RL-PLUS: Countering Capability Boundary Collapse of LLMs in Reinforcement Learning with Hybrid-policy Optimization

Yihong Dong[1,2], **Xue Jiang**[1,2], **Yongding Tao**[1], **Huanyu Liu**[1], **Kechi Zhang**[1], **Lili Mou**[3,4], **Rongyu Cao**[2], **Yingwei Ma**[2], **Jue Chen**[2], **Binhua Li**[2], **Zhi Jin**[1], **Fei Huang**[2], **Yongbin Li**[2], **Ge Li**[1]
[1] School of Computer Science, Peking University [2] Tongyi Lab, Alibaba Group
[3] Department of Computing Science, University of Alberta [4] Canada CIFAR AI Chair
dongyh@stu.pku.edu.cn lige@pku.edu.cn

Abstract

Reinforcement Learning with Verifiable Reward (RLVR) has significantly advanced the complex reasoning abilities of Large Language Models (LLMs). However, it struggles to break through the inherent capability boundaries of the base LLM, due to its essentially on-policy strategy coupled with LLM's immense action space and sparse reward. Critically, RLVR can lead to the capability boundary collapse, narrowing the LLM's problem-solving scope. To address this problem, we propose RL-PLUS, a novel hybrid-policy optimization approach for LLMs that synergizes internal exploitation with external data to achieve stronger reasoning capabilities and surpass the boundaries of base models. RL-PLUS integrates two core components, i.e., Multiple Importance Sampling to address distributional mismatch from external data, and Exploration-Based Advantage Function to guide the model towards high-value, unexplored reasoning paths. We provide both theoretical analysis and extensive experiments to demonstrate the superiority and generalizability of our approach. Compared with existing RLVR methods, RL-PLUS achieves 1) state-of-the-art performance on six math reasoning benchmarks; 2) superior performance on six out-of-distribution reasoning tasks; 3) consistent and significant gains across diverse model families, with average relative improvements up to 69.2%. Moreover, the analysis of Pass@k curves indicates that RL-PLUS effectively resolves the capability boundary collapse problem.[1]

1 Introduction

The paradigm of Reinforcement Learning with Verifiable Reward (RLVR) has significantly propelled the improvement of reasoning performance in Large Language Models (LLMs) (OpenAI, 2024; Guo et al., 2025; KimiTeam, 2025), particularly in solving complex tasks involving math and coding (Chen et al., 2021; Jiang et al., 2024; Dong et al., 2024). RLVR optimizes LLMs' performance via a reinforcement learning (RL) process guided by verifiable reward computation, e.g., determining whether an output matches a ground-truth math answer or passes unit tests for coding. This method enables LLMs to scale their computation at test time by extending Chain-of-Thought (CoT) processes and spontaneously exhibit sophisticated cognitive behaviors such as reflection and exploration. Thus, RLVR is believed to be a promising way for LLMs to achieve continuous self-evolution toward more powerful AI (Guo et al., 2025).

Despite the empirical successes, some work (Havrilla et al., 2024; Shao et al., 2024; Yue et al., 2025a) points out that current RLVR cannot enable LLMs to acquire novel reasoning abilities, but rather simply utilize reasoning patterns already in the base model. As shown in Figure 1(a), although the pass@1 performance of RLVR-trained models surpasses that of the base model, its pass@128[2] is substantially lower. This trend suggests that the underlying capability distribution of the base model is broader and that existing RLVR can collapse the base model's capability boundary, thus fundamentally limiting the acquisition of new reasoning pathways.

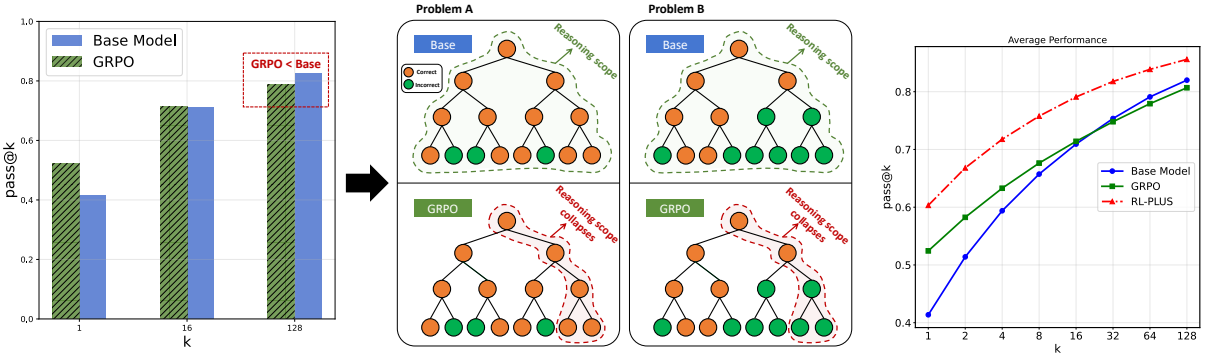
This limitation stems from an essential challenge when applying RLVR to LLMs: the potential solution space of LLMs is extremely immense with sparse reward that current RLVR techniques cannot effectively guide the model to explore new and unknown pathways, i.e., outward exploration. The challenge is particularly acute in long reasoning tasks where rewards are contingent upon the suc-

[0]Work done during Yihong Dong and Xue Jiang's internship at Tongyi Lab.

[1]https://github.com/YihongDong/RL-PLUS.

[2]The pass@k calculates the proportion of problems the model can potentially solve within a finite (k) number of attempts metric is commonly used to gauge a model's capability boundary.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 43029–43053
July 2-7, 2026



(a) Collapse problem of capability boundaries in LLMs after RLVR training. (b) Benefits of our RL-PLUS.

Figure 1: (a) The commonly used RLVR methods can lead to the collapse problem of capability boundaries in base LLMs. (b) RL-PLUS can overcome capability boundary collapse of LLMs in RLVR, consistently showing larger pass@k than base model.

cessful completion of an entire inferential chain. A single erroneous step can nullify the reward for the entire trajectory, thus failing to provide a positive signal for acquiring new knowledge. Consequently, the model is compelled to focus on inward exploitation, meaning that it refines and optimizes the knowledge and reasoning methods it already possesses, which results in a contraction of the model's exploratory range and a shrinking of its capabilities. This phenomenon not only prevents the model from acquiring new information or abilities that surpass its base model, but also significantly impedes any sustained enhancement of its overall performance.

The ancient educational principle that *"If one learns from others but does not think, one will be bewildered. If, on the other hand, one thinks but does not learn from others, one will be in peril"*[3] offers a crucial lens through which to view the limitations of current methodologies for enhancing LLM reasoning. Current RLVR can be viewed as the latter case, which excels at "thinking" through inward exploitation but demonstrates inadequate outward exploration due to its inherently on-policy strategy coupled with LLM's immense action space and sparse reward, i.e., hard to continuous "learning" of new knowledge. Conversely, approaches like Supervised Fine-Tuning (SFT) represent the former case, focusing on imitating solutions but failing to internalize the underlying reasoning principles, leading to brittleness when encountering novel problems.

This motivates us to develop novel RLVR approaches with effective external learning, but there are two key challenges that need to be addressed. First, a distributional mismatch between the model's policy and the external data source is inevitable. Standard importance sampling corrections for RL are inadequate, i.e., employing the proxy with on-policy introduces systematic bias, whereas direct using off-policy usually suffers from high variance and bias due to their significantly divergent distributions. Second, there is a challenge of efficiently extracting valuable information from this external data. Models are naturally inclined to favor high-probability tokens, thus reinforcing existing knowledge. However, the key to discovering novel reasoning often lies in exploring low-probability tokens that the model would otherwise ignore.

In this paper, we propose RL-PLUS, a novel hybrid-policy optimization approach designed to synergize internal exploitation with external data during RL process. Specifically, RL-PLUS has two core techniques. ❶ To resolve the issue of distributional mismatch, we employ Multiple Importance Sampling, which provides a lower bias and variance estimation of importance by combining information from multiple policies. ❷ To promote the discovery of new knowledge, we introduce an Exploration-Based Advantage Function, which reshapes the learning objective by prioritizing advantages for reasoning paths that are correct but are hard to explore (i.e., low probability) under the current policy. We also provide a theoretical analysis demonstrating that our approach achieves lower bias and variance compared with mainstream RLVR methods when leveraging external data.

Extensive experiments show the effectiveness

[3] A principle from the philosopher and educator Confucius.

and generalization of RL-PLUS. On six challenging math reasoning benchmarks, RL-PLUS achieves state-of-the-art (SOTA) performance, outperforming existing RLVR methods and improving upon SFT+GRPO by 5.2 average points. RL-PLUS also demonstrates superior generalization to six out-of-distribution (OOD) tasks. RL-PLUS exhibits clear and stable improvements across diverse model families, with the average relative improvements of GRPO up to 69.2%. Moreover, the analysis of Pass@k curves across multiple benchmarks indicates that RL-PLUS effectively transcends the inherent capability ceiling of the base model, thus addressing capability boundary collapse observed in prior RLVR approaches.

2 Background and Related Work

In this section, we first establish the theoretical preliminaries necessary to understand our work, and then provide a critical review of the most related work, identifying key limitations in existing methods and thereby motivating the design of our proposed RL-PLUS.

2.1 Preliminary Knowledge

LLM-based Reasoning as a Markov Decision Process. We frame the task of generating a reasoning sequence (e.g., a solution to a math problem) as a Markov Decision Process (MDP) (Puterman, 2014). At each timestep t, the state s_t consists of the initial prompt q concatenated with the sequence of previously generated tokens, $y_{<t}$. The action a_t is the selection of the next token y_t from the vocabulary. The model, or policy π_θ, maps a state to a distribution over actions. A reward $R(q, y)$ is provided only upon completion of the entire sequence y. In the context of RLVR, this reward is typically sparse and binary. For example, a score is 1 if the final answer is correct and 0 otherwise. The objective is to learn a policy π_θ that maximizes the expected cumulative reward $J(\theta) = \mathbb{E}_{y\sim\pi_\theta}[R(q, y)]$.

Policy Gradient Optimization. Policy gradient methods are the standard for optimizing LLMs in on-policy RLVR settings. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) shows exceptional performance in various tasks, especially to enable effective scaling within the RLVR paradigm. Compared to Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017), GRPO leverages group-normalized rewards to estimate advantages, eliminating the need for a value model and thereby improving computational efficiency. The standard GRPO objective is:

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{(q,y)\sim\mathcal{D}_{\text{on}}}\Bigg[\sum_{t=1}^{|y|}\min\left(r_{i,t}(\theta)A_i,\right. \\ \left.\text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)A_i\right)\Bigg] - \beta\mathcal{D}_{\text{KL}} \tag{1}$$

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \tag{2}$$

$$A_i = \frac{R_i - \text{mean}(\{R_1, R_2, \ldots, R_G\})}{\text{std}(\{R_1, R_2, \ldots, R_G\})}, \tag{3}$$

where $r_{i,t}(\theta)$ is the importance sampling ratio and A_i is the estimated advantage for an on-policy trajectory y_i. Recent work, such as Simple-rl (Zeng et al., 2025) and DAPO (Yu et al., 2025), has proposed either setting the KL coefficient β to a very small value or omitting the KL term in Equation 1 entirely. The rationale is that during the training of a model for long CoT reasoning, the model's distribution is expected to diverge significantly from the initial policy, rendering this constraint unnecessary.

Evaluating Reasoning Boundaries with pass@k. To accurately assess a model's true problem-solving capabilities, we utilize the pass@k metric (Chen et al., 2021; Dong et al., 2025a). It measures the probability of obtaining at least one correct answer within k independent samples for a given problem. Unlike mean accuracy (i.e., pass@1), pass@k provides a more comprehensive view of the model's reasoning potential and is critical for evaluating whether a method expands the set of solvable problems (Yue et al., 2025a).

