Reformulate LLM Reinforcement Learning for Efficient Training under Black-box Discrepancy

Jiashun Liu[1*] **Runze Liu**[1*] **Xu Wan**[2*] **Jing Liang**[3*]
Hongyao Tang[3] **Ling Pan**[1†]
[1]Hong Kong University of Science and Technology
[2]Zhejiang University [3]Tianjin University

Abstract

Reinforcement Learning (RL) has emerged as a pivotal post-training paradigm, yet it frequently suffers from unpredictable suboptimal performance or even training collapses. Recent findings attribute these failures to a hidden train-inference discrepancy (or mismatch), stemming from the disparate underlying engines and architecture. We find that the training policy can actively self-correct such a discrepancy when provided with an appropriate learning signal. Then, we further empirically identify a *discrepancy tolerance region*: within this region, aggressively narrowing the discrepancy can suppress policy exploration and reduce learning efficiency, whereas outside this region, reducing excessive discrepancy improves optimization consistency and raises the achievable local performance ceiling. According to such findings, we formulate this problem as **Discrepancy-Constrained LLM RL** (`DC-RL`), where reward maximization is coupled with a constraint that aligns training-Inference behavior, achieving stable dual-objective optimization. To adaptively balance performance improvement and discrepancy control, we introduce a Lagrangian relaxation mechanism that dynamically adjusts the relative weight of the two objectives according to the current degree of discrepancy violation. This enables stable dual-objective optimization: the penalty strength is adjusted according to the batch-average discrepancy relative to the tolerance budget. Empirically, `DC-RL` significantly improves the performance of 8B dense model (`Qwen-3-8b`) and 30B Mixture-of-Expert model (`Qwen-3-30bA3b`), and enables a heterogeneous training paradigm, where LLMs can be optimized in a high-fidelity training setup while being explicitly aligned for low-cost, resource-constrained inference deployment.

1 Introduction

Recent advancements have established Reinforcement Learning (RL) as a pivotal post-training paradigm [Yu et al., 2025, Guo et al., 2025, Hu et al., 2026], significantly enhancing the deep reasoning capabilities of Large Language Models (LLMs) and continuously pushing the boundaries of LLM applications in complex real-world scenarios [Hui et al., 2024]. However, a persistent and costly challenge plagues the community: RL post-training often suffers abrupt, unpredictable training collapses [Qi et al., 2025, Zheng et al., 2025, Team et al., 2025], leading to substantial waste of computational resources and time. Recent investigations have identified the underlying cause of these failures: a pervasive train-inference mismatch arising from conflicting industrial demands [Li et al., 2026]. Specifically, to balance the extreme throughput necessary for generation with the stringent numerical precision required for parameter updates, practitioners are often forced to deploy disparate underlying engines and precision formats for the exact same model weights [Zhang et al.,

*Equal contribution
†Corresponding to: Ling Pan <lingpan@ust.hk>.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

2026, Dong et al., 2026]. This discrepancy, residing entirely beneath the algorithmic layer, induces a deviation between the action probability distributions of the inference policy and the training policy.

A straightforward approach to mitigate this persistent issue is that merely aligning the precision of training and inference, by uniformly downgrading to FP16, can alleviate training collapses [Qi et al., 2025]. However, this strategy drastically shrinks both the maximum and minimum representable values and exacerbates the risk of gradient explosion due to numerical overflow [Zheng et al., 2025]. Furthermore, precision is merely one facet of the infrastructure puzzle; discrepancies in low-level operator implementations (kernels) and parallelization strategies contribute equally to the mismatch. Attempting to comprehensively align these mechanisms across hardware and software stacks requires exhaustive, bespoke optimizations to handle myriad corner cases, which are notoriously labor-intensive. An alternative research line has emerged at the algorithmic level through heuristic interventions. Li et al. [2025] found that masking the gradients generated by tokens or sentences with high discrepancy can bolster stability. Nevertheless, these methods harbor a subtle yet fundamental flaw: their primary focus is merely to artificially suppress the impact of noisy data on the training policy. They overlook the ultimate objective of LLM fine-tuning, which is to deliver superior performance on the inference policy deployed in production, not the training policy itself. Our motivating experiment suggests that generation patterns with both high rewards and low train-inference discrepancy exist in the solution space, motivating an algorithmic approach to mitigating backend-induced mismatch. Motivated by these profound limitations, we introduce a perspective: **What if we endow the training policy with the inherent capacity to dynamically perceive and reduce its discrepancy with the inference policy?**

Intuitively, this can be formulated as a dual-objective optimization problem by translating train-inference discrepancy into a step-wise penalty, jointly maximizing reward and minimizing discrepancy. The core challenge is formulating a penalty that harmonizes with the primary correctness reward while remaining robust across long-horizon generation. Through empirical and theoretical analysis, we identify the absolute token-level probability difference as an efficient penalty metric. Crucially, it remains sensitive to individual tokens with extreme discrepancy, enabling fine-grained guidance. The metric captures localized outliers rather than relying solely on a batch-level aggregate, providing timely feedback during generation. As an endogenous signal, this penalty enables the training policy to perceive mismatch and initiate self-correction, effectively preventing training collapse. Digging deeper, we empirically observe a counter-intuitive phenomenon: policies naturally tolerate minor discrepancy. Aggressive penalties within this *discrepancy tolerance region* over-optimize for a negligible gap and stifle performance gains. To exploit this region, we reformulate standard Markov Decision Process (MDP)-based LLM RL as *discrepancy-constrained LLM RL* (`DC-RL`). Specifically, penalty strength increases when batch-average discrepancy exceeds the tolerance boundary, balancing reward and discrepancy objectives, and decreases otherwise. Furthermore, we introduce an ultra-lightweight operator to adaptively balance these objectives.

Our method frames discrepancy between training and deployment (inference) policies as a feedback-driven black-box optimization problem, handling deviations from low-level infrastructure or high-level model architectures. This capability also enables heterogeneous training: policies can be optimized under a high-fidelity training backend for deployment under a lower-cost inference backend. Thus, we leverage the advantages of high-fidelity precision and lossless weight loading during training while adapting to resource-constrained, distributed inference deployments. This allows computationally powerful training configurations to discover solutions suited to the cost and hardware limits of real-world inference.

To summarize, our primary contributions are threefold:

1. We uncovered that incorporating a robust black-box penalty endows the training policy with the capacity to eradicate the pervasive risk of discrepancy-induced collapse autonomously.
2. We pioneer a paradigm shift by reformulating the LLM RL as `DC-RL` to optimally balance the dual objectives of maximizing reward and minimizing train-inference discrepancy.
3. We enable a heterogeneous training paradigm, paving the way for highly promising, resource-efficient model deployments in the real world.

2 Preliminaries

Reinforcement Learning for LLMs. The alignment of Large Language Models (LLMs) via RL is fundamentally grounded in the policy gradient theorem. Group Relative Policy Optimization (GRPO) [Shao et al., 2024] as a widely used algorithm, it retains the core clipping mechanism of PPO [Schulman et al., 2017] but entirely eliminates the Critic network. Instead of relying on a learned value baseline, GRPO samples a group of G responses $\{o_i\}_{i=1}^{G}$ for the same prompt q, and estimates the advantage A_t by normalizing their respective rewards $\{r_i\}_{i=1}^{G} : \hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_i\}_{i=1}^{G})}{\text{std}(\{r_i\}_{i=1}^{G})}$. This group-level normalization serves as a robust reward-shaping technique, effectively preserving gradient reliability even in sparse reward settings without the need for an external value estimator. The overall objective thus becomes:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\left[q\sim P(Q),\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)\right]} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left(\gamma_{i,t}(\theta)\,\hat{A}_{i,t},\ \text{clip}\left(\gamma_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\right)\,\hat{A}_{i,t}\right)\right\}. \tag{1}$$

where $\gamma_t(\theta) = \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}$ is the probability ratio. It is used by clipping to restrict the divergence between the current policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$. ϵ is the clipping range of trust region.

3 Untapped Potential: Low-Discrepancy Solutions exist in Decision Space

To algorithmically reconstruct the learning objective such that the training policy converges to an ideal parameter landscape that exhibits both high performance and low train-inference discrepancy (a region favored by the deployment policy), a fundamental prerequisite must be met: the decision space of the training policy must inherently contain latent solutions that simultaneously satisfy these dual criteria. To this end, we conduct an empirical case study to validate this underlying hypothesis.

Specifically, we employ GRPO [Shao et al., 2024], a widely adopted baseline in LLM RL, to optimize the `Deepseek-Distilled-Qwen-2.5-1.5b` model. We deliberately select mathematical reasoning tasks for evaluation that naturally possess a vast solution space composed of diverse reasoning paths. Furthermore, these tasks pose significant challenges for a 1.5B-parameter model [Liu et al., 2025b], leaving substantial headroom for improvement and thus serving as a sensitive, precise barometer for RL efficacy. During the experiments, we explicitly control the policy's exploration propensity within the decision space by sweeping the sampling temperature $t = \{0.3, 0.7, 1.0\}$. As shown in Figure 1 (Left, Middle), both the performance gains and the train-inference consistency during early-stage RL optimization exhibit a pronounced positive correlation with the degree of exploration. This provides the hypothesis: *there indeed exist valid trajectories characterized by both low inter-policy mismatch and high reward.*

However, merely relying on elevated temperatures to inject sampling diversity is not a fundamental cure, and the training will eventually collapse. To efficiently and stably guide the agent toward these optimal landscapes amidst an astronomically large decision space, an accurate and dense feedback signal is imperative. This necessitates the formulation of an idealized penalty metric.

