Uncovering What Matters in RL for LLM Mathematical Reasoning

Hantao Zhou

hantaozh@outlook.com zht8506/Easy-LLM-Post-Training

Abstract | Reinforcement learning (RL), especially RL with verifiable rewards, has become the mainstream paradigm to enhance the mathematical reasoning capability of large language models (LLMs), yet a systematic controlled-variable empirical comparison of mainstream RL algorithms is still absent. This paper systematically reviews more than ten representative RL algorithms for LLM alignment, conducts fair comparative experiments on six mathematical reasoning benchmarks under a unified setup, and analyzes the key design factors affecting algorithm performance. We verify the universal effectiveness of RL for LLM reasoning enhancement, give a clear performance ranking of mainstream algorithms, and reveal the core determinants of RL performance, providing empirical guidance for algorithm selection and optimization in LLM mathematical reasoning scenarios.

1. Introduction

Large language models (LLMs) have achieved remarkable breakthroughs in complex mathematical reasoning tasks. Recently, reinforcement learning (RL), especially Reinforcement Learning with Verifiable Rewards (RLVR), has become the dominant paradigm for enhancing the reasoning capability of LLMs. Driven by RL optimization, state-of-the-art reasoning models [Guo et al., 2025, Team et al., 2026, Zeng et al., 2026] have reached human-expert level performance on high-difficulty mathematical benchmarks, fully demonstrating the great potential of RL in aligning LLMs with complex reasoning objectives.

A series of novel RL frameworks (e.g., GRPO Shao et al. [2024], DAPO Yu et al. [2025], GSPO Zheng et al. [2025], SAPO Gao et al. [2025]) and optimization strategies have been proposed for LLM reasoning tasks, with distinct designs in advantage estimation, policy update constraints, variance reduction, and training-inference distribution mismatch mitigation. Despite the abundant algorithmic innovations, there is a lack of systematic, controlled-variable empirical comparison of these mainstream methods. The actual performance gains and core effective design tricks of different algorithms in mathematical reasoning scenarios remain largely unclarified.

To address the above gaps, this paper conducts a comprehensive comparison and in-depth analysis of mainstream RL algorithms for LLM mathematical reasoning. First, we systematically reviewed reinforcement learning algorithms for LLM alignment, covering the technical details of more than 10 representative RL algorithms. Second, we perform strictly controlled-variable experiments under a unified setup: using DeepSeek-R1-Distill-Qwen-1.5B Guo et al. [2025] as the backbone model, we evaluate all algorithms on 6 widely recognized mathematical reasoning benchmarks. Finally, we conduct an in-depth analysis of the key design factors that affect the performance of RL algorithms in mathematical reasoning tasks, including optimized advantage estimation, improved clipping mechanisms, KL regularization, and distribution mismatch mitigation strategies.

Our experimental results verify the universal effectiveness of RL in enhancing LLM reasoning performance, and provide a clear performance ranking of mainstream algorithms in mathematical reasoning scenarios. The analysis reveals the core factors that dominate the performance of RL for LLM reasoning, offering solid empirical guidance for algorithm selection.

2. Algorithms

2.1. Preliminaries

KL Divergence. KL divergence measures the difference between two probability distributions—it quantifies how much one distribution diverges from another. In LLM RL training, KL divergence is widely used as a constraint loss to prevent the updated policy from drifting too far from the original reference model. This stabilizes training and avoids catastrophic forgetting or unreasonable generation. KL divergence $\mathrm{KL}(q||p) = \mathbb{E}_{x\sim q}\left[\log \frac{q(x)}{p(x)}\right]$ has three common Monte Carlo estimators, denoted as K1, K2, K3, with distinct bias and variance properties:

- **K1 Estimator**: The K1 estimator, $\mathrm{KL}_1 = \log \frac{q(x)}{p(x)}$, is an unbiased vanilla estimator. It has zero bias but high variance.
- **K2 Estimator**: The K2 estimator, $\mathrm{KL}_2 = \frac{1}{2}\left(\log \frac{p(x)}{q(x)}\right)^2$, has low variance and tiny bias. It has a constant positive property, which is consistent with the KL divergence.
- **K3 Estimator**: The K3 estimator, $\mathrm{KL}_3 = \left(\frac{p(x)}{q(x)} - 1\right) - \log \frac{p(x)}{q(x)}$, is an unbiased and low-variance estimator. It keeps zero bias and achieves lower variance than KL_2.