This on-policy RLVR paradigm, while powerful, leads to two fundamental challenges when the goal is to surpass a base model's intrinsic capabilities: 1) an inability to effectively integrate novel, external knowledge due to the high variance and bias associated with off-policy data, and 2) the tendency for on-policy exploration to collapse into known, high-probability reasoning paths, thereby shrinking the model's reasoning boundary. These challenges directly motivate our approach.

2.2 Related Work

We position RL-PLUS by critically examining two primary lines of research: on-policy RLVR for reasoning and hybrid SFT-RL methods.

On-Policy RLVR and Its Intrinsic Limitations Reinforcement learning has become a cornerstone for enhancing LLM reasoning (Yue et al., 2025b; Liu et al., 2025c; Wang et al., 2025; Jiang et al., 2025; Dong et al., 2025b; Jiang et al., 2026). Seminal works have shown that RLVR can significantly improve performance on complex reasoning tasks by rewarding correct final answers (Guo et al., 2025; Zeng et al., 2025; Hu et al., 2025). Subsequent research has refined this paradigm; for instance, PRIME-Zero (Cui et al., 2025a) uses implicit process rewards, and Oat-Zero (Liu et al., 2025b) simplifies the advantage calculation in GRPO.

However, a growing body of evidence reveals a critical flaw in these on-policy methods: they primarily optimize existing knowledge rather than discovering new reasoning capabilities. This leads to two well-documented issues. First is the Capability Boundary Collapse problem. While RLVR models often show superior pass@1 performance, their advantage diminishes as k increases in pass@k evaluations, with base models eventually surpassing them (Yue et al., 2025a). This strongly suggests that RLVR refines the probability of known correct paths but fails to expand the overall set of solvable problems. Second, these methods suffer from Entropy Collapse, where policy entropy sharply decreases during training, making the model overly deterministic and hindering further exploration (Cui et al., 2025b). This indicates that on-policy RLVR, by its nature, is prone to inward exploitation that reinforces existing biases and limits the model's potential.

Hybrid SFT-RL Methods To overcome knowledge limitations of pure RL, researchers explored hybrid methods that combine RL with SFT on external demonstration data (Cai et al., 2025). Early approaches employed sequential, multi-stage training (SFT then RL), as seen in models like InstructGPT (Ouyang et al., 2022). While conceptually simple, this often leads to catastrophic forgetting of the SFT-learned knowledge and suffers from computational inefficiency.

More recent work has focused on unified or interleaved training frameworks. For example, ReLIFT (Ma et al., 2025) alternates between RL and online fine-tuning on difficult problems, while LUFFY (Yan et al., 2025) selectively imitates high-quality external trajectories using a mixed policy. In another example, TAPO (Wu et al., 2025) enhances RL by integrating external, high-level guidance in the form of "thought patterns" abstracted from prior data. Other methods, such as SASR (Chen et al., 2025) and SuperRL (Liu et al., 2025a), employ adaptive switches to dynamically balance SFT and RL objectives based on training state. While these methods are more sophisticated, they often rely on complex, potentially unstable heuristics for balancing the two learning signals. Moreover, simply adding an SFT loss to the RL objective, as explored in "GRPO w/ SFT Loss", can degrade performance, highlighting the difficulty of effective integration. Even advanced frameworks like UFT (Wang et al., 2024b), which aim to unify SFT and RL to accelerate convergence, do not explicitly address how to stabilize off-policy updates while simultaneously directing exploration towards novel solutions.

Motivation The foregoing analysis reveals persistent gaps in the related work. On-policy RLVR methods are constrained by the base model's inherent knowledge, while existing hybrid SFT-RL methods lack a principled mechanism to both stabilize learning from external, off-policy data and explicitly incentivize exploration of low-probability but correct reasoning pathways. RL-PLUS is designed to directly address these deficiencies.

3 RL-PLUS

RL-PLUS overcomes the LLM's capability boundaries collapse problem in RLVR by integrating externally-guided exploration with the exploitation of internal reasoning pathways.

3.1 Mitigating Distributional Mismatch with Multiple Importance Sampling

A central challenge in learning from a static dataset $\mathcal{D}_e = \{e_i\}_{i=1}^{N}$ is the distributional shift between the target policy π_θ and the unknown behavior policy π_ω. Standard importance sampling (IS) presents a dilemma for correcting this mismatch. On-policy IS estimator, which uses a proxy like $\pi_{\theta_{\text{old}}}$ in the denominator, is systematically biased when applied to external data from π_ω (**Lemma A.5**). Conversely, the theoretically correct off-policy estimator, using weights $r_t^e(\theta) = \frac{\pi_\theta(e_t|e_{<t})}{\pi_\omega(e_t|e_{<t})}$, suffers from support mismatch of π_θ (**Lemma A.6**) and prohibitively high variance as the policies diverge (**Lemma A.7**), which destabilizes training. This issue is compounded by the fact that π_ω is usually unknown, rendering direct weight computation infeasible.

To solve this, we introduce Multiple Importance Sampling to construct an estimator with lower variance and controllable bias. Instead of directly estimating π_ω, we treat the generation of an external sample as arising from a mixture policy composed of the previous policy $\pi_{\theta_{old}}$ and the external policy π_ω. Therefore, the Multiple Importance Sampling of each token can be defined as:

$$r_{i,t}^m(\theta) = \frac{2\pi_\theta(e_{i,t}|q, e_{i,<t})}{\pi_\omega(e_{i,t}|q, e_{i,<t}) + \pi_{\theta_{old}}(e_{i,t}|q, e_{i,<t})}, \tag{4}$$

where $e_{i,t}$ is the t-th token in the external data trajectory e_i. It replaces the aforementioned *explosive bias from poor proxy or support mismatch* with a controlled, bounded distortion error (**Remarks A.8 and A.9**), making the overall MIS estimator robust for stable learning from external data. The formal denominator acts as a crucial *variance guardrail*. The presence of $\pi_{\theta_{\text{old}}}$, which is intentionally kept close to π_θ, prevents the ratio from exploding even if π_ω is highly dissimilar, ensuring the estimator's variance remains bounded.

Theorem 3.1 (Variance Robustness of MIS)**.** *So long as there is at least one policy in the behavior pool $\{\pi_{\beta_k}\}$ (e.g., $\pi_{\beta_k^*}$) that is a good approximation of the target policy π_θ (i.e., $\pi_{\beta_k^*} \approx \pi_\theta$), the variance of the MIS estimator will be low. The estimator is insensitive to other arbitrarily "bad" behavior policies in the pool. (See Proof in Appendix A.4)*

A key challenge remains: the behavior policy π_ω is unknown. We require a robust method to estimate it. Instead of naively using a proxy, we derive an estimator for π_ω from a principled Bayesian perspective. We frame the estimation as a decision problem where we must balance our belief in our best available model, $\pi_{\theta_{\text{old}}}$, against a state of maximal uncertainty, represented by a non-informative uniform policy $\mathcal{U}$. This allows us to hedge against model error, leading to the following Bayes-optimal estimator.

Theorem 3.2 (Bayes-Optimal Policy Estimator)**.** *Let the model space for the unknown behavior policy π_ω be composed of two candidate models: 1) The specific proxy policy, $\pi_{\theta_{old}}$, representing our available, specific information. 2) A non-informative uniform policy, $\mathcal{U}(\tau)$, representing maximal uncertainty. Let the trajectory space $\mathcal{T}$ have a finite volume $V = \int_{\mathcal{T}} d\tau$, such that $\mathcal{U}(\tau) = 1/V$. Under the Principle of Indifference, we assign equal prior probabilities to these models, i.e., $P(\pi_\omega = \pi_{\theta_{old}}) = P(\pi_\omega = \mathcal{U}) = 1/2$. Then, the estimator $\hat{\pi}_\omega$ that minimizes the Bayes risk (expected L2 error) is the Bayesian model average: $\hat{\pi}_\omega^*(\tau) = \frac{1}{2}\pi_{\theta_{old}}(\tau) + \frac{1}{2}\mathcal{U}(\tau)$ (See Proof in Appendix A.5)*

3.2 Efficient Exploration with Exploration-Based Advantage Function

Merely incorporating external data stably is insufficient; we must also guide the model to focus on its most valuable information, especially the "new knowledge" that the model is unlikely to discover on its own. Models tend to favor high-probability tokens, whereas novel knowledge is often embedded in correct reasoning paths that the model considers to have low probability.

To this end, we design an Exploration-Based Advantage Function, $A_{i,t}^c$, which prioritizes encouraging the model to explore reasoning steps that are correct but hard to explore, defined as:

$$A_{i,t}^c = \frac{R_i - \text{mean}(\{R_1, R_2, \ldots, R_G\})}{\text{std}(\{R_1, R_2, \ldots, R_G\})} \cdot C_{i,t} \tag{5}$$

The first term is the standardized reward for all trajectories, including both internal exploration and external data, and the second term is the weight to encourage exploration. Inspired by focal loss (Lin et al., 2017), we define the weight $C_{i,t}$ as:

$$C_{i,t} = (1 - \text{detach}(\pi_\theta(e_{i,t}|q, e_{i,<t})))^\gamma, \tag{6}$$

where $\pi_\theta(e_{i,t}|q, e_{i,<t})$ represents the model's exploration probability in the correct token $e_{i,t}$ from the external data. When it is hard to explore (i.e., π_θ is small), the weight $C_{i,t}$ becomes large, amplifying the advantage signal for that timestep and compelling the model to attend to this overlooked region. γ is a hyperparameter to control $C_{i,t}$. The 'detach' function is a standard operation in Torch that prevents gradients from backpropagating through the probability calculation, which enhances training stability.

3.3 The Composite RL-PLUS Objective

To synergize internal exploitation $\mathcal{D}_o$ with external data $\mathcal{D}_e$, we formulate the final training objective of RL-PLUS as a composite function $\mathcal{J}_{\text{RL-PLUS}}(\theta)$:

Table 1: Performance of RL-PLUS against eight concurrent RLVR methods and four straightforward baselines.

Method	AIME24	AIME25	AMC	MATH-500	Minerva	Olympiad	Avg.
Qwen2.5-Math-7B	11.5	4.9	31.3	43.6	7.4	15.6	19.0
SimpleRL (Zeng et al., 2025)	27.0	6.8	54.9	76.0	25.0	34.7	37.4
OpenReasoner (Hu et al., 2025)	16.5	15.0	52.1	82.4	33.1	47.1	41.0
PRIME (Cui et al., 2025a)	17.0	12.8	54.0	81.4	39.0	40.3	40.7
Oat (Liu et al., 2025b)	33.4	11.9	61.2	78.0	34.6	43.4	43.7
DAPO (Yu et al., 2025)	23.4	15.5	66.3	86.0	40.1	49.6	46.8
TAPO (Wu et al., 2025)	33.3	18.6	77.5	83.4	38.2	46.2	49.5
LUFFY (Yan et al., 2025)	29.4	23.1	65.6	87.6	37.5	57.2	50.1
ReLIFT (Ma et al., 2025)	28.4	21.8	64.3	86.8	40.1	54.8	49.4
SFT	22.2	22.3	52.8	82.6	40.8	43.7	44.1
GRPO (Shao et al., 2024)	25.1	15.3	62.0	84.4	39.3	46.8	45.5
GRPO w/ SFT Loss	19.5	16.4	49.7	80.4	34.9	39.4	40.1
SFT+GRPO	25.8	23.1	62.7	87.2	39.7	50.4	48.2
RL-PLUS	**33.4**	**25.9**	**68.1**	**90.2**	**43.8**	**58.8**	**53.4**

Table 2: Out-of-Distribution performance on programming tasks (HumanEval, LeetCode, LiveCodeBench) and science QA (ARC-c, GPQA-diamond, MMLU-Pro).

