Historical Relative Policy Optimization for Bootstrapping LLM Reasoning

Sitong Wu[1] **Haoru Tan**[2] **Bei Yu**[1] **Xiaojuan Qi**[2] **Jiaya Jia**[1,3]*

[1]The Chinese University of Hong Kong [2]The University of Hong Kong
[3]The Hong Kong University of Science and Technology

Abstract

Reinforcement learning has become a key approach for optimizing large language models (LLMs), with Group Relative Policy Optimization (GRPO) emerging as the most popular algorithm due to its simplicity and effectiveness. However, GRPO normalizes advantages purely within the current sampling group, ignoring the dynamic evolution of the policy's performance throughout training. This renders the model susceptible to *relative deception*, *i.e.*, the model may be steadily deteriorating yet remain oblivious to its own regression. To address this issue, we propose *Historical Relative Policy Optimization (HRPO)*, which integrates two synergistic designs: (i) a *historical-aware advantage estimator* that normalizes each response against the running maximum of the policy's strongest past group-mean rewards, thereby pushing the model to continuously surpass its own historical peak (bootstrapping); and (ii) an *on-demand historical replay* that recalls historical high-reward responses as positive examples, which is triggered only when all responses in the current step fail to provide positive signals. Experiments show that HRPO consistently outperforms GRPO variants in training stability, convergence speed, and final performance across diverse models and benchmarks, demonstrating that accounting for training dynamics leads to more reliable and effective policy optimization in LLMs. The code will be released at https://github.com/JIA-Lab-research/HRPO.

1 Introduction

Reinforcement Fine-tuning (RFT) constitutes a pivotal stage in the post-training of Large Models [8, 12, 43, 54, 38, 53]. It has been widely demonstrated to play a crucial role in unlocking the models' reasoning abilities and promoting generalization performance [43, 12]. Algorithms in the GRPO family have become a major focus of research and practical deployment due to their simplicity and effectiveness. These methods eliminate the critic model found in traditional RL algorithms [42]. Instead, they estimate advantage values by sampling a set of responses and normalizing rewards to compute the relative advantage within that set.

However, in practical applications, the GRPO family [43, 55, 17, 7] of algorithms often encounters a critical instability issue. Specifically, model performance tends to follow a non-monotonic trajectory: improving during the initial phases of training but subsequently exhibiting severe oscillations or even deterioration, sometimes leading to complete policy model collapse. To mitigate this, practitioners are currently forced to rely on ad hoc strategies, such as early stopping or frequent checkpointing, to capture the optimal model before degradation occurs.

In this paper, we reveal that this phenomenon is closely associated with a fundamental mechanism we term the **relative deception effect**. Specifically, because the GRPO family computes advantage

*Corresponding author

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

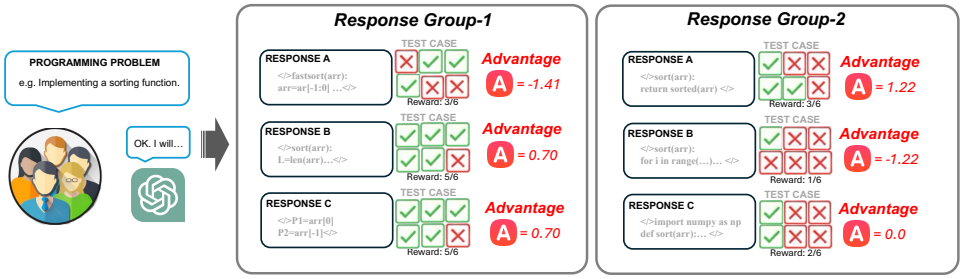

Figure 1: Illustration of the Relative Deception effect. Due to the inherent RL stochasticity and sampling randomness, the quality of on-policy response groups can vary significantly, leading to unstable and unreasonable advantage estimation. As shown in this figure, for Group-1 at the historical training step, the generated answers are of higher absolute quality (*e.g.*, responses B and C both achieve a 5/6 reward), but they only receive a maximum advantage score of 0.70. For Group-2 at the current training step, the overall quality is markedly inferior (*e.g.*, response A only achieves a 3/6 reward), yet it receives a disproportionately high advantage score of 1.22 simply by being the *best of a poor batch*.

values by considering only the group of sampled responses in the current training step, it remains agnostic to the dynamic evolution of the policy's performance throughout the training process. This creates a major loophole in the optimization objective. Due to the high variance and randomness of RL sampling, there are instances where all responses in a group are subpar. Yet, since advantages are derived from current step intra-group comparisons, the best among these poor responses is still treated as a positive signal. This *relative deception* effect forces the model to reinforce sub-optimal behaviors, establishing a vicious cycle where the normalization baseline collapses alongside the model's performance. We give a brief illustration in Fig. 1 and discuss more in Sec. 2.2.

To address these challenges, we propose **Historical Relative Policy Optimization (HRPO)**, a simple yet highly effective framework built upon two complementary mechanisms:

(1) Historical-aware Advantage Estimator: HRPO incorporates historical statistics into the advantage estimation process. Specifically, it replaces the instantaneous within-group baseline with a *historical peak anchor* $\mu(q)$, defined as the running maximum of the per-step group-mean rewards observed for the same prompt q across all past training steps. This anchor imposes an explicit bootstrapping pressure: the policy must continuously surpass its own strongest past performance to earn any positive reinforcement, which immediately disqualifies the spurious "best of a bad lot" signal at the heart of relative deception.

(2) On-Demand Historical Replay: Anchoring at the historical peak, however, may lead to a *positive-signal starvation* problem: when the current group's best reward fails to reach $\mu(q)$, all advantages collapse to negative values, leaving the optimizer aware of which directions to avoid but blind to which directions to pursue. To address this, we introduce an *On-Demand Historical Replay* (ODHR) mechanism that injects the model's own past responses with high-reward strictly exceeding $\mu(q)$ as positive examples for the current training step, reweighted with importance sampling to bridge the off-policy gap. Notably, to preserve on-policy learning as much as possible, ODHR is not always triggered; it activates only when the aforementioned positive-signal starvation occurs and qualifying historical trajectories are available in the buffer.

Empirical results on various tasks and benchmarks demonstrate that HRPO significantly outperforms existing methods like GRPO, DAPO, and REINFORCE++. Particularly, our method effectively (i) stabilizes the training, mitigating collapse and prompting bootstrap; (ii) accelerates the convergence; (iii) boosts final performance. For instance, HRPO achieves gains of +6.3 on AIME24, +5.8 on AIME25, and +4.0 on OlympiadBench for the Qwen3-8B-Base model compared with GRPO. In summary, our main contributions are as follows:

- We identify and analyze the *Relative Deception Effect* in critic-free RL algorithms, revealing how instantaneous normalization advantage estimator can lead to training instability and regression.
- We propose *HRPO*, which couples two complementary mechanisms: a *Historical-aware Advantage Estimator* with historical peak mean reward as the normalization anchor, and an *On-Demand Historical Replay* that recovers positive signals from the model's own past responses when the current step's response encounter a positive-signal starvation problem.

- We conduct extensive experiments across multiple model scales (from 1.5B to 8B) and model types (base and instruct LLMs), showing that HRPO consistently sustains upward performance trends and achieves state-of-the-art results on various reasoning tasks (math, code, logical, science, SearchQA).

2 Preliminaries

2.1 Background

Unlike traditional PPO [42], which relies on a complex online critic, critic-free variants such as GRPO [43] and REINFORCE++ [17] simplify optimization by using reward normalization. GRPO extends PPO by defining advantages through relative comparisons within a group of sampled responses for the same prompt. At t-th training step, given a prompt q, GRPO samples a group of G responses $\mathcal{O}^t(q) = \{o_i^t\}_{i=1}^G$, and the optimization objective is:

$$\mathcal{J}^{\text{GRPO}}(\theta) \;=\; \mathbb{E}_{o\in\mathcal{O}^t(q)}\Big[\, A^{\text{GRPO}}(q,o)\cdot\min\Big(\rho(q,o),\ \text{clip}\big(\rho(q,o),\ 1-\epsilon,\ 1+\epsilon\big)\Big)\,\Big] - \beta\,\text{KL}\big(\pi_\theta \,\big\|\, \pi_{\text{ref}}\big), \quad (1)$$

where $\rho(q,o) = \frac{\pi_\theta(o|q)}{\pi_{\theta^t_{\text{old}}}(o|q)}$ is the importance ratio that corrects for the distributional shift between the current policy π_θ and the behavior policy $\pi_{\theta^t_{\text{old}}}$ that generates the responses. Optimization stability is enforced by two devices: the clipping operator $\text{clip}(\cdot, 1-\epsilon, 1+\epsilon)$ that bounds per-step update magnitudes, and the KL penalty $\beta\,\text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}})$ that regularizes the policy toward the reference model π_{ref} (typically the initial checkpoint before RL), with β controlling the strictness. The advantage of the i-th response is computed by within-group reward normalization:

$$A^{\text{GRPO}}(q,o_i^t) \;=\; \frac{r_i^t(q) - \text{mean}\big(\{r_j^t(q)\}_{j=1}^G\big)}{\text{std}\big(\{r_j^t(q)\}_{j=1}^G\big)}, \quad (2)$$

where $r_i^t(q)$ is the reward of the i-th response for prompt q at t-th step.

2.2 Motivation: Relative Deception Effect

Despite the efficiency of the group-relative normalization schemes used in GRPO family [43, 17], they suffer from a fundamental flaw that we term the *Relative Deception Effect*. This phenomenon stems from the purely instantaneous and local nature of the normalization baseline in advantage estimation. For example, in GRPO (Eq. (2)), the advantage is computed by centering rewards around the mean of the current step's samples. While this effectively identifies the best responses in the current training step, it remains agnostic to the varying policy evolved throughout the training. As training progresses, the policy π_θ is dynamic; however, the normalization baseline is blind, it only sees the current step's distribution. This leads to a decoupling between the model's relative advantage and its absolute progress.

Specifically, consider a scenario where the model's absolute reasoning capability begins to degrade due to the high stochasticity of RL. In this case, the rewards $\{r_1, \ldots, r_G\}$ for the entire group may be objectively low. However, because the baseline mean$(\{r_i | i \le G\})$ also drops in tandem with the failing policy, a subpar response may still receive a positive advantage simply by being the *best of a bad lot*. Consequently, the optimization objective continues to reinforce these regressive behaviors, creating a destructive feedback loop where the model is 'deceived' into believing it is improving while its absolute performance collapses. We provide an intuitive visualization of this phenomenon in Fig. 1.

In Fig. 2, we illustrate a representative RL process experiencing training collapse. As training progresses for GRPO, the advantage values almost exhibit an upward trend, albeit with some oscillation. Notably, in the latter stages, the baseline value (mean$(\{r_i | i \le G\})$, also the mean reward) begins to decline. Despite the visible drop in the mean reward during the final phases, the GRPO advantages remain paradoxically high. This trend illustrates the "Relative Deception Effect" in action: the advantage calculation fails to recognize the systemic decline in sample quality. Because the baseline (the mean reward of the current group) is also decreasing, even poor responses can receive a positive advantage, creating a false sense of policy improvement that ultimately destabilizes the training. Notably, our HRPO is no longer susceptible to this issue.

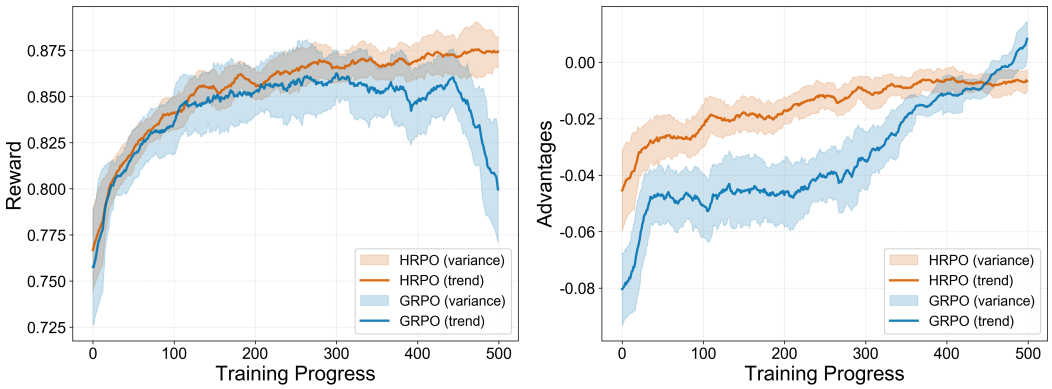

Figure 2: Illustration of Training Instability and Collapse induced by the Relative Deception Effect in group-based critic-free RL. While GRPO (blue) initially improves, its advantage signal eventually decouples from absolute quality, leading to a significant performance drop and training collapse after approximately 400 steps. In contrast, our HRPO (orange) maintains consistent reward growth and superior training stability by mitigating this relative deception. Note: Advantage curves are token-length-weighted batch means; per-group mean is zero by construction (Prop. 1, see Eq. (24)), and the non-zero drift intensifies during collapse.

