui et al., 2025b; Wang et al., 2025a).

B Limitations

While CLIPO demonstrates significant improvements in reasoning tasks, several limitations remain to be addressed in future research. The current framework employs simple linear or MLP heads. Investigating more complex architectures could potentially enhance the expressiveness of the contrastive representations and further improve performance. Moreover, CLIPO operates at the trajectory level: it biases group-relative advantages toward the structure shared by successful rollouts, but does not localize individual erroneous steps. The contrastive objective is only active in groups with $1 < |\mathcal{P}| < G$, about one third of groups in our setting (Appendix C.10), and contributes nothing to all-correct or all-incorrect groups. Within active groups the number of negatives is small (median 6), so the InfoNCE bound is loose. This reflects a trade-off: contrastive learning favors large negative

sets, but enlarging them with a memory bank of past rollouts would make the negatives off-policy and stale with respect to both the policy and the contrastive head, requiring additional machinery such as momentum encoders or importance weighting. CLIPO deliberately stays on-policy within the rollout group, and we leave the memory-bank design to future work. Additionally, our evaluation is primarily focused on mathematical reasoning. Extending CLIPO to other domains, such as code generation, multi-modal reasoning, and real-world decision-making tasks, would be valuable to assess its generality and robustness across diverse applications.

C Appendix

C.1 Implementation Details

C.1.1 Baseline Implementations

We reproduce GRPO, GSPO, DAPO, GMPO, using the public verl framework (Sheng et al., 2024). All baselines share identical data, backbone initialization, sampling strategy, batch configuration, optimization strategy, and distributed training settings; the only differences arise from their respective policy objectives and advantage aggregation schemes. We summarize the key distinctions below.

Table 10: Default Training Hyperparameters and Configuration on Track I and Track II.

Category	Parameter	Track I	Track II
General Settings			
Total epochs	total_epochs	6	9
Critic warmup epochs	critic_warmup	14 step(1 epoch)	58 step(1 epoch)
Data and Sequence Lengths			
Training batch size	train_batch_size	512	128
PPO mini-batch size	ppo_mini_batch_size	256	128
Micro batch size / GPU	micro_batch_size	32	16
Max prompt length	max_prompt_length	512	512
Max response length	max_response_length	2048	4096
Prompt overlength handling	filter_overlong_prompts	True	True
Truncation policy	truncation	error	error
Actor and PPO Optimization			
Actor learning rate	actor.optim.lr	1×10^{-6}	1×10^{-6}
Dynamic batch size	use_dynamic_bsz	True	True
Gradient checkpointing	enable_gradient_checkpointing	True	True
Rollout and Inference			
Rollout backend	rollout.name	sglang	sglang
Rollout samples (n)	rollout.n	16	16
Validation rollouts	val_rollout	1	16
Sampling during eval	do_sample	False	True
Temperature (eval)	temperature	0.6	0.6
Top-p (eval)	top_p	0.95	0.95
Distributed Training			
GPUs per node	n_gpus_per_node	8	8
Number of nodes	nnodes	1	1
Tensor model parallel size	tensor_model_parallel_size	1	1
FSDP parameter offload (actor)	fsdp.param_offload	False	False
FSDP optimizer offload (actor)	fsdp.optimizer_offload	False	False
FSDP parameter offload (ref)	ref.fsdp.param_offload	True	False

GRPO. GRPO follows a PPO-style clipped surrogate objective while incorporating an explicit KL penalty as a separate loss term. In our implementation, reward shaping does not include KL divergence, whereas an auxiliary KL loss with a fixed coefficient is enabled:

$$\mathcal{L}_{\text{GRPO}} = \mathbb{E}[\text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)A_t] - \beta\,\text{KL}(\pi \,\|\, \pi_{\text{ref}}).$$

The corresponding configuration setsuse_kl_loss=True, and kl_loss_coef=0.001. Advantage aggregation is performed via token averaging, specified by loss_agg_mode=token-mean, with a symmetric clipping factor of $\epsilon = 0.2$.

GSPO. GSPO replaces token-wise advantages with group-wise preference signals and adopts a very narrow clipping interval, as recommended by the original paper (Zheng et al., 2025a). The loss employs a sequence-level aggregation. In our setup, we configure loss_mode=gspo, loss_agg_mode=seq-mean-token-mean, and use clip_ratio_low $= 3 \times 10^{-4}$ and clip_ratio_high $= 4 \times 10^{-4}$, aligning with the suggested hyperparameters.

DAPO. DAPO maintains a PPO-like update but introduces an asymmetric clipping interval:

$$\mathcal{L}_{\text{DAPO}} = \mathbb{E}[\text{clip}(\rho_t, 1 - \epsilon_l, 1 + \epsilon_h)A_t]$$

We set $\epsilon_l = 0.2$ and $\epsilon_h = 0.28$, following the original recommendations (Yu et al., 2026). Both KL reward shaping and KL loss regularization are disabled (use_kl_in_reward=False and use_kl_loss=False).