Method	HumanEval	LeetCode	LiveCodeBench	ARC-c	GPQA-diamond	MMLU-Pro	Avg.
Base Model	42.1	22.8	14.9	18.2	13.1	30.2	23.6
SFT	55.5	8.3	8.1	75.2	24.7	42.7	35.8
GRPO	63.4	21.1	15.3	81.7	40.4	47.5	44.9
SFT+GRPO	59.8	8.34	9.7	72.4	24.2	37.7	35.4
RL-PLUS	**68.3**	**27.8**	**19.2**	**82.3**	**40.4**	**54.7**	**48.8**

$$\mathcal{J}_{\text{RL-PLUS}}(\theta) = \underbrace{\mathbb{E}_{(o_i,A_i)\sim\mathcal{D}_o}\left[r_{i,t}(\theta)A_i\right]}_{\text{Internal Exploitation (Thinking)}} + \underbrace{\mathbb{E}_{(e_i,A^c_{i,t})\sim\mathcal{D}_e}\left[r^m_{i,t}(\theta)A^c_{i,t}\right]}_{\text{External data for Exploration (Learning)}} \quad (7)$$

where the first term represents the standard policy gradient objective, which is responsible for stabilizing and improving upon the model's existing reasoning capabilities. The second term constitutes the core of our contribution, which drives the policy to external exploration. It leverages our two primary innovations: 1) Multiple Importance Sampling $r^m_{i,t}(\theta)$, which provides a low-variance, robust mechanism for integrating external data, and 2) Exploration-Based Advantage Function $A^c_{i,t}$, which re-weights the learning signal to prioritize novel yet high-value reasoning paths.

Moreover, we omit the clipping mechanism (e.g., $\text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)$), which would suppress the gradient signals corresponding to highly informative, low-probability events, i.e., the "new knowledge" we aim to acquire. By removing this constraint, RL-PLUS is empowered to take larger, more assertive optimization steps when it encounters valuable information in the external data, thus accelerating the assimilation of novel knowledge and more effectively expanding its capability boundaries in RLVR.

4 Experimental Results

In this section, we conduct extensive experiments to demonstrate the effectiveness and generalization of RL-PLUS. Detailed setup and additional experiments can be found in Appendix D - G.

Performence of RL-PLUS. Table 1 shows RL-PLUS achieves SOTA performance, comprehensively outperforming existing RLVR methods. SFT learns from external knowledge while GRPO enables self-exploration via reinforcement learning. Their combination, i.e., SFT+GRPO, yields synergistic gains, but simply adding SFT loss to RL, i.e., GRPO w/ SFT Loss, degrades performance, indicating that effective integration is non-trivial. RL-PLUS improves upon SFT+GRPO by +5.2 points on average. Compared to concurrent methods like LUFFY (which treats external data as perfect expert trajectories, introducing bias) and ReLIFT (which alternates RL and SFT training, risking knowledge loss), RL-PLUS demonstrates superior performance and more effective external knowledge integration.

Table 3: The performance of RL-PLUS based on Different LLMs.

Model	AIME 24	AIME 25	AMC	MATH-500	Minerva	Olympiad	Avg.
LLaMA-3.1-8B	4.7	0.4	18.5	46.4	19.8	13.2	17.2
SFT	2.6	0.9	29.8	50.0	21.3	16.9	20.2
GRPO	3.5	0.5	19.5	45.0	20.2	14.2	17.2
RL-PLUS	**11.7**	**2.1**	**35.5**	**64.4**	**29.4**	**31.2**	**29.1**
Deepseek-Math-7B	1.1	0.3	14.5	40.4	18.8	10.7	14.3
SFT	3.8	0.3	23.3	51.2	21.3	19.8	19.9
GRPO	2.5	0.2	17.3	47.0	20.9	14.5	17.1
RL-PLUS	**4.1**	**0.4**	**25.0**	**54.8**	**21.7**	**21.4**	**21.3**
Qwen2.5-Math-1.5B	7.2	3.6	26.4	28.0	9.6	21.2	16.0
SFT	11.7	13.2	37.8	70.6	26.8	31.3	31.9
GRPO	11.8	7.7	40.2	61.8	26.8	32.0	30.1
RL-PLUS	**20.4**	**13.6**	**50.0**	**80.4**	**33.1**	**45.2**	**40.5**
Qwen2.5-Math-7B	11.5	4.9	31.3	43.6	7.4	15.6	19.0
SFT	22.2	22.3	52.8	82.6	40.8	43.7	44.1
GRPO	25.1	15.3	62.0	84.4	39.3	46.8	45.5
RL-PLUS	**33.4**	**25.9**	**68.1**	**90.2**	**43.8**	**58.8**	**53.4**

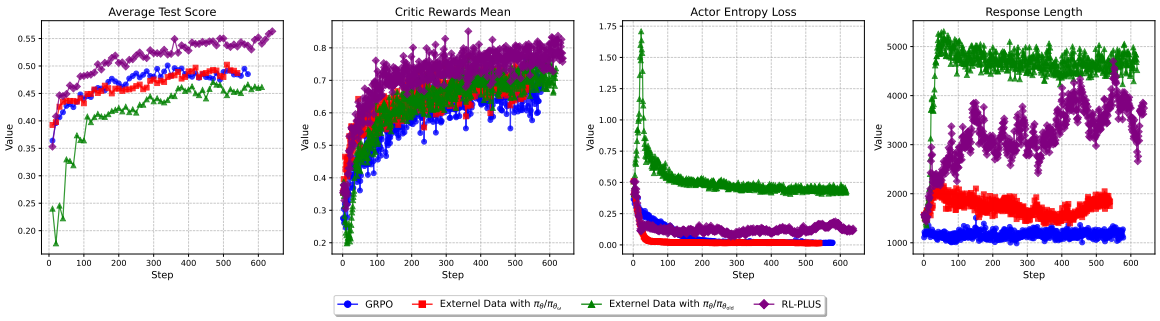


Figure 2: Training dynamics of RL-PLUS and other baselines.

Performance on OOD Tasks. Table 2 shows RL-PLUS achieves substantial improvements on OOD tasks, surpassing the next best baseline by +3.9 points on average. This demonstrates that RL-PLUS develops fundamental reasoning abilities that generalize beyond specific domains. In science QA, RL-PLUS consistently outperforms GRPO and SFT+GRPO. Under domain shift to programming tasks, RL-PLUS maintains strong performance while SFT-based methods deteriorate significantly. Combined with Table 1, a pattern emerges: SFT-based methods excel in-domain but fail to generalize in OOD scenarios. RL-PLUS resolves this trade-off by merging external knowledge acquisition with robust generalization, achieving superior performance in both settings.

Training Dynamics. Figure 2 presents training dynamics across benchmarks. RL-PLUS consistently outperforms alternatives in test accuracy and rewards, maintaining upward trends even after baselines plateau. Analyzing actor entropy, we observe that directly incorporating external data during rollouts causes "entropy explosion" and chaotic outputs, while baseline entropies collapse to zero, indicating lost exploration capability. RL-PLUS entropy remains non-zero, suggesting retained exploration capacity. Since policy performance comes at the cost of entropy (Cui et al., 2025b), RL-PLUS's non-depleted entropy indicates further improvement potential. The steadily increasing response length of RL-PLUS reflects healthy training. Directly incorporating external data also leads to long response lengths, but its low accuracy and high entropy suggest that it stems from unproductive exploration rather than meaningful reasoning.

Application on Various LLMs. We validate RL-PLUS applicability on mainstream LLMs including LLaMA-3.1-8B, Deepseek-Math-7B, and Qwen2.5-Math-1.5B/7B. Table 3 shows RL-PLUS achieves superior performance across all base models. On Qwen2.5-Math-7B, RL-PLUS reaches 53.4, significantly outperforming the base model (19.0), SFT (44.1), and GRPO (45.5). On LLaMA-3.1-8B, where GRPO struggles, RL-PLUS achieves +11.9 absolute gain. These results demonstrate RL-PLUS consistently enhances LLMs across varying

Table 4: Ablation Study of RL-PLUS.

Method	AIME 24	AIME25	AMC	MATH-500	Minerva	Olympiad	Avg.
Variants with External Data							
$\pi_\theta/\pi_{\theta_{\text{old}}}$	19.6	14.8	55.1	81.0	33.5	46.2	41.7
$\pi_\theta/\pi_{\theta_\omega}$	25.8	16.3	59.9	83.8	32.4	49.3	44.6
$\pi_\theta/\pi_{\theta_\omega}$ with Our Policy Estimation	26.1	19.2	62.3	86.8	38.6	52.0	47.5
RL-PLUS	**33.4**	**25.9**	**68.1**	**90.2**	**43.8**	**58.8**	**53.4**
- Exploration-Based Advantage Function	28.3	24.1	67.8	88.8	40.4	56.0	50.9
- Multiple Importance Sampling	25.1	15.3	62.0	84.4	39.3	46.8	45.5

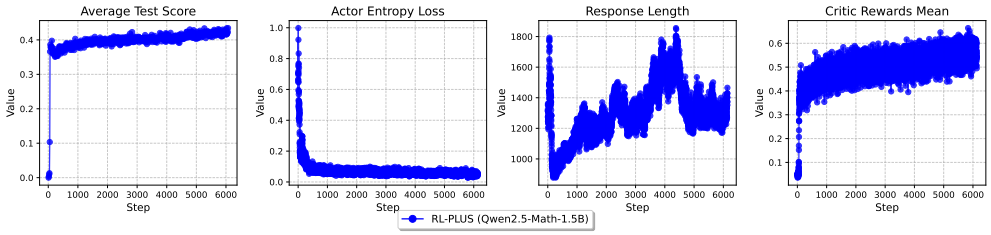


Figure 3: Training Stability of RL-PLUS with over 10 times original training steps.

architectures and scales.

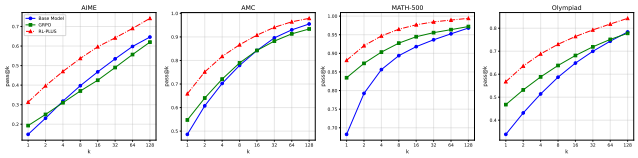


Figure 4: Pass@k curves of RL-PLUS compared with baselines across multiple benchmarks.

Acquiring Reasoning Abilities Beyond Base Model. Following work (Yue et al., 2025a), we test whether RL-PLUS acquires superior reasoning abilities beyond the base model. Figure 4 shows pass@k curves across tasks. GRPO's curve converges to the base model as k increases, sometimes dropping below it at larger k-values, consistent with work (Yue et al., 2025a). RL-PLUS maintains consistent advantage over both as k increases, demonstrating effective capability boundary expansion rather than mere optimization within inherent limits. On AMC and MATH-500, RL-PLUS accuracy plateaus near the maximum score of 1.0.

Ablation Study. Table 4 presents ablation studies analyzing RL-PLUS's effectiveness. Removing the Exploration-Based Advantage Function drops performance from 53.4 to 50.9, demonstrating the importance of efficient exploration. Removing Multiple Importance Sampling causes larger degradation to 45.5, highlighting the significance of external knowledge incorporation. We also compare against three naive integration approaches: approximating external policy π_{θ_ω} with old policy $\pi_{\theta_{\text{old}}}$, treating it as perfect oracle (probability 1, similar to LUFFY (Yan et al., 2025)), and using our policy estimation. Our policy estimation improves performance by 2.9 points over the second variant. All naive variants show significant performance gaps compared to RL-PLUS.

Training Stability of RL-PLUS. We extend training on Qwen2.5-Math-1.5B to **over 10 times** the original steps to validate RL-PLUS stability. Figure 3 shows key metrics exhibit excellent stability and continuous improvement: Average Test Score and Critic Rewards Mean display steady upward trends while Actor Entropy Loss converges to a healthy, non-zero range. This reveals an ideal balance: the model's policy, while becoming more effective (i.e., exploitation), also maintains the necessary policy stochasticity for exploration, thus avoiding premature convergence to a local optimum. These Results demonstrate RL-PLUS's outstanding stability and potential for further gains via extended training.