3 Method

In this section, we present the details of the Historical Relative Policy Optimization (HRPO) framework. We first describe the core components and algorithmic design of HRPO in Sec. 3.1. Then, in Sec. 3.2, we introduce HRPO*, a variant designed to handle history cold-start via reference-initialized seeding. To facilitate a deeper understanding of the benefits of our approach, we provide a comprehensive analysis from both intuitive (Sec. 3.3) and theoretical (Sec. 3.4) perspectives.

3.1 Historical Relative Policy Optimization

To address the limitations identified in Sec. 2.2, we propose Historical Relative Policy Optimization (HRPO). The core innovations of HRPO lie in **two complementary mechanisms**: (i) *Historical-aware Advantage Estimation* (Sec. 3.1.1). It replaces the instantaneous within-group baseline (in GRPO) with a Historical Peak Anchor $\mu(q)$ that forces the policy to continuously surpass its own strongest past performance; (ii) *On-Demand Historical Replay* (Sec. 3.1.2) that injects the model's own historical high-reward responses as positive examples whenever the current step's responses fail to provide a constructive positive learning signal, supplying the optimizer with both directions to avoid and pursue. Together, these two mechanisms form a closed-loop optimization that fundamentally reshapes policy learning dynamics, sustaining stable and continuous performance improvement throughout training. The objective function is presented in Sec. 3.1.3.

Intuitively, HRPO is no longer about picking the best of a bad lot; instead, it prompts the policy model to surpass its own historical best performance.

3.1.1 Historical-aware Advantage Estimation

HRPO computes an advantage by comparing each response's reward against a *historical peak anchor* constructed from the same prompt's rewards across all past training steps. This ensures responses are evaluated against the policy's strongest past performance rather than its (potentially degraded) instantaneous state. Formally, for the i-th response at the current t-th step (denoted by $o_i^t(q)$), let $r_i^t(q)$ be its reward and let G be the group size. We denote $\mathcal{O}^t(q) = \{o_1^t, ..., o_G^t\}$.

Historical Peak Anchor. For each prompt q, let $\mathcal{T}(q) = \{k : q \text{ appeared at step } k\}$ denote the set of training steps at which q was sampled, including the current step t. We maintain a lightweight per-prompt history $\mathcal{H}(q)$ that records the per-step group mean rewards observed so far,

$$\mathcal{H}(q) \;=\; \{\mu^k(q)\}_{k\in\mathcal{T}(q)}, \qquad \mu^k(q) = \frac{1}{G}\sum_{j=1}^{G} r_j^k(q), \tag{3}$$

and define the *historical peak anchor* as the running maximum over $\mathcal{H}(q)$:

$$\mu(q) \;=\; \max\big(\mathcal{H}(q)\big) \;=\; \max_{k\in\mathcal{T}(q)} \mu^k(q). \tag{4}$$

Advantage Estimator. The HRPO advantage of the i-th response at current step t is then defined as:

$$A^{\text{HRPO}}\big(q, o_i^t(q)\big) \;=\; r_i^t(q) - \mu(q). \tag{5}$$

The peak anchor $\mu(q)$ ensures that a response receives a positive advantage only if it surpasses the model's strongest historical group-average performance, rather than merely the current step's group average performance that may degrade. This explicitly suppresses the relative deception effect: when the policy regresses on prompt q, the current step's mean reward $\mu^t(q)$ becomes low, but $\mu(q)$ remains anchored at the historical peak, so the spurious "best of a bad lot" can no longer earn positive reinforcement. We deliberately omit the within-group standard deviation as a normalization denominator, following the analysis of Dr.GRPO [29], which shows that std-based normalization in GRPO introduces a bias.

Two Regimes of $\mu(q)$. Since $\mathcal{T}(q)$ already contains the current step t, the anchor admits the closed form $\mu(q) = \max\big(\mu_{\text{old}}(q), \mu^t(q)\big)$, where $\mu_{\text{old}}(q) = \max_{k\in\mathcal{T}(q)\setminus\{t\}} \mu^k(q)$ is the peak observed before step t. This yields a clean two-regime behavior of A^{HRPO}:

- ***Progress regime*** ($\mu^t(q) > \mu_{\text{old}}(q)$): the current group breaks the historical peak, and HRPO reduces to GRPO-style within-group centering, $A^{\text{HRPO}} = r_i^t(q) - \mu^t(q)$, rewarding stronger responses while penalizing weaker ones to push progress further.
- ***Stagnation regime*** ($\mu^t(q) \le \mu_{\text{old}}(q)$): the current group fails to surpass the peak, and the historical anchor takes over, $A^{\text{HRPO}} = r_i^t(q) - \mu_{\text{old}}(q)$, exposing the regression and shielding the optimizer from the relative deception effect.

In short, HRPO can be viewed as *GRPO equipped with a historical-peak floor*: it defers to within-group statistics when the policy is making progress, and falls back to historical awareness exactly when the current step alone is no longer trustworthy.

3.1.2 On-Demand Historical Replay

The peak anchor in Eq. (4) effectively suppresses regressive reinforcement, but it may also lead to a complementary failure mode: when the current step's best reward $\max_j r_j^t(q)$ falls at or below $\mu(q)$, all sampled responses in the current step $\{o_i^t \mid i \le G\}$ receive non-positive advantages, so that the optimizer then knows what to avoid but lacks any positive direction to pursue. We refer to this as *positive-signal starvation*. To close this loop, HRPO is paired with an On-Demand Historical Replay (ODHR) mechanism that replays high-reward responses from the model's own past steps $\mathcal{T}(q)$ as positive examples for the current step t. To preserve on-policy learning as much as possible, ODHR is activated only when the current step's responses fail to provide a constructive learning signal, specifically, when all responses in the current group fall at or below the historical peak anchor (see Trigger Condition below).

Replay Buffer. For each prompt q, we dynamically maintain a per-prompt buffer $\mathcal{B}(q)$ that stores past samples whose reward strictly exceeds the current historical peak anchor, together with the per-token probabilities of the policy at the time of sampling:

$$\mathcal{B}(q) \;=\; \Big\{ \big(o_j^k(q),\, r_j^k(q),\, \pi_{\theta_{\text{old}}^k}(o_j^k(q) \mid q)\big) \;:\; r_j^k(q) > \mu(q) \Big\}. \tag{6}$$

Storing the behavior probabilities $\pi_{\theta_{\text{old}}^k}$ is essential, as it enables principled importance-sampling correction for samples drawn from past policies. At the end of each training step t, $\mathcal{B}(q)$ is updated in two passes for every prompt q involved in current step: (1) any new sample generated in this step $(o_j^t(q), r_j^t(q), \cdot)$ with $r_j^t(q) > \mu(q)$ is inserted into $\mathcal{B}(q)$, where $\mu(q)$ is the updated anchor in this step to reflect $\mu^t(q)$; (2) any element of $\mathcal{B}(q)$ whose reward no longer exceeds the new $\mu(q)$ is removed from $\mathcal{B}(q)$. This buffer $\mathcal{B}(q)$ is maintained automatically to contain only samples that strictly outperform the current peak, so that it naturally shrinks as the policy improves and grows only when the peak is surpassed, eliminating the need for a buffer-size hyperparameter.

Trigger Condition. To preserve on-policy learning as much as possible, On-Demand Historical Replay (ODHR) is invoked only when the best response in the current group fails to surpass the

Algorithm 1 Historical Relative Policy Optimization (HRPO)

Require: Initial policy π_θ; reference policy π_{ref}; prompt set $\mathcal{D}$; group size G; total steps T; replay cap M (default $\lfloor G/2 \rfloor$); clipping ϵ; KL coefficient β; learning rate η.

Initialize: for each $q \in \mathcal{D}$, set $\mathcal{H}(q) \leftarrow \emptyset$, $\mathcal{B}(q) \leftarrow \emptyset$, $\mu(q) \leftarrow -\infty$.
for $t = 1, \ldots, T$ **do**
  Snapshot behavior policy $\theta^t_{\text{old}} \leftarrow \theta$; sample a mini-batch of prompts $\mathcal{Q}^t \subset \mathcal{D}$.
  for each prompt $q \in \mathcal{Q}^t$ **do**
    ▷ *Rollout & Compute Reward*
    Sample $\mathcal{O}^t(q) = \{o^t_j(q)\}_{j=1}^G \sim \pi_{\theta^t_{\text{old}}}(\cdot \mid q)$;
    Compute per-response rewards $\{r^t_j(q)\}_{j=1}^G$ and their mean reward $\mu^t(q) \leftarrow \frac{1}{G}\sum_j r^t_j(q)$.
    ▷ *Update History*
    $\mathcal{H}(q) \leftarrow \mathcal{H}(q) \cup \{\mu^t(q)\}$; $\mu(q) \leftarrow \max\big(\mathcal{H}(q)\big)$. ▷ update history & peak anchor
    ▷ *On-Demand Historical Replay*
    if $\max_j r^t_j(q) < \mu(q)$ **and** $\mathcal{B}(q) \neq \emptyset$ **then** ▷ $\text{Trig}^t(q) = 1$
      $K(q) \leftarrow \min\big(M, |\mathcal{B}(q)|\big)$; $\mathcal{R}(q) \leftarrow \text{TopK}\big(x \in \mathcal{B}(q), \text{key}{=}r(x), \text{num}{=}K(q)\big)$.
    else
      $\mathcal{R}(q) \leftarrow \emptyset$.
    end if
    ▷ *Historical-aware Advantage Estimation*
    for each $o \in \mathcal{O}^t(q) \cup \mathcal{R}(q)$ **do**
      $A^{\text{HRPO}}(q, o) \leftarrow r(o) - \mu(q)$.
    end for
  end for
  ▷ *Policy Update*
  $\theta \leftarrow \theta + \eta \nabla_\theta \sum_{q \in \mathcal{Q}^t} \mathcal{J}^{\text{HRPO}}(\theta)$. ▷ see Eq. (9) for $\mathcal{J}^{\text{HRPO}}(\theta)$
  ▷ *Update Buffer*
  for each prompt $q \in \mathcal{Q}^t$ **do**
    for $j = 1, \ldots, G$ **with** $r^t_j(q) > \mu(q)$ **do** ▷ insert against the updated anchor
      $\mathcal{B}(q) \leftarrow \mathcal{B}(q) \cup \big\{\big(o^t_j(q),\, r^t_j(q),\, \pi_{\theta^t_{\text{old}}}(o^t_j(q) \mid q)\big)\big\}$.
    end for
    $\mathcal{B}(q) \leftarrow \mathcal{B}(q) \setminus \big\{(o, r, \cdot) : r \leq \mu(q)\big\}$. ▷ self-prune below new anchor
  end for
end for

Ensure: Trained policy π_θ.

historical peak anchor and the buffer $\mathcal{B}(q)$ is non-empty:

$$\text{Trig}^t(q) = \mathbb{1}\big[\max_j r^t_j(q) < \mu(q)\big] \cdot \mathbb{1}\big[\mathcal{B}(q) \neq \emptyset\big]. \tag{7}$$

$\text{Trig}^t(q) = 1$ means that the current step is deemed to suffer from positive-signal starvation, and On-Demand Historical Replay (ODHR) is required for prompt q.

Recall. When triggered, we retrieve a small set of historically high-reward samples to serve as positive examples for current step t:

$$\mathcal{R}(q) \;=\; \begin{cases} \text{TopK}\Big(\mathcal{B}(q),\ \text{key} = r(x),\ \text{num} = K(q)\Big), & \text{if } \ \text{Trig}^t(q) = 1, \\ \emptyset, & \text{if } \ \text{Trig}^t(q) = 0, \end{cases}. \tag{8}$$

where $K(q) = \min\big(M,\ \big|\mathcal{B}(q)\big|\big)$ is the target number of $\text{TopK}(\cdot)$ function. The hyper-parameter cap M and $|\mathcal{B}(q)|$ are jointly applied to control the number of replayed responses, ensuring that on-policy samples remain the majority of the optimization signal. We set $M = \lfloor G/2 \rfloor$ by default. Crucially, all positive examples come from the model's own historical successes; On-Demand Historical Replay (ODHR) requires no external hints, oracle traces, or auxiliary models. Because the recalled samples share the same anchor $\mu(q)$ as the on-policy ones, the advantage estimator A^{HRPO} in Eq. (5) applies directly to both groups respectively without recomputation.