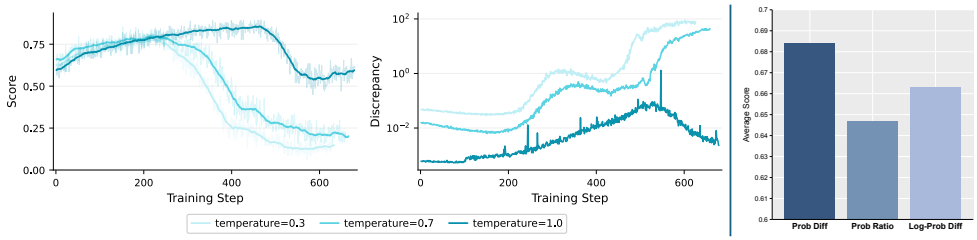

Figure 1: (Left):**Training curve**. Performance of GRPO-driven policies with different temperatures on six mathematical scenarios. (Middle): **Discrepancy ratio**. The corresponding training-inference difference ratio during training. (Right): **Probability Difference penalty achieves the best performance**. average@32 scores on six benchmarks obtained using different penalties

4 Enabling Discrepancy Compensation via a Discrepancy-Aware Penalty

Building upon the empirical verification in section 3, which confirms the latent existence of optimization landscapes satisfying both high task performance and low train-inference discrepancy, this section aims to formulate a reliable mismatch-aware penalty for exploration guidance. Given the same model weights θ_{old} and the same generated trajectory $\tau = \{o_1, \ldots, o_T\}$, the training and inference backends may instantiate two slightly different effective policy distributions, denoted as $\pi^{\text{Train}}_{\theta_{\text{old}}}(\cdot|q, o_{<t})$ and $\pi^{\text{Inf}}_{\theta_{\text{old}}}(\cdot|q, o_{<t})$, respectively. The central question is not merely whether such a discrepancy exists, but how it should be measured and penalized such that the resulting signal reflects deployment-relevant mismatch while remaining compatible with stable reinforcement learning.

We begin with three practical token-level discrepancy signals, where $p_t^{\text{Train}} = \pi^{\text{Train}}_{\theta_{\text{old}}}(o_t|q, o_{<t})$ and $p_t^{\text{Inf}} = \pi^{\text{Inf}}_{\theta_{\text{old}}}(o_t|q, o_{<t})$:

$$\begin{aligned} \texttt{Probability Difference}: &\quad \delta_t^{\text{diff}} = p_t^{\text{Train}} - p_t^{\text{Inf}} \\ \texttt{Probability Ratio}: &\quad \delta_t^{\text{ratio}} = \frac{p_t^{\text{Train}}}{p_t^{\text{Inf}}} \\ \texttt{Log-Probability Difference}: &\quad \delta_t^{\log} = \log p_t^{\text{Train}} - \log p_t^{\text{Inf}} \end{aligned} \tag{2}$$

Although all three quantities vanish when the two backends agree, they induce substantially different regularization geometries. The probability difference δ_t^{diff} constrains absolute drift in token probability. As a result, increasing the probability of a low-probability token may incur only a small penalty as long as the absolute change remains limited, even if the relative amplification is large. This makes the induced regularization highly plastic, but also potentially tail-seeking: when a low-probability token receives a large advantage, the optimizer can aggressively shift probability mass toward it. In contrast, the probability ratio δ_t^{ratio} constrains multiplicative drift. For tokens with small p_t^{Inf}, even a modest absolute increase can correspond to a large ratio change, leading to a much stronger penalty. Ratio-based regularization therefore more strictly preserves the support of the deployed inference policy and discourages abrupt amplification of tail tokens. However, this conservatism can also become restrictive: if genuinely beneficial tokens initially reside in the low-probability region of the inference policy, the ratio penalty may suppress the very adaptations needed to improve performance. $\delta_t^{\log}$ provides an intermediate geometry. In the small-drift regime, when p_t^{Train} remains close to p_t^{Inf}, we have $\log \frac{p_t^{\text{Train}}}{p_t^{\text{Inf}}} = \log\left(1 + \frac{p_t^{\text{Train}} - p_t^{\text{Inf}}}{p_t^{\text{Inf}}}\right) \approx \frac{p_t^{\text{Train}} - p_t^{\text{Inf}}}{p_t^{\text{Inf}}}$, which shows that $\delta_t^{\log}$ locally behaves like a ratio-style penalty and therefore inherits its stabilizing effect against relative mismatch. Outside this local regime, however, the logarithmic transformation grows more slowly than the raw ratio, avoiding excessive punishment of all non-negligible deviations. Theoretically, we further analyze how these three discrepancy parameterizations induce different local optimal policies, and use a one-state toy experiment to illustrate how advantage-weighted updates reshape the resulting policy distributions; details are deferred to Appendix B.

We empirically compare these three shaped discrepancy penalties under identical post-training configurations. As illustrated in Figure 1 (Right), δ_t^{dif} achieves the strongest overall performance. It consistently improves the primary verifiable reward while suppressing train-inference discrepancy. This empirical ordering indicates that effective discrepancy-aware regularization should correct deployment-relevant drift without overly constraining reward-driven exploration. δ_t^{diff} offers a more plastic penalty: it narrows train-inference discrepancy while preserving the policy's ability to shift probability mass toward useful reasoning trajectories.

The Choice of Quantization Granularity A natural alternative is to first aggregate the discrepancy over the entire generated response and then apply the penalty at the sequence level. Although such sequence-level calculations are structurally aligned with response-level rewards, they are intrinsically coarse. In long-horizon reasoning tasks, train-inference discrepancy is often highly sparse and token-localized: a small number of tokens may exhibit substantial backend disagreement and exert a disproportionate influence on subsequent generation quality. Once averaged over the whole sequence, however, these critical tokens can be diluted by a large number of benign tokens with negligible discrepancy [Liu et al., 2025a]. Therefore, we instead compute the discrepancy penalty at the token level. This design preserves the fine-grained mismatch structure along the trajectory. Consequently, tokens with large train-inference mismatch receive direct corrective feedback.

Dual-objective RL We finally incorporate the token-level probability discrepancy into the reward-maximized RL as an additional optimization objective. In practice, we use GRPO as an example. Given a prompt q and a group of responses $\{o_i\}_{i=1}^{G}$ sampled from the old policy, the standard GRPO objective encourages reward-driven policy improvement through the clipped importance ratio. In contrast, our formulation additionally penalizes the discrepancy measured on each generated token, thereby guiding the learned policy toward solutions that are not only reward-favorable but also faithfully supported by the inference policy. Formally, the per-token penalty is

$$\delta_{i,t} = \left| \pi^{\text{Train}}_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t}) - \pi^{\text{Inf}}_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t}) \right|. \tag{3}$$

Since the group-relative advantage $\hat{A}_{i,t}$ of GRPO is broadcast to every token in the response, we directly inject the discrepancy penalty at the advantage level, yielding a dual-objective advantage $\hat{A}^{\text{Dual}}_{i,t} = \hat{A}_{i,t} - \delta_{i,t}$, and the Dual-GRPO objective becomes:

$$\begin{aligned} \mathcal{J}_{\text{Dual-GRPO}}(\theta) \; = \; & \mathbb{E}_{q \sim P(Q),\, \{o_i\}_{i=1}^{G} \sim \pi^{\text{Inf}}_{\theta_{\text{old}}}(\cdot|q)} \\ & \frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left(\gamma_{i,t}(\theta)\, \hat{A}^{\text{Dual}}_{i,t},\; \text{clip}\left(\gamma_{i,t}(\theta),\, 1-\varepsilon,\, 1+\varepsilon\right)\, \hat{A}^{\text{Dual}}_{i,t} \right). \end{aligned} \tag{4}$$

5 Transitioning to discrepancy-constrained LLM RL for Stable Post-training

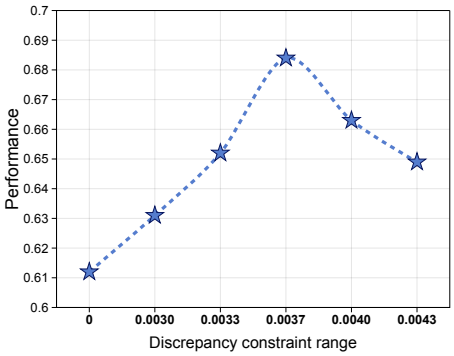

Figure 2: **Performance under various tolerance ranges.** As the region expands, the performance gradually increases, but a loose range leads to a degradation. Use the setup in Figure 1.

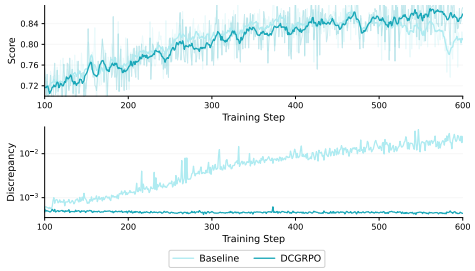

Figure 3: Compared to the naive GRPO, `DC-GRPO` controls the train-inference discrepancy at a small value, achieves more stable training and overcomes collapse, ultimately achieving better progressive gains. The experiments use the same setting in Figure 1.