REINFORCE. REINFORCE is a Monte Carlo-based policy gradient reinforcement learning algorithm that directly calculates and updates the policy network parameters. REINFORCE estimates the gradient using the trajectory rewards over a complete epoch, without estimating the value function. The gradient of REINFORCE on the policy model is:

$$\nabla_\theta \mathcal{J}_{\mathrm{REINFORCE}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi(a_t|s_t;\theta)\cdot G_t\right] \tag{1}$$

where $J(\theta)$ is the REINFORCE objective function,; $\pi(a_t|s_t;\theta)$ is the probability that policy $\pi(\theta)$ chooses action a_t when it is in state s_t. G_t is the cumulative return starting from time step t, calculated as $G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$, where r_k is the immediate reward at time step k and $\gamma \in [0,1]$ is the discount factor.

2.2. PPO

PPO [Schulman et al., 2017] is a classic critic-based on-policy reinforcement learning algorithm that balances training stability and policy update efficiency, widely used for alignment and reasoning enhancement of large language models. It uses a clipped surrogate objective to restrict the update range of the new policy relative to the old policy, avoiding performance collapse caused by excessive policy updates.

Formally, for a given prompt x sampled from the dataset distribution $\mathcal{D}$, we let y be the complete response generated from the old policy model $\pi_{\theta_{\mathrm{old}}}$. The core objective of PPO is constructed as:

$$\mathcal{J}_{\mathrm{PPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_{\theta_{\mathrm{old}}}(\cdot|x)}\left[\frac{1}{|y|}\sum_{t=1}^{|y|}\min\left(w_t(\theta)\,\hat{A}_t,\ \mathrm{clip}\big(w_t(\theta), 1-\epsilon, 1+\epsilon\big)\,\hat{A}_t\right)\right] \tag{2}$$

where $w_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}$ denotes the probability ratio between the updated policy and the previous policy and ϵ is a hyperparameter that controls the maximum allowed deviation of the new policy

from the old policy. A_t is typically calculated using the Generalized Advantage Estimation [Schulman et al., 2015] (GAE) function:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l} \tag{3}$$

where $\delta_t = \tilde{r}_t + \gamma V_{t+1} - V_t$ is the temporal difference error, and $V(\cdot)$ is the critic model. The hyperparameter $\lambda \in [0, 1]$ is used to balance the bias and variance in advantage estimation. A key design in PPO is that KL divergence is directly included in the reward, meaning the effective reward is adjusted by subtracting a KL penalty term. The adjusted reward with KL regularization is defined as:

$$\tilde{r}_t = r_t - \beta \cdot D_{\mathrm{KL}}\big(\pi_\theta(y_t|x, y_{<t}) \,\|\, \pi_{\mathrm{ref}}(y_t|x, y_{<t})\big) \tag{4}$$

where $\beta > 0$ is the KL penalty coefficient, and π_{ref} is usually the pre-trained or SFT initial model.

2.3. GRPO

GRPO [Shao et al., 2024] is a critic-free on-policy algorithm designed for stable and scalable training of large reasoning models. It replaces the value-based advantage estimation with group-relative normalization, eliminating the need for a separate critic network and improving training efficiency. Formally, for a given prompt $x \in \mathcal{D}$, let $\{y_i\}_{i=1}^{G}$ be a group of G response sequences generated by the old policy $\pi_{\theta_{\mathrm{old}}}$. The objective of GRPO is defined as:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|x)} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \min\left(w_{i,t}(\theta)\hat{A}_{i,t}, \mathrm{clip}\left(w_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right) \tag{5}$$

For the outcome supervision RL (*i.e.*, RLVR), the researchers typically broadcast the reward of sequence to each token. At this point, the group-wise advantage is:

$$\hat{A}_{i,t} = \frac{r(x, y_i) - \mathrm{mean}(\{r(x, y_j)\}_{j=1}^{G})}{\mathrm{std}(\{r(x, y_j)\}_{j=1}^{G})} \tag{6}$$

All tokens in sequence y_i share the same sequence-level advantage A_i. Unlike PPO, GRPO applies KL divergence regularization directly in the loss function rather than in the reward.

$$\mathcal{L}_{\mathrm{GRPO}} = -\mathcal{J}_{\mathrm{GRPO}}(\theta) + \beta \cdot D_{\mathrm{KL}_3}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) \tag{7}$$

GRPO stabilizes training by reducing variance through group baselines and has become the most popular algorithm for Reinforcement Learning with Verifiable Rewards (RLVR) in math, code, and general reasoning tasks.

