HINT: HELPING INEFFECTIVE ROLLOUTS NAVIGATE TOWARDS EFFECTIVENESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning (RL) has become a key driver for enhancing the long chain-of-thought (CoT) reasoning capabilities of Large Language Models (LLMs). However, prevalent methods like GRPO often fail when task difficulty exceeds the model's capacity, leading to reward sparsity and inefficient training. While prior work attempts to mitigate this using off-policy data, such as mixing RL with Supervised Fine-Tuning (SFT) or using hints, they often misguide policy updates In this work, we identify a core issue underlying these failures, which we term low training affinity. This condition arises from a large distributional mismatch between external guidance and the model's policy. To diagnose this, we introduce *Affinity*, the first quantitative metric for monitoring exploration efficiency and training stability. To improve *Affinity*, we propose HINT: **H**elping **I**neffective rollouts **N**avigate **T**owards effectiveness, an adaptive hinting framework. Instead of providing direct answers, HINT supplies heuristic hints that **guide the model to discover solutions on its own**, preserving its autonomous reasoning capabilities. Extensive experiments on mathematical reasoning tasks show that HINT consistently outperforms existing methods, achieving state-of-the-art results with models of various scales, while also demonstrating significantly more stable learning and greater data efficiency. Code is available on Github[1].

1 INTRODUCTION

RL methods, particularly GRPO (Shao et al., 2024), play a pivotal role in advancing long CoT reasoning (Wei et al., 2022). By avoiding the instability and overhead of training a separate value model, GRPO leverages group-based reward aggregation to deliver stable and efficient learning signals. Such RL approaches (Ahmadian et al., 2024; Shao et al., 2024; Hu, 2025; Yu et al., 2025) have become a key driver of progress in reasoning ability, enabling models to explore solution paths on verifiable problems. Building on these advances, recent reasoning models such as DeepSeek-R1 (Guo et al., 2025), OpenAI-o1 (Jaech et al., 2024), and Kimi-1.5 (Team et al., 2025) have achieved remarkable performance on complex tasks like mathematical problem solving (Shao et al., 2024)and programming (Jiang et al., 2024).

A critical challenge for GRPO, despite its strong empirical performance, is its tendency to generate sample groups consisting entirely of incorrect answers on tasks whose difficulty exceeds the policy model's evolving capacity (Zhao et al., 2025; Yue et al., 2025). In such cases, the learning process suffers from reward sparsity, where the feedback becomes uniform and uninformative (Yu et al., 2025), ultimately reducing training efficiency and wasting valuable data.

Leveraging external, off-policy data is a key method for addressing this issue. This method has been implemented in prior work through two main lines of remedies. (I) **Mixed-policy** (Yan et al., 2025; Zhang et al., 2025a; Fu et al., 2025b): Mixed-policy involves interleaving RL with SFT in a hybrid scheme to stabilize training by leveraging off-policy data. (II) **Using hints** (Li et al., 2025; Liu et al., 2025b; Zhang et al., 2025b): To mitigate reward sparsity and ensure continuous training updates, another common approach is to leverage prompts derived from the ground truth during the rollout phase, guiding the model's exploration along correct trajectories.

[1] https://anonymous.4open.science/r/HINT-9DD9/

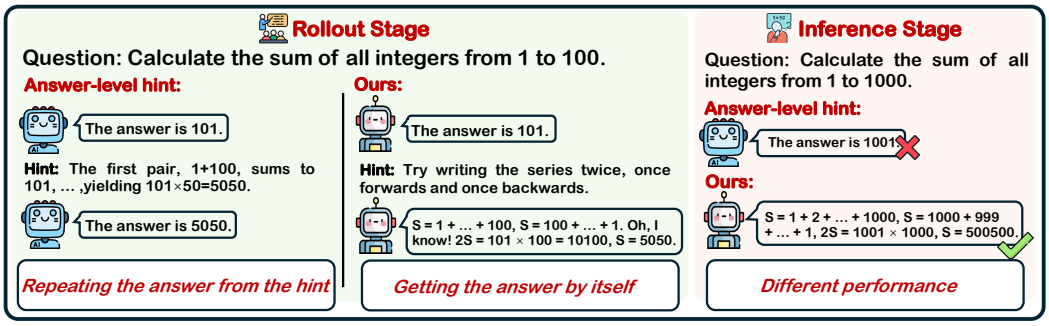

Figure 1: Comparison of Hint Mechanisms and Their Impact on Learning. The answer-level hint provides an explicit partial solution. The model can achieve a reward by simply completing this pre-defined path, which encourages learning a superficial shortcut rather than genuine reasoning. In contrast, our heuristic hint offers a high-level conceptual prompt, **compelling the model to develop its own solution path independently**.

Despite their potential benefits, both of these approaches introduce a significant drawback rooted in a **substantial distributional mismatch**. In mixed-policy training, this mismatch arises between the off-policy SFT data and the on-policy updates, which lead to conflicting gradients and training instability (Yan et al., 2025). Similarly, answer-level hints create a severe mismatch between the distribution of the ground truth and the distribution of the current policy. This results in a deceptive learning signal that, while inflating training rewards, ultimately misguides policy updates toward non-generalizable or spurious solution paths (See Figure 2).

Fundamentally, the aforementioned drawbacks stem from **a lack of what we term training affinity**. This core issue that arises from an **over-reliance** on off-policy sources, such as SFT data or answer-level hints, which inevitably creates a significant distributional mismatch with the model's current policy (Fu et al., 2025a). This mismatch, in turn, leads to excessively high variance in the importance sampling ratios, destabilizing the entire training process. This instability is such a core challenge that prominent algorithms like PPO introduce mechanisms such as clipping to manage it (Schulman et al., 2017), the behavior of which itself provides a signal of training dynamics. To leverage this insight and create a quantitative diagnostic, we define *Affinity* metric in terms of training stability, considering both the frequency of clipping and the variance of the importance sampling ratios.

To leverage off-policy data for enhancing model capability while preserving training affinity, the guiding principle must be to **help the model articulate the solution on its own, rather than being directly told the answer.** To this end, we propose HINT: **H**elping **I**neffective rollouts **N**avigate **T**owards effectiveness, an adaptive hinting framework. As illustrated in Figure 1, HINT implements this principle by providing heuristic hints instead of partial ground-truth answers. These hints serve as high-level guidance, helping the model navigate challenging problems without disclosing solutions. This dynamic is akin to the Socratic method in teaching, where guiding a student with thoughtful prompts, rather than supplying answers, is crucial for developing robust and generalizable reasoning skills.