5 Conclusion

In this paper, we proposed RL-PLUS, a novel hybrid-policy optimization approach designed to counter the "capability boundary collapse" observed in LLMs trained with RLVR. RL-PLUS addresses this problem by synergizing external data with internal exploitation through two core components: Multiple Importance Sampling to resolve distributional mismatch from external data, and

Exploration-Based Advantage Function to incentivize the discovery of correct yet low-probability reasoning paths. We provide both theoretical analysis and extensive experiments to demonstrate the superiority and generalizability of RL-PLUS. Notably, Pass@k curves and training dynamics demonstrate that our method breaks through the reasoning capability boundary of base model, leading to further performance improvements.

6 Limitations

Our work has the following two main limitations, specifically:

First, we only demonstrate the effectiveness of RL-PLUS on some mainstream reasoning tasks, including mathematical, programming, and science QA, in this paper. Its effectiveness across a broader spectrum of complex, real-world scenarios remains to be fully explored. We aim to further broaden the applicability of RL-PLUS in our future work.

Second, due to limited computational resources, our current experiments are primarily conducted on base models with around 7B parameters. However, since RL-PLUS is a model-agnostic algorithm, it can be seamlessly scaled to models with larger parameters.

7 Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grant No. 62192733, 62192730, 62192731, the National Key R&D Program under Grant No. 2023YFB4503801, and the Beijing Major Science and Technology Project under Contract No. Z251100008425005.

References

Hongyi James Cai, Junlin Wang, Xiaoyin Chen, and Bhuwan Dhingra. 2025. How much backtracking is enough? exploring the interplay of sft and rl in enhancing llm reasoning. *arXiv preprint arXiv:2505.24273*.

Jack Chen, Fazhong Liu, Naruto Liu, Yuhan Luo, Erqu Qin, Harry Zheng, Tian Dong, Haojin Zhu, Yan Meng, and Xiao Wang. 2025. Step-wise adaptive integration of supervised fine-tuning and reinforcement learning for task-specific llms. *arXiv preprint arXiv:2505.13026*.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

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

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

Yihong Dong, Jiazheng Ding, Xue Jiang, Ge Li, Zhuo Li, and Zhi Jin. 2025a. Codescore: Evaluating code generation by learning code execution. *ACM Trans. Softw. Eng. Methodol.*, 34(3):77:1–77:22.

Yihong Dong, Xue Jiang, Zhi Jin, and Ge Li. 2024. Self-collaboration code generation via chatgpt. *ACM Trans. Softw. Eng. Methodol.*, 33(7):189:1–189:38.

Yihong Dong, Xue Jiang, Jiaru Qian, Tian Wang, Kechi Zhang, Zhi Jin, and Ge Li. 2025b. A survey on code generation with llm-based agents. *CoRR*, abs/2508.00083.

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

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, et al. 2024. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.

Alexander Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, et al. 2024. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting AGI with

olympiad-level bilingual multimodal scientific problems. In *ACL (1)*, pages 3828–3850. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks*.

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

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Xue Jiang, Yihong Dong, Mengyang Liu, Hongyi Deng, Tian Wang, Yongding Tao, Rongyu Cao, Binhua Li, Zhi Jin, Wenpin Jiao, Fei Huang, Yongbin Li, and Ge Li. 2025. Coderl+: Improving code generation via reinforcement with execution semantics alignment. *CoRR*, abs/2510.18471.

Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. 2024. Self-planning code generation with large language models. *ACM Trans. Softw. Eng. Methodol.*, 33(7):182:1–182:30.

Xue Jiang, Tianyu Zhang, Ge Li, Mengyang Liu, Taozhi Chen, Zhenhua Xu, Binhua Li, Wenpin Jiao, Zhi Jin, Yongbin Li, and Yihong Dong. 2026. Think anywhere in code generation. *CoRR*, abs/2603.29957.

KimiTeam. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *NeurIPS*.

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

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *ICCV*, pages 2999–3007. IEEE Computer Society.

Yihao Liu, Shuocheng Li, Lang Cao, Yuhang Xie, Mengyu Zhou, Haoyu Dong, Xiaojun Ma, Shi Han, and Dongmei Zhang. 2025a. Superrl: Reinforcement learning with supervision to boost language model reasoning. *arXiv preprint arXiv:2506.01096*.

Zichen Liu, Changyu Chen, Wenjun Li, Tianyu Pang, Chao Du, and Min Lin. 2025b. There may not be aha moment in r1-zero-like training — a pilot study. https://oatllm.notion.site/oat-zero. Notion Blog.

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

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Bin Cui, et al. 2025. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. *arXiv preprint arXiv:2506.07527*.

OpenAI. 2024. Openai o1 system card and model report. Technical report, OpenAI o1 series, available online.

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

Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

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

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. 2025. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024a. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Zhichao Wang, Bin Bi, Zixu Zhu, Xiangbo Mao, Jun Wang, and Shiyu Wang. 2024b. Uft: Unifying fine-tuning of sft and rlhf/dpo/una through a generalized implicit reward function. *arXiv preprint arXiv:2410.21438*.

Jinyang Wu, Chonghua Liao, Mingkuan Feng, Shuai Zhang, Zhengqi Wen, Pengpeng Shao, Huazhe Xu, and Jianhua Tao. 2025. Thought-augmented policy optimization: Bridging external guidance and internal capabilities. *arXiv preprint arXiv:2505.15692*.

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

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

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

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025a. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *CoRR*, abs/2504.13837.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. 2025b. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

A Theoretical Analysis of Multiple Importance Sampling

We provide a rigorous theoretical analysis of the Multiple Importance Sampling (MIS) estimator for policy optimization. First, we dissect the bias and variance issues inherent to standard Importance Sampling (IS) when using data from a single behavior policy. Subsequently, we prove that the MIS estimator is unbiased and analyze its superior variance properties. We show that MIS is robust to the inclusion of suboptimal behavior policies, establishing it as a powerful tool for integrating diverse data sources in policy optimization.

A.1 Preliminaries and Core Assumptions

Our analysis is based on the following standard settings and assumptions. Let the objective function be $J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)]$, where τ represents a complete trajectory, $R(\tau)$ is its corresponding cumulative return, and π_θ is the target policy we aim to optimize.

Assumption A.1 (Joint Support Coverage)**.** *The support of the target policy π_θ is covered by the union of the supports of all behavior policies $\{\pi_{\beta_k}\}_{k=1}^K$. Formally,*

$$supp(\pi_\theta) \subseteq \bigcup_{k=1}^{K} supp(\pi_{\beta_k})$$

This assumption ensures that any trajectory possible under π_θ can be sampled with a non-zero probability by at least one behavior policy.

Assumption A.2 (Bounded Rewards)**.** *The trajectory returns are bounded, i.e., for all trajectories τ, there exists a constant $R_{\max}$ such that $|R(\tau)| \leq R_{\max} < \infty$. This ensures that all expectations and variances are well-defined.*

A.2 Analysis of Bias and Variance in Single-Strategy Importance Sampling

When learning from data generated by a single external behavior policy π_ω, the standard IS estimator can suffer from bias and variance problems. We analyze three primary failure modes.

A.2.1 Importance Sampling Estimators

We formally define the estimators central to our analysis. We consider a dataset of N trajectories.

Definition A.3 (Standard Importance Sampling (IS) Estimator)**.** When all data is sampled from a single behavior policy π_ω (i.e., $K = 1, \pi_{\beta_1} = \pi_\omega$), the standard IS estimator for $J(\theta)$ is:

$$\hat{J}_{\text{IS}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{\pi_\theta(\tau_i)}{\pi_\omega(\tau_i)} R(\tau_i), \quad \text{where } \tau_i \sim \pi_\omega$$

Definition A.4 (Proxy IS Estimator)**.** A biased variant of the IS estimator that uses a proxy policy $\pi_{\theta_{\text{old}}}$ in the denominator, while the data is sampled from a different policy π_ω:

$$\hat{J}_{\text{proxy}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{\pi_\theta(\tau_i)}{\pi_{\theta_{\text{old}}}(\tau_i)} R(\tau_i), \quad \text{where } \tau_i \sim \pi_\omega$$

A.2.2 Bias from a Proxy

In practice, to mitigate the high variance that occurs when the data-generating policy π_ω is far from the target policy π_θ, one might be tempted to use a different policy, $\pi_{\theta_{\text{old}}}$, as the denominator for the importance ratio. This "proxy" policy is chosen to be closer to π_θ (e.g., a previous iterate of the policy). However, this introduces a systematic bias, as it violates the fundamental principle of importance sampling.

Lemma A.5 (Bias of the IS Estimator with a Proxy)**.** *Assume trajectory data τ_i is sampled from an external policy π_ω, i.e., $\tau_i \sim \pi_\omega$. If we construct an estimator using a proxy policy $\pi_{\theta_{old}}$ in the denominator of the importance weight:*

$$\hat{J}_{proxy}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{\pi_\theta(\tau_i)}{\pi_{\theta_{old}}(\tau_i)} R(\tau_i)$$

*then this estimator is **biased** for the true objective $J(\theta)$ whenever the proxy policy $\pi_{\theta_{old}}$ is not identical to the true sampling policy π_ω. The bias is given by:*

$$\mathcal{B}(\theta, \omega, \theta_{old}) \triangleq \mathbb{E}_{\pi_\omega}[\hat{J}_{proxy}(\theta)] - J(\theta) = \int \pi_\theta(\tau) R(\tau) \left(\frac{\pi_\omega(\tau)}{\pi_{\theta_{old}}(\tau)} - 1 \right) d\tau \tag{8}$$

Proof. We compute the expectation of the proxy estimator $\hat{J}_{\text{proxy}}(\theta)$ under the true data distribution π_ω. The expectation is taken with respect to $\tau \sim \pi_\omega$.

$$\begin{aligned}
\mathbb{E}_{\pi_\omega}[\hat{J}_{\text{proxy}}(\theta)] &= \mathbb{E}_{\tau \sim \pi_\omega} \left[\frac{\pi_\theta(\tau)}{\pi_{\theta_{\text{old}}}(\tau)} R(\tau) \right] \\
&= \int \pi_\omega(\tau) \frac{\pi_\theta(\tau)}{\pi_{\theta_{\text{old}}}(\tau)} R(\tau) d\tau
\end{aligned}$$

This is the expected value that the estimator will yield. Crucially, because the sampling distribution $\pi_\omega(\tau)$ in the integral does not cancel with the denominator $\pi_{\theta_{\text{old}}}(\tau)$, this expression cannot be simplified to the true objective $J(\theta) = \int \pi_\theta(\tau) R(\tau) d\tau$.

The bias of this estimator is its expectation minus the true objective:

$$\begin{aligned}
\mathcal{B}(\theta, \omega, \theta_{\text{old}}) &= \mathbb{E}_{\pi_\omega}[\hat{J}_{\text{proxy}}(\theta)] - J(\theta) \\
&= \int \pi_\omega(\tau) \frac{\pi_\theta(\tau)}{\pi_{\theta_{\text{old}}}(\tau)} R(\tau) d\tau - \int \pi_\theta(\tau) R(\tau) d\tau \\
&= \int \left(\pi_\omega(\tau) \frac{\pi_\theta(\tau)}{\pi_{\theta_{\text{old}}}(\tau)} R(\tau) - \pi_\theta(\tau) R(\tau) \right) d\tau \\
&= \int \pi_\theta(\tau) R(\tau) \left(\frac{\pi_\omega(\tau)}{\pi_{\theta_{\text{old}}}(\tau)} - 1 \right) d\tau
\end{aligned}$$

The final expression for the bias is zero if and only if $\pi_\omega(\tau) = \pi_{\theta_{\text{old}}}(\tau)$ for all relevant trajectories. If the external data policy π_ω differs significantly from the proxy policy $\pi_{\theta_{\text{old}}}$, this ratio will deviate substantially from 1, leading to a large, systematic bias. □

A.2.3 Bias from Support Mismatch

Even when using the correct data-generating policy π_ω in the denominator, the standard IS estimator is biased if the support of π_ω does not fully cover the support of the target policy π_θ.

