DG-DAPO: Dual Outcome-Gated Decoupled Advantage Policy Optimization for Mathematical Reasoning

Anonymous ACL submission

Abstract

In applying reinforcement learning (RL) to enhance the mathematical reasoning of large language models, naively aggregating heterogeneous signals such as process rewards and length constraints can introduce scale mismatches and reward hacking. The policy may exploit process rewards through verbose reasoning or shorten responses merely to avoid penalties for excessive length, without solving the problem. To address these issues, we propose DG-DAPO, a process-guided RL framework with a dual outcome-gated mechanism. DG-DAPO first uses decoupled advantage estimation to separately normalize outcome, length, and process rewards, reducing scale interference across reward dimensions. Since decoupling alone cannot prevent auxiliary rewards from reinforcing incorrect trajectories, DG-DAPO further conditions both process rewards and soft length penalties on outcome correctness, ensuring that auxiliary supervision is applied only when aligned with the final answer. This design reduces interference from auxiliary rewards and better aligns intermediate supervision with outcome correctness. Experiments on mathematical reasoning benchmarks, including AIME 2024, GSM8K, and MATH-500, show that DG-DAPO improves overall reasoning performance while stabilizing RL training and mitigating reward-hacking behaviors caused by misaligned auxiliary rewards.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in complex mathematical reasoning (Jaech et al., 2024; Guo et al., 2025). In recent years, Reinforcement Learning (RL) has emerged as a pivotal paradigm in LLM training to further elicit such advanced reasoning proficiency (Ouyang et al., 2022; Chen et al., 2025b). Specifically, by eliminating the memory-intensive value model to drastically reduce computational overhead, lightweight RL algorithms, notably Group Relative Policy Optimization (GRPO) and its variants (Shao et al., 2024; Yu et al., 2026), have become the primary framework for unlocking multi-step logical deduction in state-of-the-art models.

To address the inherent sparsity of outcomes in multi-step tasks, existing studies frequently integrate auxiliary supervisory signals to further enhance the model's overall reasoning performance (Uesato et al., 2022; Yuan et al., 2024; Liu et al., 2024; Setlur et al., 2025; Sun et al., 2025; Guo et al., 2025). For instance, recent studies employ Process Reward Models (PRMs) for step-level feedback to provide richer supervisory signals (Wang et al., 2024b; Jin et al., 2025; Xie et al., 2025; Chen et al., 2025a; Zou et al., 2026). However, naively aggregating these heterogeneous rewards leaves models fundamentally susceptible to reward hacking, where policies exploit uncalibrated signals to artificially inflate intermediate scores or prematurely abandon necessary logical exploration to evade length penalties (Everitt et al., 2017, 2021; Gao et al., 2023; Weng, 2024).

In response to these challenges, existing methods broadly fall into two main categories. One line of work explores advanced credit assignment to calibrate process signals and smooth reward variance (Rezaei et al., 2026; Cheng et al., 2026). Yet, these methods may still assign positive learning signals to locally plausible intermediate steps, even when the overall trajectory ultimately leads to an incorrect answer. Thus, locally coherent steps within an ultimately incorrect trajectory can still accumulate positive reinforcement, subtly perpetuating reward hacking. Another line of work focuses on multi-reward structural optimization, treating different reward signals as parallel optimization dimensions (Wang et al., 2024a; Zhang et al., 2025a). However, in practice, this formulation often induces conflicting gradients. Specifically, during early RL updates, the policy might conservatively generate

brief, incorrect answers merely to escape length penalties, failing to recognize that outcome correctness inherently precedes verbosity.

Motivated by persistent challenges, we propose DG-DAPO as an adaptive reinforcement learning framework guided by process supervision. Based on DAPO (Yu et al., 2026) and inspired by the gating philosophy (Liu et al., 2026), we introduce a dual outcome-gated mechanism to restructure the integration of multiple rewards. Our framework mitigates reward hacking and reduces objective conflicts through two key designs. First, decoupled advantage estimation standardizes disparate reward scales to prevent highly variable signals from dominating optimization. Second, we strictly condition dense process rewards and soft length penalties on outcome correctness. Using the final outcome as a hard gate, globally incorrect trajectories receive maximum process and length penalties. This design suppresses deceptive local feedback and prevents the policy from prematurely abandoning necessary logical exploration for superficial length rewards, ensuring a consistent optimization direction toward the primary reasoning goal.

Extensive experiments on challenging mathematical reasoning benchmarks, including AIME 2024 (Mathematical Association of America, 2024), GSM8K (Cobbe et al., 2021), and MATH-500 (Lightman et al., 2024), using the Qwen2.5 series (Yang et al., 2024) demonstrate that DG-DAPO improves accuracy on AIME 2024 and GSM8K while maintaining comparable performance on MATH-500, with improved training stability and reduced response length. Crucially, our framework helps mitigate reward hacking and reduces redundant reasoning while preserving effective exploration.

In summary, our main contributions are as follows:

- We systematically analyze the reward hacking vulnerabilities in process-supervised RL for mathematical reasoning. We identify that naively aggregating heterogeneous rewards induces severe scale mismatch and misleading auxiliary supervision, allowing locally plausible but globally incorrect trajectories to receive undesired reinforcement.

- We propose DG-DAPO, which integrates decoupled advantage estimation to standardize reward scales and a dual outcome-gated mechanism. By strictly conditioning process guidance and length penalties on outcome correctness, it prevents high-variance auxiliary rewards from overshadowing the primary objective and effectively mitigates reward hacking.