While the reliable performance-discrepancy dual-objective formulation (Dual-GRPO) eradicates discrepancy-induced training collapses, we find a significant limitation of such a learning objective from a pratical perspective. Compared to the naive single-objective GRPO, Dual-GRPO suffers from severely degraded learning efficiency during the early stages of training and ultimately converges to a sub-optimal performance ceiling. We hypothesize that this limitation stems from over-regularization: when the inter-policy discrepancy is initially minute, an indiscriminate and strict penalty forces the agent to prematurely prioritize alignment with the inference policy at the expense of its primary objective, i.e., maximizing the verifiable task reward, thereby stifling the overall learning momentum. This observation inevitably begs the question: Does the training policy possess an inherent resistance, or a discrepancy tolerance region, against minor training-inference deviations?

To validate this conjecture, we introduce a relaxed, threshold-based truncation to the penalty, activating the discrepancy feedback only when the deviation surpasses a predefined margin c. As depicted in Figure 2, implementing this leniency improves the policy's learning efficiency and performance. In contrast, punishments that are too lenient can have a negative effect. This evidence validates the existence of the *discrepancy tolerance region*.

Driven by these empirical observations, it becomes evident that a naive, unconstrained scalarization of the two objectives is highly inefficient. Instead, a more structurally sound and mathematically

elegant modeling paradigm for LLM RL emerges: the *discrepancy-constrained LLM RL* (DC-RL). The core innovation of the DC-RL lies in the introduction of an explicit discrepancy constraint in addition to the reward-maximization objective. Specifically, it constrains the black-box train-inference discrepancy within the empirically identified *discrepancy tolerance region*, allowing harmless microscopic deviations while preventing large backend-induced distributional shifts in either direction. In this way, the DC-RL preserves the exploration flexibility of the training policy inside the tolerated region. Formally, building upon the token-level absolute probability discrepancy, the DC-RL objective can be formulated as:

$$\max_{\theta} \quad \mathcal{J}(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}^{\text{Train}}} \left[\sum_{t=1}^{|\tau|} r(o_t \mid q, o_{<t}) \right] \tag{5}$$

$$\text{s.t.} \quad \frac{1}{|\tau|} \sum_{t=1}^{|\tau|} \left| \pi_{\theta_{\text{old}}}^{\text{Train}}(o_t \mid q, o_{<t}) - \pi_{\theta_{\text{old}}}^{\text{Inf}}(o_t \mid q, o_{<t}) \right| \leq c, \quad \forall \tau \sim \pi_{\theta}^{\text{Train}}. \tag{6}$$

$r(\cdot)$ denotes the verifiable reward, and $c \geq 0$ specifies the radius of the symmetric two-bounded discrepancy tolerance region. The absolute-value operator measures deviations in both directions, i.e., both $\pi^{\text{Train}} > \pi^{\text{Inf}}$ and $\pi^{\text{Train}} < \pi^{\text{Inf}}$. To avoid maintaining a separate multiplier for each token, we use the batch-level mean discrepancy as an approximate proxy for the token-level tolerance region. This mean controls a shared multiplier λ, while the penalty is applied to all valid tokens.

Solving DC-RL via Lagrangian Relaxation. We incorporate the Lagrangian Relaxation method to achieve an adaptive and smooth equilibration of the dual objectives [Achiam et al., 2017, Ji et al., 2024]. Specifically, during RL training, a Lagrangian multiplier λ acts as a dynamic penalty weight, updated iteratively based on the average discrepancy measured across the sampled batch at each time step. When the inter-policy divergence spikes and violates the tolerance boundary, λ progressively scales up the penalty strength to forcefully pull the policy back into the safe zone. Conversely, when the discrepancy drops below the threshold, λ gently anneals toward zero. This responsive mechanism is associated with a dynamic buffer for calculation. Ultimately, using GRPO as the algorithmic backbone, our framework for solving the DC-RL can be cast as a Min-Max optimization problem:

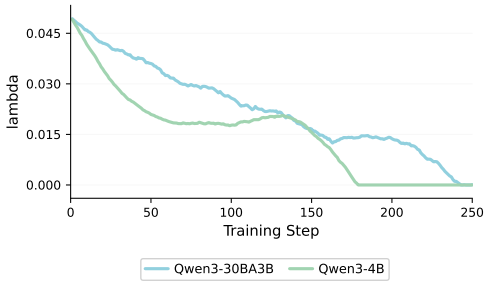

Figure 4: Depending on the adaptive optimization ability of the Lagrangian operator, we find that its behavior for different models is obviously different, which is hard to achieve by heuristics.

$$\max_{\theta} \min_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} \mathcal{L}(\theta, \lambda) = \mathcal{J}_{\text{GRPO}}(\theta) - \lambda \left(\mathbb{E}_{q, \{o_i\}_{i=1}^{G}, t} \left[\delta_{i,t} \right] - c \right), \tag{7}$$

where $\delta_{i,t} = \left| \pi_{\theta_{\text{old}}}^{\text{Train}}(o_{i,t} \mid q, o_{i,<t}) - \pi_{\theta_{\text{old}}}^{\text{Inf}}(o_{i,t} \mid q, o_{i,<t}) \right|$ is the token-level absolute probability discrepancy, λ is a learnable Lagrangian multiplier bounded within $[\lambda_{\min}, \lambda_{\max}]$ that adaptively controls the penalty strength, and c specifies the discrepancy tolerance boundary. By the linearity of expectation and the policy gradient theorem, optimizing this Lagrangian objective with respect to the policy parameters θ is mathematically equivalent to dynamically shaping the token-level advantage. In our practical implementation, at training iteration k, the penalty is injected directly into the group-normalized advantage:

$$\hat{A}_{i,t}^{\text{DC-RL}} = \hat{A}_{i,t} - \underbrace{\lambda_k \cdot \delta_{i,t}}_{\text{Adaptive Penalty}}, \quad \text{where} \quad \hat{A}_{i,t} = \frac{r_i^{\text{task}} - \text{mean}(\{r_i^{\text{task}}\}_{i=1}^{G})}{\text{std}(\{r_i^{\text{task}}\}_{i=1}^{G})}, \tag{8}$$

$$\mathcal{J}_{\text{DC-GRPO}}(\theta) = \mathbb{E}_{q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}^{\text{Inf}}(\cdot|q)} \tag{9}$$

$$\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left(\gamma_{i,t}(\theta) \, \hat{A}_{i,t}^{\text{DC-RL}}, \; \text{clip} \left(\gamma_{i,t}(\theta), \, 1-\epsilon_{\text{low}}, \, 1+\epsilon_{\text{high}} \right) \hat{A}_{i,t}^{\text{DC-RL}} \right). \tag{10}$$

The discrepancy signal is evaluated at the old policy and detached from the current parameter update. We compare this implementation with a differentiable penalty variant in Appendix D.2. Since the additive penalty can reverse the sign of the original advantage, we further examine advantage-sign reversals and gradient-direction changes during training in Appendix D.1.

The policy π_θ is then updated via the standard GRPO clipped surrogate using the penalized advantage $\hat{A}_{i,t}^{\texttt{DC-RL}}$. Simultaneously, to solve the minimization problem with respect to the dual variable, the multiplier λ is updated via dual gradient ascent at the end of each iteration based on the batch-average token-level discrepancy:

$$\lambda_{k+1} = \Pi_{[\lambda_{\min},\lambda_{\max}]}\left(\lambda_k + \eta_\lambda \left(\frac{1}{\sum_{i=1}^{B}|o_i|}\sum_{i=1}^{B}\sum_{t=1}^{|o_i|}\delta_{i,t} - c\right)\right), \tag{11}$$

where η_λ is the dual learning rate, B is the number of trajectories in the current batch, and $\Pi_{[\lambda_{\min},\lambda_{\max}]}(\cdot)$ is the projection onto the admissible range of the multiplier.

6 Experiments

6.1 Setup

Models and Baselines. We conduct experiments on two open-source LLMs spanning both dense and Mixture-of-Experts (MoE) architectures: Qwen3-8B [Yang et al., 2025] and Qwen3-30B-A3B [Yang et al., 2025] (30B total parameters with 3B activated per token). These two models cover substantially different training regimes, e.g., FSDP2 for the dense model and Megatron-LM [Shoeybi et al., 2019] with tensor/expert parallelism for the MoE model, allowing us to comprehensively evaluate the robustness of `DC-GRPO` across infrastructures that are prone to different train-inference discrepancy patterns. We compare `DC-GRPO` against the vanilla GRPO [Shao et al., 2024] baseline, which shares the identical training pipeline, data, and hyperparameters except for the discrepancy-constrained penalty.

Evaluation. We evaluate all methods on six widely-used mathematical reasoning benchmarks: AIME24 [MAA, 2024], AIME25 [MAA, 2025], AMC23 [MAA, 2023], MATH-500 [Lightman et al., 2024], Minerva Math [Lewkowycz et al., 2022], and OlympiadBench [He et al., 2024]. Evaluation uses vLLM 0.11.0 [Kwon et al., 2023] with a maximum generation length of 8,192 tokens, temperature 0.7, and top-p of 1.0. We report avg@K with $K = 32$ for AIME24/25 and AMC23, and $K = 4$ for MATH-500, Minerva, and OlympiadBench.