3.1.3 Optimization Objective

Following GRPO, we employ the PPO-style clipped objective with our HRPO advantage estimator, augmented with an off-policy term over the recall set:

$$\mathcal{J}^{\text{HRPO}}(\theta) = \mathbb{E}_{o\in\left\{\mathcal{O}^t(q)\cup\mathcal{R}(q)\right\}}\Big[\, A^{\text{HRPO}}(q,o)\cdot\min\Big(\rho(q,o),\,\text{clip}\big(\rho(q,o),1-\epsilon,1+\epsilon\big)\Big)\,\Big] - \beta\,\text{KL}\big(\pi_\theta\|\pi_{\text{ref}}\big), \tag{9}$$

where $\text{clip}(\cdot)$ is the standard PPO clipped surrogate. For $o \in \mathcal{O}^t(q)$, the on-policy ratio $\rho(q,o) = \frac{\pi_\theta(o|q)}{\pi_{\theta^t_{\text{old}}}(o|q)}$ corrects the distributional shift relative to the current step's behavior policy. For $o \in \mathcal{R}(q)$, the off-policy ratio $\rho(q,o) = \frac{\pi_\theta(o|q)}{\pi_{\theta^k_{\text{old}}}(o|q)}$ uses the buffer-stored probabilities to correct for the gap to the much older policy that produced the recalled sample. k is the index of historical step when the response o is generated. The KL divergence term $\beta\,\text{KL}(\pi_\theta\|\pi_{\text{ref}})$ regularizes the policy toward the reference model π_{ref} (typically the initial checkpoint before RL), with β controlling the strictness. When $\mathcal{R}(q) = \emptyset$ (*i.e.*, On-Demand Historical Replay (ODHR) is not triggered or yields no qualifying candidates), the objective gracefully reduces to the standard on-policy form.

3.2 HRPO*: A Variant with Reference-Initialized History Cold-Start

HRPO's mechanism is active when $\mathcal{H}(q)$ and $\mathcal{B}(q)$ are non-empty, which is not guaranteed during the first epoch of any training run (or throughout single-epoch training). For such scenarios, we provide a simple variant HRPO* that seeds $\mathcal{H}(q)$ and $\mathcal{B}(q)$ from one-shot reference policy rollouts before training, ensuring full historical awareness from the very first step. Details and experiments are deferred to Appendix C.

3.3 Intuitive Analysis

Self-Driven Bootstrapping Effect. Existing critic-free methods [43, 17] derive their normalization baseline exclusively from instantaneous reward statistics within the current training step, lacking any perception of the policy's dynamic evolution. When the policy momentarily regresses, this instantaneous baseline declines in tandem, permitting absolutely inferior responses to receive positive advantages simply because they surpass a weakened current policy, the relative deception effect. This myopic normalization fails to distinguish genuine improvement from transient superiority over a degrading policy, and the optimizer inadvertently reinforces worse behaviors, leading to training instability or outright collapse.

In contrast, HRPO anchors advantage estimation at the policy's strongest historical performance, $\mu(q) = \max(\mathcal{H}(q))$. As the policy improves, this historical peak naturally rises, establishing an implicit pressure: the future policy must exceed its own past peak to receive any positive reinforcement. This bootstrapping mechanism prevents convergence to a static equilibrium, explicitly disqualifies regressive reinforcement, and continuously raises the bar for what counts as "progress." In our experiments (Sec. 4), we observe that while GRPO often plateaus or exhibits performance decay after initial gains, HRPO consistently sustains upward trends, demonstrating how this bootstrapping effect encourages consistent and long-term improvement throughout training.

Closed-Loop Recovery from Local Collapse. The historical peak anchor alone has a complementary blind spot: when the current group fully regresses below the anchor, all advantages become negative, depriving the optimizer of any constructive direction, the policy knows what to avoid but not what to pursue. Selective Historical Replay closes this loop by retrieving the model's own past successes as positive teachers, but only on the prompts and at the moments where they are needed.

This conditional, prompt-specific intervention preserves on-policy learning whenever it is healthy, while injecting principled off-policy guidance precisely when local collapse occurs. Importantly, the positive examples are drawn from the model itself, requiring no external hints, oracle traces, or auxiliary models. The two mechanisms are thus tightly coupled: the anchor surfaces regression and demands progress, while replay supplies the building blocks needed to make that progress concrete.

3.4 Theoretical Analysis

We summarize four theoretical properties of HRPO that formalize the design intuitions in Sec. 3.3.
(1) Unbiased gradient. HRPO's advantage estimator preserves policy-gradient unbiasedness and

additionally eliminates the persistent σ-induced bias of GRPO (Sec. D.1). **(2) Anchor monotonicity.** The historical peak $\mu^{(t)}(q)$ is monotone non-decreasing in t, locking in past performance as a permanent floor for positive reinforcement (Sec. D.2). **(3) Deception immunity.** A policy regression of size Δ on any prompt yields an expected HRPO advantage at most $-\Delta + \mathcal{O}(R_{\max}/\sqrt{G})$ with high probability, while GRPO's expected advantage is identically 0 regardless of Δ (Sec. D.3). **(4) ODHR recovery.** Under positive-signal starvation, the on-policy gradient magnitude vanishes with the realized negative gap, while On-Demand Historical Replay (ODHR) injects a strictly non-vanishing per-sample multiplier $\rho_{\text{lo}}\Delta_{\text{pos}}^{\min} > 0$ (Sec. D.4). Proofs and statements are deferred to Appendix D.

4 Experiments

We validate HRPO through extensive experiments across diverse model architectures (base and instruct), scales (1.5B–8B), and reasoning tasks including math, code, logic, science, and SearchQA. These settings demonstrate the robustness and versatility of our framework. We provide the complete training and evaluation configurations in Appendix B. All the experimental results are the average values of 5 independent runs.

4.1 Main Results

Consistent Performance Superiority. Compared to existing RL algorithms, HRPO demonstrates a decisive performance advantage across varying model scales and domains. For mathematical reasoning, as shown in Tab. 2, HRPO achieves an average accuracy of 62.4 on Qwen3-8B-Base, significantly outperforming the standard GRPO (57.5) and REINFORCE++ (57.4). This superiority is even more pronounced on challenging benchmarks like AIME24, where HRPO surpasses GRPO by +6.3. Similarly, in code generation tasks (Tab. 3), HRPO consistently leads the leaderboard. On Qwen2.5-Coder-7B-Instruct, it achieves an average score of 74.7, with a notable margin on the OOD LiveCodeBench (+3.9 over GRPO). These consistent gains confirm that incorporating historical dynamics provides a robust optimization signal that universally enhances reasoning capabilities.

Breaking Data Saturation. We further explore whether HRPO can unlock the remaining value from data that is saturated under standard baselines. To this end, we resume the training of the DeepScaleR-1.5B-Preview checkpoint using its original training set, DeepScaleR-40K [32]. As shown in Fig. 3, simply continuing with GRPO leads to optimization stagnation, likely because the intra-group comparisons become ineffective when the model's response quality homogenizes. In stark contrast, switching to HRPO reignites the optimization trajectory, delivering marked gains across all benchmarks, most notably +10.0 on Minerva, +8.9 on AMC23, and +3.3 on AIME24. These results suggest that the perceived "data saturation" is largely an artifact of the algorithmic limitations. By pivoting to a historical baseline, HRPO effectively mitigates the relative deception effect, compelling the model to outperform its accumulated historical priors and thereby uncovering latent value in the dataset that remained inaccessible to standard baselines.

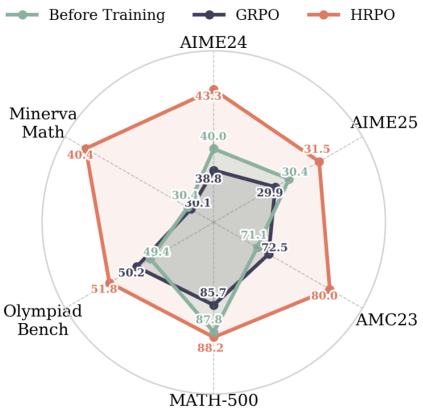

Figure 3: Performance comparison on DeepScaleR-1.5B-Preview.

Table 1: Comparison on Out-of-Domain (OOD) evaluation.

Method	In-Domain Avg.	Out-Domain GPQA	Out-Domain RiddleSense
-	38.0	35.5	66.8
GRPO	57.5	47.4	73.0
Dr.GRPO	57.0	47.6	72.4
DAPO	57.8	48.0	72.9
REINFORCE++	57.4	48.3	72.7
HRPO (Ours)	**62.4**	**51.3**	**73.5**

Out-of-Domain (OOD) Generalization. We investigate whether the reasoning improvements transfer to unseen domains. As shown in Tab. 1, HRPO achieves the highest generalization scores on GPQA (Graduate-Level Science) and RiddleSense (Commonsense Logic), significantly surpassing the baselines. This indicates that our method does not merely "reward hack" the training distribution; rather, the historical-aware optimization fosters more robust and generalized reasoning patterns that translate effectively to other domains.

Table 2: Comparison of different RL algorithms on math reasoning. All the models are trained on DAPO-17K [55] dataset. The best performance is highlighted in bold.

Model	AIME24	AIME25	AMC23	MATH-500	MinervaMath	OlympiadBench	Avg.
Qwen2.5-Math-7B-Base	2.1	2.9	11.9	19.5	7.5	6.1	8.3
+ GRPO	30.4	20.0	65.3	82.0	42.3	40.1	46.7
+ Dr.GRPO	30.8	19.5	64.9	82.5	41.8	39.6	46.5
+ DAPO	31.3	18.3	62.5	82.0	41.4	41.3	46.1
+ REINFORCE++	27.9	16.7	67.2	81.8	39.7	41.8	45.9
+ HRPO (Ours)	**34.2** (+2.9)	**23.3** (+3.3)	**74.3** (+7.1)	**83.8** (+1.3)	**45.6** (+3.3)	**44.7** (+2.9)	**50.9** (+4.2)
Qwen3-1.7B-Base	6.2	2.0	32.8	56.4	25.0	22.5	24.2
+ GRPO	17.5	12.9	47.5	72.8	32.7	32.3	35.9
+ Dr.GRPO	17.9	13.5	48.2	73.0	32.2	33.1	36.3
+ DAPO	18.3	15.4	47.5	73.4	31.3	30.8	36.1
+ REINFORCE++	16.7	12.1	50.0	72.6	33.5	32.0	36.1
+ HRPO (Ours)	**23.3** (+5.0)	**18.3** (+2.9)	**55.0** (+5.0)	**75.2** (+1.8)	**36.4** (+2.9)	**35.3** (+2.2)	**40.5** (+4.2)
Qwen3-8B-Base	12.9	9.6	52.8	73.2	39.8	39.7	38.0
+ GRPO	40.0	31.3	81.6	90.6	51.5	50.1	57.5
+ Dr.GRPO	40.5	30.9	82.5	91.2	51.0	50.8	57.8
+ DAPO	40.7	32.4	82.2	91.0	50.4	50.4	57.8
+ REINFORCE++	39.2	30.8	81.9	90.6	52.2	49.6	57.4
+ HRPO (Ours)	**46.3** (+5.6)	**37.1** (+4.7)	**90.0** (+7.5)	**92.8** (+1.6)	**54.0** (+1.8)	**54.1** (+3.3)	**62.4** (+4.6)
DeepSeek-R1-Distill-Qwen-1.5B	28.9	22.8	62.9	83.9	26.5	43.3	44.7
+ GRPO	30.2	23.5	67.0	84.1	27.3	45.1	46.2
+ Dr.GRPO	30.5	23.0	68.2	84.5	28.1	45.3	46.6
+ DAPO	31.6	24.0	72.5	85.3	30.0	46.0	48.2
+ REINFORCE++	32.1	25.0	70.0	85.5	34.3	46.3	48.9
+ HRPO (Ours)	**35.8** (+3.7)	**27.9** (+2.9)	**76.6** (+4.1)	**86.4** (+0.9)	**41.4** (+7.1)	**49.6** (+3.3)	**53.0** (+4.1)
OpenMath-Nemotron-1.5B	61.6	49.5	90.0	92.4	27.3	66.8	64.6
+ GRPO	63.3	52.9	89.1	92.3	26.4	67.2	65.2
+ Dr.GRPO	63.5	53.2	89.8	92.6	26.8	67.1	65.5
+ DAPO	62.5	52.5	90.1	92.9	27.5	67.4	65.5
+ REINFORCE++	63.0	51.6	89.4	91.6	26.5	67.2	64.9
+ HRPO (Ours)	**67.1** (+3.6)	**56.7** (+3.5)	**92.5** (+2.4)	**93.2** (+0.3)	**29.0** (+1.5)	**68.0** (+0.6)	**67.8** (+2.3)
Qwen3-1.7B	46.7	35.4	85.3	91.9	49.2	48.8	59.6
+ GRPO	53.2	42.0	88.0	92.4	50.8	52.2	63.1
+ Dr.GRPO	53.8	42.5	88.5	92.7	51.2	52.9	63.6
+ DAPO	54.0	42.8	87.6	92.6	50.5	53.5	63.5
+ REINFORCE++	52.9	41.5	88.2	92.1	51.5	51.8	63.0
+ HRPO (Ours)	**57.8** (+3.8)	**46.4** (+3.6)	**91.0** (+2.5)	**93.2** (+0.5)	**52.4** (+0.9)	**57.5** (+4.0)	**66.4** (+2.8)

4.2 Ablation Study

Effectiveness of HAE. Using only the historical-aware advantage (Variant 1, row 4) already lifts the average accuracy from 57.8 (Dr.GRPO, row 2) to 60.5 (+2.7), at a cost of only 17.8 s (7.6%) per step. Anchoring the advantage to the running historical peak, rather than the group mean, filters out the spurious positive signals caused by the relative deception effect.