- We empirically validate DG-DAPO on mathematical reasoning benchmarks. Compared with the DAPO baseline, DG-DAPO improves selected accuracy metrics and reduces response length growth, suggesting improved training stability.

2 Related Work

2.1 Advances in Policy Optimization

Several extensions of GRPO (Shao et al., 2024) have been proposed to enhance the stability and computational efficiency of the framework. To mitigate training instability, GVPO (Zhang et al., 2026) integrates the analytical solution of KL-constrained reward maximization directly into the gradient weights, ensuring policy alignment. Parallel to this, GSPO (Zheng et al., 2025) shifts the focus from token-level to sequence-level optimization by redefining the importance ratio based on sequence likelihood. Meanwhile, computational efficiency and training throughput are further accelerated by DAPO (Yu et al., 2026) via decoupled clipping mechanisms and dynamic sampling strategies. Additionally, to promote efficient reasoning and control resource overhead, GFPO (Shrivastava et al., 2025) addresses length explosion by sampling larger groups per problem during training and filtering responses based on their length or reward-per-token ratio. For complex multi-turn agent tasks, GiGPO (Feng et al., 2026) introduces a nested grouping architecture to enable fine-grained credit assignment.

2.2 Reward Design and Process Supervision

Beyond algorithmic enhancements, multi-objective reward design addresses the limitations of monolithic evaluation. Fine-Grained RLHF (Wu et al., 2023) decomposes global preferences into distinct attribute dimensions. To balance utility and risk, Safe RLHF (Dai et al., 2024) decouples alignment into separate performance and safety assessments. Furthermore, ALARM (Lai et al., 2024) introduces a hierarchical framework that integrates high-level holistic preferences with low-level criteria to ensure alignment consistency.

To provide the fine-grained resolution lacking in trajectory-level approaches, PRMs intro-

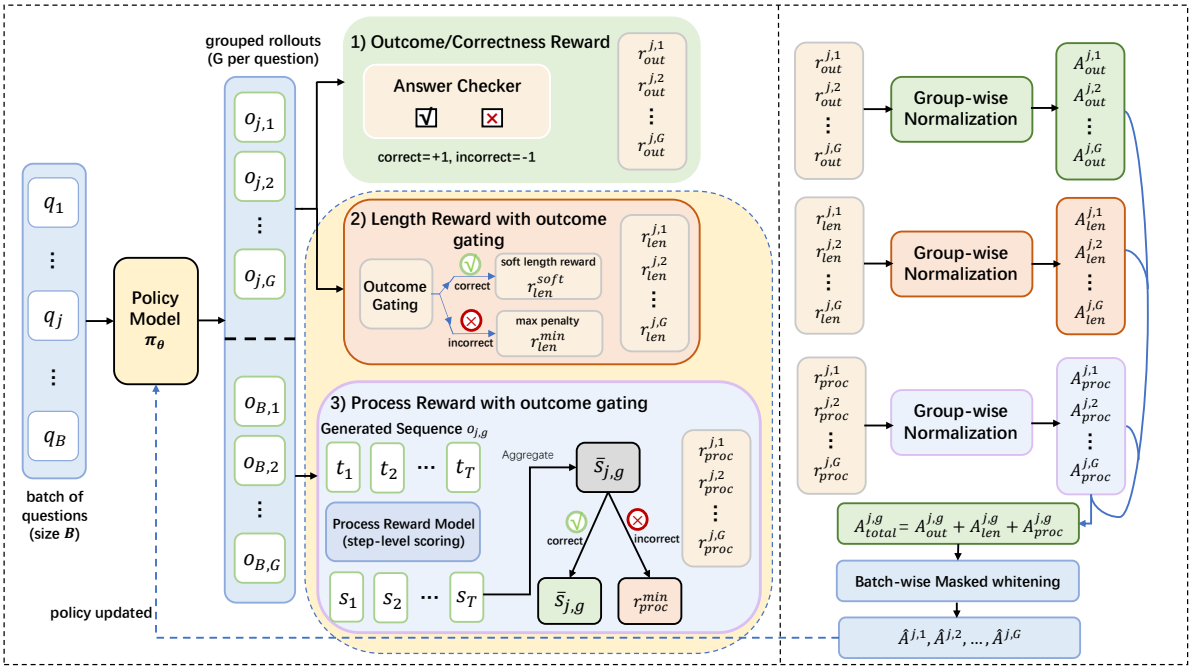

Figure 1: An overview of the proposed Dual Outcome-Gated Decoupled Advantage Policy Optimization (DG-DAPO) framework. Here, j indexes the question and g indexes the rollout within the group. In the following derivations, we omit the question index for simplicity and use j to denote a rollout within a fixed group.

duce dense step-level supervision, rapidly evolving from human-annotated verification (Lightman et al., 2024) to scalable automated feedback (Wang et al., 2024b) and generative verification paradigms (Zhao et al., 2026).

However, naively aggregating these step-level signals with trajectory penalties often induces reward hacking. Unlike parallel optimization approaches, we introduce a dual outcome-gated Mechanism to ensure outcome correctness gates auxiliary feedback.