Implementation Details. We train all methods on the DAPO-Math-17K [Yu et al., 2025] dataset using verifiable mathematical rewards computed by `math-verify`. The maximum response length is set to 8,192 tokens. Our training pipeline guarantees tokens-in-tokens-out (TITO) rollouts. The train batch size is 64 prompts, with a PPO mini-batch size of 16. We use AdamW with learning rate 1×10^{-6}, betas $(0.9, 0.95)$, weight decay 0, and optimizer epsilon 1×10^{-15} following [Qi et al., 2025]. We use $\varepsilon_{\text{high}}$=0.24 and disable the KL loss. The dense Qwen3-8B is trained on 1×8 NVIDIA GPUs under the FSDP2 backend, while the MoE Qwen3-30B-A3B is trained on 4×8 GPUs under the Megatron backend (TP=2, EP=8, rollout TP=4) with BF16 weights.

The discrepancy penalty is instantiated as the token-level absolute probability difference $\delta_{i,t} = |\pi_{\theta_{\text{old}}}^{\text{Train}} - \pi_{\theta_{\text{old}}}^{\text{Inf}}|$ computed from the rollout (vLLM, inference backend) and recomputation (training backend) log-probabilities on the exact same token sequences. The Lagrangian multiplier λ is initialized at $\lambda_0 = 0.1$ for Qwen3-8B and $\lambda_0 = 0.05$ for Qwen3-30B-A3B, with dual learning rate $\eta_\lambda = 1.0$. The discrepancy tolerance budget c is set to 0.0037 for Qwen3-8B and 0.0050 for Qwen3-30B-A3B, reflecting the empirically higher baseline mismatch of MoE rollouts due to expert routing non-determinism. The penalty is injected on the token-level advantage as in Eq. (8), and λ is updated once per training step.

6.2 Main Results

BF16 main results. Table 1 reports mathematical reasoning results for Qwen3-8B and Qwen3-30B-A3B under the setting of BF16. Across both backbones, `DC-GRPO` consistently improves the overall average performance for each backbone, demonstrating that enforcing discrepancy-tolerance constraints not only preserves, but can even enhance the primary reward-driven optimization signal.

These results suggest that DC-RL provides a unified framework for mitigating both architecture-induced and infrastructure-induced instability. This highlights an algorithmic advantage of our approach: instead of explicitly tracing and modeling the underlying source of train-inference mismatch, DC-RL treats the mismatch as a black-box optimization.

Table 1: Evaluation results on mathematical benchmarks under the BF16 training/inference regime. The results of DC-GRPO are shaded and the highest values are **bolded**.

Method	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	Avg.
Qwen3-8B							
↳ GRPO	58.8	40.0	90.4	93.7	**51.0**	70.1	67.3
↳ DC-GRPO	**63.2**	**52.5**	**91.8**	**94.2**	50.1	**74.4**	**71.0**
Qwen3-30B-A3B							
↳ GRPO	60.2	43.6	91.4	95.5	52.0	71.3	69.0
↳ DC-GRPO	**66.8**	**49.1**	**91.9**	**95.8**	**52.9**	**74.5**	**71.8**

Heterogeneous FP8 deployment. A central motivation of DC-GRPO is to enable *heterogeneous* training: the policy is updated under a high-fidelity training backend (BF16) while being optimized for a low-cost deployment backend (FP8 vLLM). Table 2 reports the results in this heterogeneous setting, where GRPO suffers noticeable degradation or outright collapse on several benchmarks due to the amplified train-inference gap, whereas DC-GRPO preserves (and often exceeds) the BF16-only performance by natively closing the backend gap. This validates the core claim of our method: by framing the train-inference gap as a black-box feedback signal, DC-GRPO turns infrastructure-level heterogeneity from a liability into a controllable optimization knob.

Table 2: Evaluation results on mathematical benchmarks under the heterogeneous training (BF16) + FP8 deployment regime. The results of DC-GRPO are shaded and the highest values are **bolded**.

Method	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	Avg.
Qwen3-8B							
↳ GRPO	53.2	35.2	89.5	92.9	50.0	67.4	64.7
↳ DC-GRPO	**55.0**	**44.1**	**90.2**	**94.4**	**54.0**	**70.9**	**68.1**
Qwen3-30B-A3B							
↳ GRPO	52.4	37.5	87.3	93.8	49.6	70.5	65.2
↳ DC-GRPO	**56.6**	**41.2**	**91.8**	**94.0**	**50.5**	**72.0**	**67.7**

Additional baseline comparisons. We further evaluate DC-GRPO against multiple mismatch-mitigation baselines on Qwen3-8B under both the BF16-training/BF16-inference and BF16-training/FP8-inference settings, with five-seed results reported in Appendix C.1. For Qwen3-30B-A3B, we additionally compare DC-GRPO with Rollout Routing Replay and evaluate their combination to examine compatibility with MoE routing alignment (Appendix C.2).

6.3 Ablation Study

To disentangle the contribution of each design choice of DC-GRPO, we conduct ablations on Qwen3-8B by varying the three core hyperparameters of the Lagrangian relaxation. All other training settings are kept identical to the main experiments.

Initial value of λ. We sweep $\lambda_0 \in \{0.05, 0.1, 0.2\}$. Overall, DC-GRPO is not very sensitive to λ_0: all three settings converge to competitive and stable performance, because the dual variable λ is updated online and quickly adjusts to the prevailing discrepancy level. The remaining differences are confined to early training: an overly small λ_0 (e.g., 0.05) provides only a weak initial penalty and therefore offers limited early containment of backend-induced spikes, whereas an overly large λ_0 (e.g., 0.2) over-regularizes the policy at the very beginning and slows down early-stage reward maximization. λ_0=0.1 achieves a good middle-ground and is used as our default. As shown in Table 3, the final-performance gap between $\lambda_0 = 0.1$ and $\lambda_0 = 0.2$ is small.

Dual learning rate of λ. We compare $\eta_\lambda \in \{0.5, 1.0, 1.5\}$. Similar to λ_0, DC-GRPO remains robust to a broad range of dual learning rates. Only when η_λ is very small (e.g., well below 0.5) does the multiplier react too slowly to discrepancy spikes, allowing constraint violations to accumulate and noticeably hurting final performance. Moderately larger values such as η_λ=1.5 still work well:

Table 3: Ablation on the initial value of λ for Qwen3-8B. All other settings follow the main experiments (c=0.0037, η_λ=1.0). Results are shaded for the default.

λ_0	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	Avg.
0.05	59.8	43.8	91.5	94.7	52.0	71.8	68.9
0.10 (default)	63.2	52.5	91.8	94.2	50.1	74.4	71.0
0.20	64.9	45.3	92.8	95.4	52.0	72.0	70.4

although larger η_λ in principle induces faster oscillations of λ around the tolerance boundary, the projection onto $[\lambda_{\min}, \lambda_{\max}]$ and the smoothing provided by the batch-averaged discrepancy keep the training stable in practice. η_λ=1.0 is chosen as our default for its clean convergence behavior. Detailed numbers are shown in Table 4.

Table 4: Ablation on the dual learning rate η_λ for Qwen3-8B. All other settings follow the main experiments (c=0.0037, λ_0=0.1). Results are shaded for the default.

η_λ	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	Avg.
0.5	61.5	43.6	90.5	94.4	50.0	71.5	68.6
1.0 (default)	63.2	52.5	91.8	94.2	50.1	74.4	71.0
1.5	63.1	46.0	94.2	95.7	52.5	72.9	70.7

Tolerance budget c. When c is set too small, the batch-average discrepancy persistently exceeds c throughout training, causing λ to monotonically increase and saturate at its upper bound; the resulting overly strong penalty dominates the advantage and impedes the primary reward-maximization objective of RL. Conversely, when c is too large, the discrepancy rarely crosses the boundary and λ keeps decreasing toward $\lambda_{\min}$, so the penalty effectively vanishes and `DC-GRPO` degenerates to vanilla GRPO, losing its constraint-control capability. The middle setting c=0.0037 yields the best downstream performance (Table 5).

Table 5: Ablation on the tolerance budget c for Qwen3-8B. All other settings follow the main experiments (λ_0=0.1, η_λ=1.0). Results are shaded for the default.

c	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	Avg.
0.0030	61.3	49.5	92.3	95.2	52.3	72.9	70.6
0.0035	63.0	48.8	92.8	94.9	52.1	72.7	70.7
0.0037 (default)	63.2	52.5	91.8	94.2	50.1	74.4	71.0
0.0040	62.9	44.8	92.3	95.2	51.0	74.1	70.1

We further examine the sensitivity to c across model sizes under BF16 training and FP8 inference. The results suggest $c \in [0.0033, 0.0037]$ as a practical starting range for the tested configurations; detailed results and guidance for selecting c are provided in Appendix C.3.