Lemma A.6 (Bias of the Standard IS Estimator from Support Mismatch)**.** *When using data sampled from an external policy π_ω to estimate the objective $J(\theta)$, if the support condition $supp(\pi_\theta) \not\subseteq supp(\pi_\omega)$ is not met, the standard IS estimator $\hat{J}_{IS}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{\pi_\theta(\tau_i)}{\pi_\omega(\tau_i)} R(\tau_i)$ (where $\tau_i \sim \pi_\omega$) is biased. The bias relative to the true objective is:*

$$\mathcal{B}(\theta, \omega) \triangleq \mathbb{E}_{\pi_\omega}[\hat{J}_{IS}(\theta)] - J(\theta) = - \int_{\tau \in supp(\pi_\theta) \setminus supp(\pi_\omega)} \pi_\theta(\tau) R(\tau) d\tau \tag{9}$$

Proof. The expectation of the IS estimator is calculated as follows:

$$\begin{aligned}
\mathbb{E}_{\pi_\omega}[\hat{J}_{\text{IS}}(\theta)] &= \mathbb{E}_{\tau \sim \pi_\omega} \left[\frac{\pi_\theta(\tau)}{\pi_\omega(\tau)} R(\tau) \right] \\
&= \int_{\tau \in \text{supp}(\pi_\omega)} \pi_\omega(\tau) \frac{\pi_\theta(\tau)}{\pi_\omega(\tau)} R(\tau) d\tau \\
&= \int_{\tau \in \text{supp}(\pi_\omega) \cap \text{supp}(\pi_\theta)} \pi_\theta(\tau) R(\tau) d\tau
\end{aligned}$$

The true objective $J(\theta)$ can be decomposed over the same domains:

$$\begin{aligned} J(\theta) &= \int_{\tau \in \text{supp}(\pi_\theta)} \pi_\theta(\tau) R(\tau) d\tau \\ &= \int_{\tau \in \text{supp}(\pi_\theta) \cap \text{supp}(\pi_\omega)} \pi_\theta(\tau) R(\tau) d\tau + \int_{\tau \in \text{supp}(\pi_\theta) \setminus \text{supp}(\pi_\omega)} \pi_\theta(\tau) R(\tau) d\tau \end{aligned}$$

The bias is the difference between these two quantities. This term represents the expected return from trajectories possible under π_θ but not under π_ω, and it is zero if and only if the support condition holds. □

A.2.4 Variance Divergence of the Importance Ratio

Lemma A.7 (Variance of the IS Ratio)**.** *Even if the support condition is satisfied, the variance of the importance ratio $r^\omega(\tau) = \frac{\pi_\theta(\tau)}{\pi_\omega(\tau)}$ can become extremely large when the target policy π_θ and behavior policy π_ω are dissimilar. Precisely, the variance is equal to the **Chi-squared divergence** between the two policies:*

$$\text{Var}_{\pi_\omega}(r^\omega) = \chi^2(\pi_\theta, \pi_\omega)$$

Proof. The variance of the ratio is $\text{Var}_{\pi_\omega}(r^\omega) = \mathbb{E}_{\pi_\omega}[(r^\omega)^2] - (\mathbb{E}_{\pi_\omega}[r^\omega])^2$. Under the support coverage condition, the expectation of the ratio is $\mathbb{E}_{\pi_\omega}[r^\omega] = 1$. We compute the second moment:

$$\mathbb{E}_{\pi_\omega}[(r^\omega)^2] = \int \pi_\omega(\tau) \left(\frac{\pi_\theta(\tau)}{\pi_\omega(\tau)} \right)^2 d\tau = \int \frac{\pi_\theta(\tau)^2}{\pi_\omega(\tau)} d\tau.$$

By noting that $\chi^2(\pi_\theta, \pi_\omega) = \int \frac{(\pi_\theta(\tau) - \pi_\omega(\tau))^2}{\pi_\omega(\tau)} d\tau = \int \frac{\pi_\theta(\tau)^2}{\pi_\omega(\tau)} d\tau - 2 \int \pi_\theta(\tau) d\tau + \int \pi_\omega(\tau) d\tau = \mathbb{E}_{\pi_\omega}[(r^\omega)^2] - 2 + 1 = \mathbb{E}_{\pi_\omega}[(r^\omega)^2] - 1$, we have:

$$\mathbb{E}_{\pi_\omega}[(r^\omega)^2] = \chi^2(\pi_\theta, \pi_\omega) + 1.$$

Therefore, the variance is:

$$\text{Var}_{\pi_\omega}(r^\omega) = (\chi^2(\pi_\theta, \pi_\omega) + 1) - 1^2 = \chi^2(\pi_\theta, \pi_\omega).$$

Both the χ^2-divergence and the more commonly known KL-divergence ($\mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_\omega)$) are measures of dissimilarity between distributions (both are instances of f-divergences). A large value in one typically implies a large value in the other. Therefore, as the policies diverge, there are often regions where $\pi_\theta(\tau) \gg \pi_\omega(\tau)$. In these regions, the ratio $r^\omega(\tau)$ becomes extremely large, causing the variance to explode. □

A.3 Bias Advantage of the MIS Estimator

The standard MIS estimator is proven to be unbiased. In practice, a common and highly practical scenario involves using external data collected from the behavior policy, π_ω, which may be far from the target policy π_θ. To stabilize estimates, one can introduce a proxy policy, $\pi_{\theta_{\text{old}}}$ (e.g., a previous iterate of π_θ), into the denominator of the importance weight. This creates a powerful estimator that deliberately accepts a small, controlled bias in exchange for a substantial reduction in variance. We now formally analyze the bias advantage of this practical MIS estimator compared to the aforementioned approaches.

Remark A.8 (Controlled Bias vs. Explosive Bias of Proxy IS). This estimator is motivated by variance reduction. While biased, its bias is far more controlled than that of the proxy estimator from Lemma A.5, which uses only $\pi_{\theta_{\text{old}}}$ in the denominator. A comparison of their bias-inducing factors is revealing:

- **Proxy IS Factor:** $f_{\text{proxy}}(\tau) = \frac{\pi_\omega(\tau) - \pi_{\theta_{\text{old}}}(\tau)}{\pi_{\theta_{\text{old}}}(\tau)}$
- **Practical MIS Factor:** $f_{\text{MIS}}(\tau) = \frac{\pi_\omega(\tau) - \pi_{\theta_{\text{old}}}(\tau)}{\pi_\omega(\tau) + \pi_{\theta_{\text{old}}}(\tau)}$

When $\pi_{\theta_{\text{old}}}(\tau) \to 0$ for a trajectory that is plausible under π_ω, the proxy IS factor can become arbitrarily large, leading to an uncontrolled, potentially infinite bias. In contrast, the practical MIS factor is a normalized difference and is strictly bounded within $(-1, 1)$. The presence of the true sampling distribution $\pi_\omega(\tau)$ in the denominator acts as a crucial **guardrail**, preventing the weights from exploding and ensuring the bias remains bounded.

Remark A.9 (Overcoming Support Mismatch). The practical MIS estimator also offers a robust solution to the critical problem of support mismatch (Lemma A.6), where $\text{supp}(\pi_\theta) \not\subseteq \text{supp}(\pi_\omega)$. The practical MIS estimator mitigates this by relying on the weaker joint support assumption, $\text{supp}(\pi_\theta) \subseteq \text{supp}(\pi_\omega) \cup \text{supp}(\pi_{\theta_{\text{old}}})$. By including $\pi_{\theta_{\text{old}}}$, it explicitly covers the full support of π_θ and eliminates the truncation error. In its place, it introduces a **distortion error**, given by the bounded bias term derived above. In essence, this estimator replaces a potentially infinite and unrecoverable truncation error, i.e,

$$\mathcal{B}_{\text{support}} = -\int_{\tau \in \text{supp}(\pi_\theta) \setminus \text{supp}(\pi_{\beta_1})} \pi_\theta(\tau) R(\tau) d\tau$$

, with a manageable and bounded distortion error, making it a far more robust choice for real-world applications.

A.4 Variance Advantage and Robustness of the MIS Estimator

The core advantage of MIS lies in its variance control and robustness, and we formally analyze below.

Theorem A.10 (Variance Robustness of MIS)**.** *So long as there is at least one policy in the behavior pool $\{\pi_{\beta_k}\}$ (e.g., $\pi_{\beta_k^*}$) that is a good approximation of the target policy π_θ (i.e., $\pi_{\beta_k^*} \approx \pi_\theta$), the variance of the MIS estimator will be low. The estimator is insensitive to other arbitrarily "bad" behavior policies in the pool.*

Proof. We qualitatively analyze the behavior of the MIS weight $w(\tau) = \frac{\pi_\theta(\tau)}{\sum_j \alpha_j \pi_{\beta_j}(\tau)}$, whose magnitude directly drives the variance.

Dilemma of Standard IS: Assume we only use a "bad" policy π_{β_m}, for which the probability density approaches zero in some region $\mathcal{S}_{\text{bad}}$ ($\pi_{\beta_m}(\tau) \to 0$), while the target policy has non-negligible density there ($\pi_\theta(\tau) > \epsilon$). In this case, the standard IS ratio $\frac{\pi_\theta(\tau)}{\pi_{\beta_m}(\tau)}$ would diverge in $\mathcal{S}_{\text{bad}}$, causing the variance to explode.

Advantage of MIS: Now, we add a "good" policy $\pi_{\beta_k^*}$ to the pool, satisfying $\pi_{\beta_k^*} \approx \pi_\theta$. The denominator of the MIS weight is a mixture density: $\sum_j \alpha_j \pi_{\beta_j}(\tau)$. Even in the problematic region $\mathcal{S}_{\text{bad}}$, the denominator contains at least one term, $\alpha_{k^*} \pi_{\beta_{k^*}}(\tau) \approx \alpha_{k^*} \pi_\theta(\tau)$, which is positive and non-negligible. The MIS weight is therefore effectively bounded:

$$w(\tau) = \frac{\pi_\theta(\tau)}{\alpha_{k^*} \pi_{\beta_{k^*}}(\tau) + \sum_{j \neq k^*} \alpha_j \pi_{\beta_j}(\tau)} \approx \frac{\pi_\theta(\tau)}{\alpha_{k^*} \pi_\theta(\tau) + \ldots} \leq \frac{\pi_\theta(\tau)}{\alpha_{k^*} \pi_{\beta_{k^*}}(\tau)} \approx \frac{1}{\alpha_{k^*}} = \frac{N}{n_{k^*}}$$

The weight is bounded from above by a constant that does not depend on the ratio of policies. The summation in the denominator acts as a **"variance guardrail",** preventing the sampling deficiencies of any single policy from destabilizing the entire estimate. □

Remark A.11 (Practical Implications). The robustness of MIS is especially critical when combining internal data (from an old policy π_{old}) and external data (from π_ω). The policy π_{old} ensures that the KL-divergence from the current policy π_θ is kept within a controllable range. This ensures that there is always a "good" policy in the pool. Therefore, even if the external policy π_ω is far from π_θ, the MIS estimator can stabilize the variance through the presence of π_{old}. MIS achieves a "soft", unbiased form of variance control by mixing policy densities in the denominator. This adaptive weighting mechanism makes MIS a theoretically sound and highly effective choice for integrating heterogeneous data sources in policy optimization.

A.5 Optimal Bayesian Estimation of the Behavior Policy under Model Uncertainty

We need a method to construct a robust estimator for π_ω that acknowledges our uncertainty. We propose a principled approach based on Bayesian decision theory to derive an optimal estimator for π_ω that explicitly balances our belief in the proxy model $\pi_{\theta_{\text{old}}}$ with a model of maximal uncertainty.

We frame the task of selecting an estimator $\hat{\pi}_\omega$ as a Bayesian decision problem.

- **State of Nature**: The true, unknown behavior policy π_ω.
- **Action**: Our choice of an estimator $\hat{\pi}_\omega$ for π_ω.
- **Model Space** $\mathcal{M}$: The set of candidate models for π_ω. Given our limited knowledge, we define a minimal, discrete model space that captures the dichotomy between our specific knowledge and our uncertainty.
- **Loss Function** $L(\hat{\pi}_\omega, \pi_\omega)$: A function that quantifies the error of our estimator. A standard choice is the squared L2-error, $L(\hat{\pi}_\omega, \pi_\omega) = \int \left(\hat{\pi}_\omega(\tau) - \pi_\omega(\tau)\right)^2 d\tau$.