Our contributions can be summarized as follows:

- We introduce the first formal definition of low training affinity, a key failure mode in RL methods that incorporate off-policy data. Building on this formalization, we propose *Affinity*, a quantitative metric that enables the continuous monitoring of these critical training dynamics.
- To effectively enhance the model's reasoning capabilities while preserving high *Affinity*, we propose HINT, a framework that adaptively providing heuristic hints. HINT guides the model towards successful trajectories without compromising its autonomous exploration and reasoning capabilities.
- Extensive experiments validate our approach. HINT consistently outperforms methods based on mixed-policy and answer-level hints, achieving state-of-the-art results with models of various scales across multiple datasets. Furthermore, our method demonstrates robustness and superior generalization.

2 METHODS

2.1 THE ILLUSION OF HIGH REWARD

A central challenge in RL is discovering successful trajectories under a limited sampling budget. Although most approaches rely on the reward signal during training to evaluate learning quality, this signal is not always reliable or accurate. To demostrate this, we conduct a simple experiment where we train Qwen2.5-7B (Team, 2024) on the DAPO-Math-170K (Yu et al., 2025), with periodic evaluation on MATH-500 (Hendrycks et al., 2021) test set. During the training phase, if all of its rollouts for a problem are incorrect, we will give an answer-level hint to the model.

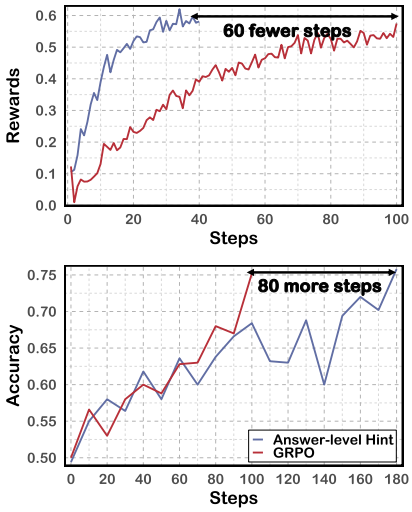

Figure 2: A comparison of training rewards (top) and test accuracy (bottom). High rewards during training do not necessarily lead to high test accuracy, indicating that reward signals may be misleading indicators of model generalization.

Figure 2 shows the outcome of this experiments. Answer-level hint rapidly boosts rewards, creating the illusion of faster convergence. However, the plot on the bottom reveals a different story, as this apparent improvement does not translate into better generalization, with test accuracy stagnating at a low level. Furthermore, providing more detailed hints does not necessarily yield better outcomes, since excessive bias may cause the model's behavior to deviate substantially from its current policy and potentially destabilize training.

The discrepancy between high training rewards and stagnant test accuracy raises a critical question: **why does an apparently strong learning signal fail to produce a generalizable policy?** Our analysis reveals that this problem originates from the severe answer leakage caused by answer-level hints. At a mechanistic level, these hints encourage large deviations from the current policy, generating updates with high importance ratios. These updates are then frequently clipped, which nullifies much of the potential learning signal. While this points to the importance of clipping, we find that its frequency alone is an incomplete indicator of training quality. The stability and diversity of the updates that survive clipping are also crucial for effective learning. To properly diagnose these dynamics, we must quantify both how much of the learning signal survives clipping and the variability of those surviving updates. This motivates our proposal of a new set of metrics to evaluate exploration efficiency and quality.

2.2 QUANTIFYING EXPLORATION EFFICIENCY AND QUALITY

The foundation for our new metrics is a direct analysis of the clipping mechanism, which constrains policy updates within a trust region (Schulman et al., 2015). While clipping improves stability, it also suppresses part of the original learning signal, making it difficult to evaluate how effectively the model leverages sampled trajectories. To quantitatively assess this, we focus on two factors that critically influence training quality: (1) the frequency with which policy updates are clipped, and (2) the variability of importance ratios. The first determines how much of the learning signal survives clipping, while the second reflects how stably the surviving updates are distributed. Building on these considerations, we introduce two complementary metrics: Effective Update Ratio (EUR) and Update Consistency (UC).

Effective Update Ratio (EUR). EUR quantifies the proportion of policy updates that remain within the trust region, thereby preserving the original learning signal. Formally, it is defined as

$$\text{EUR} = \frac{\sum_i w_i \mathbf{1}|\ell_i| \leq \delta}{\sum_i w_i}, \quad w_i = |A_i|, \quad \ell_i = \log \frac{\pi_\theta(a_i \mid s_i)}{\pi_{\theta_{\text{old}}}(a_i \mid s_i)} \tag{1}$$

In this definition, A_i denotes the advantage of sample i, w_i is the absolute advantage serving as its weight, and ℓ_i is the log-importance ratio between the new policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$. A

higher EUR indicates that most updates fall within the trust region and are therefore not suppressed, allowing the model to retain more informative gradients.

Update Consistency (UC). While EUR measures the proportion of valid updates, it does not capture the variability of those updates. To address this, we focus on the set of samples whose log-importance ratios remain within the trust region, i.e.,

$$\mathcal{I} = \{i : |\ell_i| \leq \delta\}$$

UC is then defined as the weighted standard deviation of the log-importance ratios within this set:

$$\mathrm{UC} = \sqrt{\frac{\sum_{i\in\mathcal{I}} w_i(\ell_i - \mu_\ell)^2}{\sum_{i\in\mathcal{I}} w_i}}, \quad \mu_\ell = \frac{\sum_{i\in\mathcal{I}} w_i \ell_i}{\sum_{i\in\mathcal{I}} w_i} \tag{2}$$

Here, μ_ℓ represents the weighted mean of the log-importance ratios within the trust region, and $\mathcal{I}$ denotes the corresponding index set. A low UC implies that the updates are conservative and tightly concentrated around the mean, whereas an excessively high UC suggests unstable updates driven by values approaching the trust region boundary.