7 Conclusion

In this paper, we reformulate LLM RL as *discrepancy-constrained LLM RL* (`DC-RL`) to tackle the long-standing train-inference discrepancy that underlies unpredictable collapses: using the token-level absolute probability difference as a black-box discrepancy signal, we inject an adaptive Lagrangian penalty into the advantage, with its strength adjusted according to the batch-average discrepancy relative to the empirically-identified tolerance budget, thereby balancing reward-driven exploration with discrepancy control; experiments on Qwen3-8B and Qwen3-30B-A3B across various benchmarks show that our method outperforms GRPO under the widely used BF16 training setup, and the heterogeneous BF16-training + FP8-deployment regime, validating that train-inference gap can be mitigated at algorithmic level.

Limitations. Our empirical scope is limited to BF16 and FP8 precisions and extending `DC-RL` to other backend combinations is left for future work. In addition, although we verify the effectiveness of our method on large-sized models as well as advanced architecture models, e.g., MoE, there is a lack of evaluation on more open source model families and analysis on different model sizes. We will further improve the related research in the future.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. Pmlr, 2017.

Yiming Dong, Kun Fu, Haoyu Li, Xinyuan Zhu, Yurou Liu, Lijing Shao, Jieping Ye, and Zheng Wang. Probing rlvr training instability through the lens of objective-level hacking. *arXiv preprint arXiv:2602.01103*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=NFM8F5cV0V.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. Omnisafe: An infrastructure for accelerating safe reinforcement learning research. *Journal of Machine Learning Research*, 25(285):1–6, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/18abbeef8cfe9203fdf9053c9c4fe191-Paper-Conference.pdf.

Yingru Li, Jiacai Liu, Jiawei Xu, Yuxuan Tong, Ziniu Li, Qian Liu, and Baoxiang Wang. Trust region masking for long-horizon llm reinforcement learning. *arXiv preprint arXiv:2512.23075*, 2025.

Yuhang Li, Reena Elangovan, Xin Dong, Priyadarshini Panda, and Brucek Khailany. Qurl: Efficient reinforcement learning with quantized rollout. *arXiv preprint arXiv:2602.13953*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Hanbing Liu, Lang Cao, Yuanyi Ren, Mengyu Zhou, Haoyu Dong, Xiaojun Ma, Shi Han, and Dongmei Zhang. Bingo: Boosting efficient reasoning of llms via dynamic and significance-based reinforcement learning. *arXiv preprint arXiv:2506.08125*, 2025a.

Jiashun Liu, Johan S. Obando-Ceron, Han Lu, Yancheng He, Weixun Wang, Wenbo Su, Bo Zheng, Pablo Samuel Castro, Aaron C. Courville, and Ling Pan. Asymmetric proximal policy optimization: mini-critics boost llm reasoning. *ArXiv*, abs/2510.01656, 2025b. URL https://api.semanticscholar.org/CorpusID:281724081.

MAA. American mathematics contest 12 (amc 12), November 2023. URL https://artofproblemsolving.com/wiki/index.php/AMC_12_Problems_and_Solutions. Accessed: 2026-04-30.

MAA. American invitational mathematics examination (aime), February 2024. URL https://artofproblemsolving.com/wiki/index.php/2024_AIME_I. Accessed: 2026-04-30.

MAA. American invitational mathematics examination (aime), February 2025. URL https://artofproblemsolving.com/wiki/index.php/2025_AIME_I. Accessed: 2026-04-30.

Penghui Qi, Zi-Yan Liu, Xiangxin Zhou, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Defeating the training-inference mismatch via fp16. *ArXiv*, abs/2510.26788, 2025. URL https://api.semanticscholar.org/CorpusID:282591916.

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, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your efficient rl framework secretly brings you off-policy rl training, August 2025. URL https://fengyao.notion.site/off-policy-rl.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Yaxiang Zhang, Yingru Li, Jiacai Liu, Jiawei Xu, Ziniu Li, Qian Liu, and Haoyuan Li. Beyond precision: Training-inference mismatch is an optimization problem and simple lr scheduling fixes it. *arXiv preprint arXiv:2602.01826*, 2026.

Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Huiqiang Jiang, Junrong Lin, Yuqiong Liu, Hao Lin, Chencan Wu, Feng Hu, et al. Stabilizing reinforcement learning with llms: Formulation and practices. *arXiv preprint arXiv:2512.01374*, 2025.

A Full Algorithm

We summarize the full training procedure of DC-GRPO in Algorithm 10. At each iteration, the same rollout policy weights θ_{old} are executed under two backends: the high-throughput inference backend $\pi^{\text{Inf}}_{\theta_{\text{old}}}$ (e.g., vLLM) that generates the sampled trajectories, and the high-fidelity training backend $\pi^{\text{Train}}_{\theta_{\text{old}}}$ (e.g., FSDP2/Megatron) that recomputes the per-token probabilities. The absolute token-level probability difference between the two backends constitutes the black-box discrepancy signal $\delta_{i,t}$, which (i) is injected into the GRPO advantage to regularize the policy toward the discrepancy tolerance region, and (ii) drives the dual gradient ascent on the Lagrangian multiplier λ. The resulting update rule requires only a single extra forward pass per step on the training backend beyond the standard GRPO pipeline, and introduces no additional learnable parameters besides the scalar λ.

Algorithm 1: DC-GRPO: Discrepancy-Constrained Group Relative Policy Optimization.

Input: Initial policy θ_0; initial multiplier λ_0; tolerance budget c; dual learning rate η_λ; multiplier range $[\lambda_{\min}, \lambda_{\max}]$; clipping range $(\epsilon_{\text{low}}, \epsilon_{\text{high}})$; group size G; total steps K.

Output: Trained policy parameters θ_K.

for $k = 0, 1, \ldots, K-1$ **do**

- Sample a prompt batch $\{q_b\}_{b=1}^{B} \sim P(Q)$;
- **Rollout (inference backend):** for each q_b, sample G responses $\{o_{b,i}\}_{i=1}^{G} \sim \pi^{\text{Inf}}_{\theta_{\text{old}}}(\cdot \mid q_b)$ and record per-token probabilities $\pi^{\text{Inf}}_{\theta_{\text{old}}}(o_{b,i,t} \mid q_b, o_{b,i,<t})$;
- **Recompute (training backend):** run a forward pass on the same tokens to obtain $\pi^{\text{Train}}_{\theta_{\text{old}}}(o_{b,i,t} \mid q_b, o_{b,i,<t})$;
- **Discrepancy:** $\delta_{b,i,t} \leftarrow \left|\pi^{\text{Train}}_{\theta_{\text{old}}}(o_{b,i,t} \mid q_b, o_{b,i,<t}) - \pi^{\text{Inf}}_{\theta_{\text{old}}}(o_{b,i,t} \mid q_b, o_{b,i,<t})\right|$;
- **Rewards & GRPO advantage:** compute $r^{\text{task}}_{b,i}$ via the verifiable verifier and $\hat{A}_{b,i} = \dfrac{r^{\text{task}}_{b,i} - \text{mean}(\{r^{\text{task}}_{b,i}\}_{i=1}^{G})}{\text{std}(\{r^{\text{task}}_{b,i}\}_{i=1}^{G})}$;
- **Discrepancy-penalized advantage:** $\hat{A}^{\texttt{DC-RL}}_{b,i,t} \leftarrow \hat{A}_{b,i} - \lambda_k \cdot \delta_{b,i,t}$;
- **Policy update (primal):** perform a GRPO clipped surrogate step on θ using $\hat{A}^{\texttt{DC-RL}}$:

$$\theta_{k+1} \leftarrow \theta_k + \nabla_\theta \left[\frac{1}{BG} \sum_{b,i} \frac{1}{|o_{b,i}|} \sum_{t} \min\Big(\gamma_{b,i,t}(\theta) \hat{A}^{\texttt{DC-RL}}_{b,i,t},\ \text{clip}(\gamma_{b,i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}) \hat{A}^{\texttt{DC-RL}}_{b,i,t} \Big) \right];$$

- **Multiplier update (dual):** $\bar{\delta}_k \leftarrow \dfrac{\sum_{b,i,t} \delta_{b,i,t}}{\sum_{b,i} |o_{b,i}|}$,;
- $\lambda_{k+1} \leftarrow \Pi_{[\lambda_{\min}, \lambda_{\max}]}\big(\lambda_k + \eta_\lambda(\bar{\delta}_k - c)\big)$;

end

B Optimal Policies and Toy Analysis for Alternative Discrepancy Models

B.1 Local Optimal Policies under Three Discrepancy Models

At a fixed decoding state s, consider the train-side policy $\pi^{\text{Train}}(\cdot \mid s)$ and the deployed inference policy $\pi^{\text{Inf}}(\cdot \mid s)$. We analyze the local surrogate

$$\max_{\pi^{\text{Train}}(\cdot|s) \in \Delta} \left\{ \sum_a \pi^{\text{Train}}(a \mid s) A(s,a) - \lambda\, \Omega_s\big(\pi^{\text{Train}}(\cdot \mid s), \pi^{\text{Inf}}(\cdot \mid s)\big) \right\}, \tag{12}$$

where Δ is the action simplex, $A(s,a)$ is the advantage, and $\lambda \geq 0$ is the penalty weight.