Our goal is to find the estimator $\hat{\pi}_\omega$ that minimizes the **Bayes risk**, which is the expected loss with respect to our prior beliefs about the state of nature.

Theorem A.12 (Bayes-Optimal Policy Estimator)**.** *Let the model space for the unknown behavior policy π_ω be composed of two candidate models:*

- *The specific proxy policy, $\pi_{\theta_{old}}$, representing our available, specific information.*
- *A non-informative uniform policy, $\mathcal{U}(\tau)$, representing maximal uncertainty.*

*Let the trajectory space $\mathcal{T}$ have a finite volume $V = \int_{\mathcal{T}} d\tau$, such that $\mathcal{U}(\tau) = 1/V$. Under the **Principle of Indifference**, we assign equal prior probabilities to these models, i.e., $P(\pi_\omega = \pi_{\theta_{old}}) = P(\pi_\omega = \mathcal{U}) = 1/2$. Then, the estimator $\hat{\pi}_\omega$ that minimizes the Bayes risk (expected L2 error) is the Bayesian model average:*

$$\hat{\pi}^*_\omega(\tau) = \frac{1}{2}\pi_{\theta_{old}}(\tau) + \frac{1}{2}\mathcal{U}(\tau)$$

Proof. The Bayes risk of an estimator $\hat{\pi}_\omega$ is the expectation of the loss function over the prior distribution of π_ω:

$$\begin{aligned} R(\hat{\pi}_\omega) &= \mathbb{E}_{\pi_\omega}\left[L(\hat{\pi}_\omega, \pi_\omega)\right] \\ &= \sum_{\pi' \in \{\pi_{\theta_{\text{old}}}, \mathcal{U}\}} L(\hat{\pi}_\omega, \pi') P(\pi_\omega = \pi') \\ &= \frac{1}{2}\int \left(\hat{\pi}_\omega(\tau) - \pi_{\theta_{\text{old}}}(\tau)\right)^2 d\tau + \frac{1}{2}\int \left(\hat{\pi}_\omega(\tau) - \mathcal{U}(\tau)\right)^2 d\tau \end{aligned}$$

To find the optimal estimator $\hat{\pi}^*_\omega$ that minimizes this risk, we can use the calculus of variations or simply note that the integrand is a sum of squared errors, which is minimized point-wise. For any given trajectory τ, we seek to minimize:

$$f(\hat{\pi}_\omega(\tau)) = \left(\hat{\pi}_\omega(\tau) - \pi_{\theta_{\text{old}}}(\tau)\right)^2 + \left(\hat{\pi}_\omega(\tau) - \mathcal{U}(\tau)\right)^2$$

This is a simple quadratic function of the scalar value $\hat{\pi}_\omega(\tau)$. We find the minimum by taking the derivative with respect to $\hat{\pi}_\omega(\tau)$ and setting it to zero:

$$\begin{aligned} \frac{\partial f}{\partial \hat{\pi}_\omega(\tau)} &= 2\left(\hat{\pi}_\omega(\tau) - \pi_{\theta_{\text{old}}}(\tau)\right) + 2\left(\hat{\pi}_\omega(\tau) - \mathcal{U}(\tau)\right) = 0 \\ 2\hat{\pi}_\omega(\tau) - \pi_{\theta_{\text{old}}}(\tau) - \mathcal{U}(\tau) &= 0 \\ \hat{\pi}_\omega(\tau) &= \frac{1}{2}\left(\pi_{\theta_{\text{old}}}(\tau) + \mathcal{U}(\tau)\right) \end{aligned}$$

This result gives the point-wise minimizer. Integrating over all τ confirms that the optimal estimator function is:

$$\hat{\pi}^*_\omega(\tau) = \frac{1}{2}\pi_{\theta_{\text{old}}}(\tau) + \frac{1}{2}\mathcal{U}(\tau)$$

This estimator is known as the **Bayes estimator** under quadratic loss for this prior. It is optimal in the sense that no other estimator has a lower expected error, given our stated beliefs about the possible models for π_ω. It is straightforward to verify that $\hat{\pi}^*_\omega(\tau)$ is a valid probability density function, as $\int \hat{\pi}^*_\omega(\tau)d\tau = \frac{1}{2}\int \pi_{\theta_{\text{old}}}(\tau)d\tau + \frac{1}{2}\int \mathcal{U}(\tau)d\tau = \frac{1}{2}(1) + \frac{1}{2}(1) = 1$. □

Assumption A.13 (Unit-Volume Trajectory Space)**.** *For analytical tractability, we assume the trajectory space $\mathcal{T}$ is normalized to have unit volume, i.e., $\int_{\mathcal{T}} d\tau = 1$. Under this assumption, the maximum-entropy (uniform) distribution is $\mathcal{U}(\tau) = 1$ for all $\tau \in \mathcal{T}$.*

Remark A.14 (Robustness and Connection to Regularization). Theorem A.12 provides a rigorous justification for what is, in essence, a form of regularization. The resulting estimator $\hat{\pi}^*_\omega$ is a mixture model that hedges against the deficiencies of $\pi_{\theta_{\text{old}}}$. The uniform component $\mathcal{U}(\tau)$ acts as a **"safety net"** or a **"defensive distribution"**. By ensuring that $\hat{\pi}^*_\omega(\tau) \geq \frac{1}{2V} > 0$ for all τ, it guarantees that the importance sampling ratio's denominator is strictly positive and bounded away from zero. This prevents the variance of the importance weights from exploding, a critical property for stable off-policy learning.

The assumption $P = 1/2$ reflects a state of maximal ambiguity between the specific information we have ($\pi_{\theta_{\text{old}}}$) and the general uncertainty we face ($\mathcal{U}$). It is the most conservative and robust choice when we cannot quantify our confidence in $\pi_{\theta_{\text{old}}}$. Thus, forming the estimator as their mean is the theoretically optimal strategy to navigate this uncertainty.

B Theoretical Analysis of the Exploration-Based Advantage

We provide a theoretical justification for the proposed Exploration-Based Advantage function. We prove that it adaptively focuses the policy gradient updates on high-value, hard-to-explore actions.

B.1 Gradient Analysis

We now analyze the effect of this advantage function on the policy gradient.

Lemma B.1 (Gradient Contribution of a Single Timestep)**.** *The gradient update for the policy parameters θ induced by the action $e_{i,t}$ from a correct, high-reward trajectory i is given by:*

$$\Delta\theta_{i,t} \propto \nabla_\theta \log \pi_\theta(e_{i,t}|q, e_{i,<t}) \cdot A_i \cdot (1 - \pi_\theta(e_{i,t}|q, e_{i,<t}))^\gamma$$

Proof. The gradient update for the policy objective at timestep t is proportional to $\nabla_\theta \log \pi_\theta(e_{i,t}|\dots) \cdot A^c_{i,t}$. Substituting the definition of $A^c_{i,t}$, we have:

$$\Delta\theta_{i,t} \propto \nabla_\theta \log \pi_\theta(e_{i,t}|\dots) \cdot A_i \cdot C_{i,t}(\theta)$$

By the definition of $C_{i,t}(\theta)$ and the properties of the detach operator, the term $C_{i,t}(\theta)$ is treated as a scalar weight during backpropagation. Substituting its definition yields the result. □

This lemma establishes the precise form of the gradient update. We now prove our main result: that this form adaptively focuses learning.

Theorem B.2 (Adaptive Gradient Focusing)**.** *Given a high-reward trajectory where $A_i > 0$, the gradient magnitude of the update induced by $A^c_{i,t}$ is inversely related to the policy's confidence $\pi_\theta(e_{i,t}|\dots)$. The update is amplified for "hard" (low-probability) actions and suppressed for "easy" (high-probability) actions.*

Proof. We analyze the asymptotic behavior of the scaling factor on the gradient, based on Lemma B.1. Let $p_t = \pi_\theta(e_{i,t}|\dots)$ denote the policy's probability for the correct action at time step t. The gradient is scaled by the factor $A_i \cdot (1 - p_t)^\gamma$. We consider two cases for the value of p_t.

Case 1: Hard-to-Explore Correct Action. In this case, the policy assigns a low probability to the correct action, i.e., $p_t \to 0$. The exploration weight becomes:

$$\lim_{p_t \to 0} C_{i,t}(\theta) = \lim_{p_t \to 0} (1 - p_t)^{\gamma} = 1$$

The resulting gradient update, $\Delta\theta_{i,t} \propto \nabla_\theta \log p_t \cdot A_i$, retains its full magnitude. The learning signal from this valuable, unexplored action is preserved.

Case 2: Easy-to-Explore Correct Action. In this case, the policy is already confident about the correct action, i.e., $p_t \to 1$. The exploration weight becomes:

$$\lim_{p_t \to 1} C_{i,t}(\theta) = \lim_{p_t \to 1} (1 - p_t)^{\gamma} = 0$$

The resulting gradient update vanishes: $\Delta\theta_{i,t} \to 0$. The model effectively ignores updates from examples it has already mastered.

Conclusion. It demonstrates that the optimization process is focused on the gradients from actions where the policy is incorrect or uncertain, thereby prioritizing the learning of new knowledge. This proves that the advantage function leads to adaptive gradient focusing. □

C Extended Background and Related Work

C.1 Extended Preliminary Knowledge

LLM-based Reasoning as a Markov Decision Process. We frame the task of generating a reasoning sequence (e.g., a solution to a math problem) as a Markov Decision Process (MDP) (Puterman, 2014). At each timestep t, the state s_t consists of the initial prompt q concatenated with the sequence of previously generated tokens, $y_{<t}$. The action a_t is the selection of the next token y_t from the vocabulary. The model, or policy π_θ, maps a state to a distribution over actions. A reward $R(q, y)$ is provided only upon completion of the entire sequence y. In the context of RLVR, this reward is typically sparse and binary. For example, a score is 1 if the final answer is correct and 0 otherwise. The objective is to learn a policy π_θ that maximizes the expected cumulative reward $J(\theta) = \mathbb{E}_{y\sim\pi_\theta}[R(q, y)]$.

Evaluating Reasoning Boundaries with pass@k. To accurately assess a model's true problem-solving capabilities, we utilize the 'pass@k' metric (Chen et al., 2021; Dong et al., 2025a). It measures the probability of obtaining at least one correct answer within k independent samples for a given problem. Unlike mean accuracy ('pass@1'), 'pass@k' provides a more comprehensive view of the model's reasoning potential and is critical for evaluating whether a method expands the set of solvable problems (Yue et al., 2025a).

C.2 Extended Related Work

RL-PLUS builds upon two significant lines of research in enhancing LLMs: on-policy RLVR for improving reasoning, and hybrid methods that integrate SFT with RL. In this section, we provide a comprehensive review of these areas, highlighting their advancements and the critical limitations that motivate our approach.

C.2.1 On-Policy RLVR and Its Intrinsic Limitations

RLVR has emerged as a dominant paradigm for augmenting the reasoning capabilities of LLMs, particularly in domains like mathematics and programming, where solution correctness can be automatically verified (OpenAI, 2024; Guo et al., 2025; KimiTeam, 2025). This approach treats the LLM as a policy that generates a sequence of tokens (a reasoning trajectory), and a reward is provided based on the final outcome. Seminal works have demonstrated that by optimizing for rewards on correct final answers, LLMs can learn to generate longer, more complex, and more accurate chains of thought (Zeng et al., 2025; Hu et al., 2025; Yue et al., 2025b).

A key algorithmic advancement in this area is GRPO (Shao et al., 2024), which has proven highly effective. Unlike PPO (Schulman et al., 2017), GRPO forgoes a separate value model and instead estimates the advantage function by normalizing rewards across a batch of sampled trajectories. This design simplifies the training process and improves computational efficiency, making it well-suited for the large-scale optimization of LLMs. Other variants, such as those that simplify or remove the KL-divergence penalty, have also been proposed to better accommodate the significant policy shifts required for long-form reasoning (Zeng et al., 2025; Yu et al., 2025).