2.4. GSPO

GSPO [Zheng et al., 2025] elevates importance sampling and clipping from token-level to sequence-level, improving training stability for Mixture of Experts (MoE) large language models.

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|x)} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \min\left(w_i(\theta)A_i, \mathrm{clip}\left(w_i(\theta), 1-\epsilon, 1+\epsilon\right)A_i\right) \tag{8}$$

with sequence-wise importance weight:

$$w_i(\theta) = \left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\mathrm{old}}}(y_i|x)}\right)^{\frac{1}{|y_i|}} = \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\log\frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(y_{i,t}|x, y_{i,<t})}\right) \tag{9}$$

By avoiding per-token bias in importance sampling, GSPO achieves smoother convergence and better scalability for large scale models.

2.5. DAPO

DAPO [Yu et al., 2025] improves GRPO with asymmetric clipping (clip-higher) and dynamic sampling, enabling more aggressive exploration for low-probability but high-reward tokens and balance the loss weights of long and short samples.

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \sum_{i=1}^{G} \frac{1}{|S|} \sum_{t=1}^{|y_i|} \min\left(w_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\left(w_{i,t}(\theta),\ 1-\epsilon_{\text{low}},\ 1+\epsilon_{\text{high}}\right)\hat{A}_{i,t}\right) \quad (10)$$

where $|S| = \sum_{i=1}^{G} |y_i|$ denotes the total number of tokens in the group of sequences. In addition, DAPO employs dynamic sampling and soft overlong punishment to increase the proportion of effective gradients and prevent unreasonable truncation penalties.

2.6. Dr. GRPO

Dr. GRPO [Liu et al., 2025] (GRPO Done Right) removes the response-length normalization and group-wise reward standardization components in GRPO, as these mechanisms result in undesirably long yet incorrect responses and imbalanced optimization. The objective is formulated as:

$$\mathcal{J}_{\text{Dr. GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \frac{1}{G} \sum_{i=1}^{G} \sum_{t=1}^{|y_i|} \min\left(w_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\left(w_{i,t}(\theta),\ 1-\epsilon,\ 1+\epsilon\right)\hat{A}_{i,t}\right) \quad (11)$$

The revised group-wise advantage is:

$$\hat{A}_{i,t} = r(x, y_i) - \text{mean}(\{r(x, y_j)\}_{j=1}^{G}) \quad (12)$$

2.7. RLOO

RLOO [Ahmadian et al., 2024] is a critic-free algorithm that uses a leave-one-out baseline to construct unbiased advantage estimates, reducing variance more effectively than simple mean baselines. RLOO employs a REINFORCE-style reinforcement learning objective.

$$\nabla_\theta \mathcal{J}_{\text{RLOO}}(\theta) = \frac{1}{k} \sum_{i=1}^{k} \left[r(x, y_{(i)}) - \frac{1}{k-1} \sum_{j\neq i} r(x, y_{(j)}) \right] \nabla_\theta \log \pi_\theta(y_{(i)}|x) \quad (13)$$

2.8. GMPO

Instead of optimizing the arithmetic mean of token-level importance-weighted rewards, GMPO [Zhao et al., 2025] maximizes the geometric mean, which is inherently less sensitive to outliers and suppresses extreme values in importance sampling ratios during training. This design yields more stable policy updates, lower variance in gradients, and sustained exploration via higher token entropy throughout training. The GMPO objective can be formulated as:

$$\mathcal{J}_{\text{GMPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \frac{1}{G} \sum_{i=1}^{G} \left\{ \prod_{t=1}^{|y_i|} \left| \min\left[w_{i,t}(\theta)\hat{A}_i,\ \text{clip}\left(w_{i,t}(\theta), \epsilon_{\text{low}}, \epsilon_{\text{high}}\right)\hat{A}_i \right] \right| \right\}^{\frac{1}{|y_i|}} \cdot \text{sgn}(\hat{A}_i) \quad (14)$$

where $\text{sgn}(\hat{A}_i)$ ensures the correct optimization direction, returning 1 when $\hat{A}_i$ is positive and -1 otherwise. Due to the stability of the geometric mean, DMPO can use a larger clipping thresholds, encouraging greater exploration and improving performance.