To derive closed-form or semi-closed-form optimal policies, we use the squared mismatch as the local constraint surrogate. Compared with the absolute-value form, the quadratic penalty is smooth and differentiable at zero, yields cleaner stationarity conditions, and penalizes larger discrepancies more strongly, making the geometry of the optimum analytically more tractable. This local analysis uses differentiable discrepancy surrogates; the detached implementation used in our main experiments is analyzed in Appendix D.2.

Theorem B.1. *Suppose $\pi^{\mathrm{Train}}(\cdot \mid s)$ is an interior maximizer of (12). Under the quadratic discrepancy models*

$$\Omega_s^{\mathrm{diff}} = \sum_a \pi^{\mathrm{Inf}}(a \mid s)\big(\pi^{\mathrm{Train}}(a \mid s) - \pi^{\mathrm{Inf}}(a \mid s)\big)^2, \tag{13}$$

$$\Omega_s^{\mathrm{ratio}} = \sum_a \frac{\big(\pi^{\mathrm{Train}}(a \mid s) - \pi^{\mathrm{Inf}}(a \mid s)\big)^2}{\pi^{\mathrm{Inf}}(a \mid s)}, \tag{14}$$

and

$$\Omega_s^{\log} = \sum_a \pi^{\mathrm{Inf}}(a \mid s) \left(\log \frac{\pi^{\mathrm{Train}}(a \mid s)}{\pi^{\mathrm{Inf}}(a \mid s)}\right)^2, \tag{15}$$

the corresponding stationary policies satisfy

$$\pi^*_{\mathrm{diff}}(a \mid s) = \pi^{\mathrm{Inf}}(a \mid s) + \frac{A(s,a) - \nu(s)}{2\lambda\, \pi^{\mathrm{Inf}}(a \mid s)}, \tag{16}$$

$$\pi^*_{\mathrm{ratio}}(a \mid s) = \pi^{\mathrm{Inf}}(a \mid s) + \frac{A(s,a) - \nu(s)}{2\lambda}\, \pi^{\mathrm{Inf}}(a \mid s), \tag{17}$$

and

$$\pi^*_{\log}(a \mid s) = -\pi^{\mathrm{Inf}}(a \mid s) \frac{2\lambda}{A(s,a) - \nu(s)} W\left(-\frac{A(s,a) - \nu(s)}{2\lambda}\right), \tag{18}$$

where $\nu(s)$ is the normalization multiplier and $W(\cdot)$ is the Lambert-W function. If any interior solution violates non-negativity, the optimum is obtained by the usual KKT projection back to the simplex.

Proof. Introduce the Lagrangian

$$\mathcal{L}_s(\pi, \nu) = \sum_a \pi^{\mathrm{Train}}(a \mid s) A(s,a) - \lambda\, \Omega_s\big(\pi^{\mathrm{Train}}(\cdot \mid s), \pi^{\mathrm{Inf}}(\cdot \mid s)\big) - \nu(s) \left(\sum_a \pi^{\mathrm{Train}}(a \mid s) - 1\right). \tag{19}$$

For the probability-difference model (13),

$$\frac{\partial \mathcal{L}_s}{\partial \pi^{\mathrm{Train}}(a \mid s)} = A(s,a) - 2\lambda\, \pi^{\mathrm{Inf}}(a \mid s)\big(\pi^{\mathrm{Train}}(a \mid s) - \pi^{\mathrm{Inf}}(a \mid s)\big) - \nu(s) = 0,$$

which yields (16). For the probability-ratio model (14),

$$\frac{\partial \mathcal{L}_s}{\partial \pi^{\mathrm{Train}}(a \mid s)} = A(s,a) - 2\lambda\, \frac{\pi^{\mathrm{Train}}(a \mid s) - \pi^{\mathrm{Inf}}(a \mid s)}{\pi^{\mathrm{Inf}}(a \mid s)} - \nu(s) = 0,$$

which yields (17). For the log-probability model (15),

$$\frac{\partial \mathcal{L}_s}{\partial \pi^{\mathrm{Train}}(a \mid s)} = A(s,a) - 2\lambda\, \pi^{\mathrm{Inf}}(a \mid s) \frac{\log(\pi^{\mathrm{Train}}(a \mid s)/\pi^{\mathrm{Inf}}(a \mid s))}{\pi^{\mathrm{Train}}(a \mid s)} - \nu(s) = 0.$$

Let $c(s,a) \triangleq A(s,a) - \nu(s)$ and $y(a) \triangleq \pi^{\mathrm{Train}}(a \mid s)/\pi^{\mathrm{Inf}}(a \mid s)$. Then the stationarity condition becomes

$$c(s,a)\, y(a) = 2\lambda \log y(a),$$

or equivalently

$$y(a) \exp\left(-\frac{c(s,a)}{2\lambda} y(a)\right) = 1.$$

Applying the Lambert-W function gives

$$y(a) = -\frac{2\lambda}{c(s,a)}\, W\left(-\frac{c(s,a)}{2\lambda}\right),$$

and substituting back yields (18). The simplex correction follows from the KKT conditions whenever the interior solution becomes negative. □

The theorem makes the geometry difference explicit. The probability-difference model scales the correction by $1/\pi^{\mathrm{Inf}}(a \mid s)$ and therefore allows more aggressive movement toward low-probability but high-advantage actions. The probability-ratio model scales the correction by $\pi^{\mathrm{Inf}}(a \mid s)$, which is markedly more conservative in the tail. The log-probability model is locally close to the ratio form but remains globally non-linear, providing an intermediate geometry between these two extremes.

Table 6: Toy action patterns for the one-state analysis.

Action	Pattern	Setting
a_1	**High** Prob. **Neg.** Adv.	$\pi^{\text{Inf}}(a_1 \mid s) = 0.30,\ A(s, a_1) = -0.1$
a_2	**High** Prob. **Pos.** Adv.	$\pi^{\text{Inf}}(a_2 \mid s) = 0.30,\ A(s, a_2) = +0.1$
a_3	**Medium** Prob. **Neg.** Adv.	$\pi^{\text{Inf}}(a_3 \mid s) = 0.15,\ A(s, a_3) = -0.5$
a_4	**Medium** Prob. **Pos.** Adv.	$\pi^{\text{Inf}}(a_4 \mid s) = 0.15,\ A(s, a_4) = +0.5$
a_5	**Low** Prob. **Neg.** Adv.	$\pi^{\text{Inf}}(a_5 \mid s) = 0.05,\ A(s, a_5) = -1.0$
a_6	**Low** Prob. **Pos.** Adv.	$\pi^{\text{Inf}}(a_6 \mid s) = 0.05,\ A(s, a_6) = +1.0$

B.2 A One-State Toy Experiment

To complement the theorem, we consider a one-state toy problem that isolates how the three discrepancy models reshape the policy under the same advantage signal. The action set is designed to cover six distinct patterns, pairing sampling probability with advantage sign: one **High** Prob. **Pos.** Adv. action and one **High** Prob. **Neg.** Adv. action; one **Medium** Prob. **Pos.** Adv. action and one **Medium** Prob. **Neg.** Adv. action; and one **Low** Prob. **Pos.** Adv. action and one **Low** Prob. **Neg.** Adv. action. Table 6 summarizes these patterns.

For each discrepancy family, we solve

$$\pi^*(\lambda) = \arg\max_{\pi \in \Delta} \left\{ \langle \pi, A \rangle - \lambda\, \Omega(\pi, \pi^{\text{Inf}}) \right\}, \tag{20}$$

and choose λ so that the corresponding quadratic budget is active, $\Omega(\pi^*(\lambda), \pi^{\text{Inf}}) \approx C$. Figures 5 and 6 visualize the resulting optima under two-sided and one-sided penalties, respectively.

Several patterns are worth emphasizing. First, the direction of reallocation is largely determined by the sign of the advantage: all three penalties consistently suppress the **Neg.** Adv. actions (a_1, a_3, a_5) and promote the **Pos.** Adv. actions (a_2, a_4, a_6). In particular, the paired **Low** Prob. actions provide a clean control: the **Low** Prob. **Neg.** Adv. action a_5 is suppressed, while the **Low** Prob. **Pos.** Adv. action a_6 is promoted.

The more important difference is the magnitude of this flow. Under probability difference, moving mass onto the strongest **Low** Prob. **Pos.** Adv. action a_6 is comparatively cheap, so the optimizer concentrates probability on a_6 most aggressively and tends to produce a sparser solution. Probability ratio spreads the reallocation more gradually across a_2, a_4, and a_6, preserving broader support because tail amplification is much more expensive. Log-probability difference remains intermediate: around the inference policy it behaves similarly to the ratio penalty, but once larger relative drifts become beneficial, its non-linearity allows more movement toward a_6 than pure ratio regularization while still avoiding the most aggressive tail-seeking behavior of probability difference.

Take-Away

At the same quadratic budget level, the three discrepancy models agree on the *direction* of advantage-weighted reallocation, moving mass away from **Neg.** Adv. actions and toward **Pos.** Adv. actions, but they differ sharply in the *magnitude* of that reallocation. When high advantage lies on low-probability actions, probability difference makes those moves cheapest, probability ratio makes them most expensive, and log-probability difference provides an intermediate compromise.

This toy analysis highlights the practical message behind the theorem: once the advantage signal is fixed, the discrepancy model acts as a geometric filter on which reward-improving reallocations are considered cheap or expensive. Consequently, the choice among probability difference, probability ratio, and log-probability difference is not merely a matter of metric design; it directly determines how the learned policy redistributes probability mass under advantage-weighted updates.