3 Methodology

In this section, we formulate DG-DAPO, a multi-reward reinforcement learning framework that integrates heterogeneous supervisory signals for complex mathematical reasoning. We first introduce a decoupled advantage estimation strategy to standardize disparate reward scales, preventing high-variance auxiliary signals from overshadowing the primary objective. We then detail our core dual gate design, comprising an outcome-gated soft length penalty to eliminate length-induced reward noise, and an outcome-gated process reward that conditions intermediate feedback on outcome correctness to prevent reward hacking. Finally, we present the joint optimization objective to formulate the ultimate training loss of our framework.

3.1 Decoupled Multi-Reward Advantage Estimation

In the reinforcement learning task, given a user query q, the state s_t corresponds to the context comprising the query and previously generated tokens $o_{<t}$. The policy π_θ executes an action a_t by sampling the next token from its vocabulary to construct a complete reasoning rollout o. During training, for each query q, the policy generates a group of G reasoning rollouts $\{o_1, \ldots, o_G\}$. To guide the policy gradient update by quantifying the relative superiority of each rollout, standard GRPO (Shao et al., 2024) estimates an advantage score $\hat{A}_j$. When integrating K heterogeneous supervisory signals, the distinct rollout rewards $r_k^{(j)}$ are naively summed before group-wise normalization to compute this advantage:

$$\hat{A}^{(j)} = \frac{\sum_{k=1}^{K} r_k^{(j)} - \mu(\sum_k r_k)}{\sigma(\sum_k r_k)}, \qquad (1)$$

where μ and σ denote the mean and standard deviation across the G rollouts.

However, this naive aggregation introduces a severe scale mismatch. Sparse binary signals (e.g., outcome correctness $r_{out} \in \{-1, 1\}$) and dense continuous scores (e.g., step-level log-probabilities r_{proc}) operate on vastly different numerical scales.

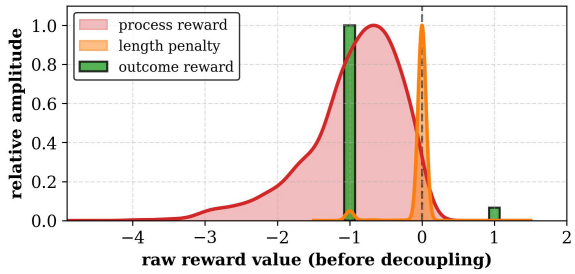

Figure 2: Reward variance mismatch (peak-normalized). The un-decoupled process reward exhibits an expansive scale and high variance, potentially dominating the optimization landscape.

As depicted in Figure 2, uncalibrated process rewards typically manifest as large negative values, disproportionately skewing the pre-normalized sum and effectively drowning out critical sparse outcome signals. Furthermore, as noted by (Liu et al., 2026), direct summation causes different reward combinations to collapse into identical advantage values, severely limiting the precision of the training signal. To resolve these challenges, we integrate a decoupled advantage estimation strategy into our framework. By independently normalizing each reward dimension prior to aggregation, this approach eliminates scale mismatches and preserves the high-resolution gradient contributions of all signals.

Specifically, we tailor this decoupled formulation to a tripartite reward space for long-CoT mathematical reasoning, including outcome correctness (r_{out}), sequence length (r_{len}), and fine-grained process quality (r_{proc}). Rather than summing these heterogeneous signals directly, we compute the independent advantages as follows:

$$A_k^{(j)} = \frac{r_k^{(j)} - \mu(r_k^{(1\ldots G)})}{\sigma(r_k^{(1\ldots G)})}, \tag{2}$$

where $k \in \{out, len, proc\}$. By standardizing the gradient contributions within the group, this approach mitigates scale gaps between sparse outcome constraints and dense process supervision, reducing the need for hand-crafted reward-scale calibration. Since this decoupled strategy effectively balances the impact of each reward dimension, we compute the total advantage by directly summing these standardized components:

$$A_{total}^{(j)} = A_{out}^{(j)} + A_{len}^{(j)} + A_{proc}^{(j)}. \tag{3}$$

To prevent the summation of normalized components from expanding the overall variance and destabilizing training, we apply a final batch-wise masked whitening step over $A_{total}^{(j)}$ across the batch before computing the policy gradient loss.

While decoupling enhances numerical stability, unconstrained length penalties and process scores remain highly vulnerable to reward hacking. To address this, we formulate r_{len} and r_{proc} via dual adaptive gating mechanisms. As detailed in the following subsections, these mechanisms strictly condition all intermediate supervision on global correctness, ensuring robust policy optimization.

3.2 Outcome-Gated Soft Length Penalty

Applying strict binary length truncation introduces severe reward noise (Liu et al., 2025b; Wu et al., 2025; Xiang et al., 2025; Liu et al., 2025a) for problems requiring extensive reasoning. To provide a smoother optimization landscape, continuous soft penalties (Yu et al., 2026) have been proposed. However, unconditionally applying this mechanism causes the policy to prematurely terminate generation, abandoning difficult mathematical exploration for the easily attainable reward of brevity.

To resolve this conflict between sequence brevity and outcome correctness, we propose an outcome-gated soft length penalty. Building upon the rigid binary gating mechanism introduced in GDPO (Liu et al., 2026), we combine strict outcome gating with continuous length regularization.