2.9. CISPO

CISPO [Chen et al., 2025] (Clipped Importance Sampling-weight Policy Optimization) revises the clipping mechanism by clipping importance sampling (IS) weights instead of token-level updates, which retains gradient contributions from all tokens and avoids discarding critical low-probability reasoning tokens in long chain-of-thought (CoT) generation.

$$\mathcal{J}_{\text{CISPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \text{sg}\left(\hat{w}_{i,t}(\theta)\right)\hat{A}_{i,t}\log\pi_\theta(y_{i,t}|x,y_{i,<t}) \tag{15}$$

Note that CISPO uses REINFORCE as its baseline, not PPO. CISPO also adopts the same group-wise relative advantages as GRPO but discards KL constraints.

2.10. SAPO

Soft Adaptive Policy Optimization [Gao et al., 2025] (SAPO) replaces hard clipping with a temperature-controlled soft gate to deliver sequence-coherent and token-adaptive policy updates, enhancing training stability and sample efficiency for large language models, especially Mixture of Experts (MoE) architectures. The objective of GRPO can be formulated as:

$$\mathcal{J}_{\text{SAPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} f_{i,t}\left(w_{i,t}(\theta)\right)\hat{A}_{i,t} \tag{16}$$

The soft gating function with asymmetric temperature for positive/negative advantages is defined as:

$$f_{i,t}(x) = \sigma\left(\tau_{i,t}(x-1)\right)\cdot\frac{4}{\tau_{i,t}}, \quad \tau_{i,t} = \begin{cases}\tau_{\text{pos}}, & \hat{A}_{i,t} > 0\\ \tau_{\text{neg}}, & \text{otherwise}\end{cases} \tag{17}$$

Negative advantage updates tend to elevate logits of numerous irrelevant tokens in the large vocabulary of LLMs, introducing far more instability than positive updates. To mitigate this, SAPO adopts asymmetric temperature settings with $\tau_{\text{neg}} > \tau_{\text{pos}}$, making gradients of negative tokens decay more rapidly.

2.11. TIS

Truncated Importance Sampling (TIS) is an algorithm-level correction strategy to resolve the rollout-training distribution mismatch in LLM reinforcement learning (RL) frameworks. This mismatch stems from hybrid inference-training backends (e.g., vLLM for rollout sampling, FSDP for model training), which breaks the on-policy assumption by generating inconsistent token probabilities between the sampler policy π_{sampler} and learner policy π_{learner}. TIS calibrates the policy gradient with a truncated importance ratio to eliminate bias and stabilize training. The core truncated importance ratio of TIS is formulated as:

$$w_{\text{TIS}} = \min\left(\frac{\pi_{\text{learner}}(y_{i,t}\mid x)}{\pi_{\text{sampler}}(y_{i,t}\mid x)}, C\right) \tag{18}$$

TIS can be seamlessly applied to various reinforcement learning algorithms. Here I demonstrate the application of TIS to GRPO:

$$\mathcal{J}_{\text{TIS-GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_{\text{sampler}}(\cdot|x)} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} w_{\text{TIS}}\cdot\min\left(w_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(w_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right) \tag{19}$$

2.12. IcePop

IcePop [Team et al., 2025] is a lightweight on-policy RL optimization method tailored for Mixture-of-Experts (MoE) models. It targets the training-inference probability mismatch caused by MoE's dynamic routing, where minor precision gaps lead to inconsistent expert selection, compounding divergence, and training collapse. Unlike TIS that uses mild coefficient scaling, IcePop adopts double-sided masking to discard noisy gradients and retain valid updates, stabilizing training and boosting reasoning performance of MoE models. Applying IcePop to GRPO can be represented as follows:

$$\mathcal{J}_{\text{IcePop-GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\mathcal{M}\left(\frac{\pi_{\text{train}}}{\pi_{\text{infer}}}\right)\cdot\min\left(r_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\left(r_{i,t}(\theta),1-\epsilon,1+\epsilon\right)\hat{A}_{i,t}\right)\right] \tag{20}$$