Despite these successes, a growing body of evidence suggests a fundamental limitation: on-policy RLVR primarily refines and exploits the knowledge already present in the base model rather than enabling the discovery of genuinely new reasoning abilities (Havrilla et al., 2024; Yue et al., 2025a). This leads to two critical problems: 1) Capability Boundary Collapse: As highlighted in Figure 1(a), while RLVR-trained models often exhibit improved performance on the pass@1 metric, their advantage diminishes and can even become negative when evaluated with a larger budget of attempts (pass@k for large k). The base model often shows a higher pass@k, indicating it possesses a wider, albeit less refined, range of potential problem-solving strategies. This suggests that RLVR optimizes the probability of known correct paths but fails to expand the set of solvable problems, effectively causing the model's capability boundary to contract (Yue et al., 2025a). 2) Entropy Collapse: During on-policy RLVR training, the policy's entropy can decrease sharply as the model learns to favor high-reward trajectories (Cui et al., 2025b). This makes the model's output more deterministic, hindering its ability to explore diverse reasoning paths. This phenomenon of "inward exploitation" reinforces existing biases and severely limits the model's potential for continuous self-improvement and discovery.

In essence, on-policy RLVR methods are constrained by the inherent knowledge of the base model, making it difficult to achieve breakthroughs beyond its initial capabilities.

C.2.2 Hybrid SFT-RL Methods

To overcome the knowledge limitations of pure on-policy RL, researchers have explored hybrid approaches that combine RL with SFT on external datasets of high-quality demonstrations (Cai et al., 2025). These methods aim to infuse the model with new knowledge and reasoning patterns that it might not discover through self-exploration alone.

Early hybrid methods often employed a sequential, multi-stage training pipeline: first performing SFT on an external dataset, followed by RL fine-tuning. This approach, used in models like InstructGPT (Ouyang et al., 2022), is conceptually simple but suffers from significant drawbacks, including catastrophic forgetting of the SFT-learned knowledge during the RL phase and high computational overhead from the sequential training stages.

More recent work has shifted towards unified or interleaved training frameworks that attempt to combine SFT and RL more dynamically. These approaches can be broadly categorized as follows. 1) Alternating/Mixed Objectives: Methods like ReLIFT (Ma et al., 2025) alternate between RL and online fine-tuning on problems the model finds difficult. LUFFY (Yan et al., 2025) uses a mixed policy to selectively imitate high-quality trajectories from an external dataset. 2) Adaptive Balancing: SASR (Chen et al., 2025) and SuperRL (Liu et al., 2025a) introduce adaptive mechanisms to dynamically switch or balance the SFT and RL loss contributions based on the model's training state or performance. 3) Guidance from Abstracted Knowledge: TAPO (Wu et al., 2025) moves beyond direct imitation by abstracting "thought patterns" from external data and using them as high-level guidance to shape the RL exploration process. 4) Unified Frameworks: UFT (Wang et al., 2024b) proposes a unified framework to merge SFT and RL updates, aiming to accelerate convergence and improve stability.

While these hybrid methods represent a significant step forward, they often rely on complex heuristics to balance the SFT and RL objectives, which can be unstable and difficult to tune. More importantly, they do not fully address the two core technical challenges of integrating external data into an RL framework. 1) Distributional Mismatch: There is an unavoidable and often significant distributional gap between the model's current on-policy distribution and the distribution of the external (off-policy) data. Naively applying an SFT loss (i.e., imitation learning) can disrupt the RL objective, and standard importance sampling corrections are often inadequate, suffering from high variance and bias when the distributions are far apart. 2) Inefficient Knowledge Extraction: Simply adding an SFT loss does not guarantee that the model will internalize the underlying reasoning principles from the external data. The model may be inclined to focus on high-probability tokens that align with its existing knowledge, rather than exploring the low-probability but potentially crucial tokens that represent novel reasoning steps.

D Effect of hyperparameter γ

Our systematic investigation into the hyperparameter γ in RL-PLUS, illustrated in Figure 5, reveals two key findings. First, the model demonstrates considerable robustness, as its performance fluctuates only small across the tested range of γ. Second, RL-PLUS consistently surpasses the GRPO baseline across all math reasoning benchmarks, irrespective of the specific value of γ. Further analysis highlights a distinct trend: the model uniformly achieves its peak performance when γ=0.5. This optimal value holds true not only for the Average test score but also across all individual benchmarks, including AMC, Olympiad, AIME, Minerva, and Math. This suggests that an intermediate value for γ strikes an effective balance in the model's learning process. While the model is not highly sensitive to this parameter, the clear peak establishes γ=0.5 as a strong default, and there is still potential room for improvement with fine-grained tuning of γ.

E Complete Experimental Analysis

Performence of RL-PLUS. As shown in Table 1, RL-PLUS comprehensively outperforms existing RLVR methods across all evaluated applications, achieving SOTA performance. A comparison with several straightforward baselines clearly demonstrates the benefits of RL-PLUS. SFT can be viewed as a means of learning from external knowledge, while GRPO enables the model to explore solutions on its own through reinforcement learning. The combined "SFT+GRPO" approach yields synergistic gains, illustrating the value of integrating both external knowledge and self-exploration. However, the "GRPO w/ SFT Loss" baseline, which simply adds an SFT loss to the RL training, shows a decline in performance. This suggests that effectively merging these two

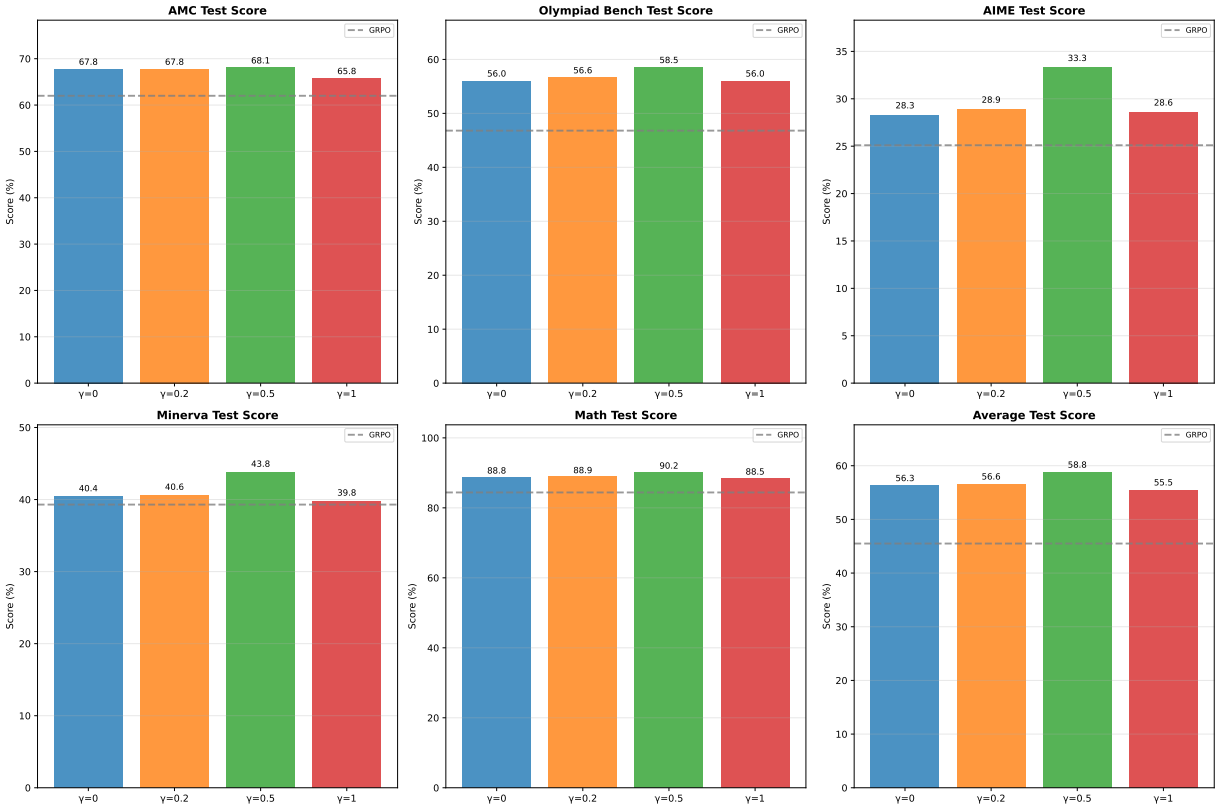


Figure 5: Effect of hyperparameter γ in RL-PLUS.

learning paradigms is a non-trivial challenge. RL-PLUS significantly improves upon "SFT+GRPO" by an average of +5.2 points, showcasing a more potent strategy for this integration. Furthermore, when compared to concurrent methods like LUFFY and ReLIFT, which also incorporate external examples into their training process in some form (LUFFY treats external data as perfect expert trajectories with probability 1, which introduces significant bias. ReLIFT alternates training between RL and SFT, but this may lead to knowledge loss and suboptimal balance), RL-PLUS also achieves superior performance, which indicates RL-PLUS offers a more effective way for learning from external knowledge.

Performance on OOD Tasks. The results on OOD tasks are presented in Table 2, which show that RL-PLUS achieves substantial improvements over all baselines, including the mainstream method SFT+GRPO. It surpasses the next best baseline by an average of +3.9 points. This indicates that RL-PLUS not only enhances capabilities within a specific domain but also develops more fundamental reasoning abilities that generalize to other domains. In the domain of science QA, RL-PLUS consistently outperforms both GRPO and SFT+GRPO across all benchmarks. More notably, under a significant domain shift to programming tasks, our approach maintains its strong performance and advantage. In contrast, the performance of SFT and SFT+GRPO deteriorates significantly in this area. Considering this alongside the in-domain results from Table 1, a clear pattern emerges: while SFT-based methods provide a strong boost for in-domain tasks, they fail to generalize and perform worse than RL-based methods in OOD scenarios. RL-PLUS resolves this trade-off. By effectively merging the external knowledge acquisition of SFT with the robust generalization of RL, it achieves superior performance in both in-domain and out-of-distribution settings, outclassing methods reliant on either paradigm alone.

	Avg (Table 1)	OOD Avg (Table 2)
LUFFY	50.1 ± 0.9	38.9 ± 0.9
SFT+GRPO	48.2 ± 1.0	35.4 ± 1.1
Our	$\mathbf{53.4 \pm 0.8}$	$\mathbf{48.8 \pm 0.6}$

Training Dynamics. In Figure 2, we present the training dynamics of our proposed method and baselines on various benchmarks. As illustrated,

RL-PLUS consistently outperforms the alternatives in terms of test accuracy and rewards throughout the training process. Notably, RL-PLUS continues to show a clear upward trend in performance even after the baselines have plateaued. We further analyze the changes in actor entropy during training. We observe that directly incorporating external data during rollouts (the green line in Figure 2) leads to an "entropy explosion", causing the model's outputs to become chaotic. In contrast, the entropy of the baseline models collapses to zero over the course of training, indicating a loss of exploratory capability. The entropy of RL-PLUS, however, does not diminish to zero, which suggests that our trained model retains a considerable capacity for exploration. Prior research (Cui et al., 2025b) has established that policy performance is achieved at the cost of policy entropy, and the depletion of entropy marks the upper limit of performance. This implies that RL-PLUS still possesses potential for further improvement. Additionally, the response length can reflect the test-time scaling performance of a method. The steadily increasing response length of RL-PLUS is the indicator of a healthy and robust training state. In contrast, while directly incorporating external data also leads to long response lengths, its low accuracy and high policy entropy suggest that this length stems from unproductive exploration rather than meaningful reasoning.