GMPO. GMPO adopts a geometric-mean policy objective in place of additive advantage accumulation, which promotes multiplicative update dynamics and mitigates domination by outlier advantages. GMPO uses a single symmetric clipping factor with $\epsilon = 0.4$, enforced by setting clip_ratio_low=clip_ratio_high=0.4, and applies token-level advantage aggregation via loss_agg_mode=token-mean.

Shared Training Protocol. All baselines are trained under a unified protocol, sharing identical hyperparameters for learning rate, rollout sampling configuration (temperature, top-p, number of rollouts), batch size, sequence length, and distributed memory optimization settings. This ensures that performance differences can be attributed solely to the distinct policy objectives and advantage aggregation methods employed by each algorithm. The complete set of shared training hyperparameters is detailed in Table 10.

Contrastive Head Parameters We detail the architecture and hyperparameters of the Contrastive Head in Table 11. The input dimension D of the head matches the hidden size of the respective base model, and the output embeddings are L_2-normalized. For Track II, we increase the output size from 512 to 2048 to provide the higher representational capacity required for competition-level reasoning. The head is optimized jointly with the policy using AdamW with a learning rate of 1×10^{-3} and a weight decay of 0.01. The global batch size is 512 for Track I and 128 for Track II (Table 10).

Table 11: Default Contrastive Head Parameters on Track I and Track II.

Parameter	Track I	Track II
Contrastive LM Head Settings		
Loss type	infonce_loss	infonce_loss
Distance measure	cosine_distance	cosine_distance
Head type	linear	linear
Output size	512	2048
Contrastive temperature	0.02	0.02
Learning rate	0.001	0.001
Weight decay	0.01	0.01
λ	0.2	0.2
Negative Selection	group_level	group_level

C.1.2 Prompt Construction

For all datasets, each training and evaluation example is converted into a unified instruction-following format. Given an original problem statement q from dataset d, we construct the user prompt as

$$\texttt{Prompt}(q, d) = q \parallel \texttt{Instruction}(d), \tag{12}$$

where $\parallel$ denotes string concatenation with a newline separator.

The dataset-specific instruction `Instruction`(d) is appended to explicitly encourage step-by-step reasoning and constrain the answer format. All prompts are assigned the `user` role and no system or assistant messages are used.

Instruction Templates. For most mathematical reasoning datasets, including GSM8K, MATH, AMC, AIME, and their variants, we use the following instruction:

```
Let's think step by step and output the final answer
within \boxed{}.
```

For multiple-choice datasets, the instruction additionally specifies the expected answer type. For example, for MMLU:

```
Let's think step by step and output the final answer
(eg, A, B, C, D) within \boxed{}.
```

We summarize the complete set of dataset-specific prompt instructions in Table 12.

Table 12: Prompt Construction and Instruction Templates

Dataset	**Instruction Template**
GSM8K / GSM-P1 / GSM-P2 / GSM-Symbolic	Let's think step by step and output the final answer within \boxed{}.
MATH / AMC / AIME / AIME2025	Let's think step by step and output the final answer within \boxed{}.
TheoremQA	Let's think step by step and output the final answer within \boxed{}.
MMLU	Let's think step by step and output the final answer (eg, A, B, C, D) within \boxed{}.
TruthfulQA	Let's think step by step and output the final answer (eg, A, B, C, D, ...) within \boxed{}.
CommonsenseQA	Let's think step by step and output the final answer (eg, A, B, C, D, E) within \boxed{}.

C.1.3 Verifiable Reward Models

Following prior work (Shao et al., 2024; Aggarwal & Welleck, 2025), we employ rule-based verifiable reward models to compute rewards based on the final answers generated by the LLM. Specifically, we extract the final answer enclosed within the `\boxed{}` delimiters. This extracted answer, denoted as LLM_{pred}, is then compared against the ground truth answer GT to determine correctness. The reward r is 0 for incorrect answers and 1 for correct answers.

C.2 Contrastive Head Representation and Ablation

To gain insights into the representations learned by the contrastive head, we conduct a qualitative analysis of the semantic embeddings it produces. We randomly sample a set of responses generated during training and compute their embeddings using the contrastive head.

As illustrated in Figure 6, the embeddings exhibit no clear separation: correct and incorrect trajectories are mixed and lack coherent structure at the start of training. After contrastive training, the embeddings exhibit a clear clustering structure. Embeddings corresponding to correct rollouts tend to cluster closely together, while those from incorrect rollouts are relatively separated. This structure aligns with the goal of the contrastive reward, which encourages higher-quality solutions to be close in representation space while pushing away suboptimal ones. Importantly, this signal is more fine-grained than binary reward feedback, since embeddings reflect relative similarity of trajectories even when they share the same correctness outcome.