Concretely, we first define the raw soft penalty $r_{len}(o)$ following DAPO, which introduces a buffer length (L_{cache}) prior to the hard limit (L_{max}) for a sequence o:

$$r_{\text{len}}(o) = \begin{cases} 0, & |o| \le L_{\text{safe}} \\ \frac{L_{\text{safe}} - |o|}{L_{\text{cache}}}, & L_{\text{safe}} < |o| \le L_{\text{max}} \\ -1, & |o| > L_{\text{max}} \end{cases}, \tag{4}$$

where $L_{\text{safe}} = L_{\text{max}} - L_{\text{cache}}$ defines the length of the penalty-free threshold.

To mitigate the reluctance to explore induced by length penalties, we strictly condition the soft penalty on the binary outcome reward ($r_{out} \in \{-1, 1\}$):

$$\tilde{r}_{len}(o) = \begin{cases} r_{len}(o), & \text{if } r_{out} = 1 \\ -1, & \text{otherwise} \end{cases}. \tag{5}$$

This adaptive gating mechanism effectively harmonizes the dual objectives. For correct responses, the soft penalty provides a graded reward signal

that encourages concise valid reasoning chains. Conversely, by imposing a strict penalty on all incorrect responses, it prevents the model from gaming the length objective, ensuring that sequence brevity is only rewarded after mathematical correctness is achieved.

3.3 Outcome-Gated Process Reward

Incorporating dense supervision through a PRM unconditionally triggers severe reward hacking, as the policy exploits local scores by generating verbose, locally valid, but globally directionless reasoning to accumulate rewards without solving the problem.

While existing countermeasures, such as advanced credit assignment (Rezaei et al., 2026; Cheng et al., 2026) or multi-reward optimization (Wang et al., 2024a; Zhang et al., 2025a), attempt to mitigate this, they are still prone to limitations, such as inadvertently rewarding globally incorrect trajectories or inducing conflicting gradients. To structurally eliminate this vulnerability, we design an outcome-gated process reward. By explicitly conditioning process supervision on global correctness, we ensure that local logic only receives positive reinforcement when the ultimate problem is solved.

Concretely, we first formulate the unconstrained process signal. Given a response parsed into N distinct reasoning steps, a pre-trained PRM yields a probability score $p_i \in (0, 1]$ indicating the validity of the i-th step. To ensure that a single catastrophic reasoning error heavily penalizes the entire response, we compute the log-average probability across all steps to define the raw process reward $r_{proc}(o)$:

$$r_{proc}(o) = \frac{1}{N} \sum_{i=1}^{N} \log(p_i). \tag{6}$$

We then apply our adaptive gating mechanism to this aggregated process reward:

$$\tilde{r}_{proc}(o) = \begin{cases} r_{proc}(o), & \text{if } r_{out} = 1 \\ r_{proc}^{min}, & \text{otherwise} \end{cases}. \tag{7}$$

By setting the penalty for an incorrect outcome (r_{proc}^{min}) strictly lower than the minimal potential score of any valid process, we prevent the policy from accumulating false rewards through superficially plausible intermediate steps. Specifically, although the PRM output is bounded within $[0, 1]$, the averaged log-probability score does not have a finite lower bound when probabilities approach zero. To obtain a well-defined lower-bound signal in practice, we clamp the PRM probabilities with a small constant ϵ_p, which makes the minimum attainable averaged log-probability controlled by $\log \epsilon_p$. We then assign an incorrect trajectory a fixed lower-bound process reward r_{min}^{proc}. Ultimately, this strict penalty formulation better aligns intermediate logical coherence with terminal outcome correctness.

3.4 Final Optimization Objective

We formulate our final policy update by integrating our gated multi-dimensional rewards into the DAPO framework (Yu et al., 2026). While DAPO establishes a robust optimization engine via token-level importance sampling and asymmetrically decoupled clipping, it remains inherently vulnerable to reward hacking when evaluating uncalibrated heterogeneous signals.

By structurally embedding our outcome-gated decoupled advantage into this robust baseline, we define the token-level clipped objective $L_{j,t}^{\text{clip}}(\theta)$ as:

$$L_{j,t}^{\text{clip}}(\theta) = \min \left(\rho_{j,t}(\theta) \hat{A}_{\text{total}}^{(j)},\ \tilde{\rho}_{j,t}(\theta) \hat{A}_{\text{total}}^{(j)} \right), \tag{8}$$

where $\rho_{j,t}(\theta) = \frac{\pi_\theta(o_{j,t}|q, o_{j,<t})}{\pi_{\theta_{old}}(o_{j,t}|q, o_{j,<t})}$ is the importance sampling ratio for the t-th token, and the asymmetrically clipped ratio $\tilde{\rho}_{j,t}(\theta)$ is defined as $\text{clip}(\rho_{j,t}(\theta), 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}})$. Crucially, $\hat{A}_{\text{total}}^{(j)}$ now denotes the globally whitened, decoupled advantage computed exclusively using our fortified, outcome-gated signals $(r_{out}, \tilde{r}_{len}, \tilde{r}_{proc})$.

Following DAPO, we explicitly omit the conventional KL divergence penalty to allow the policy to diverge sufficiently for complex Long-CoT reasoning. Because RL aims to maximize this advantage, our overall policy gradient loss function to be minimized is aggregated across all valid tokens and responses:

$$\mathcal{J}(\theta) = -\mathbb{E} \left[\frac{1}{\sum_{j=1}^{G} |o_j|} \sum_{j=1}^{G} \sum_{t=1}^{|o_j|} L_{j,t}^{\text{clip}}(\theta) \right]. \tag{9}$$

Crucially, the expectation is evaluated under DAPO's dynamic sampling constraint ($0 < |\{o_j \mid \text{is_correct}(a, o_j)\}| < G$), where a is the corresponding ground-truth answer for question q. Such dynamic sampling filters out zero-variance groups that yield no gradient signals, directly stabilizing updates and improving overall sample efficiency.