While the EUR and UC provide critical insights into training, neither metric is sufficient on its own to guarantee high-quality exploration. For instance, a high EUR, which indicates that most updates are being utilized, could be deceptive if those updates are highly inconsistent (a high UC), suggesting an unstable policy on the verge of divergence. Conversely, perfect consistency (a low UC) is of little value if very few updates are effective to begin with (a low EUR), a scenario that would indicate stalled or overly conservative learning. An ideal training process must therefore achieve a balance: leveraging a high volume of effective updates that are also highly consistent. Therefore, to capture this essential synergy in a single, holistic measure, we define our unified metric, *Affinity*, as the combination of EUR and UC:

$$\textit{Affinity} = \mathrm{EUR} \cdot \exp\Big(-\tfrac{\mathrm{UC}}{\tau}\Big), \quad \tau = \delta/2 \tag{3}$$

This multiplicative formulation ensures that *Affinity* is high only when both conditions hold simultaneously: a substantial fraction of updates remain within the trust region, and their variability is moderate. As such, *Affinity* serves as a holistic indicator of exploration efficiency and training stability under online RL.

In Section 3.3, we report Affinity curves alongside reward learning curves and demonstrate that our method consistently achieves higher Affinity compared to baseline approaches, indicating more stable and effective online policy updates.

2.3 HINT: Helping Ineffective rollouts Navigate Towards effectiveness

The preceding analysis, formalized by the *Affinity* metric, reveals a central dilemma in RL which strong external guidance often degrades training quality by causing frequent clipping (low EUR) or destabilizing updates (high UC). An ideal method must therefore provide guidance that is potent enough to prevent unproductive exploration but gentle enough to maintain high *Affinity*.

To achieve this delicate balance, we introduce HINT. As illustrated in Figure 3, HINT is an adaptive mechanism that steers the model toward productive reasoning paths. HINT achieves this by sourcing heuristic hints from a stronger "teacher" model. These hints represent a higher form of guidance, operating on a conceptual level to spark a reasoning process. This methodology is designed not to provide answers directly, but to equip the model with the strategic insight needed to **formulate the solution autonomously**, a philosophy akin to the principle of "teaching one to fish".

Formally, the HINT framework operates as a two-stage process. The first stage mirrors a standard GRPO update cycle. On the rollout stage, for a given problem q, the model begins by sampling a set of trajectories $\{o_1, o_2, \ldots, o_G\}$ using its current policy. These trajectories are then evaluated by a reward model or predefined rules to obtain a set of rewards $\{r_1, r_2, \ldots, r_G\}$. If these rewards are not sparse (i.e., at least one trajectory is correct), the process proceeds identically to the GRPO algorithm. The non-sparse rewards are used to compute advantages and perform a normal policy update.

The second stage, the hint-augmented rollout, is activated **only if the initial rewards from the first stage are sparse** (i.e., all trajectories are incorrect). In this scenario, where GRPO would

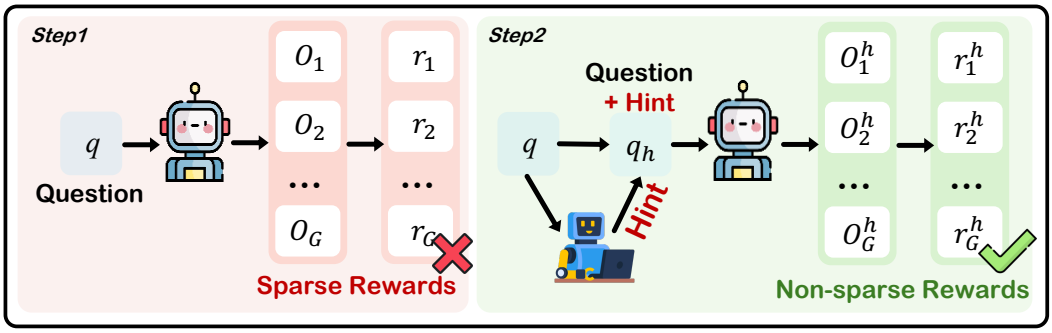

Figure 3: The HINT Framework: An Adaptive Two-Stage Rollout Process. HINT operates in two stages. **(1) Standard Rollout:** The model first samples trajectories from the original problem. If the rewards are non-sparse (at least one is correct), the process follows the standard GRPO update path. **(2) Hint-Augmented Rollout:** If, however, the rewards are sparse (all trajectories are incorrect), the hint mechanism is activated. The model then re-rolls out conditioned on a heuristic hint from a "teacher model". This stage is designed to produce non-sparse rewards, turning a failed sample into a valuable learning opportunity.

stall due to a lack of learning signal, HINT intervenes. A pre-defined hint h is used to construct a hint-augmented query q_h. The model is then prompted to resample a new set of trajectories $\{o_1^h, o_2^h, \ldots, o_G^h\}$, this time conditioned on q_h. These new, hinted trajectories are re-evaluated to produce a new set of rewards $\{r_1^h, r_2^h, \ldots, r_G^h\}$. This rescue mechanism thus turns a failed rollout into a valuable learning opportunity. By providing a heuristic hint, it is intended to enable a meaningful gradient update, which enhances training efficiency. This is accomplished while the hint itself is carefully constructed to avoid degrading training Affinity.

Mathematically, HINT optimizes the model's behavior through the following objective function:

$$\mathcal{J}_{\text{HINT}}(\theta) = \mathbb{E}_{(q,a)\sim D,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(\cdot|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big(\min(r_{i,t}(\theta)A_{i,t}, \text{clip}\left(r_{i,t}(\theta), 1\pm\epsilon\right)A_{i,t}) - \beta D_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}})\Big)\right] \tag{4}$$

where

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q^*, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q^*, o_{i,<t})}, \quad q^* = \begin{cases} q, & \sum_{i=1}^{n} f(a, o_i) > 0, \\ q_h, & \text{otherwise.} \end{cases} \tag{5}$$

In addition, we decouple the rollout prompt from the policy prompt. the rollout prompt may include the hint-augmented problem, while the policy prompt is restricted to the original problem only. This separation ensures that hints are used solely to stabilize exploration during rollouts, without leaking into the policy optimization stage, thereby preventing the model from developing a systematic reliance on hints after training.