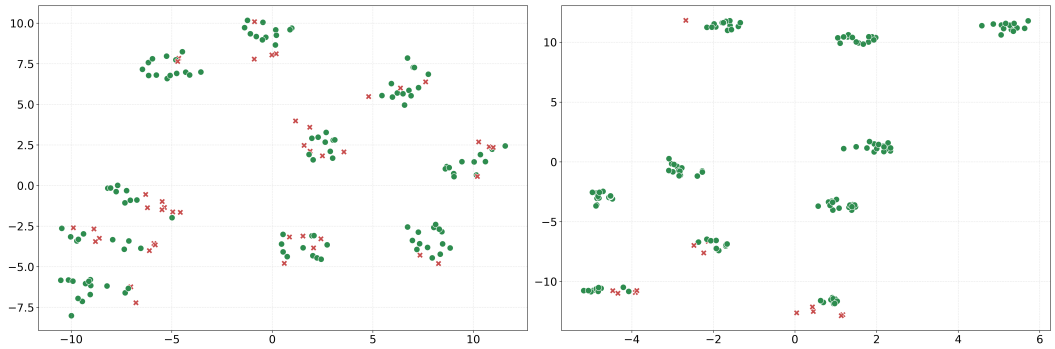


Figure 6: The t-SNE visualization of semantic embeddings produced by the contrastive head at the start of training (left) and after 3 epochs of training (right). Green points represent embeddings from correct rollouts, while red points correspond to incorrect ones. After training, correct responses cluster closely together, forming more distinct group clusters. Within these clusters, correct and incorrect responses also exhibit some separation.

We note that this visualization illustrates what the contrastive objective optimizes by construction, and the separation may also partly reflect factors such as prompt identity, answer string, length or difficulty. It should therefore be read as an illustration rather than as evidence of reasoning quality; direct evidence on the reasoning process is provided by the step-level evaluation in Section 4.2 and Appendix C.8.

To validate that the learned geometry is not solely inherited from the pretrained LLM backbone, we conduct an ablation where the contrastive head is fixed and not updated during training (denoted as CLIPO-fixed).

Table 13: Performance comparison of fixed-head ablation on Track I. CLIPO-fixed shows performance drops compared to CLIPO across most benchmarks.

Methods	GSM8K	SYM	P1	P2	cQA	tQA	thrmQA	MMLU	M-Avg.	G-Avg.	Avg.
CLIPO	88.02	84.62	74.60	54.16	76.90	31.52	27.58	68.64	75.35	51.16	63.26
CLIPO-fixed	88.63	85.12	73.48	51.68	75.59	30.11	26.77	68.49	74.73	50.24	62.48
Δ	0.61	0.50	-1.12	-2.48	-1.31	-1.41	-0.80	-0.15	-0.62	-0.92	-0.77

Table 14: Performance comparison of fixed-head ablation on Track II. CLIPO-fixed shows performance drops compared to CLIPO across most benchmarks.

Methods	Math	Math_S	Math_H	AMC	AIME	AIME25	M-Avg.	C-Avg.	Avg.
CLIPO	77.49	67.21	38.97	49.17	18.13	10.63	61.22	25.97	43.60
CLIPO-fixed	76.76	64.83	38.99	49.77	17.92	7.50	60.19	25.06	42.63
Δ	-0.73	-2.38	0.02	0.60	-0.21	-3.13	-1.03	-0.91	-0.97

As shown in Table 13, freezing the head leads to consistent performance drops across all three average metrics, particularly on distribution-shifted and perturbed datasets (P1, P2), with decreases of -1.12 and -2.48, respectively. Table 14 shows a similar trend, where fixing the head results in overall performance degradation, with decreases of -1.03, -0.91, and -0.97 on M-Avg, C-Avg, and Avg, respectively. These results suggest that the mapping learned by the contrastive head contributes to the gains.

However, CLIPO-fixed is not a fully clean control for the learned geometry. With $\lambda = 0.2$ and the lower clip at -0.5, r^{CL} saturates whenever $\mathcal{L}_{\mathrm{CL}} \geq 2.5$, which is close to the value $\log(G-1) \approx 2.71$ produced by near-uniform similarities of a randomly initialized head. Since r^{CL} is only assigned to correct rollouts, full saturation yields $r' = 0.5\,r$, a positive affine rescaling that group-relative normalization removes exactly, so the advantages become identical to those of GRPO. CLIPO-fixed therefore partially reduces to GRPO rather than isolating the effect of the learned geometry.