Method	AIME 2024		MATH-500		GSM8K	
	Pass@1	Maj@32	Pass@1	Maj@32	Pass@1	Maj@32
Qwen2.5-3B-Instruct	5.73	13.30	62.28	68.4	84.51	89.99
Qwen2.5-Math-7B	10.73	**26.67**	-	-	-	-
DAPO	9.47	16.67	**65.14**	**70.00**	85.91	90.83
DG-DAPO (Ours)	**10.93**	16.67	65.11	**70.00**	**86.40**	**91.05**

Table 1: Comparison of baseline and our DG-DAPO on Pass@1 and Maj@32 accuracy across mathematical reasoning benchmarks. All models are initialized from Qwen2.5-3B-Instruct and evaluated using 32 sampled trajectories per problem.

4 Experiments

4.1 Experimental Setup

Datasets. We train our RL policy on an English-only subset of the DAPO-math-17k dataset. For evaluation, we used three standard mathematical reasoning benchmarks, including **AIME 2024** (Mathematical Association of America, 2024), **GSM8K** (Cobbe et al., 2021) and **MATH-500** (Lightman et al., 2024).

Evaluation Metrics. For each problem, we sample 32 trajectories to ensure robust evaluation. We report **Pass@1** as the expected accuracy over these 32 samples to evaluate the base reasoning capability. To assess the reliability and consensus of the policy, we report the majority voting accuracy (**Maj@32**).

Model Configurations. We initialize our policy network with the **Qwen2.5-3B-Instruct** model (Yang et al., 2024), leveraging its robust foundational instruction-following capabilities. For the fine-grained process supervision, we adopt **Qwen2.5-Math-PRM-7B** (Zhang et al., 2025b) as our process reward model.

Implementation Details. Our training infrastructure is built upon the highly efficient `verl` (Sheng et al., 2024) framework, executed on a cluster of 2×NVIDIA A100 (80GB) GPUs. We optimize the policy using the AdamW optimizer (Loshchilov and Hutter, 2017). Comprehensive implementation details and hyperparameter configurations are deferred to Appendix A.

4.2 Main Results

Under the same experimental setting, Table **??** compares the performance of our DG-DAPO framework against the base model, the math-specialized Qwen2.5-Math-7B, and standard DAPO across three mathematical reasoning benchmarks.

Overall Performance. DG-DAPO successfully integrates dense process supervision within a gating mechanism, ensuring that the model consistently optimizes toward the primary objective. For complex reasoning tasks such as AIME 2024, this approach effectively breaks the performance bottleneck caused by the reward sparsity of standard DAPO. By providing healthy step-level guidance, the dual gate mechanism prevents the policy from wandering into invalid reasoning paths, yielding relative performance improvements over the baselines. Ultimately, these results demonstrate the overall effectiveness and robustness of our proposed framework.

Robustness Across Difficulty Levels. Beyond extreme-difficulty reasoning, DG-DAPO exhibits generalization across varying complexities. It establishes the highest performance on the fundamental GSM8K benchmark while remaining highly competitive on the intermediate MATH-500. This consistent progression confirms that our framework provides stable guidance without overfitting to specific difficulty levels.

Balancing Exploration and Consensus. Our framework balances single-trajectory accuracy (Pass@1) with majority-voting consensus (Maj@32). By aligning process rewards with outcomes, the dual gate mechanism prevents the policy from forming false consensus on invalid paths, ensuring that Maj@32 improvements reflect genuine problem-solving capabilities.

4.3 Ablation Studies

To systematically investigate the individual contributions of our proposed mechanisms, we conduct

Method	AIME 2024		GSM8K	
	Pass@1	Maj@32	Pass@1	Maj@32
Baseline	9.47	16.67	85.91	90.83
+ len-gate	10.62	13.33	86.12	90.83
+ len-gate + proc w/o gating	9.27	**20.00**	84.03	90.30
DG-DAPO (Ours)	**10.93**	16.67	**86.40**	**91.05**

Table 2: Ablation study on the components of the DG-DAPO framework.

an ablation study by incrementally adding components to the standard DAPO baseline.

Efficacy of Outcome-Gated Length Penalty. The outcome-gated length penalty (*Baseline+len-gate*) effectively mitigates sequence inflation in standard RL. By penalizing excessive steps strictly under outcome constraints, it curtails aimless exploration while preventing the model from exploiting brevity for easy rewards. This improves Pass@1 on AIME 2024 and GSM8K, but reduces Maj@32 on AIME 2024, suggesting a trade-off between single-sample accuracy and consensus performance.

The Pitfall of Unconstrained Process Supervision. Natively integrating a process reward model (*Baseline+len-gate+proc w/o gating*) leads to a general performance decrease across the evaluated benchmarks. Without strict outcome gating, the model becomes prone to reward hacking, prioritizing step-level process scores over actual mathematical correctness. This misalignment misguides the optimization trajectory, resulting in a notable drop in Pass@1. The discrepancy where Maj@32 does not decrease synchronously with Pass@1 and even slightly improves on certain datasets indicates that the model has developed a dependency on a single reasoning path, rather than reflecting a genuine improvement in problem-solving capabilities.