Application on Various LLMs. To validate the applicability of RL-PLUS on various LLMs, we conduct experiments on several mainstream open-source LLMs, including LLaMA-3.1-8B, Deepseek-Math-7B, and the 1.5B and 7B versions of Qwen2.5-Math. The detailed results are presented in Table 3. The results indicate that RL-PLUS achieves comprehensively superior performance, regardless of the base model. Notably, on Qwen2.5-Math-7B model, RL-PLUS elevates the average score to 53.4, significantly outperforming the base model of 19.0 and other methods such as SFT of 44.1 and GRPO of 45.5. Furthermore, on LLaMA-3.1-8B, where methods like GRPO struggled to yield improvements, RL-PLUS successfully trained the model to achieve an absolute gain of 11.9 points. These findings provide evidence that RL-PLUS can consistently enhance LLMs of varying architectures and scales, significantly boosting their reasoning capabilities.

Acquiring Reasoning Abilities Beyond Base Model. The fundamental goal of incorporating an external policy into the RL-PLUS method is to expand the model's capability boundary by continuously introducing knowledge. Following the experimental setup of (Yue et al., 2025a), we test whether RL-PLUS acquires superior reasoning abilities relative to the base model. Figure 4 displays the pass@k performance curves for different methods across multiple tasks. A clear trend is observable where the performance curve of the GRPO method gradually converges with that of the base model as k increases. In some instances, GRPO's performance even drops below the base model at larger k-values, a finding consistent with that of (Yue et al., 2025a). In contrast, our approach maintains a consistent performance advantage over both the base model and GRPO as k-values increase. This sustained outperformance provides strong evidence that RL-PLUS effectively breaks through the capability boundary of the base model, rather than merely optimizing performance within its inherent ability range. On the AMC and MATH-500 tasks, the accuracy of RL-PLUS eventually plateaus because its performance is approaching the maximum possible score of 1.0.

Ablation Study. To analyze the sources of RL-PLUS's effectiveness, we conduct a series of ablation studies, with the results presented in Table 4. We first ablate the two core components of our approach: Multiple Importance Sampling and the Exploration-Based Advantage Function. The experimental results show that removing the Exploration-Based Advantage Function causes the model's average performance to decrease from 53.4 to 50.9, which demonstrates the importance of efficient exploration for reinforcement learning. Furthermore, removing Multiple Importance Sampling leads to a more significant performance degradation, with the average score dropping substantially to 45.5, highlighting the significance of incorporating external knowledge. Additionally, we compare our method against three naive approaches for integrating external knowledge. The first variant approximates the external policy π_{θ_ω} using the old policy $\pi_{\theta_{\text{old}}}$. The second variant, an approach also seen in LUFFY (Yan et al., 2025), approximates the external policy's probability as 1, treating it as a perfect oracle. When using our policy estimation as the external policy, i.e., the third variant, the performance improves by 2.9 points, demonstrating the

effectiveness of our policy estimation. Due to the improper integration methods, these variants all show a significant performance gap compared to RL-PLUS.

Training Stability. To validate the training stability of the RL-PLUS method, we extended the number of training steps on the Qwen2.5-Math-1.5B model to **over 10 times** the original setup. As shown in Figure 3, the model's key metrics demonstrate excellent stability and continuous performance improvement as training progresses. Specifically, the Average Test Score and Critic Rewards Mean both show a steady upward trend, while the Actor Entropy Loss rapidly converges and stabilizes in a healthy, non-zero range. This reveals an ideal balance: the model's policy, while becoming more effective (i.e., exploitation), also maintains the necessary policy stochasticity for exploration, thus avoiding premature convergence to a local optimum. These results strongly demonstrate that the RL-PLUS framework possesses outstanding training stability and has the potential for further performance gains through extended training.

F Case Study

Figure 6 presents a typical case study that visually contrasts the performance of RL-PLUS with the baseline methods, GRPO and SFT+GRPO. In this case, RL-PLUS demonstrates a significant advantage in both logical rigor and computational precision. Specifically, GRPO, while touching upon a part of the core issue by identifying 'multiples of 5' as part of the losing positions, demonstrates an incomplete understanding. It fails to identify the other critical condition, thus arriving at an incorrect conclusion. SFT+GRPO's approach is fundamentally flawed. It completely misinterprets the game-theoretic model of the problem, erroneously applying an irrelevant 'modulo 3' logic, causing its reasoning to be incorrect from the outset. The performance of RL-PLUS is exemplary. It begins by accurately identifying the problem as a game of finding P-positions (second-player winning positions). Subsequently, through deductive reasoning, it successfully derives the complete pattern for the set of losing positions: when $n \equiv 0$ or $2 \ mod \ 5$. Finally, it proceeds with a clear, step-by-step calculation for both conditions, sums them accurately, and arrives at the correct answer. This case provides compelling evidence that RL-PLUS possesses a more profound and comprehensive multi-step reasoning capability.

G Experimental Setup

Training Details. All experiments are conducted on 8 NVIDIA A100 80G GPUs. For our training, we use the dataset from previous work (Yan et al., 2025), which contains 45,000 prompts from OpenR1-Math-220k with correct reasoning trajectories annotated by Deepseek-R1. In implementing the RL algorithm, we leverage the VeRL framework (Sheng et al., 2024). We set the batch size to 128, the mini-batch size to 64, and the maximum training epoch to 2. For each problem, we generate 8 rollouts, with a maximum response length of 8192 tokens. For our approach, we replace one of the model-generated rollouts with a correct reasoning trajectory from training dataset. It is important to note that we ensure all other RL algorithms maintain the same parameter settings as RL-PLUS to guarantee a fair comparison. For the hyperparameter γ, we set it to 0.5 in all experiments by default. By default, we use the Qwen2.5-Math-7B model (Yang et al., 2024) as base model, following previous work (Yan et al., 2025). To validate the applicability of RL-PLUS on various base LLMs, we additionally employ LLaMA-3.1-8B-instruct, Deepseek-Math-7B-instruct, and Qwen2.5-Math-1.5B as our base models. Each result is reported as the average pass@1 score over 5 runs by default.

Evaluation. In line with established practices, we evaluate the performance of RL-PLUS on a comprehensive suite of standard mathematical reasoning benchmarks, including GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024), as well as on competition-level benchmarks such as AIME 2024 (Li et al., 2024) and AMC 2023 (Li et al., 2024). Additionally, although our training focuses on math, we extend our evaluation to out-of-domain (OOD) tasks to assess the robustness and generalization capabilities of our approach. The OOD datasets include ARC-c (Clark et al., 2018)(Open-Domain Reasoning), GPQA-diamond (Rein et al., 2024) (Science Graduate Knowledge), MMLU-Pro (Wang et al., 2024a) (Reasoning-focused Questions from Academic Exams and Textbooks), as well as three code generation datasets: HumanEval (Chen et al., 2021), LeetCode (Guo et al., 2024), and LiveCodeBench (Jain et al., 2024). During evaluation, we set the sam-

Case Study

Prompt: [System Prompt] This is the problem: Alice and Bob play the following game. A stack of n tokens lies before them. The players take turns with Alice going first. On each turn, the player removes either 1 token or 4 tokens from the stack. Whoever removes the last token wins. Find the number of positive integers n less than or equal to 2024 for which there exists a strategy for Bob that guarantees that Bob will win the game regardless of Alice's play.

Correct Answer: 809

GRPO

Response: To determine the number of positive integers (n) less than or equal to 2024 for which there exists a strategy for Bob that guarantees that Bob will win the game regardless of Alice's play, we find the number of multiples of 5 up to 2024.

The largest multiple of 5 less than or equal to 2024 is (2020 = 5 \times 404), so there are 405 multiples of 5 from 0 to 2024 (including 0). Therefore, the number of such (n) is (405).

The final answer is (\boxed{405}).

SFT+GRPO

Response: To determine the number of positive integers (n) less than or equal to 2024 for which Bob can always win the game regardless of Alice's play, we need to analyze the losing positions. A position is a losing position if every move from it leads to a winning position for the opponent.

Through recursive analysis, we find that the losing positions are those where (n \equiv 0) or (n \equiv 2 \pmod{3}). This means the numbers that are congruent to 0 or 2 modulo 3 are losing positions for the player about to move.

To count the numbers from 1 to 2024 that are not congruent to 1 modulo 3, we calculate the number of numbers in each congruence class modulo 3:
1. Numbers congruent to 0 modulo 3 between 1 and 2024: (\left\lfloor \frac{2024}{3} \right\rfloor = 674)
2. Numbers congruent to 2 modulo 3 between 1 and 2024: (\left\lfloor \frac{2024 - 2}{3} \right\rfloor + 1 = 674 + 1 = 675)
Adding these together gives the total number of losing positions:
[674 + 675 = 1349]

Thus, the number of positive integers (n) less than or equal to 2024 for which Bob can always win is (\boxed{1349}).

RL-PLUS

Response: To determine the number of positive integers (n) less than or equal to 2024 for which Bob has a strategy to guarantee a win regardless of Alice's play, we analyze the game where Alice and Bob take turns removing either 1 or 4 tokens from a stack of (n) tokens. The key is to identify the losing positions for Alice, i.e., positions where Bob can always win.

We use the concept of winning and losing positions. A position is a losing position if the player whose turn it is will lose if the opponent plays optimally. We find that the losing positions for Alice are those where (n \equiv 0) or (2 \mod 5). This is because:
- If (n \equiv 0 \mod 5), any move (subtracting 1 or 4) results in a position (n - 1) or (n - 4), both of which are winning positions.
- If (n \equiv 2 \mod 5), any move results in a position (n - 1) or (n - 4), both of which are winning positions.

Thus, the losing positions for Alice are (n \equiv 0) or (2 \mod 5). We count the number of such (n) up to 2024.
1. **Counting (n \equiv 0 \mod 5)**:
   - The sequence is (5, 10, 15, \ldots, 2020).
   - The number of terms is (\left\lfloor \frac{2024}{5} \right\rfloor = 404).
2. **Counting (n \equiv 2 \mod 5)**:
   - The sequence is (2, 7, 12, \ldots, 2022).
   - The number of terms is (\left\lfloor \frac{2024 - 2}{5} \right\rfloor + 1 = 405).

Adding these counts together, the total number of positive integers (n \leq 2024) for which Bob has a winning strategy is (404 + 405 = 809).

Thus, the final answer is (\boxed{809}).

Figure 6: A case of RL-PLUS compared with baselines GRPO and SFT+GRPO.

pling temperature to 0.6 and report the average pass@1 score over 5 runs by default.

Baselines. We compare our approach with two categories of baselines, all trained upon the same base model. The first category comprises eight recently proposed RLVR methods, including: 1) **SimpleRL** (Zeng et al., 2025) and 2) **OpenReasoner-Zero** (Hu et al., 2025) are two open-source RL implementations that train starting from the base model using rule-based rewards. SimpleRL employs a token-level, length-rectified GRPO algorithm, while OpenReasoner-Zero utilizes the PPO algorithm. 3) **PRIME** (Cui et al., 2025a) introduces an implicit process reward based on outcome

labels during RL. 4) **Oat-Zero** (Liu et al., 2025b) modifies GRPO algorithm by removing the standard deviation from the advantage computation and eliminating token-level normalization in the policy loss calculation. 5) **DAPO** (Yu et al., 2025) optimize the GRPO algorithm by introducing four operations: Clip-Higher, Dynamic Sampling, a Token-Level Policy Gradient Loss, and Overlong Reward Shaping. 6) **LUFFY** (Yan et al., 2025) leverages off-policy reasoning trajectories to augment GRPO. 7) **TAPO** (Wu et al., 2025) integrates reasoning templates into GRPO sampling process to enhance the model's internal reasoning capabilities. 8) **ReLIFT** (Ma et al., 2025) performs RL and SFT alternately during training. The second category consists of four straightforward baselines: 1) **SFT**, supervised fine-tuning using external reasoning trajectory data. 2) **GRPO** (Shao et al., 2024), training with GRPO algorithm on question-answer pairs. 3) **SFT+GRPO**, a common RL cold-start approach that performs SFT before RL training. 4) **GRPO w/ SFT Loss**, jointly optimizes the GRPO objective and SFT loss during training.