Consistently, it does not interpolate monotonically between GRPO and CLIPO: across the 14 per-benchmark results in Tables 13 and 14, it lies between the two in 7 cases, above both in 4, and below both in 3. The same clipping mechanism is consistent with the temperature results: $\tau = 0.2$, where clipping is most frequent, performs on par with GRPO (42.13 vs. 42.25 Avg.), whereas $\tau = 0.02$ gives the best result (43.60).

Table 15: Performance comparison of Contrastive Loss Variants on Track I. The improvements (Δ) over base method(GRPO) are indicated with positive (green) and negative (red) colors.

Methods	GSM8K	SYM	P1	P2	cQA	tQA	thrmQA	MMLU	M-Avg.	G-Avg.	Avg.
GRPO	87.79	84.04	73.12	50.80	74.94	30.48	27.71	68.19	73.94	50.33	62.13
CLIPO-InfoNCE	88.02	84.62	74.60	54.16	76.90	31.52	27.58	68.64	75.35	51.16	63.26
Δ	0.23	0.58	1.48	3.36	1.97	1.04	−0.13	0.46	1.41	0.83	1.12
CLIPO-SupCon	87.79	85.04	74.00	51.12	77.07	30.23	27.04	69.25	74.49	50.90	62.69
Δ	0.00	1.00	0.88	0.32	2.13	−0.24	−0.67	1.06	0.55	0.57	0.56
CLIPO-SoftNN	87.11	84.70	74.06	53.92	75.27	29.99	27.31	68.59	74.95	50.29	62.62
Δ	−0.68	0.66	0.94	3.12	0.33	−0.49	−0.40	0.41	1.01	−0.04	0.48

Table 16: Performance comparison of Contrastive Loss Variants on Track II. The improvements (Δ) over base method(GRPO) are indicated with positive (green) and negative (red) colors.

Methods	Math	Math_S	Math_H	AMC	AIME	AIME25	M-Avg.	C-Avg.	Avg.
GRPO	76.46	64.83	37.86	46.84	17.92	9.58	59.72	24.78	42.25
CLIPO-InfoNCE	77.49	67.21	38.97	49.17	18.13	10.63	61.22	25.97	43.60
Δ	1.03	2.38	1.11	2.33	0.21	1.04	1.50	1.19	1.35
CLIPO-SupCon	78.18	65.90	38.97	48.42	17.71	6.88	61.01	24.33	42.67
Δ	1.72	1.06	1.11	1.58	−0.21	−2.71	1.30	−0.45	0.43
CLIPO-SoftNN	78.17	65.78	38.02	49.85	20.21	9.58	60.66	26.55	43.60
Δ	1.71	0.95	0.16	3.01	2.29	0.00	0.94	1.77	1.35

C.3 Contrastive Loss Variants

We have also implemented additional loss functions for the Contrastive LM Head, including Soft Nearest Neighbor (SoftNN) loss and Supervised Contrastive (SupCon) loss. We refer to the corresponding methods as CLIPO-SoftNN and CLIPO-SupCon, respectively. The main experimental results for these methods on GSM8K and MATH datasets are presented in Tables 15 and 16.

One observation is that SoftNN exhibits a performance drop on G-Avg. To further investigate this phenomenon, we conduct additional ablation experiments using SoftNN loss across different base methods. The complete experimental results are presented in Table 17.

C.4 Temperature Results

The complete experimental results across different temperature settings on Track II are presented in Table 18. Figure 7 visualizes the average cosine similarity of positive pairs within the same group during training, under different temperature settings. The plot illustrates that higher temperature values lead to greater fluctuations in the similarity of positive pairs, indicating increased instability in distinguishing positive and negative examples.

C.5 Group Size Results

The complete experimental results across different group size settings are presented in Table 19. From the results, we observe that larger group sizes generally lead to better performance across different temperature settings. Part of this trend is expected from the contrastive objective, whose activation rate increases with G (Appendix C.10); however, a larger G also improves the group-relative baseline and exploration of the underlying RLVR method, so these results alone do not isolate the contribution of contrastive learning.

Table 17: Performance comparison of different RLVR methods. CLIPO-SoftNN improves the average performance based on four baseline methods, with the main improvements in M-Avg. tasks, while the improvements in G-Avg. tasks are limited, and even some decreases.