C Additional Experiments

This section supplements the main experiments with comparisons to mismatch-mitigation baselines, an evaluation of compatibility with MoE routing alignment, a budget-sensitivity study across model sizes, and analyses of discrepancy penalties and policy updates. The baseline comparisons use the

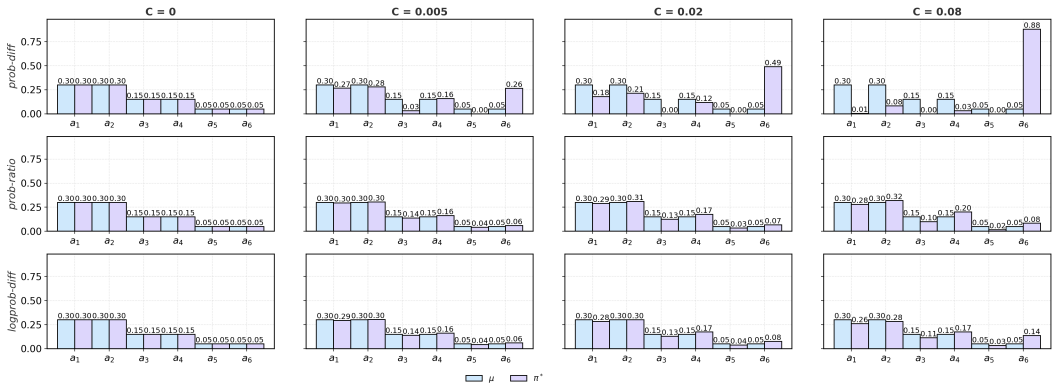

Figure 5: Two-sided toy comparison under a shared quadratic mismatch budget. Columns sweep the budget level C; rows correspond to probability difference, probability ratio, and log-probability difference penalties.

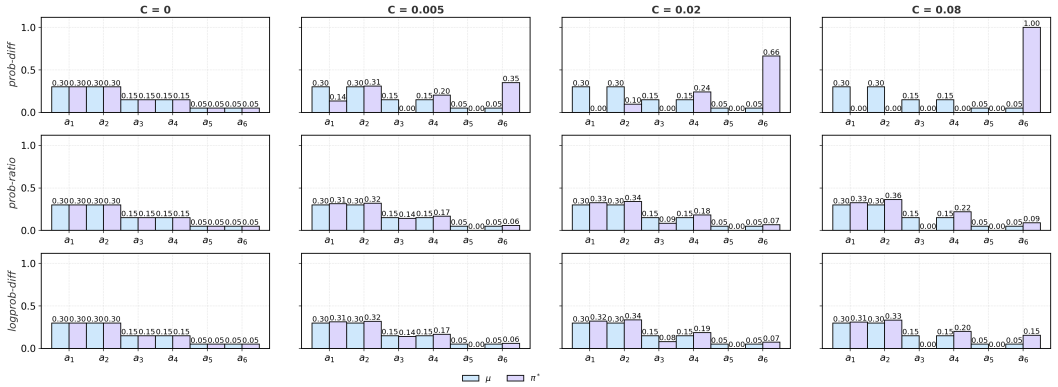

Figure 6: One-sided toy comparison under a shared quadratic mismatch budget. Columns sweep the budget level C; rows correspond to probability difference, probability ratio, and log-probability difference penalties.

BF16/BF16 and BF16/FP8 settings of the main experiments. FP16 alignment is a baseline method, not the shared experimental setting. We state the evaluation setting and repetition count separately for each experiment.

C.1 Comparisons with Mismatch-Mitigation Baselines

We compare DC-GRPO with TIS [Yao et al., 2025], MIS [Li et al., 2025], learning-rate scheduling [Zhang et al., 2026], and FP16 alignment [Qi et al., 2025].

We evaluate Qwen3-8B-Base under the two precision settings used in the main experiments: (i) BF16 training with BF16 inference, and (ii) BF16 training with FP8 inference. The FP16 alignment baseline changes both training and inference precision to FP16 and is included in the first comparison. It is not included in the second comparison, which requires FP8 inference. Both comparisons report averages over five seeds. For TIS, MIS, and learning-rate scheduling, we use the parameter settings recommended in the corresponding papers.

DC-GRPO achieves the highest reported average score among the compared methods in both settings. The larger average margin in the latter comparison supports the utility of the discrepancy penalty under the tested heterogeneous precision configuration. These aggregate improvements do not imply that DC-GRPO outperforms every baseline on every individual benchmark.

Table 7: Evaluation results on mathematical benchmarks for Qwen3-8B under the BF16 training/inference and heterogeneous BF16-training/FP8-inference regimes. Results are averaged over five seeds. The FP16 alignment baseline switches both training and inference to FP16. The results of DC-GRPO are shaded and the highest mean values within each regime are **bolded**.

Method	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	Avg.
BF16 training / BF16 inference							
↳ TIS	60.2 ± 0.3	46.1 ± 0.6	91.3 ± 0.3	92.2 ± 0.2	**52.1** ± 0.4	71.9 ± 0.6	69.0
↳ MIS	61.4 ± 0.1	47.6 ± 0.2	90.8 ± 0.6	91.6 ± 0.3	50.8 ± 0.7	70.8 ± 0.2	68.8
↳ LR scheduler	59.7 ± 0.1	47.5 ± 0.3	92.3 ± 0.8	90.4 ± 0.6	49.2 ± 0.5	71.3 ± 0.8	68.4
↳ FP16 alignment	60.4 ± 0.2	46.8 ± 0.4	89.2 ± 0.9	91.8 ± 0.5	50.9 ± 0.7	71.1 ± 0.2	68.4
↳ DC-GRPO	**64.8** ± 0.4	**52.2** ± 0.2	**93.1** ± 0.3	**94.6** ± 0.3	51.4 ± 0.2	**74.1** ± 0.4	**71.7**
BF16 training / FP8 inference							
↳ TIS	52.6 ± 0.7	36.3 ± 0.9	89.6 ± 0.4	92.3 ± 0.6	51.4 ± 0.1	68.2 ± 0.4	65.1
↳ MIS	53.8 ± 0.1	37.2 ± 0.3	90.1 ± 0.5	92.8 ± 0.1	51.2 ± 0.4	68.7 ± 0.3	65.6
↳ LR scheduler	53.1 ± 0.2	35.2 ± 0.9	88.2 ± 0.4	92.4 ± 0.7	51.3 ± 0.2	67.6 ± 0.5	64.6
↳ DC-GRPO	**56.5** ± 0.6	**47.3** ± 0.9	**91.2** ± 0.4	**94.7** ± 0.3	**55.1** ± 0.7	**71.2** ± 0.1	**69.3**

Table 8: Compatibility with routing alignment on Qwen3-30B-A3B. Each value is the average score over six benchmarks from a single run.

Method	Avg.
Rollout Routing Replay	66.4
DC-GRPO	67.7
DC-GRPO + Rollout Routing Replay	72.5

C.2 Compatibility with MoE Routing Alignment

We evaluate whether discrepancy-aware policy updates can be combined with a method that addresses routing inconsistency in MoE models. On Qwen3-30B-A3B, we compare DC-GRPO, Rollout Routing Replay, and their combination.

The combined method obtains an average score of 72.5, exceeding both DC-GRPO alone (67.7) and Rollout Routing Replay alone (66.4). These results show that DC-GRPO can be combined with Rollout Routing Replay to obtain further performance gains.

C.3 Sensitivity to the Discrepancy Budget across Model Sizes

To examine whether the budget-selection guidance extends beyond the Qwen3-8B ablation in Table 5, we evaluate the sensitivity to c across three model sizes: 1.7B-Base, 8B-Base, and 30B-A3B. Under BF16 training and FP8 inference, we sweep $c \in \{0.0030, 0.0033, 0.0037, 0.0040, 0.0043\}$ and report the average score across the six mathematical reasoning benchmarks.

Table 9: Ablation on the tolerance budget c across model sizes under BF16 training and FP8 inference. Each entry is the average score across six mathematical benchmarks. The results for $c = 0.0037$ are shaded and the highest values are **bolded**.

c	1.7B-Base	8B-Base	30B-A3B
0.0030	61.5	67.3	65.8
0.0033	62.5	68.7	66.3
0.0037	**62.9**	**69.4**	**67.7**
0.0040	60.8	68.3	66.9
0.0043	60.4	67.2	64.8

Table 9 shows that $c = 0.0037$ achieves the highest average score for all three tested models. Smaller and larger budgets reduce performance relative to this value, supporting the need to balance discrepancy control with reward optimization. The results also show that $c = 0.0033$ provides competitive performance across the tested model sizes. These observations suggest $c \in [0.0033, 0.0037]$ as a useful initial search range for the tested BF16-training/FP8-inference configurations.

D Mechanism and Gradient Analysis of Discrepancy-Aware Updates

This section explains how the discrepancy penalty influences policy updates, its distinction from importance-sampling-based correction, and the relationship between the detached implementation and a differentiable discrepancy objective. We also provide empirical analyses of update behavior and compare the detached and differentiable implementations.