3 Experiments

3.1 Setup

Experimental Setup. Our experiments are conducted using Qwen2.5-7B and Qwen2.5-3B (Team, 2024) as backbone models. To ensure a fair and controlled comparison, we constructed a high-quality training set derived from the DAPO-Math-170K dataset (Yu et al., 2025). This process involved using Qwen2.5-72B-Instruct (Team, 2024) to generate four distinct reasoning trajectories for each problem. These outputs were then validated for correctness with Math Verify[2], from which we retained 30k fully correct samples to form our final training data. For baseline methods that require a ground-truth reference solution, we designated the shortest of the four correct trajectories for each problem.

[2]https://github.com/huggingface/Math-Verify

Benchmarks. We evaluate the generalization ability of HINT on seven datasets, covering both in-distribution and out-of-distribution scenarios, without using any hint during evaluation. For mathematical reasoning, we adopt AIME24[3], MATH-500 (Hendrycks et al., 2021), OlympiadBench (He et al., 2024), and Minerva (Lewkowycz et al., 2022), which are widely used benchmarks. Since the test sets of AIME24 are relatively small, we report avg@32, while for the other datasets we use pass@1. To assess complex reasoning and out-of-distribution generalization, we further evaluate on ARC-Challenge (Clark et al., 2018), GPQA-Diamond (Rein et al., 2024), and MMLU-Pro (Wang et al., 2024). To demonstrate HINT effectiveness, we conduct systematic experiments across multiple benchmarks.

Baselines. We compare HINT against several existing methods designed to improve rollout accuracy rate or rollout efficiency in GRPO. The baselines include: (1)**LUFFY** (Yan et al., 2025): A hybrid approach that combines on-policy and off-policy training, ensuring that each sampled batch contains at least one correct trajectory. (2)**CHORD** (Zhang et al., 2025a): A method dynamically integrating SFT as a weighted objective within on-policy RL. (3)**GHPO** (Liu et al., 2025b): A method that adaptively adjusts the hint length based on the ground-truth solution. If a shorter hint fails to solve the problem, the hint length is progressively increased until the correct answer is obtained. (4)**QuestA** (Li et al., 2025): A method constructs the hint by using the initial 50% of a reasoning trajectory generated by a larger, more capable model. (5)**BREAD** (Zhang et al., 2025b): A binary search–based method that identifies a hint length such that the model's rollouts are neither all correct nor all incorrect, and uses this balanced point as the hint for training.

A comprehensive overview of our experimental configuration, including detailed prompts, hyperparameters, and implementation settings for all methods, can be found in the Appendix A for full reproducibility.

3.2 Main results

We benchmarked our proposed method against several mainstream approaches, including both mixed-policy strategies and other hint-based methods. These experiments were conducted on two scales of backbone models: Qwen2.5-7B and Qwen2.5-3B. We report our results in Table 1. Our analysis reveals the following key findings:

HINT enhances In-Distribution reasoning and teaches problem-solving skills. HINT significantly enhances the reasoning capabilities of models, achieving state-of-the-art performance on multiple in-distribution benchmarks. Models trained with HINT demonstrate substantial gains, with Qwen2.5-7B and Qwen2.5-3B showing average improvements of 9.0% and 6.8%, respectively, underscoring the effectiveness of our approach. We also observed an interesting emergent behavior during training: when a model encountered two similar, challenging problems, it would often rely on a hint for the first but then solve the second independently by applying the same reasoning pattern. This observation provides strong evidence that our heuristic and minimal hints teach the model how to reason about a class of problems, rather than simply encouraging it to memorize a solution path for a single instance.

HINT generalizes to Out-of-Distribution problems by optimizing reasoning paths. HINT also demonstrates strong generalization, enhancing the model's ability to tackle novel problems. Even on out-of-distribution (OOD) test sets, models trained with HINT showed marked improvements. On the OOD test sets, models trained with HINT demonstrated strong generalization, with Qwen2.5-7B and Qwen2.5-3B achieving average performance gains of 7.4% and 1.6%, respectively, highlighting the method's robust ability to generalize. This strong OOD performance is explained by a deeper phenomenon observed in our case studies. We found that the model successfully reapplies high-level reasoning methods from our hints, such as Proof by Contradiction to solve new OOD problems. This demonstrates that our method operates on a conceptual level, effectively teaching the model transferable problem-solving paradigms rather than just answers. It is this acquisition of new, abstract reasoning skills that drives the model's robust generalization.

The effectiveness of HINT scales with model size. Our results show that the benefits of HINT are more pronounced in larger models, with the performance gains for Qwen2.5-7B consistently outpacing those for Qwen2.5-3B across all evaluations. To understand the mechanism behind this

[3]https://huggingface.co/datasets/math-ai/aime24

Table 1: Main Performance Comparison of HINT against Baselines. HINT demonstrates significant performance gains on in-distribution datasets, improving the Qwen2.5-7B and Qwen2.5-3B models by **13.5%** and **6.8%**, respectively. The method also **shows strong generalization capabilities on out-of-distribution data**.

Methods	In-Distribution				Avg	Out-of-Distribution			Avg
	AIME	Math	Olympaid	Minerva		ARC	GPQA	MMLU	
Qwen2.5-7B									
Vanilla	9.8	50.2	34.0	19.5	28.4	85.3	25.6	46.0	52.3
GRPO	12.8	75.2	40.8	**31.2**	40.0	87.3	30.8	56.6	58.2
CHORD	13.2	74.4	40.0	31.2	39.7	86.6	30.1	51.2	56.0
LUFFY	12.6	70.2	38.6	30.8	38.1	87.2	**32.2**	46.8	55.4
GHPO	13.1	75.6	42.2	30.0	40.2	87.0	32.0	50.0	56.3
QuestA	13.1	73.6	38.8	28.6	38.5	88.0	26.6	53.2	55.9
BREAD	11.7	72.8	41.8	29.2	38.9	85.0	29.4	48.8	54.4
HINT	**13.3**	**79.6**	**43.6**	31.0	**41.9**	**88.8**	31.8	**58.4**	**59.7**
Qwen2.5-3B									
Vanilla	2.9	39.8	12.0	9.8	16.1	44.8	11.4	28.8	28.3
GRPO	4.3	44.0	18.2	12.2	19.7	45.0	11.8	28.0	28.3
CHORD	4.5	46.6	20.2	13.0	21.1	40.0	11.0	26.4	25.8
LUFFY	3.3	40.0	18.0	13.2	18.6	40.8	11.2	24.0	25.3
GHPO	4.0	42.2	19.6	12.8	19.7	45.5	**12.0**	28.2	28.6
QuestA	3.9	42.0	19.6	12.4	19.5	44.8	**12.0**	29.0	28.6
BREAD	4.1	44.4	**20.4**	**13.4**	20.6	45.5	11.8	29.2	28.8
HINT	**4.9**	**48.6**	20.2	**13.4**	**21.8**	**48.8**	11.8	**30.2**	**29.9**

trend, we analyzed the training rollouts and found a clear difference in how effectively each model leveraged the provided hints. A quantitative analysis confirmed that out of 100 randomly sampled rollouts where hints were provided to each model, Qwen2.5-7B produced a successful trajectory following the hint 34.0% more often than Qwen2.5-3B did. This superior efficacy in converting hints into successful outcomes directly explains the more pronounced performance gains, indicating that the greater capacity of larger models allows them to better capitalize on the abstract guidance offered by HINT.