Effectiveness of ODHR. As shown in Tab. 4 (rows 3 vs. 4), removing the On-Demand Historical Replay (ODHR) leads to a significant performance drop of 1.9. This highlights that when the current policy encounters "positive-signal starvation," recalling high-quality historical trajectories is essential for providing constructive reinforcement and guiding the model out of performance plateaus. Since replay is activated only on demand, ODHR adds merely 9.2 s (3.7%) per step over Variant 1.

Impact of Advantage Normalization. Comparing row 5 with row 3 in Tab. 4, we observe that omitting the standard deviation in advantage normalization yields a 0.4 improvement (62.0 $\rightarrow$ 62.4; likewise for GRPO vs. Dr.GRPO, rows 1–2). This aligns with our theoretical analysis that within-group variance normalization in GRPO can introduce batch-wise bias; using historical-aware reward offsets instead provides a more stable and unbiased optimization signal for policy bootstrapping.

Sensitivity of Replay Capacity M. The bottom group of Tab. 4 illustrates how the maximum replay capacity M balances corrective strength with distribution consistency. We find that $M = \lfloor G/2 \rfloor$ achieves the optimal accuracy. While a small M ($\lfloor G/4 \rfloor$) provides insufficient positive guidance, an excessively large M (G) introduces older off-policy samples that deviate significantly from the current policy, leading to a slight performance degradation. Notably, every tested M still beats GRPO and Dr.GRPO by at least 3.4 points, whereas $M = G$ is 5.3% slower yet 0.5 points worse than $M = \lfloor G/2 \rfloor$.

Table 3: Comparison of different RL algorithms on code generation benchmarks. All the models are trained on LeetCodeDataset [50]. The best performance is highlighted in bold.

Model	LiveCodeBench	HumanEval	HumanEval+	MBPP	MBPP+	Avg
Qwen2.5-7B-Instruct	22.5	86.4	80.5	75.6	66.7	66.3
+ GRPO	26.9	87.5	84.0	80.0	68.1	69.3
+ Dr.GRPO	26.5	87.8	83.2	79.8	67.9	69.0
+ DAPO	25.0	86.8	84.1	80.5	69.0	69.1
+ REINFORCE++	26.4	87.3	83.6	80.5	68.7	69.3
+ HRPO (Ours)	**31.2** (+4.3)	**88.8** (+1.0)	**85.2** (+1.1)	**83.5** (+3.0)	**73.2** (+4.2)	**72.4** (+3.1)
Qwen2.5-Coder-7B-Instruct	30.4	86.0	83.2	82.5	69.7	70.4
+ GRPO	32.5	86.5	83.4	85.1	70.3	71.6
+ Dr.GRPO	32.8	86.8	83.6	84.8	71.5	71.9
+ DAPO	32.3	86.5	83.7	85.3	72.8	72.1
+ REINFORCE++	31.8	86.2	83.9	85.0	73.1	72.0
+ HRPO (Ours)	**36.4** (+3.6)	**88.6** (+1.8)	**85.6** (+1.7)	**87.0** (+1.7)	**76.6** (+3.5)	**74.8** (+2.7)
Qwen3-1.7B	33.4	73.8	68.9	70.2	64.5	62.2
+ GRPO	35.3	85.3	79.8	83.1	71.1	70.9
+ Dr.GRPO	35.5	86.5	80.5	83.4	72.4	71.7
+ DAPO	36.0	86.2	81.0	84.0	72.8	72.0
+ REINFORCE++	35.8	86.0	79.5	83.5	71.6	71.3
+ HRPO (Ours)	**37.7** (+1.7)	**89.4** (+2.9)	**85.5** (+4.5)	**85.3** (+1.3)	**74.9** (+2.1)	**74.6** (+2.6)

Table 4: Ablation study of HRPO on Qwen3-8B-Base model and math reasoning. The first group evaluates the contribution of individual components, while the second group investigates the sensitivity of the replay capacity M. Speed means the time cost for each training step.

	Category	Configuration	HAE	ODHR	Div. Std	M	Avg (%)	Speed (s)
1	Component Ablation	GRPO	–	–	✓	–	57.5	233.4
2		Dr.GRPO	–	–	✗	–	57.8	233.4
3		**HRPO**	✓	✓	✗	$\lfloor G/2 \rfloor$	**62.4**	260.4
4		Variant 1	✓	✗	✗	–	60.5	251.2
5		Variant 2	✓	✓	✓	$\lfloor G/2 \rfloor$	62.0	260.6
6	Sensitivity of M	Small Capacity	✓	✓	✗	$\lfloor G/4 \rfloor$	61.2	252.8
7		Medium Capacity	✓	✓	✗	$\lfloor G/2 \rfloor$	**62.4**	260.4
8		Large Capacity	✓	✓	✗	G	61.9	274.1

5 Conclusion

In this work, we identify and analyze the relative deception effect, a fundamental instability in critic-free RFT algorithms like GRPO. However, by relying solely on instantaneous intra-group normalization, these methods inadvertently reinforce suboptimal behaviors during sampling dips, leading to policy collapse. To address this, we propose HRPO, a framework combining a historical-aware advantage estimator with on-demand historical replay. The former anchors advantages to running peak performance to filter spurious signals, while the latter dynamically injects high-reward historical trajectories during positive-signal starvation. Extensive experiments demonstrate that HRPO consistently mitigates training collapse and achieves state-of-the-art performance across reasoning benchmarks. Ablations further show that the two components are complementary, together bringing +4.6 average accuracy over Dr.GRPO with only 11.6% extra training time over GRPO, and the gains hold across model scales and domains.

Acknowledgements

This work was supported in part by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R. The work has been supported by Hong Kong Research Grant Council-General Research Fund Scheme (Grant No. 17202422, 17212923, 17215025), Theme-based Research (Grant No. T45-701/22-R), and Strategic Topics Grant (Grant No. STG3/E-605/25-N). Part of the described research work is conducted in the JC STEM Lab of Robotics for Soft Materials, funded by The Hong Kong Jockey Club Charities Trust.

References

[1] AIME: American invitational mathematics examination (2024), https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions

[2] AIME: American invitational mathematics examination (2025), https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions

[3] AMC: American mathematics competitions (2023), https://artofproblemsolving.com/wiki/index.php/2023_AMC_12A

[4] Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al.: Program synthesis with large language models. arXiv preprint arXiv:2108.07732 (2021)

[5] Cao, M., Zhang, S., Chang, X.W., Precup, D.: Scar: Shapley credit assignment for more efficient rlhf. arXiv preprint arXiv:2505.20417 (2025)

[6] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P.D.O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al.: Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374 (2021)

[7] Chu, X., Huang, H., Zhang, X., Wei, F., Wang, Y.: Gpg: A simple and strong reinforcement learning baseline for model reasoning (2025), https://arxiv.org/abs/2504.02546

[8] Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)

[9] Cui, G., Yuan, L., Wang, Z., Wang, H., Zhang, Y., Chen, J., Li, W., He, B., Fan, Y., Yu, T., et al.: Process reinforcement through implicit rewards. arXiv preprint arXiv:2502.01456 (2025)

[10] DeepMind, G.: Gemini 2.5 flash (2025), https://deepmind.google/technologies/gemini/flash/

[11] Fei, W., Kong, H., Liang, S., Lin, Y., Yang, Y., Tang, J., Chen, L., Hua, X.: Self-guided process reward optimization with redefined step-wise advantage for process reinforcement learning. arXiv e-prints pp. arXiv–2507 (2025)

[12] Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)

[13] He, B., Qu, Z., Liu, Z., Chen, Y., Zuo, Y., Qian, C., Zhang, K., Chen, W., Xiao, C., Cui, G., et al.: Justrl: Scaling a 1.5 b llm with a simple rl recipe. arXiv preprint arXiv:2512.16649 (2025)

[14] He, C., Luo, R., Bai, Y., Hu, S., Thai, Z.L., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., et al.: Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. arXiv preprint arXiv:2402.14008 (2024)

[15] He, H., Lab, T.M.: Defeating nondeterminism in llm inference. Thinking Machines Lab: Connectionism (2025), https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/

[16] Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., Steinhardt, J.: Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874 (2021)

[17] Hu, J.: Reinforce++: A simple and efficient approach for aligning large language models. arXiv preprint arXiv:2501.03262 (2025)

[18] Hui, B., Yang, J., Cui, Z., Yang, J., Liu, D., Zhang, L., Liu, T., Zhang, J., Yu, B., Lu, K., et al.: Qwen2. 5-coder technical report. arXiv preprint arXiv:2409.12186 (2024)

[19] Jain, N., Han, K., Gu, A., Li, W.D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., Stoica, I.: Livecodebench: Holistic and contamination free evaluation of large language models for code. arXiv preprint arXiv:2403.07974 (2024)

[20] Ji, Y., Zhao, S., Tian, X., Wang, H., Chen, S., Peng, Y., Zhao, H., Li, X.: How difficulty-aware staged reinforcement learning enhances llms' reasoning capabilities: A preliminary experimental study. arXiv preprint arXiv:2504.00829 (2025)

[21] Lee, A., Tong, H.: Token-efficient rl for llm reasoning. arXiv preprint arXiv:2504.20834 (2025)

[22] Lee, J.H., Yang, J.Y., Heo, B., Han, D., Kim, K., Yang, E., Yoo, K.M.: Token-supervised value models for enhancing mathematical problem-solving capabilities of large language models. arXiv preprint arXiv:2407.12863 (2024)

[23] Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al.: Solving quantitative reasoning problems with language models. Advances in neural information processing systems **35**, 3843–3857 (2022)

[24] Li, J., Sun, S., Yuan, W., Fan, R.Z., Zhao, H., Liu, P.: Generative judge for evaluating alignment (2023), https://arxiv.org/abs/2310.05470

[25] Li, X., Zou, H., Liu, P.: Limr: Less is more for rl scaling. arXiv preprint arXiv:2502.11886 (2025)

[26] Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.G., Chen, W.: Making large language models better reasoners with step-aware verifier (2023), https://arxiv.org/abs/2206.02336

[27] Lin, B.Y., Wu, Z., Yang, Y., Lee, D.H., Ren, X.: Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP 2021): Findings (2021), to appear

[28] Liu, J., Xia, C.S., Wang, Y., Zhang, L.: Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. Advances in Neural Information Processing Systems **36**, 21558–21572 (2023)

[29] Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W.S., Lin, M.: Understanding r1-zero-like training: A critical perspective. arXiv preprint arXiv:2503.20783 (2025)

[30] Lu, J., Dou, Z., Wang, H., Cao, Z., Dai, J., Wan, Y., Guo, Z.: Autopsv: Automated process-supervised verifier (2024), https://arxiv.org/abs/2405.16802

[31] Luo, L., Liu, Y., Liu, R., Phatale, S., Guo, M., Lara, H., Li, Y., Shu, L., Zhu, Y., Meng, L., Sun, J., Rastogi, A.: Improve mathematical reasoning in language models by automated process supervision (2024), https://arxiv.org/abs/2406.06592

[32] Luo, M., Tan, S., Wong, J., Shi, X., Tang, W.Y., Roongta, M., Cai, C., Luo, J., Li, L.E., Popa, R.A., Stoica, I.: Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2 (2025), notion Blog

[33] Luo, M., Tan, S., Wong, J., Shi, X., Tang, W.Y., Roongta, M., Cai, C., Luo, J., Li, L.E., Popa, R.A., Stoica, I.: Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2 (2025), notion Blog

[34] Lyu, C., Gao, S., Gu, Y., Zhang, W., Gao, J., Liu, K., Wang, Z., Li, S., Zhao, Q., Huang, H., et al.: Exploring the limit of outcome reward for learning mathematical reasoning. arXiv preprint arXiv:2502.06781 (2025)

[35] Ma, Q., Zhou, H., Liu, T., Yuan, J., Liu, P., You, Y., Yang, H.: Let's reward step by step: Step-level reward model as the navigators for reasoning (2023), https://arxiv.org/abs/2310.10080

[36] McAleese, N., Pokorny, R.M., Uribe, J.F.C., Nitishinskaya, E., Trebacz, M., Leike, J.: Llm critics help catch llm bugs (2024), https://arxiv.org/abs/2407.00215

[37] Moshkov, I., Hanley, D., Sorokin, I., Toshniwal, S., Henkel, C., Schifferer, B., Du, W., Gitman, I.: Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset (2025), https://arxiv.org/abs/2504.16891

[38] OpenAI: Learning to reason with llms (2024), https://openai.com/index/learning-to-reason-with-llms/