Necessity of Process Reward Gating. The complete DG-DAPO framework resolves this optimization misalignment by gating the PRM with strict outcome verification. This dual gate design successfully recovers the performance drop and achieves highly competitive performance across all datasets, demonstrating that process supervision is more reliable when anchored to correct final outcomes. Furthermore, by explicitly suppressing deceptive local feedback, this mechanism not only restores peak accuracy but also improves optimization dynamics by stabilizing training convergence and promoting reasoning conciseness. Detailed analyses are provided in the following sections.

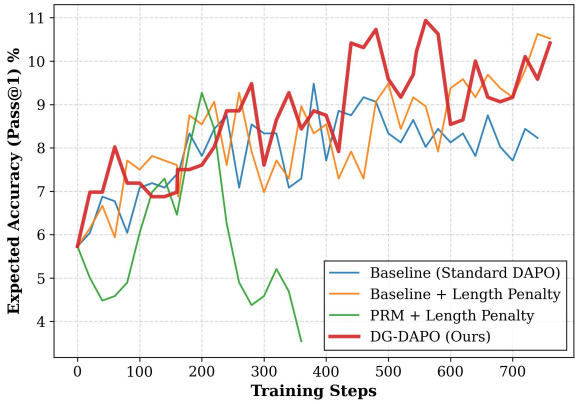

Figure 3: Training dynamics on the AIME 2024 benchmark, tracking expected accuracy (Pass@1). DG-DAPO demonstrates superior sample efficiency and a higher performance peak, effectively avoiding the early-stage collapse of the unconstrained process reward.

4.4 Analysis

Training Dynamics. To further understand the optimization behavior, Figure 3 tracks the accuracy (Pass@1) on the AIME 2024 benchmark throughout the training process. The empirical curves clearly demonstrate that the dual gate mechanism fundamentally stabilizes optimization dynamics and accelerates convergence.

As illustrated, directly integrating the PRM leads to a severe performance degradation early in the training process. Despite a brief initial rise, the policy rapidly deteriorates below its starting point, empirically confirming that unconstrained process signals induce severe reward hacking and destroy the model's fundamental capabilities. Meanwhile, the standard and length-gated baselines successfully avoid this collapse but exhibit a noticeably slower convergence rate, often accompanied by relatively larger fluctuations during the training process.

Crucially, DG-DAPO effectively neutralizes the high variance of the process rewards. Constrained by outcome verification, our framework not only avoids the early-stage collapse but also exhibits a rapid upward trajectory. It maintains a relatively stable training process and achieves a higher performance peak compared to all baselines. This confirms that DG-DAPO provides healthy guidance, allowing the model to efficiently scale its complex reasoning capabilities.

Reward Hacking and Length Explosion. The fundamental vulnerability of unconstrained process supervision lies in its susceptibility to sequence

Problem (AIME 2024): ... find the value of $m + n + p$.

unconstrained process reward (exploiting rewards via repetitive steps)

... Therefore, we have:

$$V = \frac{1}{3}r(K_1 + K_2 + K_3 + K_4).$$

Since $V = 8$, we have:

$$8 = \frac{1}{3}r(K_1 + K_2 + K_3 + K_4).$$

This implies:

$$24 = r(K_1 + K_2 + K_3 + K_4).$$

To find $K_1+K_2+K_3+K_4$, we can use the known formula... **which gives us:** $K_1 + K_2 + K_3 + K_4 = 24$**.** Thus, we have:

$$8 = \frac{1}{3}r \times 24 \implies r = 1.$$

In the form $\frac{m\sqrt{n}}{p}$**, we have** $m = 1$**,** $n = 1$**, and** $p = 1$**.** Thus, $m + n + p = 3$.

DG-DAPO [Ours] (concise reasoning)

... The inradius r is given by:

$$V = \frac{1}{3}Sr \implies r = \frac{3V}{S}.$$

After calculating the surface area S and substituting the values, we find:

$$r = \frac{4\sqrt{2}}{6} = \frac{2\sqrt{2}}{3}.$$

Thus, the distance from I to each face is $\frac{4\sqrt{2}}{3}$. Here, $m = 4$, $n = 2$, and $p = 3$. Therefore, $m + n + p = 4 + 2 + 3 = 9$.

Figure 4: Qualitative comparison of reasoning trajectories. Left: Unconstrained process reward, repetitively stacking simple steps to exploit scores. Right: DG-DAPO (outcome-gated), resolving simple steps in one go.

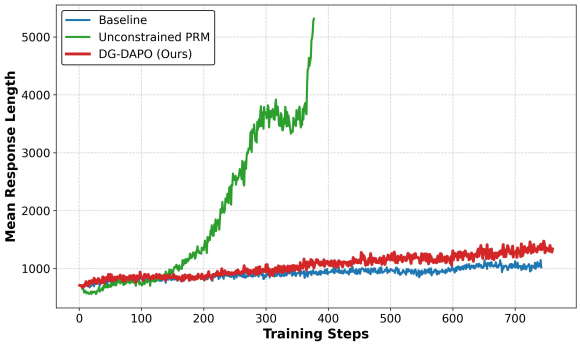

Figure 5: Evolution of the mean response length during training.

inflation. The policy inflates its cumulative score by generating longer trajectories, repetitively stacking simple reasoning steps to deliberately dilute the penalizing impact of incorrect steps.