3.3 Training Dynamics

To investigate the impact of various off-policy strategies, we tracked the EUR, UC, and *Affinity* metrics for our method alongside several key baselines which detailed in Section 3.1, with the full training dynamics plotted in Figure 4. This analysis led to the following key observations.

In the early stages of training, the model shows strong resistance to off-policy data. As illustrated in the left plot of Figure 4, all three off-policy methods exhibit a sharp drop in EUR, indicating that clipping occurs very frequently at this stage. We call this initial period the "EUR Collapse Stage", where the model is highly resistant to the off-policy data and the clipping frequency is consequently high. With more training steps, the model gradually adapts, leading to reduced clipping frequency and eventual stabilization. Notably, compared to GHPO and LUFFY, HINT achieves a higher steady-state EUR, demonstrating its superior ability to help the model accommodate and leverage off-policy data.

Over-reliance on off-policy data often prevents the model from converging. As shown in the middle plot of Figure 4, both GHPO and LUFFY quickly reach high UC values at the beginning of training and remain at that level. This indicates persistently large variance in importance sampling, which results in unstable model updates and hampers convergence. In contrast, the UC of HINT does not spike early on but instead indicates that our heuristic hints avoid casing large distributional shifts, allowing the policy updates to remain centered around a stable learning direction.

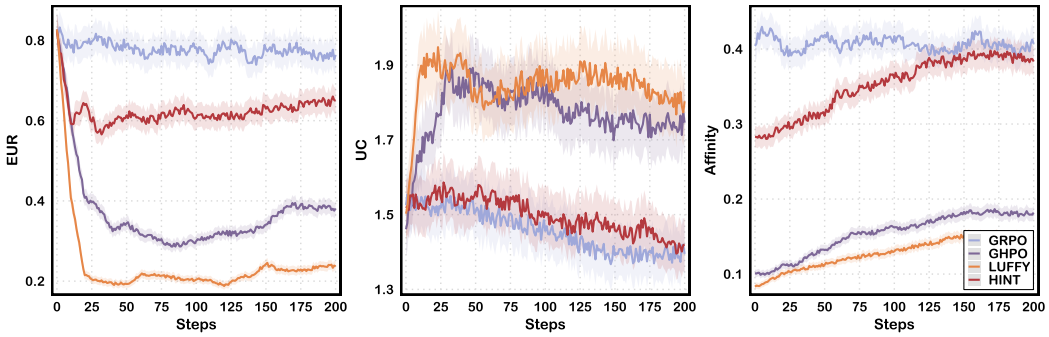

Figure 4: We record the EUR, UC, and *Affinity* metrics across different training processes to investigate the impact of various off-policy strategies on training. **Left:** EUR during training; **Middle:** UC during training; **Right:** *Affinity* during training. Overall, HINT most effectively alleviates the EUR collapse, avoids persistently high UC, and achieves higher *Affinity*, thereby enabling more stable and efficient training.

HINT enables the model to genuinely absorb the knowledge provided by hints. As presented in the right plot of Figure 4, the *Affinity* of HINT gradually approaches that of GRPO as training progresses. This implies that the model becomes increasingly capable of identifying which hints are truly useful. In other words, HINT enhances training efficiency and sample utilization in the early stages, while maintaining convergence trends consistent with GRPO in the later stages, thereby balancing early gains with eventual stability.

3.4 In-depth Analysis

Does hinting truly enhance training effectiveness? We measured the number of valid samples (i.e., rollouts that are not entirely incorrect) generated by GRPO and HINT under an equal computational budget (8 hours of training). As shown in the top of Figure 5, although HINT produced slightly fewer total samples than GRPO, it yielded a greater number of valid samples. This indicates that HINT achieves higher training efficiency under the same time constraints, suggesting that hints guide the model toward more productive exploration trajectories rather than wasting updates on implausible rollouts.

From a broader perspective of the entire training process, the proportion of valid samples with HINT is higher than that of GRPO by 18.9%, further confirming that hinting improves the signal-to-noise ratio of training data. In other words, the gradient updates induced by HINT are more likely to be based on partially correct reasoning chains, thereby amplifying useful supervision signals and mitigating the destabilizing effects of noisy rollouts.

The dominance of valid rollouts under HINT suggests that hints not only improve rollout quality but also reshape the global optimization landscape by steering policy learning toward regions where correct reasoning is more likely to occur. This mechanism explains why HINT can achieve sustained improvements even without relying on answer leakage, ultimately leading to more robust and generalizable training outcomes.

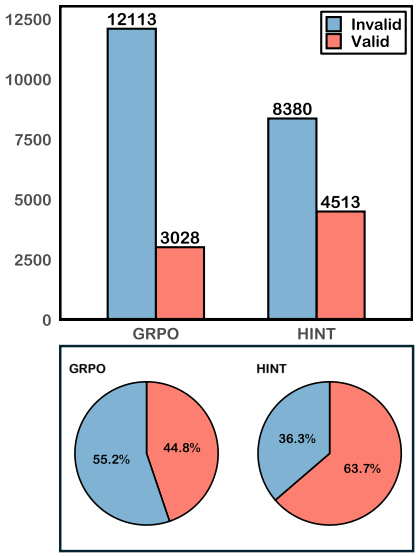

Figure 5: Sampling Efficiency of HINT and GRPO at Different Training Stages. Under an equal budget, HINT yields **1,485 more valid samples** (top) and achieves a **18.9% higher final proportion of valid samples** (bottom).