D.1 Discrepancy-Aware Updates and Comparison with Importance Sampling

Mechanism of discrepancy-aware advantage shaping. The discrepancy penalty is incorporated into the advantage as

$$\widehat{A}_{i,t}^{\mathrm{DC}} = \widehat{A}_{i,t} - \lambda_k \delta_{i,t}. \tag{21}$$

Although $\delta_{i,t}$ is detached from the current policy parameters, it directly affects the advantage used in the policy update. When the training–inference discrepancy is sufficiently large, the penalty can turn an initially positive advantage into a negative one, changing the corresponding token from being promoted to being suppressed. The resulting update therefore incorporates both the task reward and feedback about training–inference discrepancy, guiding the policy toward generation patterns with high rewards and low mismatch.

Distinction from importance-sampling-based correction. Importance-sampling-based correction and masking adjust the weight of token-level policy-gradient contributions. In particular, non-negative reweighting does not reverse the sign of the original advantage. By contrast, the additive penalty in Eq. (21) can reverse that sign according to the measured training–inference discrepancy. This provides a mechanism for explicitly suppressing sampled actions with large discrepancy, even when their original reward-derived advantages are positive.

In this sense, discrepancy awareness refers to incorporating backend disagreement directly into the advantage calculation. Self-correction refers to the resulting adjustment of policy updates: the penalty can suppress high-discrepancy actions and guide the policy toward generation patterns that jointly support reward maximization and discrepancy reduction.

Changes in optimization direction. We compare the reported frequency of gradient-direction changes between consecutive training steps for DC-GRPO, TIS, MIS, and learning-rate scheduling. As shown in Table 10, DC-GRPO exhibits a higher average frequency than the compared methods. This observation complements the mechanism above, in which the discrepancy signal can modify the reward-driven update through an additive advantage penalty.

Table 10: Gradient-direction changing frequencies between training steps t and $t-1$, averaged over five runs. The result of `DC-GRPO` is shaded.

Method	Frequency
↳ TIS	0.13 ± 0.09
↳ MIS	0.16 ± 0.13
↳ LR scheduler	0.11 ± 0.07
↳ `DC-GRPO`	0.21 ± 0.05

Advantage-sign reversal. We further measure how frequently the discrepancy penalty changes the sign of the initial advantage during training. Table 11 compares DC-GRPO with MIS [Li et al., 2025]. DC-GRPO exhibits a nonzero reversal frequency, whereas MIS does not reverse the original advantage sign. These results demonstrate that the discrepancy penalty can reverse reward-driven advantages, providing an explicit mechanism for redirecting updates on high-discrepancy tokens.

Table 11: Advantage-sign reversal frequencies for `DC-GRPO` and MIS. The result of `DC-GRPO` is shaded.

Method	Reversal frequency
↳ MIS	0
↳ `DC-GRPO`	0.13 ± 0.06

D.2 Detached and Differentiable Gradient Analysis

The detached black-box implementation and the differentiable discrepancy formulation must be distinguished when interpreting the gradient analysis. Both are motivated by the discrepancy-constrained Lagrangian

$$\max_{\theta} \min_{\lambda \geq 0} \left[J(\theta) - \lambda\big(g(\theta) - c\big) \right], \tag{22}$$

but differ in whether a direct gradient through the discrepancy is retained.

For a parameter-dependent discrepancy cost evaluated on sampled trajectories, differentiation gives

$$\begin{aligned} &\nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta} \left[\delta_{i,t}(\theta) \right] \\ &\quad = \mathbb{E}_{\tau \sim \pi_\theta} \left[\delta_{i,t}(\theta) \nabla_\theta \log \pi_\theta(\tau) + \nabla_\theta \delta_{i,t}(\theta) \right]. \end{aligned} \tag{23}$$

The first term is the policy-gradient, or score-function, term. The second term is the direct gradient through the discrepancy. Detaching the discrepancy removes the second term while retaining the first.

Detached gradient analysis. The main implementation computes the discrepancy using the old policy and applies stop-gradient:

$$\begin{aligned} \delta_{i,t} = \mathrm{sg}\Big[\Big| &\pi^{\mathrm{Train}}_{\theta_{\mathrm{old}}}(o_{i,t} \mid q, o_{i,<t}) \\ &- \pi^{\mathrm{Inf}}_{\theta_{\mathrm{old}}}(o_{i,t} \mid q, o_{i,<t}) \Big| \Big], \qquad \nabla_\theta \delta_{i,t} = 0. \end{aligned} \tag{24}$$

Here, $\mathrm{sg}[\cdot]$ denotes stop-gradient. The direct-gradient term therefore vanishes, while the policy-gradient contribution remains:

$$\nabla_\theta J_{\text{DC-GRPO}} = \mathbb{E}\left[\sum_{i,t} \mathbf{1}^{\mathrm{clip}}_{i,t} \gamma_{i,t}(\theta) \big(\widehat{A}_{i,t} - \lambda_k \delta_{i,t} \big) \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}) \right]. \tag{25}$$

The clipping indicator $\mathbf{1}^{\mathrm{clip}}_{i,t}$ identifies the active policy-gradient contributions in the clipped surrogate. Thus, the detached discrepancy acts as a black-box cost optimized through policy gradient. It can down-weight or reverse updates associated with high-discrepancy tokens without directly differentiating the backend probability gap.

Differentiable discrepancy formulation. If the discrepancy is retained as a differentiable function of the current parameters,

$$\begin{aligned} \delta_{i,t}(\theta) = \Big| &\pi^{\mathrm{Train}}_{\theta}(o_{i,t} \mid q, o_{i,<t}) \\ &- \pi^{\mathrm{Inf}}_{\theta}(o_{i,t} \mid q, o_{i,<t}) \Big|, \end{aligned} \tag{26}$$

then the direct-gradient term is retained:

$$\begin{aligned} \nabla_\theta \delta_{i,t}(\theta) = &\operatorname{sgn}\left(\pi^{\mathrm{Train}}_{\theta} - \pi^{\mathrm{Inf}}_{\theta} \right) \\ &\cdot \left(\nabla_\theta \pi^{\mathrm{Train}}_{\theta} - \nabla_\theta \pi^{\mathrm{Inf}}_{\theta} \right), \end{aligned} \tag{27}$$

where the token and context arguments are omitted for brevity. In the Lagrangian update, this contributes $-\lambda_k \nabla_\theta \delta_{i,t}(\theta)$, which directly penalizes the probability gap.

Keeping the discrepancy in the computation graph yields the corresponding surrogate-gradient expression

$$\begin{aligned} \nabla_\theta J^{\mathrm{attach}}_{\text{DC-GRPO}} = \mathbb{E}\Bigg[&\sum_{i,t} \mathbf{1}^{\mathrm{clip}}_{i,t} \gamma_{i,t}(\theta) \widehat{A}^{\mathrm{DC}}_{i,t}(\theta) \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}) \\ &- \lambda_k \sum_{i,t} \mathbf{1}^{\mathrm{clip}}_{i,t} \gamma_{i,t}(\theta) \nabla_\theta \delta_{i,t}(\theta) \Bigg], \end{aligned} \tag{28}$$

where $\widehat{A}^{\mathrm{DC}}_{i,t}(\theta) = \widehat{A}_{i,t} - \lambda_k \delta_{i,t}(\theta)$. The distinction between Eqs. (25) and (28) makes explicit which gradient contribution is absent from the detached implementation and retained in the differentiable variant.

Empirical comparison. We additionally compare the detached implementation with the differentiable variant under BF16 training and FP8 inference. Table 12 reports the average performance over five runs and six mathematical benchmarks. The two variants achieve similar average scores: 69.4 for the detached implementation and 69.1 for the differentiable variant, both outperforming standard GRPO.

Table 12: Comparison of detached and differentiable discrepancy penalties under BF16 training and FP8 inference. Results are averaged over five runs and six mathematical benchmarks. The result of the detached implementation used by `DC-GRPO` is shaded and the highest value is **bolded**.

Variant	**Avg.**
Differentiable penalty	69.1
Detached penalty	**69.4**

E The Use of Large Language Models

In this paper, LLMs are only used to polish the writing of some paragraphs to improve clarity and grammar. The key ideas, theoretical analysis, method design, figures, and experimental results are all from the human authors' contributions, and LLMs do not constitute a core component of the proposed methodology.

F Licenses

The licenses of used assets in this paper are listed as follows:

- MATH: MIT License
- AMC23: Apache-2.0 License
- Minerva Math: No License
- OlympiadBench: MIT License
- verl: Apache-2.0 License

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Introduction for our contribution and experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See our limitation at final section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Seemethod part for the theory and its proof.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See section 6.1 for our experiment settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use data that's totally open-sourced.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See hyperparameters in experiment part.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See section 6.1 for details

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that this research conforms to all its guidelines.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This research focuses on fundamental algorithmic improvements in reinforcement learning. The models and data used (e.g., standard benchmarks) do not pose a high risk of misuse as defined by the guidelines.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by re-

quiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See setup part for details.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: See main paper for details.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This research does not involve crowdsourcing or experiments with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve human subjects, personal data, or any experiments requiring IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were only used for language editing and proofreading of the manuscript, which does not impact the core methodology, scientific rigor, or originality of the research as per the conference guidelines.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.

- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.