Methods	GSM	SYM	P1	P2	cQA	tQA	thrmQA	MMLU	M-Avg.	G-Avg.	Avg.
Base	86.05	82.40	70.90	47.96	73.63	29.99	24.90	68.09	71.83	49.15	60.49
GRPO	87.79	84.04	73.12	50.80	74.94	30.48	27.71	68.19	73.94	50.33	62.13
CLIPO-SoftNN	87.11	84.70	74.06	53.92	75.27	29.99	27.31	68.59	74.95	50.29	62.62
Δ	−0.68	0.66	0.94	3.12	0.33	−0.49	−0.40	0.41	1.01	−0.04	0.48
GSPO	87.19	83.94	73.28	51.36	74.86	30.23	27.38	67.68	73.94	50.04	61.99
CLIPO-SoftNN	87.87	83.92	75.14	51.60	76.33	30.97	27.64	68.44	74.63	50.85	62.74
Δ	0.68	−0.02	1.86	0.24	1.47	0.73	0.27	0.76	0.69	0.81	0.75
DAPO	87.41	84.56	74.70	52.32	76.00	31.58	27.11	69.00	74.75	50.92	62.84
CLIPO-SoftNN	88.25	84.48	76.10	53.52	75.84	31.82	27.24	67.68	75.59	50.65	63.12
Δ	0.83	−0.08	1.40	1.20	−0.16	0.24	0.13	−1.32	0.84	−0.28	0.28
GMPO	86.81	84.02	73.90	49.80	76.33	30.97	26.57	68.49	73.63	50.59	62.11
CLIPO-SoftNN	87.87	84.14	73.94	51.32	75.84	31.52	26.24	67.63	74.32	50.31	62.31
Δ	1.06	0.12	0.04	1.52	−0.49	0.55	−0.33	−0.86	0.69	−0.28	0.20

Table 18: Performance comparison of different temperature settings on Track II.

τ	Math	Math_S	Math_H	AMC	AIME	AIME25	M-Avg.	C-Avg.	Avg.
0.2	77.13	66.12	36.44	47.67	17.92	7.50	59.90	24.36	42.13
0.1	76.90	65.01	37.36	**51.05**	**18.75**	**10.63**	59.76	**26.81**	43.28
0.05	77.18	67.14	**39.53**	48.72	16.04	9.79	**61.28**	24.85	43.07
0.02	**77.49**	**67.21**	38.97	49.17	18.13	**10.63**	61.22	25.97	**43.60**
0.015	76.79	66.42	38.72	50.15	17.71	7.71	60.64	25.19	42.91

C.6 Base Model Variants

Beyond Qwen2.5, we also validate the effectiveness of CLIPO on other base models, including DS-7B and Llama-8B. The results are presented in Table 20.

C.7 Multi-Seed Results

Table 21 lists the paired differences between GRPO+CLIPO and GRPO over three independent training seeds, visualized in Figure 3(a).

C.8 Details of the Step-Level Process Evaluation

Table 22 lists the exact values shown in Figure 3(b) and (c).

Protocol. We follow the ProcessBench protocol (Zheng et al., 2025b). Each outcome-correct rollout is split into steps, and the judge (DeepSeek-V4-Flash-0423) receives only the problem, the reference answer and the step-split solution; it returns the index of the earliest erroneous step (or none) and an error type. The judge is blind to the method that produced the rollout. We sample two rollouts per prompt and method from GRPO and GRPO+CLIPO (Qwen2.5-7B-Instruct, Track II) at training steps 116, 203 and 406, giving 10,573 outcome-correct rollouts over 2,686 prompt-matched pairs. As a validity check, the same judge flags 94.7% of 416 outcome-incorrect rollouts, where an error exists by construction.

Per-checkpoint results. The flawed-but-correct rate is 30.5% / 33.8% / 30.8% for GRPO and 24.5% / 26.8% / 24.2% for GRPO+CLIPO at steps 116 / 203 / 406, i.e., reductions of -6.0 / -7.0 / -6.6 pp.

Robustness to surface features. Three estimators agree. (i) A paired per-prompt cluster bootstrap over the 2,686 prompts gives -6.52 pp (95% CI $[-8.21, -4.80]$). (ii) Stratifying by the quartile of the number of judged steps gives -5.6 pp, negative in all four strata (-4.5 / -5.6 / -6.9 / -6.1). (iii) A logistic regression with $\log(\#\text{steps})$, number of characters, number of paragraphs, display- and inline-math density, and checkpoint and benchmark fixed effects, with prompt-clustered

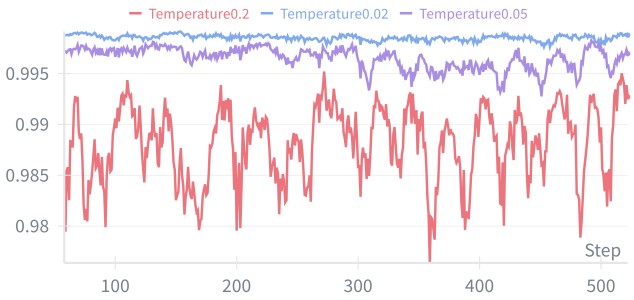


Figure 7: The plot illustrates the average cosine similarity of positive pairs within the same group during training, under different temperature settings. We observe that higher temperature values lead to greater fluctuations in the similarity of positive pairs.