Does hinting affect the diversity of model's outputs? Entropy serves as a key metric for measuring generation diversity (Cheng et al., 2025; Zheng et al., 2025). Building on the training processes for HINT and the GHPO baseline detailed in Section 3.1, we further compared their dynamics by analyzing the average entropy of reasoning trajectories throughout the training period. For each method, we separately computed the mean entropy on samples with and without hints.

Table 2: We compare the average entropy for different methods on samples both with and without hints. The results consistently show that **HINT promotes higher entropy than answer-level hints across both scenarios**.

	w/ hint	w/o hint	All
GRPO	–	0.203	0.203
GHPO	0.123	0.141	0.129
HINT	0.188	0.198	0.193

As illustrated in Table 2, on the subset requiring hints, the entropy of HINT is notably higher than GHPO, which is answer-level hints. This is because answer-level hints often provide a "half-completed" reasoning trajectory, forcing the model to follow a predetermined path with limited exploration. In contrast, ours do not disclose specific solution steps, leaving the reasoning process entirely up to the model and thereby encouraging broader exploration across different trajectories.

Even more surprisingly, we find that on samples where no hints are needed, GHPO still yield the lowest entropy compared to both GRPO and HINT. This suggests that long-term exposure to answer-level hints suppresses diversity at a deeper level: even when no hints are provided, the model's ability to generate diverse reasoning paths is diminished.

4 Related Work

Reinforcement Learning for Large Language Model Reasoning. Recent advances in RL approaches have significantly enhanced the reasoning capabilities of LLMs. Large reasoning Models (LRMs) such as OpenAI-o1 (Jaech et al., 2024), DeepSeek-R1 (Guo et al., 2025), and Kimi-1.5 (Team et al., 2025) achieve state-of-the-art performance on complex reasoning tasks (e.g., mathematics, coding, scientific problem solving) by leveraging Reinforcement Learning from Verifiable Rewards (RLVR) (Liu et al., 2025a; Hu et al., 2025; Cui et al., 2025), where automatically checkable rules provide supervision signals. Compared to earlier methods like SFT or reinforcement learning from human feedback (RLHF), RLVR has shown superior generalization and robustness (Chu et al., 2025; Snell et al., 2025). Building on this paradigm, subsequent studies have proposed improved optimization strategies and structured prompting techniques that further strengthen reasoning capabilities (Schulman et al., 2017; Wang et al., 2020). Despite this progress, a critical failure mode for existing RL methods is reward sparsity, which occurs when all rollouts in a sample fail. Overcoming this challenge is essential for enhancing the stability and sample efficiency of training large reasoning models.

Improving Rollout Efficiency in RL for LLMs. A well-known challenge in methods such as GRPO is the vanishing gradient issue. This problem occurs when all trajectories in a sample group are incorrect, as the group advantage collapses to zero, yielding no gradient for policy updates (Shao et al., 2024; Guo et al., 2025). To mitigate this, some works have focused on injecting external, off-policy data to improve training efficiency and stability. This has been explored through two main strategies. Some methods use mixed-policy, replacing a portion of on-policy rollouts with complete, high-quality trajectories from off-policy datasets (Yan et al., 2025; Lin et al., 2025; Xu et al., 2025; Wang et al., 2025). Others employ partial supervision, providing segments of a ground truth to rescue failed rollouts (Li et al., 2025; Liu et al., 2025b; Zhang et al., 2025b). While these approaches effectively improve rollout efficiency, their over-reliance on off-policy data can misguide policy updates, steering the model toward non-generalizable or spurious solution paths.

5 Conclusion

In this work, we identify the problem of low training affinity caused by an over-reliance on off-policy data and propose HINT, an adaptive framework to resolve this trade-off. HINT significantly outperforms strong baselines on competitive math benchmarks and demonstrates robust out-of-distribution generalization. Our work showcases a scalable and principled path toward more capable, self-improving reasoning models, with future work pointing towards extending HINT to new domains and modalities.

6 Ethics Statement

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets used were sourced in compliance with relevant usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

7 Reproducibility Statement

We have made every effort to ensure that the results presented in this paper are reproducible. All code and datasets have been made publicly available in an anonymous repository to facilitate replication and verification. The experimental setup, including training steps, model configurations, and hardware details, is described in detail in the paper.

Additionally, All datasets are publicly available, ensuring consistent and reproducible evaluation results.

We believe these measures will enable other researchers to reproduce our work and further advance the field.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

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

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

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

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

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, et al. Areal: A large-scale asynchronous reinforcement learning system for language reasoning. *arXiv preprint arXiv:2505.24298*, 2025a.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning. *arXiv preprint arXiv:2506.19767*, 2025b.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

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

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

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

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

Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*, 2024.

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

Jiazheng Li, Hong Lu, Kaiyue Wen, Zaiwen Yang, Jiaxuan Gao, Hongzhou Lin, Yi Wu, and Jingzhao Zhang. Questa: Expanding reasoning capacity in llms via question augmentation. *arXiv preprint arXiv:2507.13266*, 2025.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.

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

Ziru Liu, Cheng Gong, Xinyu Fu, Yaofang Liu, Ran Chen, Shoubo Hu, Suiyun Zhang, Rui Liu, Qingfu Zhang, and Dandan Tu. Ghpo: Adaptive guidance for stable and efficient llm reinforcement learning. *arXiv preprint arXiv:2507.10628*, 2025b.

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

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

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

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

Yuhui Wang, Hao He, and Xiaoyang Tan. Truly proximal policy optimization. In *Uncertainty in artificial intelligence*, pages 113–122. PMLR, 2020.

Zhenting Wang, Guofeng Cui, Yu-Jhe Li, Kun Wan, and Wentian Zhao. Dump: Automated distribution-level curriculum learning for rl-based llm post-training. *arXiv preprint arXiv:2504.09710*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yixuan Even Xu, Yash Savani, Fei Fang, and Zico Kolter. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.

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

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

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

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. *arXiv preprint arXiv:2508.11408*, 2025a.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. Bread: Branched rollouts from expert anchors bridge sft & rl for reasoning. *arXiv preprint arXiv:2506.17211*, 2025b.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. *arXiv preprint arXiv:2504.07912*, 2025.

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, et al. First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025.