$\mathcal{M}$ is a double-sided masking function that discards gradient updates of tokens with excessive training-inference probability discrepancy, preserving only healthy updates within the threshold range $[\alpha,\beta]$.

$$\mathcal{M}(k) = \begin{cases} k & k\in[\alpha,\beta] \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

2.13. Reinforce++

REINFORCE++ [Hu, 2025] is a critic-free on-policy reinforcement learning framework for LLM alignment and RLHF, which solves the theoretical bias, training instability and overfitting problems of prompt-level local normalization (adopted by GRPO) with Global Advantage Normalization. It contains two task-specific variants: REINFORCE++ for general-domain RLHF ($k \geq 1$) and REINFORCE++w/ Baseline for complex reasoning/agentic tasks ($k > 1$).

For the basic REINFORCE++ ($k \geq 1$), the advantage with global normalization is calculated as:

$$A_{q,o_t}^{\text{norm}} = \frac{A_{q,o_t} - \text{mean}\left(A \mid A\in\mathcal{D}_{\text{batch}}\right)}{\text{std}\left(A \mid A\in\mathcal{D}_{\text{batch}}\right)+\epsilon} \tag{22}$$

For REINFORCE++w/ Baseline ($k > 1$), it first subtracts the group mean for reward reshaping, then applies global normalization:

$$A'_{q,o_t} = r(x,y_i) - \text{mean}\left(\{r(x,y_j)\}_{j=1}^{G}\right) \tag{23}$$

$$A_{q,o_t}^{\text{norm}} = \frac{A'_{q,o_t} - \text{mean}_{\text{batch}}(A')}{\text{std}_{\text{batch}}(A')+\epsilon} \tag{24}$$

REINFORCE++ adopts PPO-style optimization objectives:

$$\mathcal{J}_{\text{REINFORCE++}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\left(w_{i,t}(\theta)A_{i,t}^{\text{norm}},\text{clip}\left(w_{i,t}(\theta),1-\epsilon,1+\epsilon\right)A_{i,t}^{\text{norm}}\right) \tag{25}$$

Regarding the KL divergence, RE applies KL term (KL_1) to the reward function, as in PPO, while RER applies KL term (KL_2) to the loss function, as in GRPO.

3. Experiments

This section systematically evaluates the performance of mainstream reinforcement learning (RL) algorithms for large language model (LLM) reasoning, presents comprehensive benchmark results, and

conducts in-depth analysis of key design tricks that contribute to reasoning performance improvement. All experiments are conducted under controlled variable settings to ensure fair comparison across different algorithms.

3.1. Experimental Setup

3.1.1. Model and Datasets

We use DeepSeek-R1-Distill-Qwen-1.5B as the backbone model for all experiments, which is a supervised fine-tuned (SFT) distilled reasoning model serving as both the initial policy network and the fixed reference model for KL divergence regularization.

To comprehensively evaluate the mathematical reasoning capability of models optimized by different RL algorithms, we select 6 widely recognized reasoning benchmarks covering difficulty levels from high school competitions to olympiad-level problems. The mathematical reasoning benchmarks include AIME2024 [Li et al., 2024], AIME2025 [Balunović et al., 2025], AMC2023 [Li et al., 2024], MATH-500 [Hendrycks et al., 2021], Minerva Math [Lewkowycz et al., 2022], Olympiad-Bench [He et al., 2024].

3.1.2. Training Configuration

All algorithms are trained with identical core hyperparameters and training pipelines to ensure a fair comparison, with only algorithm-specific modules modified. We adopt Reinforcement Learning with Verifiable Rewards (RLVR) with outcome supervision, where a binary reward is assigned at the sequence level: a positive reward is given if the final answer of the generated response is correct, and zero reward otherwise.

The shared training hyperparameters are set as follows:

- **Group size** G: $G = 8$ for all group-based on-policy algorithms.
- **Clipping threshold** ϵ: Unless otherwise specified, $\epsilon = 0.2$; other specialized cropping thresholds follow the original paper.
- **Batch Size**: Train Batch Size is 128 and PPO Mini Batch Size is 64.
- **Context Length**: Max Prompt Length is 1K and Max Response Length is 8K.
- **Optimizer settings**: Learning rate = 1e-6, warm-up steps = 10, total training steps = 675.