[39] Pan, S., Lialin, V., Muckatira, S., Rumshisky, A.: Let's reinforce step by step (2023), https://arxiv.org/abs/2311.05821

[40] Qi, P., Liu, Z., Zhou, X., Pang, T., Du, C., Lee, W.S., Lin, M.: Defeating the training-inference mismatch via fp16. arXiv preprint arXiv:2510.26788 (2025)

[41] Rein, D., Hou, B.L., Stickland, A.C., Petty, J., Pang, R.Y., Dirani, J., Michael, J., Bowman, S.R.: Gpqa: A graduate-level google-proof q&a benchmark. In: First Conference on Language Modeling (2024)

[42] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)

[43] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)

[44] Tan, H., Wu, X., Wu, S., Zhang, S., Chen, Y., Sun, X., Shen, J., QI, X.: Understanding data influence in reinforcement finetuning. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025), https://openreview.net/forum?id=mjhCFB3HLQ

[45] Team, O.R.: Openr1-math-220k (2025), https://huggingface.co/datasets/open-r1/OpenR1-Math-220k

[46] Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., Higgins, I.: Solving math word problems with process- and outcome-based feedback (2022), https://arxiv.org/abs/2211.14275

[47] Wang, P., Li, L., Shao, Z., Xu, R.X., Dai, D., Li, Y., Chen, D., Wu, Y., Sui, Z.: Math-shepherd: Verify and reinforce llms step-by-step without human annotations (2024), https://arxiv.org/abs/2312.08935

[48] Wang, T., Kulikov, I., Golovneva, O., Yu, P., Yuan, W., Dwivedi-Yu, J., Pang, R.Y., Fazel-Zarandi, M., Weston, J., Li, X.: Self-taught evaluators (2024), https://arxiv.org/abs/2408.02666

[49] Wang, Y., Yang, Q., Zeng, Z., Ren, L., Liu, L., Peng, B., Cheng, H., He, X., Wang, K., Gao, J., Chen, W., Wang, S., Du, S.S., Shen, Y.: Reinforcement learning for reasoning in large language models with one training example. arXiv preprint arXiv:2504.20571 (2025)

[50] Xia, Y., Shen, W., Wang, Y., Liu, J.K., Sun, H., Wu, S., Hu, J., Xu, X.: Leetcodedataset: A temporal dataset for robust evaluation and efficient training of code llms. arXiv preprint arXiv:2504.14655 (2025)

[51] Xie, G., Shi, Y., Tian, H., Yao, T., Zhang, X.: Capo: Towards enhancing llm reasoning through verifiable generative credit assignment. arXiv e-prints pp. arXiv–2508 (2025)

[52] Yan, J., Li, Y., Hu, Z., Wang, Z., Cui, G., Qu, X., Cheng, Y., Zhang, Y.: Learning to reason under off-policy guidance. arXiv preprint arXiv:2504.14945 (2025)

[53] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)

[54] Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al.: Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115 (2024)

[55] Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al.: Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476 (2025)

[56] Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Fan, T., Liu, G., Liu, L., Liu, X., et al.: Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476 (2025)

[57] Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., He, J.: Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. arXiv preprint arXiv:2503.18892 (2025)

[58] Zhang, X., Wu, S., Zhu, Y., Tan, H., Yu, S., He, Z., Jia, J.: Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. arXiv preprint arXiv:2510.19807 (2025)

[59] Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging llm-as-a-judge with mt-bench and chatbot arena (2023), https://arxiv.org/abs/2306.05685

Broader impacts

This paper introduces a novel reinforcement fine-tuning algorithm designed for the post-training phase of large models. By significantly improving training efficiency and reducing both computational overhead and iterative trial-and-error costs, our approach lowers energy consumption and accelerates the development of more capable models that can positively benefit society. Nevertheless, like any advanced AI technology, this method carries dual-use potential and could theoretically be misapplied to develop systems for unethical or inhumane surveillance. In light of this, we strongly urge international regulatory and legislative bodies to establish collaborative frameworks that ensure the responsible development and deployment of such technologies, safeguarding against harmful applications while maximizing their societal benefits.

LLM Usage Statement

During the preparation of this manuscript, large language models were used exclusively for editorial purposes, including typo correction, grammar checking, and stylistic polishing. All research conceptualization, data analysis, results interpretation, and scientific writing were performed independently by the authors. The authors assume full responsibility for the accuracy, integrity, and originality of this work.

Limitations

While our proposed reinforcement fine-tuning algorithm demonstrates strong stability and improved training efficiency, this study is subject to computational constraints. Due to limited hardware resources, we were unable to evaluate the method in real-world industrial settings or on ultra-large language models with hundreds of billions of parameters. We acknowledge this as an important avenue for future research and plan to conduct large-scale empirical studies and domain-specific deployments as distributed training infrastructure and computational resources become more accessible.

Ethics Statement

This research adheres strictly to established guidelines for responsible AI development. Our work focuses on improving the computational efficiency and convergence stability of reinforcement learning for large reasoning models, without introducing new data collection pipelines. All experiments utilize exclusively publicly available benchmarks and datasets; we do not collect, process, or store any personally identifiable information (PII) or sensitive user data. We acknowledge the dual-use nature of accelerated reasoning capabilities and emphasize the importance of deploying such optimizations alongside robust safety and alignment frameworks.

Reproducibility Statement

We provide complete implementation details to ensure independent verification, including model configurations, training pipelines, optimization hyperparameters, and computational environments, all documented in the main text and Supplementary Material. To facilitate replication and future research, we will publicly release the full source code, training scripts, and configuration files upon paper acceptance. All reported results can be reproduced using the provided artifacts and standard academic computing resources.

A Related Work

Reinforcement Fine-Tuning (RFT) stands as a cornerstone in the recent evolution of LLM performance [38, 12, 54, 10]. By treating model optimization as a reinforcement learning problem, this paradigm employs reward-based guidance to calibrate the model's generative behavior. Notably, the DeepSeek-R1 model [12] demonstrates the potency of this approach by integrating GRPO [43] with deterministic rewards. To better understand the landscape of RFT advancements, we propose a taxonomy consisting of three distinct thematic categories.

The first group of works aims to design advanced reward signals. Early RFT research, exemplified by DeepSeek-R1 [12], focuses on binary outcome rewards to verify correctness in structured domains like math and coding [38, 54, 18, 53, 10, 45], while recent studies [48, 24, 36, 59] have expanded this to unstructured tasks (*e.g.*, general writing) by employing LLMs to perform automated outcome evaluation. However, outcome rewards suffer from a severe limitation: reward sparsity. Subsequently, an increasing body of research has shifted the granularity of supervision signals toward the process-level [46, 26, 35, 39, 47, 31, 30] and even token-levels [9, 34, 21, 11, 5, 51, 22].

Second, several recent works have introduced data-centric paradigms [25, 33, 56, 49] to further improve RFT. By leveraging indicators such as the Historical Variance Score [49] or difficulty levels [57, 20], these approaches prioritize high-quality data to maximize the efficiency of reinforcement learning. Additionally, recent work [44] leverages influence functions to assess the impact of training samples on the RFT process, utilizing these insights to guide the selection of the most effective data.

The third category comprises algorithmic improvements, which aim to further streamline or enhance the pipelines of the currently prevailing GRPO-based methods. For example, Dr.GRPO [29] analyzes DeepSeek-R1-Zero's [12] training pipeline, uncovering biases in GRPO and proposing solutions to enhance token efficiency and final performance. To provide effective learning signals for extremely challenging problems, [52, 58] introduce auxiliary hints to guide the model toward sampling correct trajectories. This approach enables the model to successfully learn from challenging data samples for which it originally struggled to generate correct responses. Other works aim to simplify GRPO [56, 7, 13], *e.g.*, by removing the KL-divergence term to achieve more robust empirical results [7]. Recently, several studies have focused on the inherent stability of the RFT process. These works address fundamental algorithmic implementations to resolve issues such as reproducibility across multiple runs [15] and numerical instability encountered during low-precision FP16 training [40].

B Experiment Settings

Models. We evaluate HRPO on a comprehensive suite of models varying in size and distinct training stages:

- Base Models: To assess performance from scratch (post-pretraining), we use Qwen2.5-Math-7B-Base [54] and Qwen3-Base (1.7B and 8B) [53].
- SFT Models: We employ DeepSeek-R1-Distill-Qwen-1.5B [12] and OpenMath-Nemotron-1.5B [37] as representative instruction-tuned models.
- RL-Post-trained Models: To test if HRPO can further boost converged models, we include the Qwen3 Series (1.7B and 4B) [53] and DeepScaleR-1.5B-Preview [32], which have previously undergone both SFT and RL processes.

Training Dataset. We train the models on the DAPO-17K [55] dataset for math reasoning experiments, and the LeetCodeDataset [50] for code generation experiments.

Training Settings. We optimize all models using the AdamW optimizer with a constant learning rate of 1e-6 and a weight decay of 0.01. During the data collection phase, we sample $G = 8$ trajectories for each prompt with a temperature of $\tau = 1.0$. The training employs a global batch size of 256 for Base LLMs, while for the LongCoT models (*i.e.* DeepSeek-R1-Distill, OpenMath-Nemotron, and DeepScaleR), the batch size is reduced to 128. The PPO mini-batch size is fixed at 64 across all experiments. Regarding regularization, we set the KL divergence coefficient $\beta = 0$ for base models, whereas for other models, we apply a coefficient of β=1e-4. All models are trained for a total of 500 steps. All experiments are conducted on 8 $\times$ A100 GPUs.

Evaluation Benchmarks. Our evaluation protocol encompasses both in-domain tasks and out-of-distribution (OOD) generalization tests. For mathematical reasoning, we employ six challenging benchmarks: AIME24 [1], AIME25 [2], AMC23 [3], MATH-500 [16], MinervaMath [23], and

OlympiadBench [14]. For coding tasks, we evaluate on five code generation benchmarks, including HumanEval [6], HumanEval+ [28], MBPP [4], MBPP+ [28], and LiveCodeBench [19]. For OOD evaluation, we test on GPQA [41] (graduate-level STEM questions) and RiddleSense [27] (logical puzzles).

Evaluation Setting. During inference, we generate 16 trajectories per question and report the average Pass@1 accuracy. We use a temperature of 0.6 and top-p=0.95. Regarding top-k, we adhere to the official setting of top-k=20 for the Qwen3 models, while disabling it (top-k=-1) for all other models.

C HRPO*: HRPO with Reference-Initialized History Cold-Start

C.1 Motivation: The History Cold-Start Problem

The defining benefit of HRPO is its ability to leverage accumulated historical information: as a prompt q is revisited, $\mathcal{H}(q)$ grows, the historical peak $\mu(q)$ tightens, and $\mathcal{B}(q)$ collects above-anchor responses usable by ODHR. However, this benefit is unavailable whenever a prompt has not yet been visited more than once. Two concrete scenarios trigger this *history cold-start problem*:

- *Single-epoch training.* When the dataset is traversed only once, each prompt is sampled at most at a single step t, so $\mathcal{T}(q) = \{t\}$ throughout training.
- *The first epoch of multi-epoch training.* Even in standard multi-epoch runs, every prompt experiences exactly one visit during the first pass over the dataset, before any cross-epoch revisit accumulates meaningful history.

In both cases, at the moment prompt q is sampled for the first time, $\mathcal{T}(q) = \{t\}$, and both core mechanisms of HRPO lose their historical grounding:

- the historical peak anchor reduces to the current group's mean, $\mu(q) = \mu^t(q)$, so the advantage signal carries no information beyond the current step's intra-group ordering;
- the per-prompt buffer $\mathcal{B}(q) = \emptyset$, so the trigger $\mathrm{Trig}^t(q) \cdot \mathbb{1}\big[\mathcal{B}(q) \neq \emptyset\big] = 0$ and ODHR never activates.

We address this with a small, one-shot pre-processing step that injects a virtual "step 0" from the reference policy into $\mathcal{H}(q)$ and $\mathcal{B}(q)$ for every prompt before training starts, providing each prompt with a non-trivial historical baseline at the first epoch.

C.2 Algorithm

The reference policy π_{ref} (typically the checkpoint before RL that also serves as the KL anchor in Eq. (9)) is used to seed the per-prompt history and buffer before training starts. Concretely, for every $q \in \mathcal{D}$ we sample G_0 reference responses, compute their rewards, and use them to initialize:

$$\mathcal{H}(q) \leftarrow \{\mu^0(q)\}, \quad \mu(q) \leftarrow \mu^0(q), \quad \mathcal{B}(q) \leftarrow \big\{\big(o_j^0,\, r_j^0,\, \pi_{\text{ref}}(o_j^0 \mid q)\big) : r_j^0 > \mu^0(q)\big\}, \tag{10}$$

where $\mu^0(q) = \frac{1}{G_0}\sum_{j=1}^{G_0} r_j^0(q)$ is the reference baseline. Training then proceeds exactly as in vanilla HRPO (Algorithm 1), only with the empty initialization on its line 1 replaced by Eq. (10). Algorithm 2 summarizes the full procedure.