Table 19: Performance comparison of different group size.

τ	GroupSize	Math	Math_S	Math_H	AMC	AIME	AIME25	M-Avg.	C-Avg.	Avg.
0.05	$\|G\| = 8$	76.56	64.92	38.09	46.69	17.50	**10.00**	59.86	24.73	42.29
	$\|G\| = 16$	77.18	**67.14**	**39.53**	48.72	16.04	9.79	**61.28**	24.85	43.07
	$\|G\| = 32$	**77.92**	64.90	39.37	**49.55**	**21.04**	7.08	60.73	**25.89**	**43.31**
0.02	$\|G\| = 8$	77.41	64.90	38.49	47.67	15.42	8.54	60.27	23.87	42.07
	$\|G\| = 16$	**77.49**	**67.21**	38.97	49.17	18.13	**10.63**	**61.22**	25.97	43.60
	$\|G\| = 32$	77.26	66.03	**39.89**	**50.45**	**20.83**	8.33	61.06	**26.54**	**43.80**

sandwich standard errors ($n = 10{,}486$; 2,686 clusters), gives an average marginal effect of -5.50 pp ($z = -6.46$, $p = 1.1 \times 10^{-10}$), while every surface control is individually non-significant ($|z| < 1.4$). Across nine surface features, the largest between-method effect size is $|d| = 0.19$, and response length is nearly identical (1,401 vs. 1,404 characters, $d = 0.004$). GRPO rollouts contain slightly more judged steps on average (8.67 vs. 7.79), which is why we report the stratified and regression-adjusted estimates alongside the raw difference.

C.9 Pass@k Analysis

To examine whether aligning successful trajectories reduces solution diversity, we report Pass@16 ($k = G = 16$) on all six Track II benchmarks in Figure 8(a) and Table 23. On average, Pass@16 is retained (59.95% $\rightarrow$ 60.53%), while Pass@1 improves 2.3$\times$ faster ($+1.35$ vs. $+0.58$), and the share of solvable problems solved on the first attempt rises on 5 of 6 benchmarks. Diversity collapse would show the opposite signature, with Pass@16 decreasing as Pass@1 increases. We read Pass@16 as a negative control rather than as a claimed gain: with 30 to 500 problems per benchmark, no per-benchmark difference is individually resolvable (the two negative cells correspond to 0.7 and 0.3 problems), but a collapse large enough to matter would be clearly visible. Structurally, the contrastive loss is only applied when $1 < |\mathcal{P}| < G$, so all-correct groups are never pulled together.

C.10 Activation of the Contrastive Objective

Since the contrastive loss is only applied when $1 < |\mathcal{P}| < G$, we measure how often it is active from the rollout logs (1,193 groups of $G = 16$ rollouts per checkpoint). As shown in Table 24 and Figure 8(b), the objective is active in 32–37% of groups, decreasing slightly over training as more groups become saturated, and the activation rate increases with the group size. Across benchmarks, activation is lowest in-distribution (Math, 30.4%, where 56.9% of groups are already saturated) and higher on the perturbed and AMC benchmarks (42.4%, 42.6% and 53.0% for Math_S, Math_H and AMC), where the accuracy gains are also largest (Figure 8(c)).

The intermediate-pass-rate condition $1 < |\mathcal{P}| < G$ partly coincides with headroom, so this correspondence could in principle reflect a generic property of group-relative methods. The two, however, come apart: AIME and AIME25 have the most headroom but the lowest activation (23.3% and 16.7%), and across the six Track II benchmarks the gain of CLIPO over GRPO correlates with activation (Spearman $\rho = +0.60$) but not with headroom ($\rho = -0.03$), whereas the gain of GRPO over the

Table 20: Performance comparison on various base models.

Methods	Math	Math_S	Math_H	AMC	AIME	AIME25	M-Avg.	C-Avg.	Avg.
DeepSeek-R1-Distill-Qwen-7B									
GRPO	89.70	83.98	64.48	72.59	36.88	26.67	79.39	45.38	62.38
CLIPO	90.08	83.94	63.83	73.80	39.38	26.46	79.28	46.54	62.91
Δ	0.38	-0.05	-0.65	1.20	2.50	-0.21	-0.11	1.17	0.53
Llama3.1-8B-Instruct									
GRPO	52.04	40.10	12.39	23.64	10.63	0.00	34.84	11.42	23.13
CLIPO	54.14	42.87	14.28	25.38	9.79	0.21	37.10	11.79	24.44
Δ	2.10	2.76	1.90	1.73	-0.83	0.21	2.25	0.37	1.31

Table 21: Improvement of GRPO+CLIPO over GRPO across three independent training seeds, with paired 95% confidence intervals of the per-seed differences. Significant improvements are highlighted in bold.