3.1.3. Evaluation Protocol

We use the standard mean@k metric to evaluate the reasoning performance of all models. For AIME24, AIME25, and AMC23, we adopt @32 sampling (32 responses generated per prompt), while @4 sampling is used for the MATH, Minerva, and OlympiadBench datasets, following the common evaluation protocol in LLM reasoning research. The final average score (Avg) is calculated as the arithmetic mean of the scores across all 6 benchmarks, providing a comprehensive measure of the model's overall reasoning capability.

3.2. Main Experimental Results

Table 1 summarizes the performance of all evaluated RL algorithms on the 6 reasoning benchmarks, with the SFT model without RL optimization as the baseline.

Model	AIME24	AIME25	AMC23	MATH	Minerva	Olympiad	Avg
Baseline	19.58	19.69	54.9	74.95	23.90	34.98	38.00
GRPO	29.27	24.69	73.28	84.1	29.60	46.03	47.83
DAPO	24.69	25.94	68.75	81.7	29.32	43.66	45.68
CISPO	28.65	25.42	72.81	81.75	26.27	43.51	46.4
Reinforce++	27.4	23.54	73.05	83.65	29.41	45.99	47.17
GRPO-IcePop	28.44	24.17	72.27	83.65	29.69	45.25	47.25
GRPO-TIS	28.02	24.06	72.97	83.8	29.6	45.81	47.38
GSPO	28.85	25.83	71.56	83.5	28.77	46.48	47.50
PPO	30.42	24.48	74.61	83.1	28.68	46.18	47.91
GRPO w/o KL	29.38	24.58	75.39	84.1	30.51	45.36	48.22
RLOO	27.6	24.9	76.17	84.3	30.24	46.14	48.23
DrGRPO	29.9	25.1	74.22	84.3	30.5	47.51	48.59
GMPO	30.42	24.06	75.31	83.95	29.78	47.7	48.54
Reinforce†++	29.9	26.25	74.77	85.05	29.14	46.44	48.59
SAPO	30.83	25.42	75.31	85.15	29.69	47.03	48.91

Table 1 | Reasoning performance of different RL algorithms on DeepSeek-R1-Distill-Qwen-1.5B backbone. Reinforce†++ means Reinforce++ with baseline.

The key observations from the experimental results are summarized as follows:

- **Universal Effectiveness of RL for Reasoning Enhancement**: All RL algorithms achieve significant performance improvements over the SFT baseline, with the average score increasing by 7.34 to 10.91 points. This verifies that RL is a powerful paradigm for enhancing the multi-step reasoning capability of LLMs, even for models that have already been fine-tuned for reasoning tasks.
- **State-of-the-Art Algorithm Comparison**: SAPO achieves the highest average score of 48.91, outperforming all other algorithms and setting a new state-of-the-art on AIME24, AMC23, and MATH datasets. Dr. GRPO and REINFORCE++ w/ Baseline follow closely with an average score of 48.59, with Dr. GRPO achieving the top performance on the high-difficulty OlympiadBench dataset.
- **Classic Algorithm Comparison**: The classic PPO and widely adopted GRPO achieve comparable performance, with average scores of 47.91 and 47.83 respectively. Remarkably, GRPO matches the performance of PPO without requiring a separate critic network, demonstrating its superior computational efficiency for LLM reasoning tasks. In practice, GRPO reduces training time by nearly 30% compared to PPO.
- **Removing KL term**: In our ablation study, GRPO w/o KL achieves an average score of 48.22 across all benchmarks, outperforming the original KL-regularized GRPO (47.83). Beyond the GRPO ablation, multiple state-of-the-art RL algorithms discard KL regularization entirely, including SAPO, GMPO and DrGRPO. This is because KL imposes limitations on model exploration capabilities.