C.3 Properties

Augmented anchor. Throughout training, the running anchor admits the closed form

$$\mu(q) \;=\; \max\big(\mu^0(q),\; \mu_{\text{old}}(q),\; \mu^t(q)\big), \tag{11}$$

where $\mu_{\text{old}}(q)$ is the pre-step-t peak. HRPO* thus assigns positive advantage only to responses that surpass the strongest of (i) the reference policy's average reward on q, (ii) the policy's own historical peak, and (iii) the current group mean.

Reduction relations. HRPO* smoothly subsumes vanilla HRPO across training regimes:

- *(a) HRPO* with $G_0 = 0$.* No reference rollout is performed and $\mu^0(q) = -\infty$ by convention; HRPO* collapses to vanilla HRPO.
- *(b) Vanilla HRPO under multi-epoch training.* Equivalent to HRPO* with $\mu^0(q) = -\infty$, since prompt revisits supply the historical signal that $\mu^0(q)$ provides in HRPO*.

Algorithm 2 HRPO*: Reference-Initialized HRPO

Require: Reference policy π_{ref}; reference rollout size G_0 (default $G_0 = G$); remaining inputs same as Algorithm 1.

▷ *Reference initialization (one-shot, before training)*

for each prompt $q \in \mathcal{D}$ **do**

  Sample $\{o_j^0(q)\}_{j=1}^{G_0} \sim \pi_{\text{ref}}(\cdot \mid q)$ and compute rewards $\{r_j^0(q)\}_{j=1}^{G_0}$.

  $\mu^0(q) \leftarrow \frac{1}{G_0}\sum_{j=1}^{G_0} r_j^0(q)$; $\mathcal{H}(q) \leftarrow \{\mu^0(q)\}$; $\mu(q) \leftarrow \mu^0(q)$.

  $\mathcal{B}(q) \leftarrow \big\{\big(o_j^0(q),\, r_j^0(q),\, \pi_{\text{ref}}(o_j^0(q) \mid q)\big) : r_j^0(q) > \mu^0(q)\big\}$.

end for

▷ *Training phase*

Run Algorithm 1, replacing the empty initialization on its line 1 with the seeded $\{\mathcal{H}(q),\, \mu(q),\, \mathcal{B}(q)\}$ above.

Ensure: Trained policy π_θ.

- *(c) Strict single-epoch streaming.* Vanilla HRPO loses access to historical information (Section C.1), whereas HRPO* retains a non-trivial anchor and a non-empty buffer for every prompt at the moment of its first (and only) visit.

Theoretical extensions. All theoretical properties stated for HRPO in Sec. 3.4 carry over to HRPO* verbatim by treating $\mu^0(q)$ as a virtual "step 0" entry of $\mathcal{H}(q)$. In particular, the anchor remains monotone non-decreasing, and the deception-immunity bound holds against the elementwise max of the reference and self-peak baselines.

Computational overhead. The pre-processing cost is $|\mathcal{D}| \cdot G_0$ inference samples under π_{ref}, performed once before training. Relative to total training rollouts (which contribute $|\mathcal{Q}^t| \cdot G \cdot T$ samples in expectation), this overhead is $\frac{|\mathcal{D}| \cdot G_0}{|\mathcal{Q}^t| \cdot G \cdot T}$. For typical configurations (e.g., $|\mathcal{D}| = 12\text{K}$, $|\mathcal{Q}^t| = 64$, $G_0 = G = 8$, $T = 500$), the overhead is below 1%, and the pre-processing can be parallelized with model loading.

C.4 Experiments

In Tab. 5, we evaluate the impact of reference-initialized seeding (HRPO*) across different training horizons. In the single-epoch setting (Total Epoch = 1), standard HRPO yields performance identical to GRPO (61.4) because the historical anchors and buffers are initially empty and only begin to accumulate data after the first pass. In contrast, HRPO* achieves a significant improvement (63.6), demonstrating that seeding history from reference policy rollouts effectively mitigates the "cold-start" problem and provides immediate optimization guidance. As training progresses to multiple epochs ($\sim$3.8), the performance of standard HRPO improves significantly as it self-accumulates sufficient historical statistics, narrowing the gap with HRPO* (66.4 vs. 66.7). Nonetheless, HRPO* remains the top performer, underscoring that a high-quality initial historical anchor benefits both early-stage convergence and long-term training stability.

Table 5: Effect of HRPO* with different training epochs.

Model	Total Epoch	Avg.
Qwen3-1.7B	-	59.6
+ GRPO	1	61.4
+ HRPO	1	61.4
+ HRPO*	1	**63.6**
+ GRPO	$\sim$3.8	63.1
+ HRPO	$\sim$3.8	66.4
+ HRPO*	$\sim$3.8	**66.7**

C.5 Discussion

HRPO* remains entirely self-contained: it requires no external demonstrations, oracle traces, or auxiliary teacher models. The reference policy π_{ref} is the same SFT-initialized checkpoint already used as the KL anchor in Eq. (9), and the only addition is one inference pass over the full dataset before training starts, which is cheaper than a single training step's rollout in typical configurations. We recommend HRPO* as the default variant when the training budget is tight (single-epoch or a

small number of epochs), as it ensures HRPO's historical-aware mechanisms are active from the very first step for every prompt. For long multi-epoch runs where prompts are revisited many times, vanilla HRPO naturally accumulates sufficient history and HRPO* provides only a marginal warm-start benefit.

D Theoretical Analysis of HRPO

This appendix collects the full statements and proofs that formalize the design intuitions of HRPO presented in Sec. 3.3. We establish four properties of HRPO's advantage estimator and its induced optimization dynamics:

- Sec. D.1 shows that the HRPO advantage estimator yields an unbiased policy gradient, in contrast to GRPO whose standard-deviation normalization breaks unbiasedness.
- Sec. D.2 establishes the monotonicity of the historical peak anchor and the resulting *ratchet effect*.
- Sec. D.3 formalizes HRPO's immunity to the relative deception effect via a high-probability lower bound on the negative advantage during policy regression.
- Sec. D.4 characterizes the non-vanishing gradient signal recovered by ODHR under positive-signal starvation, where vanilla on-policy updates lose all directional information.

Throughout this appendix we adopt the notation of Sec. 3.1: π_θ is the current policy, $\pi_{\theta^t_{\text{old}}}$ the behavior policy at step t, $\mathcal{O}^t(q) = \{o^t_j(q)\}_{j=1}^G$ the on-policy group, $r^t_j(q)$ the per-response reward, $\mu^t(q) = \frac{1}{G}\sum_j r^t_j(q)$ the current step's group mean, $\mathcal{H}(q) = \{\mu^k(q)\}_{k\in\mathcal{T}(q)}$ the per-prompt history, and $\mu(q) = \max(\mathcal{H}(q))$ the historical peak anchor.

D.1 Unbiased Policy Gradient Estimator

We show that HRPO's advantage estimator (Eq. (5)) preserves the unbiasedness of the underlying policy gradient, modulo the same $\mathcal{O}(1/G)$ leave-one-out approximation that the entire GRPO family relies on. In contrast, GRPO's standard-deviation normalization (Eq. (2)) introduces an additional, distinct source of bias that HRPO sidesteps by design.

Setup. Fix a prompt q and the parameters of the current policy π_θ. Let $J(\theta; q) := \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}[r(o)]$ be the expected reward of π_θ on q. The vanilla policy gradient is given by the score-function identity

$$\nabla_\theta J(\theta; q) \;=\; \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\big[\, r(o)\,\nabla_\theta \log \pi_\theta(o \mid q)\,\big]. \tag{12}$$

For the proof we work with the canonical on-policy form ($\pi_{\theta^t_{\text{old}}} = \pi_\theta$, no clipping), which isolates the effect of the advantage normalization; importance sampling and clipping introduce only standard PPO-style off-policy corrections that are orthogonal to the present analysis.

Auxiliary lemma. A baseline $b(q)$ that is measurable with respect to q alone (i.e., does not depend on the random sample o) does not affect the policy gradient in expectation. This is the well-known control-variate property:

$$\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\big[\, b(q)\,\nabla_\theta \log \pi_\theta(o \mid q)\,\big] \;=\; b(q)\,\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\big[\nabla_\theta \log \pi_\theta(o \mid q)\big] \;=\; 0, \tag{13}$$

where the last equality uses $\mathbb{E}_{o\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(o \mid q)] = \nabla_\theta \int \pi_\theta(o \mid q)\,\mathrm{d}o = \nabla_\theta 1 = 0$.

Theorem 1 (Unbiased policy gradient under a strict historical baseline). *At training step t, define the strict historical peak*

$$\mu_{\text{hist}}(q) \;:=\; \max_{k\in\mathcal{T}(q)\setminus\{t\}} \mu^k(q), \tag{14}$$

i.e., the running maximum over per-step group means observed strictly before step t (with the convention $\mu_{\text{hist}}(q) = -\infty$ if $\mathcal{T}(q) = \{t\}$, in which case the statement holds trivially with the baseline replaced by 0). Let $\tilde{A}(q, o) := r(o) - \mu_{\text{hist}}(q)$ denote the HRPO advantage with this strict baseline. Then

$$\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\big[\, \tilde{A}(q, o)\,\nabla_\theta \log \pi_\theta(o \mid q)\,\big] \;=\; \nabla_\theta J(\theta; q). \tag{15}$$

Proof. By linearity of expectation,

$$\mathbb{E}_{o\sim\pi_\theta}\big[\tilde{A}(q, o)\,\nabla_\theta \log \pi_\theta(o \mid q)\big] \;=\; \mathbb{E}_{o\sim\pi_\theta}\big[r(o)\,\nabla_\theta \log \pi_\theta(o \mid q)\big] - \mathbb{E}_{o\sim\pi_\theta}\big[\mu_{\text{hist}}(q)\,\nabla_\theta \log \pi_\theta(o \mid q)\big].$$

The first term equals $\nabla_\theta J(\theta; q)$ by Eq. (12). The second term vanishes: $\mu_{\text{hist}}(q)$ depends only on q and on rewards collected at past steps $\{1, \ldots, t-1\}$, all of which are jointly independent of

$o \sim \pi_\theta(\cdot \mid q)$ (a fresh sample drawn at step t). Hence $\mu_{\text{hist}}(q)$ acts as a baseline measurable in q alone and Eq. (13) applies. Combining the two, Eq. (15) follows.

Remark 1: relation to the deployed anchor. The deployed anchor $\mu(q) = \max(\mu_{\text{hist}}(q),\, \mu^t(q))$ used in Eq. (4) can equal $\mu^t(q)$ in the *progress regime* of Sec. 3.1.1, in which case the baseline depends on the current step's sample $\mathcal{O}^t(q)$. This is exactly the situation faced by every member of the GRPO family: using the empirical group mean as a baseline introduces a deterministic $\mathcal{O}(1/G)$ *leave-one-out* bias because $r_i^t(q)$ contributes to the very mean against which it is centered. A strictly unbiased counterpart in this regime is the leave-one-out variant

$$\mu_{-i}^t(q) \;:=\; \frac{1}{G-1}\sum_{j\neq i} r_j^t(q), \tag{16}$$

which is independent of $o_i^t(q)$ and recovers exact unbiasedness via the same argument as Theorem 1. The bias gap between $\mu^t(q)$ and $\mu_{-i}^t(q)$ is $\frac{1}{G}(\mu^t(q) - r_i^t(q)) = \mathcal{O}(1/G)$, and is identical to the GRPO baseline's leave-one-out bias. HRPO inherits this approximation but introduces no additional bias relative to GRPO.

Remark 2: the missing σ as a bias source in GRPO. The GRPO advantage in Eq. (2) divides the centered reward by $\sigma^t(q) = \text{std}(\{r_j^t(q)\}_{j=1}^G)$, which is itself a random function of the same group $\mathcal{O}^t(q)$. Because $\sigma^t(q)$ is neither deterministic nor independent of $r_i^t(q)$, the standard baseline argument fails:

$$\mathbb{E}_{o\sim\pi_\theta}\left[\frac{r(o) - \text{mean}(\{r_j\})}{\text{std}(\{r_j\})}\,\nabla_\theta \log \pi_\theta(o \mid q)\right] \;\neq\; \nabla_\theta J(\theta; q),$$

in general, with the discrepancy persisting at every G and not vanishing as $G \to \infty$. This is the bias formalized by [29], and it is structurally distinct from the leave-one-out bias of Remark 1: the former is a multiplicative coupling between the score function and a sample-dependent denominator, the latter is a $\mathcal{O}(1/G)$ additive perturbation of an otherwise valid baseline. By omitting the σ denominator, HRPO eliminates the multiplicative bias entirely while retaining only the standard, GRPO-equivalent leave-one-out term.