Track	Benchmark	Δ	95% CI
Track I	GSM8K (in-dist.)	$+0.28$	$[-0.33, +0.88]$
	P2	$\mathbf{+3.98}$	$[+2.28, +5.69]$
	M-Avg.	$\mathbf{+1.63}$	$[+0.10, +3.16]$
	Avg.	$\mathbf{+1.33}$	$[+0.53, +2.13]$
Track II	Math (in-dist.)	$+0.63$	$[-0.24, +1.50]$
	Math_S	$\mathbf{+2.73}$	$[+1.48, +3.97]$
	Math_H	$+1.01$	$[-0.47, +2.49]$
	AMC	$\mathbf{+3.63}$	$[+0.43, +6.83]$
	M-Avg.	$\mathbf{+1.66}$	$[+0.87, +2.44]$

base model shows the opposite pattern ($\rho = -0.09$ with activation, $+0.26$ with headroom). With only six benchmarks, these correlations are not individually significant (permutation $p = 0.24$ for $\rho = +0.60$) and should be read as supporting rather than conclusive evidence.

Table 24(d) also quantifies the cost of computing the contrastive loss within the on-policy rollout group: among active groups, the median number of negatives is 6, and one third of them have at most 3 negatives, a regime in which the InfoNCE bound is loose. We discuss this trade-off in Appendix B.

C.11 Training Dynamics

To analyze the training dynamics of CLIPO, we track the entropy of the policy, the contrastive reward r^{CL}, and the contrastive loss $\mathcal{L}_{\text{CL}}$ throughout the training process. As shown in Figure 9, we observe that the policy entropy gradually decreases over time, indicating that the model is becoming more confident in its predictions as it learns from the reward signals. The contrastive reward r^{CL} shows an increasing trend, suggesting that the model is learning to produce responses that are more similar to high-quality solutions while pushing away low-quality ones in the representation space. Meanwhile, the contrastive loss $\mathcal{L}_{\text{CL}}$ decreases over time, indicating that the model is effectively learning to distinguish between positive and negative samples, which contributes to the overall performance improvement. These training dynamics provide insights into how CLIPO enhances the learning process by providing more informative and fine-grained reward signals, leading to better performance on mathematical reasoning tasks.

C.12 Overhead Analysis

To evaluate the computational overhead introduced by CLIPO, we measure the wall-clock training time with and without CLIPO. Overall, CLIPO incurs only marginal additional cost.

Table 22: Step-level evaluation of outcome-correct rollouts on Track II. *Flawed* denotes the percentage of rollouts that reach the correct answer but contain at least one erroneous step, as identified by a method-blind step-level judge (lower is better). The lower block breaks down the flawed rate by error type.

Slice	**GRPO**	**GRPO+CLIPO**	**Δ (pp)**
Overall (2,686 prompt pairs)	31.7	**25.2**	−6.5
Math	27.8	**22.6**	−5.2
Math_S	34.4	**27.7**	−6.8
Math_H	37.7	**29.3**	−8.4
AMC	32.8	**21.7**	−11.1
AIME25	36.4	**27.3**	−9.1
Error type			
Answer forcing	4.4	**2.0**	−2.4
Logic	12.2	**10.0**	−2.2
Calculation	15.1	**13.2**	−1.9

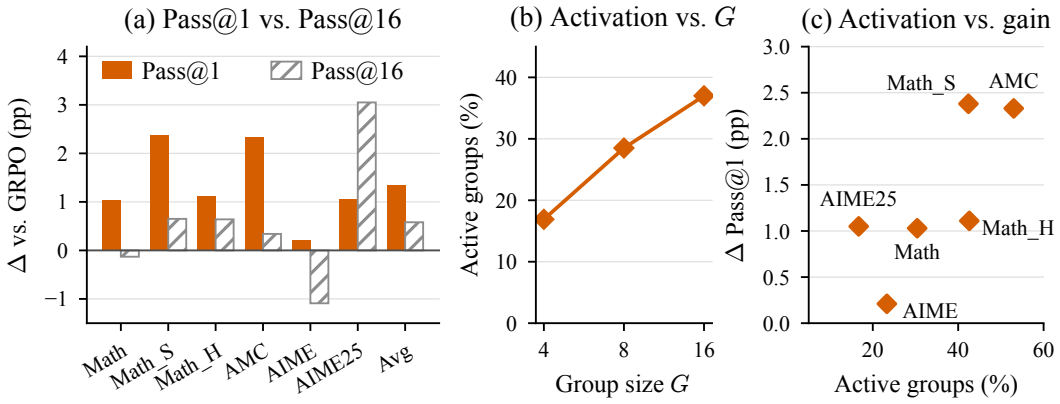


Figure 8: CLIPO retains solution coverage, and its contrastive objective is active more often with larger groups and on benchmarks with larger gains. (a) Improvement of GRPO+CLIPO over GRPO in Pass@1 and Pass@16 on Track II. (b) Share of rollout groups where the contrastive loss is active ($1 < |\mathcal{P}| < G$) for different group sizes. (c) Per-benchmark activation rate versus Pass@1 improvement. Exact numbers are in Tables 23 and 24.