APPENDIX

A EXPERIMENTAL DETAILS

A.1 DETAILED SETUP

Platform. All of our experiments are conducted on workstations equipped with 8 NVIDIA A100 PCIe GPUs with 80GB memory.

Training Data. The training was performed using a carefully selected subset of the DAPO-Math-170K dataset (Yu et al., 2025). As the original dataset lacks ground-truth solutions, we curated our own by first using Qwen2.5-72B-Instruct to generate four reasoning trajectories for each problem. After validating the final answers with *Math-verify*, we compiled a high-quality training set of 30k problems for which all four generated trajectories were correct. For baselines requiring a ground truth, the most token-efficient of these four correct trajectories was designated as the ground truth. For our methods, we pre-generated the required heuristic hints for the entire 30k-sample training set using Qwen2.5-72B-Instruct. The prompts used in the above process will be detailed in Section A.2.

Important Parameters of HINT. HINT is implemented based on the open-source Rl framework lsrl[4]. The RL algorithm employs the GRPO advantage estimator with no KL penalty (kl_coef is set to 0.0). The clipping parameter ϵ is set to 0.2. For each group, 8 answers are generated, and the training batch size is set to 2. Distributed training utilizes the DeepSpeed library with the *AdamW* optimizer and a learning rate of 1e-6. The *train batch size* is set to 8, *gen batch size* is set to 32, *accum steps* is set to 64, *gen update steps* is set to 128, *temperature* is set to 0.9, *max response* is set to 4096. Mixed-precision training with BF16 is enabled. Memory optimization employs ZeRO Stage 2, with optimizer state offloading to CPU.

Important Parameters of Other Baselines. For baselines with publicly available code repositories, we utilized their official implementations and the parameters specified in their respective publications. For methods without public code, such as BREAD(Zhang et al., 2025b) and QuestA(Li et al., 2025), we reproduced their results using the lsrl framework, strictly adhering to the experimental parameters detailed in their papers.

Reward Setup. For our experiments, we employ a sparse, binary reward function. The reward is determined exclusively by the correctness of the final answer in a model's generated trajectory. We use the *Math-Verify* tool for automatic verification, assigning a reward of **+1** for a correct final answer and **0** for an incorrect one.

A.2 PROMPT LIST

Prompt Template for GRPO

System: You are a helpful AI assistant. A conversation takes place between the User and the Assistant. The User asks a question, and the Assistant solves it. Please help me solve this question. Wrap only the final answer in $\backslash\backslash boxed\{\}$.

Question: [Question]

User:

[4] https://github.com/lsdefine/lsrl

Prompt Template for HINT

System: You are a helpful AI assistant. A conversation takes place between the User and the Assistant. The User asks a question, and the Assistant solves it. Please help me solve this question. Wrap only the final answer in $\backslash\backslash boxed\{\}$.

Hint: Here are some key information provided to assist you in solving the problem: [Hint]

Question: [Question]

User:

Prompt Template for Generating hints

System:
* Role and Goal
You are a top-tier problem-solving expert and a master educator. Your goal is not to solve the problem, but to distill the single most critical "Core Insight" or "Aha! Moment" required to find the solution.
* Core Task
You will be given a [Question] and its final [Answer]. Your sole job is to reverse-engineer the most likely solution path and identify the crucial "mental bridge"—the non-obvious insight, change in perspective, or core principle—that unlocks the problem.
* Thinking Framework
Analyze the Gap: First, understand the [Question] and look at the [Answer]. The core difficulty lies in the conceptual space between them. What makes bridging this gap non-trivial? Reconstruct the "Hidden" Step: Mentally construct the most elegant solution path. In that path, what is the single most pivotal, non-obvious leap of logic or application of a principle that a student is most likely to miss? Distill the Insight: Condense this pivotal leap into an extremely short, potent, and core-focused sentence. This sentence is the key that unlocks the door, not the map of the room.
* Constraints
Absolute Brevity: The insight must be a single sentence, ideally under 20 words. No Spoilers: The insight must not reveal any part of the [Answer] or the specific numbers used to calculate it. Inspirational, Not Instructional: It should inspire thought ("heuristic"), not provide a step-by-step recipe ("algorithmic"). Target the Crux: It must address the most critical linchpin that makes the entire solution possible.
* Output Format
Directly output the single, distilled "Core Insight". Do not include any other explanations, headings, or conversational text.

User:
Question:
[Question]
Answer:
[Answer]

Prompt Template for Generating Ground Truth

System: You are a helpful AI assistant. A conversation takes place between the User and the Assistant. The User asks a question, and the Assistant solves it. Please help me solve this question. Wrap only the final answer in $\backslash\backslash boxed\{\}$.

Question: [Question]

User:

Prompt Template for Evaluation

System: You are a helpful AI assistant. A conversation takes place between the User and the Assistant. The User asks a question, and the Assistant solves it. Please help me solve this question. Wrap only the final answer in $\backslash\backslash boxed\{\}$.

Question: [Question]

User:

B Further Analysis

B.1 Details of HINT's Entropy

HINT Encourages Sustained Exploration. The entropy of the generation distribution serves as a key indicator of exploration diversity. As illustrated in Figure 6, HINT avoids the rapid entropy collapse observed in GRPO during the early stages of training. Instead, HINT maintains a consistently high level of entropy, indicating that the model actively explores when first introduced to the hints. This period of high exploration corresponds directly to the "EUR collapse" phase (discussed in Section 3.3), explaining that while the model initially resists the off-policy guidance, it is nevertheless engaged in a productive and diverse search of the solution space.

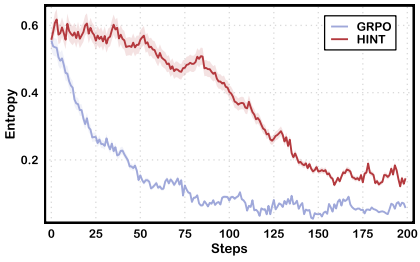

Figure 6: **HINT Prevents Entropy Collapse and Encourages Sustained Exploration.** HINT maintains a high entropy level, especially in the early stages, and stabilizes at a significantly higher value. This demonstrates that HINT's heuristic guidance fosters more continuous and diverse exploration, preventing premature policy convergence.