3.3. In-Depth Analysis: Key Tricks for Reasoning Performance Improvement

3.3.1. Optimized Advantage Estimation Reduces Bias and Variance

Advantage estimation is the core component of policy gradient algorithms, as it directly determines the bias and variance of gradient updates, which are critical for training stability and final reasoning performance. Our experiments reveal that refined advantage estimation strategies are among the most effective tricks for boosting reasoning capability:

- **Reducing the bias and variance of Advantage estimation**: RLOO uses a leave-one-out baseline instead of the simple group mean to construct an unbiased advantage estimator. This design achieves an average score of 48.23, outperforming the vanilla GRPO. The leave-one-out baseline removes correlation between sampling trajectories and the baseline, providing a theoretically unbiased advantage estimate. REINFORCE++ replaces the prompt-level local normalization in GRPO with global batch-wise normalization, which addresses the theoretical bias issue of local normalization. The REINFORCE++ w/ Baseline variant, which combines group-wise centering and global normalization, achieves 48.59, outperforming the basic GRPO. This indicates that global normalization can reduce optimization bias between different prompts, resulting in a more stable advantage estimate.
- **Removal of Unnecessary Reward Standardization**: Dr. GRPO removes the group-wise standard deviation normalization in the original GRPO, retaining only mean centering for advantage calculation. This modification leads to a 0.76 improvement in average score over GRPO. This is because introducing standard deviation normalization will cause the model to favor easy and difficult problems, resulting in training imbalance.

3.3.2. Robust Gradient Clipping Mechanism Improve RL Performance

SAPO and GMPO both enhance reasoning performance by addressing the core limitation of vanilla PPO and GRPO: over-reliance on narrow clip thresholds to suppress unstable gradients. Instead of restricting updates with tight clipping, these two methods use more robust mechanisms to stabilize policy optimization while preserving useful learning signals. GMPO stabilizes training via geometric mean aggregation, which naturally damps extreme importance weights and outliers without harsh truncation, enabling more consistent gradient signals for complex reasoning. SAPO replaces hard clipping with asymmetric soft gating, which smoothly down-weights noisy gradients rather than discarding them, maintaining adaptive exploration while avoiding instability from excessive updates. Together, they demonstrate that robust outlier-resilient gradient processing—rather than narrow clip constraints—effectively balances exploration and stability, yielding stronger and more reliable improvements in multi-step reasoning tasks.

3.3.3. Distribution Mismatch Mitigation Need Further Validation

Training-inference distribution mismatch is a common challenge in LLM RL training, especially for MoE models and hybrid inference-training backends. We evaluated two representative mitigation strategies, namely TIS and IcePop, and found that their performance is comparable to that of vanilla GRPO. In addition, GSPO, which is used to improve the stability of reinforcement learning training of MoE models, has a similar effect to GRPO. We conjecture that these strategies can demonstrate effectiveness when applied to MoE models and with longer training steps.

4. Conclusion

This paper systematically reviews more than ten mainstream reinforcement learning (RL) algorithms for large language model (LLM) mathematical reasoning, and conducts a fair comparative experiment under a unified controlled-variable setup on six widely recognized mathematical reasoning benchmarks, verifying the universal effectiveness of RL in enhancing the multi-step mathematical reasoning capability of LLMs. The experimental results give a clear performance ranking of mainstream algorithms in mathematical reasoning scenarios, and reveal that optimized advantage estimation, robust gradient update constraints, and reasonable regularization strategy design are the core determinants of algorithm performance. This study fills the gap of systematic controlled-variable empirical comparison in this field, and provides empirical guidance for algorithm selection and optimization in LLM mathematical reasoning tasks.

Limitation This study has several limitations. First, all experiments are conducted solely on the DeepSeek-R1-Distill-Qwen-1.5B model, and the generalizability of the empirical findings to larger and MoE language model has not been verified. Second, the training steps are relatively limited, and long-term training dynamics and stability under extended optimization are not explored.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, 2024.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

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*, 2025.

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

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

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. *arXiv preprint arXiv:2103.03874*, 2021.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv e-prints*, pages arXiv–2501, 2025.

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

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

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

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.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Ling Team, Anqi Shen, Baihui Li, Bin Hu, Bin Jing, Cai Chen, Chao Huang, Chao Zhang, Chaokun Yang, Cheng Lin, et al. Every step evolves: Scaling reinforcement learning for trillion-scale thinking model. *arXiv preprint arXiv:2510.18855*, 2025.

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

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

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

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*, 2025.