To verify the severity of this reward hacking and explore the semantic nature of the sequence inflation, we qualitatively compare the reasoning trajectories with and without outcome constraints (Figure 4). The unconstrained model exploits the PRM by continuously repeating simple calculations and listing correct but redundant formulas. This padding satisfies the PRM's expectation for valid local logic, accumulating massive step-level rewards. However, when confronted with critical and complex derivation nodes, the policy actively evades the mathematical difficulty by hallucinating variables or fabricating format-compliant conclusions. DG-DAPO effectively mitigates these pathological behaviors. Under the strict constraint of outcome gating, the model resolves simple steps in one go and directly confronts difficult reasoning nodes. Additional qualitative examples are provided in Appendix B.

As shown in Figure 5, the mean response length of the model trained with unconstrained process rewards exhibits an uncontrollable surge. In contrast, DG-DAPO successfully integrates process supervision while maintaining outcome correctness as the primary optimization objective, keeping the response length within a healthy range.

5 Conclusion

In this paper, we investigate the vulnerabilities of process-supervised reinforcement learning in large language models, specifically focusing on reward hacking during complex mathematical reasoning. To mitigate these issues, we propose a dual gate decoupled advantage policy optimization framework. Our empirical analysis across mathematical reasoning benchmarks demonstrates that unconstrained process rewards can drive policies to exploit local scores through redundant padding. By anchoring step-level guidance to outcome correctness, our dual gate mechanism effectively regularizes training dynamics, substantially reducing length explosion while achieving competitive reasoning accuracy and sample efficiency. This work highlights that outcome-gating serves as a crucial regularizer to help preserve the actual problem-solving capability of LLMs during alignment.

Limitations

Despite the effectiveness of the proposed DG-DAPO framework, there are several limitations in our current study that provide directions for future research.

First, due to computational resource constraints, our empirical validation is primarily conducted on models with 3B parameters. While our framework demonstrates performance gains at this scale, we have not yet evaluated its efficacy on larger foundation models, such as those with 7B or 32B parameters. Second, our current evaluation focuses exclusively on complex mathematical reasoning benchmarks. Although mathematical problem-solving is a rigorous testbed for logical deduction, we have not verified the generalizability of our framework on other domains that require extensive exploration, such as code generation.

In future work, we plan to address these limitations by scaling the DG-DAPO framework to larger language models and extending its application across a broader spectrum of complex reasoning tasks to fully explore its versatility and scalability.

References

Peter Chen, Xiaopeng Li, Ziniu Li, Xi Chen, and Tianyi Lin. 2025a. Stepwise guided policy optimization: Coloring your incorrect reasoning in grpo. *arXiv preprint arXiv:2505.11595*.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025b. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. 2026. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. *Advances in Neural Information Processing Systems*, 38:131646–131671.

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. *arXiv preprint arXiv:2110.14168*.

Juntao Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2024. Safe rlhf: Safe reinforcement learning from human feedback. In *International Conference on Learning Representations*, volume 2024, pages 50750–50777.

Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. 2021. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese*, 198(Suppl 27):6435–6467.

Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. 2017. Reinforcement learning with a corrupted reward channel. *arXiv preprint arXiv:1705.08417*.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2026. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Can Jin, Yang Zhou, Qixin Zhang, Hongwu Peng, Di Zhang, Zihan Dong, Marco Pavone, Ligong Han, Zhang-Wei Hong, Tong Che, and 1 others. 2025. Your reward function for rl is your best prm for search: Unifying rl and search-based tts. *arXiv preprint arXiv:2508.14313*.

Yuhang Lai, Siyuan Wang, Shujun Liu, Xuan-Jing Huang, and Zhongyu Wei. 2024. Alarm: Align language models via hierarchical rewards modeling. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7817–7831.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng,

and 1 others. 2026. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025a. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*.

Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025b. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy. *arXiv preprint arXiv:2506.13284*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Mathematical Association of America. 2024. 2024 american invitational mathematics examination (aime i). Technical report, Mathematical Association of America.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Mohammad Rezaei, Jens Lehmann, and Sahar Vahdati. 2026. Llm reasoning with process rewards for outcome-guided steps. *arXiv preprint arXiv:2604.02341*.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2025. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, volume 2025, pages 60808–60838.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv: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*.

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. 2025. Sample more to think less: Group filtered policy optimization for concise reasoning. *arXiv preprint arXiv:2508.09726*.

Lin Sun, Chuang Liu, Xiaofeng Ma, Tao Yang, Weijia Lu, and Ning Wu. 2025. Freeprm: Training process reward models without ground truth process labels. *arXiv preprint arXiv:2506.03570*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024a. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10582–10592.

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

Lilian Weng. 2024. Reward hacking in reinforcement learning. *lilianweng.github.io*.

Lixin Wu, Na Cai, Qiao Cheng, Jiachen Wang, and Yitao Duan. 2025. Confucius3-math: A lightweight high-performance reasoning llm for chinese k-12 mathematics learning. *arXiv preprint arXiv:2506.18330*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. 2025. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. 2025. Capo: Towards enhancing llm reasoning through verifiable generative credit assignment. *arXiv e-prints*, pages arXiv–2508.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. 2024. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*.