Remark 3: extension to the off-policy term. The recall set $\mathcal{R}(q)$ in Eq. (8) consists of samples drawn from a past behavior policy $\pi_{\theta^k_{\text{old}}}$. The corresponding term in the optimization objective (Eq. (9)) carries an importance ratio $\rho(q, o) = \pi_\theta(o \mid q)/\pi_{\theta^k_{\text{old}}}(o \mid q)$, which is the standard self-normalized importance-sampling correction; without clipping, it yields an unbiased estimator of $\nabla_\theta J(\theta; q)$ via the same baseline argument as Theorem 1 applied under the off-policy measure. PPO-style clipping introduces a known small bias whose treatment is identical to vanilla PPO and orthogonal to the HRPO design.

Takeaway. HRPO is unbiased in the strongest sense available within the GRPO family: under a strict historical baseline (Eq. (14)), the gradient estimator is exactly unbiased; under the deployed anchor (Eq. (4)), it inherits the same $\mathcal{O}(1/G)$ leave-one-out bias as every other GRPO-style method, and *no more*. In particular, HRPO removes the multiplicative bias that the standard-deviation normalization introduces in GRPO, answering the natural reviewer question of whether incorporating historical statistics could itself be a new bias source: it is not.

D.2 Anchor Monotonicity and the Ratchet Effect

We now formalize a structural property that follows directly from HRPO's running-maximum construction: the per-prompt anchor sequence is monotone non-decreasing in time, so the bar a sample must clear to receive any positive reinforcement only ever rises. We call this the *ratchet effect*: HRPO mechanically locks in every level of group-average performance the policy has ever attained on a given prompt, and no future step can re-earn positive advantage by merely returning to a previously surpassed level.

Setup. For each prompt q and integer $t \geq 0$, let

$$\mathcal{H}^{(t)}(q) \;:=\; \big\{\mu^k(q) \,:\, k \in \mathcal{T}(q),\, k \leq t\big\}, \qquad \mu^{(t)}(q) \;:=\; \max\big(\mathcal{H}^{(t)}(q)\big), \tag{17}$$

denote the per-prompt history and the historical peak anchor as they stand at the end of step t, with the convention $\mathcal{H}^{(0)}(q) = \emptyset$ and $\mu^{(0)}(q) = -\infty$. This notation makes explicit the time-indexing that is left implicit in $\mathcal{H}(q)$ and $\mu(q)$ used in the main text. Note that if prompt q is not sampled at step t, then $\mathcal{H}^{(t)}(q) = \mathcal{H}^{(t-1)}(q)$ by definition; the anchor evolves only when q itself is revisited.

Theorem 2 (Anchor monotonicity). *For every prompt q and every $t \geq 1$,*

$$\mu^{(t)}(q) \;\geq\; \mu^{(t-1)}(q). \tag{18}$$

Hence the sequence $\{\mu^{(t)}(q)\}_{t\geq 0}$ is monotone non-decreasing in t.

Proof. By construction, $\mathcal{H}^{(t)}(q) = \mathcal{H}^{(t-1)}(q) \cup \{\mu^t(q)\}$ if $q \in \mathcal{Q}^t$ (the mini-batch sampled at step t), and $\mathcal{H}^{(t)}(q) = \mathcal{H}^{(t-1)}(q)$ otherwise. In both cases $\mathcal{H}^{(t)}(q) \supseteq \mathcal{H}^{(t-1)}(q)$, and the maximum over a superset is at least as large as the maximum over the subset:

$$\mu^{(t)}(q) \;=\; \max\big(\mathcal{H}^{(t)}(q)\big) \;\geq\; \max\big(\mathcal{H}^{(t-1)}(q)\big) \;=\; \mu^{(t-1)}(q).$$

Iterating from $t = 1$ yields the global monotonicity claim.

Corollary 1 (Ratchet effect on positive advantages). *Let $o_i^t(q)$ be any sample drawn at step t. If its HRPO advantage is positive, $A^{\mathrm{HRPO}}(q, o_i^t(q)) > 0$, then its reward must strictly exceed every per-step group mean attained on prompt q at any earlier step:*

$$A^{\mathrm{HRPO}}\big(q, o_i^t(q)\big) \;>\; 0 \quad\Longrightarrow\quad r_i^t(q) \;>\; \mu^{(t-1)}(q) \;=\; \max_{k\in\mathcal{T}(q),\, k<t} \mu^k(q). \tag{19}$$

Proof. By definition, $A^{\mathrm{HRPO}}(q, o_i^t(q)) = r_i^t(q) - \mu^{(t)}(q)$, so $A^{\mathrm{HRPO}} > 0$ is equivalent to $r_i^t(q) > \mu^{(t)}(q)$. By Theorem 2, $\mu^{(t)}(q) \geq \mu^{(t-1)}(q)$, and the chain $r_i^t(q) > \mu^{(t)}(q) \geq \mu^{(t-1)}(q)$ gives the claim.

Corollary 2 (No-regression reinforcement). *Suppose at some past step $k < t$ the policy attained group mean $\mu^k(q)$ on prompt q. Then for every subsequent step $s > k$ at which q is sampled, every sample with $r_i^s(q) \leq \mu^k(q)$ receives a non-positive HRPO advantage.*

Proof. For any $s > k$ with $q \in \mathcal{Q}^s$, $\mu^k(q) \in \mathcal{H}^{(s-1)}(q)$ (since $k < s$ and the history is never pruned), hence $\mu^{(s-1)}(q) \geq \mu^k(q)$. By Theorem 2, $\mu^{(s)}(q) \geq \mu^{(s-1)}(q) \geq \mu^k(q)$. Therefore $r_i^s(q) \leq \mu^k(q) \leq \mu^{(s)}(q)$, which gives $A^{\mathrm{HRPO}}(q, o_i^s(q)) \leq 0$.

Corollary 3 (Anchor convergence). *If rewards are bounded, $r(o) \in [0, R_{\max}]$ for some $R_{\max} < \infty$, then for every prompt q the limit $\mu^{(\infty)}(q) := \lim_{t\to\infty} \mu^{(t)}(q)$ exists and satisfies $\mu^{(\infty)}(q) \leq R_{\max}$.*

Proof. Each $\mu^k(q)$ is an average of bounded rewards, so $\mu^{(t)}(q) \leq R_{\max}$ for all t. Combined with Theorem 2, $\{\mu^{(t)}(q)\}_{t\geq 0}$ is monotone non-decreasing and bounded above, hence converges by the monotone convergence theorem.

Remark 1: contrast with GRPO's free-floating baseline. In GRPO, the per-step baseline $\mathrm{mean}(\{r_j^t(q)\}_{j=1}^G)$ is itself a function only of the current group $\mathcal{O}^t(q)$ and inherits all of its volatility. In particular, when the policy regresses on q between steps k and $t > k$, the group mean can drop, $\mathrm{mean}(\{r_j^t\}) < \mathrm{mean}(\{r_j^k\})$, and a sample at step t with reward $r_i^t \leq \mu^k(q)$ may still be assigned positive GRPO advantage as long as it exceeds the new (lower) mean. Theorem 2 and Corollary 2 formalize precisely the structural difference: the HRPO anchor only ratchets up, so positive reinforcement is gated by the strongest historical level, not by an instantaneous baseline that follows the policy down.

Remark 2: bounded buffer growth. Theorem 2 also yields a structural bound on the replay buffer $\mathcal{B}(q)$: every stored sample satisfies $r > \mu^{(t)}(q)$ at insertion, and the self-pruning rule (Eq. (6)) discards any sample whose reward drops at or below the updated anchor. Combined with monotonicity, the lifetime of a stored sample is bounded by the time until the anchor surpasses its reward; in particular, the buffer can only grow when $\mu^{(t)}(q)$ strictly increases, and is automatically pruned without an explicit size hyperparameter.

Remark 3: building block for deception immunity. Theorem 2 is the structural backbone for the high-probability deception-immunity bound in Sec. D.3: the bound there controls the gap between the current expected reward $V^{\pi_{\theta_t}}(q)$ and the historical peak $\mu^{(t)}(q)$, and Theorem 2 ensures that the latter is at least as large as $\mu^{(k)}(q)$ for any past step k, so the bound is automatically anchored at the strongest observed historical performance.

Takeaway. HRPO's running-maximum construction turns the advantage estimator into a ratchet: the bar for positive reinforcement is monotone non-decreasing on every prompt. Operationally, this means HRPO will never reinforce a sample whose reward fails to exceed the strongest level the policy

has previously attained on that prompt, irrespective of how the current step's group statistics happen to evolve. Theorem 2 also serves as a lightweight structural lemma that simplifies subsequent proofs (Sec. D.3, Sec. D.4).

D.3 High-Probability Immunity to the Relative Deception Effect

We now formalize the central claim of the paper: HRPO's expected advantage is provably negative whenever the current policy has regressed against any prior peak on a prompt, while GRPO's expected advantage is identically zero regardless of the policy's evolution. This sharply quantifies what Sec. 2.2 calls the *relative deception effect*: GRPO's optimizer is blind to policy regression in expectation, whereas HRPO's optimizer receives a calibrated negative signal whose magnitude scales with the size of the regression.

Setup. Fix a prompt q. For each step $k \in \mathcal{T}(q)$, denote by $\pi_k := \pi_{\theta^k_{\text{old}}}$ the behavior policy at step k (the snapshot from which the group $\mathcal{O}^k(q) = \{o^k_j(q)\}_{j=1}^G$ is sampled), and let

$$V_k(q) \;:=\; V^{\pi_k}(q) \;=\; \mathbb{E}_{o\sim\pi_k(\cdot|q)}[\,r(o)\,] \tag{20}$$

denote its expected reward on q. The empirical group mean $\mu^k(q) = \frac{1}{G}\sum_{j=1}^G r^k_j(q)$ is then the G-sample Monte Carlo estimator of $V_k(q)$ under π_k. We assume rewards are bounded, $r(o) \in [0, R_{\max}]$, throughout.

Auxiliary lemma (Hoeffding). For any fixed step k, conditional on θ^k_{old}, the rewards $\{r^k_j(q)\}_{j=1}^G$ are i.i.d. in $[0, R_{\max}]$, so for any $\delta \in (0,1)$,

$$\Pr\left[\,\left|\mu^k(q) - V_k(q)\right| \;\geq\; R_{\max}\sqrt{\frac{\log(2/\delta)}{2G}} \;\middle|\; \theta^k_{\text{old}}\right] \;\leq\; \delta, \tag{21}$$

by Hoeffding's inequality applied to the bounded i.i.d. sample mean.

Theorem 3 (High-probability deception immunity). *Fix a prompt q, a current step t, and a confidence level $\delta \in (0,1)$. Suppose there exists a past step $k^* \in \mathcal{T}(q)$ with $k^* < t$ such that the current policy has regressed in expected reward by at least $\Delta \geq 0$ relative to step k^*:*

$$V_{k^*}(q) \;-\; V_t(q) \;\geq\; \Delta. \tag{22}$$

Then, with probability at least $1-\delta$ over the sampling of $\mathcal{O}^{k^*}(q)$ (conditional on $\theta^{k^*}_{\text{old}}$),

$$\mathbb{E}_{o\sim\pi_t(\cdot|q)}\big[\,A^{\text{HRPO}}(q,o)\,\big] \;\leq\; -\Delta \;+\; R_{\max}\sqrt{\frac{\log(2/\delta)}{2G}}. \tag{23}$$

Proof. We unpack the expected HRPO advantage in three steps.

Step 1 (decomposition). By definition of the HRPO advantage (Eq. (5)) and the linearity of expectation,

$$\mathbb{E}_{o\sim\pi_t}[A^{\text{HRPO}}(q,o)] \;=\; \mathbb{E}_{o\sim\pi_t}[r(o)] \;-\; \mu^{(t)}(q) \;=\; V_t(q) \;-\; \mu^{(t)}(q),$$

where $\mu^{(t)}(q)$ is the deployed anchor at step t in the time-indexed notation of Sec. D.2.