- First, the additional trainable parameters introduced by the contrastive head are lightweight compared with the LLM backbone. For example, on Qwen2.5-7B, the contrastive module contains only approximately 7.34M parameters, which is negligible relative to the 7B backbone.
- Second, the memory overhead is minimal. The original RL training pipeline already stores rollout hidden states with complexity $O(BTD)$, where B, T, and D denote batch size, sequence length, and hidden dimension, respectively. CLIPO only introduces an additional $O(BD)$ memory cost

Table 23: Pass@16 and Pass@1 on Track II. *Coverage realized* is Pass@1/Pass@16, i.e., the share of solvable problems that are solved on the first attempt.

Benchmark	**#Problems**	**GRPO Pass@16**	**GRPO+CLIPO Pass@16**	**Δ Pass@16**	**Δ Pass@1**	**Coverage realized**
Math	500	**88.62**	88.48	−0.13	+1.03	86.3% → **87.6%**
Math_S	276	79.23	**79.88**	+0.65	+2.38	81.8% → **84.1%**
Math_H	277	60.84	**61.49**	+0.64	+1.11	62.2% → **63.4%**
AMC	83	70.11	**70.45**	+0.34	+2.33	66.8% → **69.8%**
AIME	30	**33.74**	32.65	−1.09	+0.21	53.1% → **55.5%**
AIME25	30	27.16	**30.21**	+3.05	+1.05	**35.3%** → 35.2%
Avg.	–	59.95	**60.53**	+0.58	+1.35	64.3% → **65.9%**

Table 24: Activation statistics of the contrastive objective for GRPO+CLIPO on Track II. A group is *active* when $1 < |\mathcal{P}| < G$, and *saturated* when $|\mathcal{P}| = G$.

(a) Over training ($G = 16$)					
Training step	**Active**	**Saturated**		**Mean $\lvert\mathcal{P}\rvert$**	
116	37.0%	36.3%		9.28	
203	35.2%	38.9%		9.60	
406	32.2%	41.7%		9.65	
(b) Group size (step 116)					
G	4	8		16	
Active	16.9%	28.5%		37.0%	
(c) Per benchmark ($G = 16$)					
Math 30.4%	Math_S 42.4%	Math_H 42.6%	AMC 53.0%	AIME 23.3%	AIME25 16.7%
(d) Composition of groups ($G = 16$)					
Median $\lvert\mathcal{P}\rvert$ / median #negatives among active groups				10 / 6	
Active groups with ≤ 3 negatives				33%	
Groups with $\lvert\mathcal{P}\rvert = 0$ (no contribution from CLIPO)				∼22%	

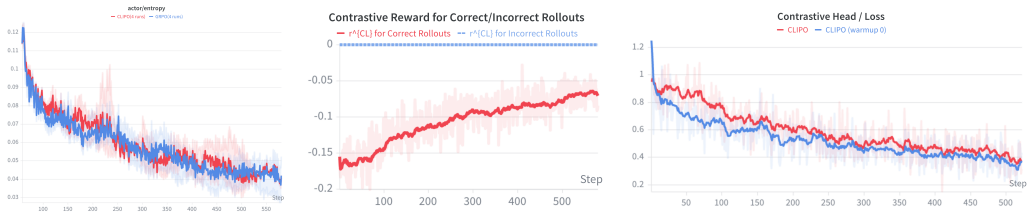


Figure 9: Training dynamics of CLIPO throughout the training process.

for contrastive representations. Under a typical reasoning setting with $T = 4096$, the additional memory usage is approximately 0.024% of the original training memory footprint.

- Third, the computational overhead introduced by the contrastive objective scales linearly with sequence length, i.e., $O(T)$, which is negligible compared with the dominant $O(T^2)$ complexity of transformer-based autoregressive decoding.
- Finally, we report the practical runtime overhead in terms of wall-clock step time in Table 25. On Qwen2.5-7B training, the baseline GRPO implementation requires 84.83s (±18.1s) per optimization step, while CLIPO introduces only an additional 1.03s (±0.1s). These results demonstrate that CLIPO achieves consistent performance gains with negligible training overhead.

Table 25: Training runtime overhead introduced by CLIPO.

Metric	GRPO Runtime	CLIPO Extra Runtime
Step Time	84.83s (±18.1s)	1.03s (±0.1s)