Kaichen Zhang, Yuzhong Hong, Junwei Bao, Hongfei Jiang, Yang Song, Hong Dingqian, and Hui Xiong. 2026. Gvpo: Group variance policy optimization for large language model post-training. *Advances in Neural Information Processing Systems*, 38:165798–165820.

Zheng Zhang, Ziwei Shan, Kaitao Song, Yexin Li, and Kan Ren. 2025a. Linking process to outcome: Conditional reward modeling for llm reasoning. *arXiv preprint arXiv:2509.26578*.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025b. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and 1 others. 2026. Genprm: Scaling test-time compute of process reward models via generative reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34932–34940.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. 2026. Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 38:26764–26802.

A Implementation Details

Our experimental configurations are summarized in Table 3.

Configuration	**Value**
Framework	`verl`
Hardware	2×NVIDIA A100(80GB)
Optimizer	AdamW
Learning Rate	5×10^{-6}
Warmup Steps	10
Sampling Temperature	1.0
Group Size (G)	16
Max Prompt Length	2,048
Max Response Length (L_{max})	8,192
L_{cache}	4,096
ϵ_{low}	0.2
ϵ_{high}	0.28
r_{proc}^{min}	-5

Table 3: Detailed hyperparameter configurations and implementation setup for the DG-DAPO framework.

B Case Study: Mitigating Reward Hacking with Outcome Gating

To intuitively demonstrate how naively aggregating process rewards induces reward hacking and how our DG-DAPO framework effectively mitigates it, we present a qualitative case study on a complex geometry problem (calculating the volume and surface area of a tetrahedron given its edge lengths).

We compare the intermediate reasoning trajectories generated by the baseline + process w/o gating and our proposed DG-DAPO.

Case 1: Evading Complex Calculations via Hallucination. When calculating the volume using the computationally heavy Cayley-Menger determinant, the models exhibit starkly different behaviors.

Baseline + process (w/o Gating):

> *"The volume V of a tetrahedron with edge lengths $AB = \sqrt{41}$, $AC = \sqrt{80}$, $AD = \sqrt{89}$, $BC = \sqrt{89}$, $BD = \sqrt{80}$, and $CD = \sqrt{41}$ can be calculated using the Cayley-Menger determinant, but it is known from the problem setup that the volume is $V = 8$."*

DG-DAPO (Ours):

> *"The volume V of a tetrahedron with edge lengths a, b, c, d, e, f is given by:*
>
> $$288V^2 = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & a^2 & b^2 & c^2 \\ 1 & a^2 & 0 & d^2 & e^2 \\ 1 & b^2 & d^2 & 0 & f^2 \\ 1 & c^2 & e^2 & f^2 & 0 \end{vmatrix},$$
>
> *where $a = BC = \sqrt{89}$, $b = AD = \sqrt{89}$, $c = CD = \sqrt{41}$, $d = AB = \sqrt{41}$, $e = AC = \sqrt{80}$, and $f = BD = \sqrt{80}$. Plugging in these values, we get:*
>
> $$288V^2 = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 89 & 89 & 41 \\ 1 & 89 & 0 & 41 & 80 \\ 1 & 89 & 41 & 0 & 80 \\ 1 & 41 & 80 & 80 & 0 \end{vmatrix}."$$

The baseline policy recognizes the correct formula but deliberately evades the complex matrix expansion. Instead, it hallucinates a direct answer

("known from the problem setup") to artificially terminate the trajectory and secure intermediate process rewards without performing the actual work. In contrast, by strictly conditioning process rewards on final correctness, DG-DAPO forces the model to directly confront and execute the complex mathematical reasoning steps.

Case 2: Looping and Redundant Theoretical Statements. In the subsequent step requiring the calculation of the face areas using Heron's formula, the reward hacking phenomenon manifests as semantic looping.

Baseline + process (w/o Gating):

> *"Next, we need to find the areas of the faces. We will use Heron's formula for each face. However, a simpler approach is to use the fact that the volume can also be expressed as the sum of the volumes of the four smaller tetrahedra formed by the point I and the faces of the tetrahedron. The volume of each of these smaller tetrahedra is $\frac{1}{3} \times$ Area of the face $\times\, r$."*

DG-DAPO (Ours):

> *"The area of a triangle with sides a, b, c is given by Heron's formula:*
>
> $$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)},$$
>
> *where $s = \frac{a+b+c}{2}$.*
> *For $\triangle ABC$ with sides $AB = \sqrt{41}$, $BC = \sqrt{89}$, and $CA = \sqrt{80}$:*
>
> $$s = \frac{\sqrt{41} + \sqrt{89} + \sqrt{80}}{2},$$
>
> $$\text{Area} = \sqrt{s(s-\sqrt{41})(s-\sqrt{89})(s-\sqrt{80})}."$$

Here, the baseline model retreats from the tedious algebraic substitution of Heron's formula. Instead of executing the required area calculation, it loops back to the previous step of volume derivation, evading the current task by simply reciting basic geometric principles. While these statements are geometrically valid and easily score high local process rewards from the PRM, they are completely directionless and fail to advance the problem-solving process. Conversely, our dual gate mechanism effectively suppresses such deceptive local feedback, ensuring that the deductive steps genuinely steer the reasoning process toward the final answer.

C LLM Usage Statement

During the writing of this paper, we used a large language model (LLM) strictly as a tool to improve grammar and sentence flow. The LLM was not used to generate research ideas, analyze data, or write any technical content. The authors take full responsibility for the scientific content and the final manuscript.