During the middle stages of training, HINT's entropy does not decrease monotonically. It exhibits periodic increases. We attribute this to the model encountering novel types of hints and adapting its exploratory behavior to learn how to utilize them. Crucially, even after the policy stabilizes in the later stages, HINT maintains a significantly higher entropy level than GRPO. This provides strong evidence that HINT's heuristic guidance successfully fosters more continuous and diverse exploration, preventing the policy from prematurely converging to a deterministic state.

B.2 Details of HINT's Accuracy

Our results reveal an interesting trade-off: while the off-policy guidance from HINT may initially slow the rate of convergence, it ultimately enables the model to achieve a higher performance ceiling. As shown in Figure 7, HINT initially exhibits a slower rate of accuracy improvement compared to GRPO. This initial lag is consistent with the early training stages where the model shows resistance to the heuristic hints and has not yet learned to leverage them effectively. However, as training progresses, the model begins to adapt and utilize the guidance. This leads to an accelerated learning rate after approximately 100 steps, with HINT's accuracy eventually surpassing GRPO's and reaching a higher final value. This dynamic suggests that the model requires an adaptation period to master the use of heuristic hints, but once learned, this skill allows it to develop stronger and more robust capabilities.

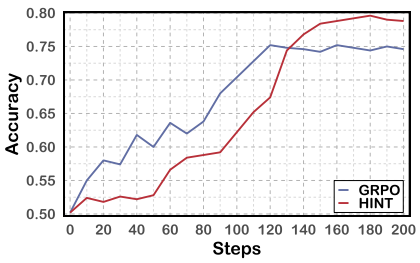

Figure 7: Accuracy of Different Methods. **HINT Achieves Higher Final Accuracy Despite Slower Initial Convergence.**

B.3 Case Study

Case Study

Question Let a and b be positive integers such that all but 2009 positive integers are expressible in the form $ma + nb$, where m and n are nonnegative integers. If 1776 is one of the numbers that is not expressible, find $a + b$.

Hint: How many positive integers cannot be expressed in the form $ma+nb$ with nonnegative integers m, n?

Before the Hint:

By the two-coin Frobenius result, for coprime positive integers a, b the number of positive integers not representable as $ma + nb$ equals $\frac{(a-1)(b-1)}{2}$.

So

$$\frac{(a-1)(b-1)}{2} = 2009$$

$$(a-1)(b-1) = 4018$$

Factor $4018 = 41 \cdot 98$. Take $a = 42, b = 99$.

Hence $a + b = 141$.

Answer: $\boxed{141}$ ✖

After the Hint:

By the two-coin Frobenius result, for coprime positive integers a, b the number of positive integers not representable as $ma + nb$ equals $\frac{(a-1)(b-1)}{2}$.

So

$$\frac{(a-1)(b-1)}{2} = 2009$$

$$(a-1)(b-1) = 4018$$

Candidates:

$(2, 4019), (3, 2010), (8, 575),$

$(15, 288), (42, 99), (50, 83).$

Require $\gcd(a, b) = 1$,

leaving $(2, 4019), (8, 575), (50, 83)$.

Check 1776: representable for first two, not for $(50, 83)$.

Thus $a + b = 133$.

Answer: $\boxed{133}$ ✔

Why the hints works in this question?

The key to solving this problem is identifying it as an instance of the **Frobenius Coin Problem**. This theorem states that for two coprime positive integers, a and b, the number of positive integers not expressible in the form $ma + nb$ is precisely $\frac{(a-1)(b-1)}{2}$. Without guidance, the model correctly sets up the equation $(a-1)(b-1) = 4018$ but may hastily select a factor pair, such as (42, 99), while neglecting the crucial precondition that a and b must be coprime. As shown in the "Before the Hint" example, this oversight leads to an incorrect answer.

The provided hint is designed to steer the model away from this pitfall. By asking about the *number* of non-representable integers, the hint explicitly directs the model's attention toward the Frobenius formula. This encourages a more rigorous, systematic approach: first, finding all possible integer pairs for (a, b); second, filtering these candidates by checking the essential coprimality condition ($\gcd(a, b) = 1$); and finally, verifying which of the remaining valid pairs satisfies the constraint that 1776 is non-representable. This structured reasoning process, prompted by the hint, is effective because it signals the specific theoretical framework needed to solve the problem, thereby preventing common errors and guiding the model to the correct solution.

B.4 Algorithm Details

Algorithm 1 HINT: Helping Ineffective rollouts Navigate Towards effectiveness

Input: initial policy model $\pi_{\theta_{\text{init}}}$; reward models r_ϕ; task prompts $\mathcal{D}$; hints $\mathcal{H}$; hyperparameters ϵ, β, μ
policy model $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration = 1, ..., I **do**
  reference model $\pi_{\text{ref}} \leftarrow \pi_\theta$
  for step = 1, ..., M **do**
    Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
    Update the old policy model $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
    ▷ *Stage 1: Standard Rollout*
    Sample G outputs $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$ for each $q \in \mathcal{D}_b$
    Compute rewards $\{r_{ij}\}_{i=1}^{G}$ for each o_i by running r_ϕ
    ▷ *Stage 2: Hint-Augmented Rollout (if necessary)*
    if all rewards $\{r_{ij}\}$ are sparse (e.g., zero) **then**
      Get hint $h \in \mathcal{H}$ for problem q
      Construct hint-augmented query q_h
      Resample G new outputs $\{o_i^h\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q_h)$
      Compute new rewards $\{r_{ij}^h\}_{i=1}^{G}$
      Let $\{o_i\} \leftarrow \{o_i^h\},\ \{r_{ij}\} \leftarrow \{r_{ij}^h\}$
    end if
    Compute $\hat{A}_{i,t}$ for each token t of o_i using final rewards
    for HINT iteration = 1, ..., μ **do**
      Update π_θ by maximizing GRPO objective
    end for
    Update r_ϕ via replay training
  end for
end for
Output: π_θ

C LLM Usage

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

It is important to note that the LLM was not involved in the ideation, research methodology, or experimental design. All research concepts, ideas, and analyses were developed and conducted by the authors. The contributions of the LLM were solely focused on improving the linguistic quality of the paper, with no involvement in the scientific content or data analysis.

The authors take full responsibility for the content of the manuscript, including any text generated or polished by the LLM. We have ensured that the LLM-generated text adheres to ethical guidelines and does not contribute to plagiarism or scientific misconduct.