Step 2 (lower-bound the anchor by the past empirical mean). By Theorem 2 (anchor monotonicity), and because $k^* < t$ together with $k^* \in \mathcal{T}(q)$ implies $\mu^{k^*}(q) \in \mathcal{H}^{(t)}(q)$,

$$\mu^{(t)}(q) \;=\; \max(\mathcal{H}^{(t)}(q)) \;\geq\; \mu^{k^*}(q).$$

Step 3 (concentrate the empirical mean to its expectation). By the Hoeffding bound Eq. (21) applied at step k^*, with probability at least $1-\delta$,

$$\mu^{k^*}(q) \;\geq\; V_{k^*}(q) \;-\; R_{\max}\sqrt{\frac{\log(2/\delta)}{2G}}.$$

Chaining the three displays and using the regression assumption Eq. (22),

$$\begin{aligned}\mathbb{E}_{o\sim\pi_t}\left[A^{\text{HRPO}}(q,o)\right] &= V_t(q) - \mu^{(t)}(q) \leq V_t(q) - \mu^{k^*}(q)\\ &\leq V_t(q) - V_{k^*}(q) + R_{\max}\sqrt{\frac{\log(2/\delta)}{2G}} \leq -\Delta + R_{\max}\sqrt{\frac{\log(2/\delta)}{2G}}.\end{aligned}$$

Corollary 1 (Active gradient under non-trivial regression). *If $\Delta > R_{\max}\sqrt{\log(2/\delta)/(2G)}$, then with probability at least $1-\delta$ the expected HRPO advantage is strictly negative, and consequently HRPO's policy-gradient estimator (in the on-policy regime of Theorem 1) descends a strictly non-zero direction that pushes the policy away from π_t.*

Proposition 1 (GRPO's blindness to regression). *Under the same setup, the GRPO advantage estimator (Eq. (2)) satisfies*

$$\mathbb{E}_{o\sim\pi_t(\cdot|q)}\big[\, r(o) - \mu^t(q) \,\big] \;=\; 0, \qquad \text{for every } \pi_t \text{ and every } \Delta \geq 0, \tag{24}$$

where $\mu^t(q) = \frac{1}{G}\sum_j r_j^t(q)$ is the current step's group mean. In particular, the regression magnitude Δ leaves no trace in the expected (numerator of the) GRPO advantage.

Proof of Proposition 1. The current group $\mathcal{O}^t(q)$ is sampled from $\pi_t = \pi_{\theta^t_{\text{old}}}$, so the group mean satisfies $\mathbb{E}_{\mathcal{O}^t\sim\pi_t}[\mu^t(q)] = V_t(q)$. For an independent sample $o \sim \pi_t(\cdot \mid q)$,

$$\mathbb{E}_{o\sim\pi_t}[r(o) - \mu^t(q)] \;=\; V_t(q) - \mathbb{E}[\mu^t(q)] \;=\; V_t(q) - V_t(q) \;=\; 0,$$

independent of any past policy or of Δ. Dividing by $\sigma^t(q)$ does not change the sign of the expected numerator and the conclusion that GRPO's advantage signal carries no information about Δ in expectation.

Remark 1: this is the deception effect, formalized. Theorem 3 and Proposition 1 together pin down the precise gap between HRPO and GRPO that motivates this paper. GRPO's expected advantage is identically zero on every prompt at every step, regardless of how far the policy has drifted from any historical peak: the optimizer cannot tell, from the average advantage signal, that any regression has occurred. HRPO's expected advantage, in contrast, is bounded above by $-\Delta + \mathcal{O}(R_{\max}/\sqrt{G})$, so any regression of size $\Delta \gg R_{\max}/\sqrt{G}$ produces a strictly negative expected signal that reliably penalizes the regressed policy. The ratchet provided by Theorem 2 ensures this protection persists for the remainder of training.

Remark 2: tightness in G and δ. The Hoeffding rate $\sqrt{\log(2/\delta)/(2G)}$ is the standard non-asymptotic concentration of a G-sample bounded mean and is tight up to constants. The bound Eq. (23) therefore decays at the canonical $\mathcal{O}(1/\sqrt{G})$ rate in the group size: doubling the group size halves the deception slack, while the high-probability framing decouples the bound from any distributional structure beyond bounded support. For typical RLHF settings ($G = 8$, $R_{\max} = 1$, $\delta = 0.05$), the slack term evaluates to roughly 0.48, so a regression of size $\Delta \gtrsim 0.5$ already triggers a strictly negative expected advantage with 95% confidence; for $G = 16$ the threshold drops below 0.34.

Remark 3: extension to multiple historical reference steps. Theorem 3 is stated against a single past step k^*, but the construction of $\mu^{(t)}(q)$ as a running maximum over *all* past steps means the bound Eq. (23) holds simultaneously for every $k \in \mathcal{T}(q)$ with $k < t$, by a union bound or by simply taking the worst (largest) Δ. In particular, defining $\Delta^*(q) := \max_{k\in\mathcal{T}(q),\, k<t} \big(V_k(q) - V_t(q)\big)$ as the worst regression against any historical peak, Theorem 3 implies

$$\mathbb{E}_{o\sim\pi_t}[A^{\text{HRPO}}(q,o)] \;\leq\; -\Delta^*(q) + R_{\max}\sqrt{\frac{\log(2/\delta)}{2G}} \quad \text{w.p. } \geq 1-\delta.$$

Remark 4: complementarity with ODHR. Theorem 3 controls the *negative* side of the HRPO advantage during regression: it guarantees that the optimizer is correctly told what *not* to do. However, when every sample in the current group falls below the anchor, the on-policy gradient may still vanish in magnitude (the negative direction is detected, but with arbitrarily small step). Sec. D.4 establishes the complementary positive-side guarantee: ODHR injects a non-vanishing positive gradient via replayed buffer samples, ensuring the optimizer also receives a constructive direction to pursue.

Takeaway. HRPO converts the qualitative claim "GRPO is blind to relative deception, HRPO is not" into a quantitative high-probability bound: every Δ-magnitude regression on a prompt produces an expected HRPO advantage at most $-\Delta + \mathcal{O}(R_{\max}/\sqrt{G})$, while the corresponding GRPO advantage is identically zero in expectation. The bound is tight up to constants, anchored at the strongest historical peak (via Theorem 2), and is structurally distinct from anything achievable by within-step normalization alone.

D.4 ODHR: Non-Vanishing Gradient under Positive-Signal Starvation

Theorem 3 guarantees that HRPO's expected advantage correctly reports a regression by giving a calibrated negative signal, telling the optimizer what to avoid. However, when the entire current group falls below the anchor (the *positive-signal starvation* regime defined in Sec. 3.1.2), the on-policy gradient magnitude can shrink toward zero: the optimizer knows "not here," but has no positive direction to descend toward. We now show that ODHR provides exactly the missing positive-side guarantee, contributing a strictly positive learning signal whose magnitude is bounded below by the buffer invariant and is independent of how marginal the starvation becomes.

Setup. We adopt the notation of Sec. D.1 –D.3: $\pi_t = \pi_{\theta^t_{\text{old}}}$ is the behavior policy at step t, $\mathcal{O}^t(q) = \{o_j^t(q)\}_{j=1}^G$ is the on-policy group with empirical mean $\mu^t(q) = \frac{1}{G}\sum_j r_j^t(q)$, $\mu^{(t)}(q)$ is the deployed anchor at step t, and $\mathcal{R}(q) \subseteq \mathcal{B}(q)$ is the recall set produced by ODHR (Eq. (8)). We assume rewards lie in $[0, R_{\max}]$ throughout. We say that strict starvation holds at (t, q) when

$$\max_{j\in[G]} r_j^t(q) \;<\; \mu^{(t)}(q) \quad \text{and} \quad \mathcal{B}(q) \neq \emptyset, \tag{25}$$

which is a slightly strengthened version of the trigger condition in Eq. (7). Define the two key gaps directly from observed group/buffer statistics:

$$\bar{\Delta}_{\text{neg}}(q) \;:=\; \mu^{(t)}(q) - \mu^t(q) \;>\; 0, \qquad \Delta^{\min}_{\text{pos}}(q) \;:=\; \min_{x\in\mathcal{R}(q)} \big(r(x) - \mu^{(t)}(q)\big) \;>\; 0, \tag{26}$$

where the strict positivity of $\bar{\Delta}_{\text{neg}}$ follows from $\mu^t(q) \le \max_j r_j^t(q) < \mu^{(t)}(q)$ under Eq. (25), and the strict positivity of $\Delta^{\min}_{\text{pos}}$ follows from the buffer invariant $r(x) > \mu^{(t)}(q)$ for every $x \in \mathcal{B}(q)$ (Eq. (6)). Intuitively, $\bar{\Delta}_{\text{neg}}$ measures how far the current group's mean falls below the anchor (the realized negative-side gap), while $\Delta^{\min}_{\text{pos}}$ measures the smallest strictly-positive gap among replayed samples.

Auxiliary lemma (Strict baseline under starvation). *Whenever Eq. (25) holds, the deployed anchor reduces to the strict historical peak: $\mu^{(t)}(q) = \mu_{\text{old}}(q) := \max_{k\in\mathcal{T}(q),\, k<t} \mu^k(q)$. In particular, $\mu^{(t)}(q)$ is independent of the current group $\mathcal{O}^t(q)$.*

Proof. By Eq. (25), $\mu^t(q) = \frac{1}{G}\sum_j r_j^t(q) \le \max_j r_j^t(q) < \mu^{(t)}(q)$, so $\mu^{(t)}(q) \neq \mu^t(q)$. Combined with $\mu^{(t)}(q) = \max(\mu_{\text{old}}(q), \mu^t(q))$, this forces $\mu^{(t)}(q) = \mu_{\text{old}}(q)$, which depends only on past steps and not on $\mathcal{O}^t(q)$.

Theorem 4 (Strict positive replay signal under starvation). *Suppose strict starvation Eq. (25) holds at (t, q). Then:*

*(a) **On-policy advantage shrinks toward zero.** The realized average HRPO advantage over the current group is exactly $-\bar{\Delta}_{\text{neg}}(q)$:*

$$\frac{1}{G}\sum_{j=1}^{G} A^{\text{HRPO}}(q, o_j^t(q)) \;=\; \mu^t(q) - \mu^{(t)}(q) \;=\; -\bar{\Delta}_{\text{neg}}(q) \;<\; 0. \tag{27}$$

Furthermore, every individual on-policy advantage is non-positive: $A^{\text{HRPO}}(q, o_j^t(q)) = r_j^t(q) - \mu^{(t)}(q) \le \max_j r_j^t(q) - \mu^{(t)}(q) < 0$. In the marginal-starvation limit ($\mu^t(q) \to \mu^{(t)}(q)$, equivalently $\bar{\Delta}_{\text{neg}}(q) \to 0$), the realized average advantage tends to 0.

*(b) **Replay advantage is uniformly bounded below.** Every replayed sample $x \in \mathcal{R}(q)$ has strictly positive HRPO advantage:*

$$A^{\text{HRPO}}(q, x) \;=\; r(x) - \mu^{(t)}(q) \;\ge\; \Delta^{\min}_{\text{pos}}(q) \;>\; 0, \qquad \textit{for all } x \in \mathcal{R}(q). \tag{28}$$

This bound is uniform in x, independent of $\bar{\Delta}_{\text{neg}}(q)$, and persists no matter how marginal the starvation becomes.

Proof. For (a): direct calculation, $\frac{1}{G}\sum_j A^{\text{HRPO}}(q, o_j^t) = \frac{1}{G}\sum_j (r_j^t - \mu^{(t)}) = \mu^t - \mu^{(t)} = -\bar{\Delta}_{\text{neg}}(q)$, which is strictly negative under starvation by the definition in Eq. (26). The per-sample bound follows from $r_j^t \le \max_j r_j^t < \mu^{(t)}$.

For (b): by the buffer invariant $\mathcal{B}(q) = \{(o, r, \cdot) : r > \mu^{(t)}(q)\}$ (Eq. (6)), every $x \in \mathcal{R}(q) \subseteq \mathcal{B}(q)$ satisfies $r(x) > \mu^{(t)}(q)$, hence $A^{\text{HRPO}}(q, x) = r(x) - \mu^{(t)}(q) > 0$. Taking the minimum over $\mathcal{R}(q)$ gives $A^{\text{HRPO}}(q, x) \ge \Delta^{\min}_{\text{pos}}(q) > 0$ for all $x \in \mathcal{R}(q)$. Uniformity is by construction.

Corollary 1 (Vanishing on-policy gradient magnitude). *Suppose the score function is bounded,* $\|\nabla_\theta \log \pi_\theta(o \mid q)\| \le S_{\max}$ *for all* o, q. *Define the realized on-policy gradient component contributed by prompt* q*:*

$$\hat{g}_{\text{van}}(q) \;:=\; \frac{1}{G}\sum_{j=1}^{G} A^{\text{HRPO}}\big(q, o_j^t(q)\big)\, \nabla_\theta \log \pi_\theta\big(o_j^t(q) \mid q\big). \tag{29}$$

Then under strict starvation, $\|\hat{g}_{\text{van}}(q)\| \le S_{\max} \cdot \bar{\Delta}_{\text{neg}}(q)$*, which can be made arbitrarily small as* $\bar{\Delta}_{\text{neg}}(q) \to 0$.

Proof. Under starvation, every per-sample advantage satisfies $A^{\text{HRPO}}(q, o_j^t) = r_j^t - \mu^{(t)} \le \max_j r_j^t - \mu^{(t)} < 0$ (Theorem 4(a)), so $|A^{\text{HRPO}}(q, o_j^t)| = \mu^{(t)} - r_j^t$. By the triangle inequality and the bounded-score assumption,

$$\begin{aligned}\|\hat{g}_{\text{van}}(q)\| &\le \frac{1}{G}\sum_{j=1}^{G} \big|A^{\text{HRPO}}(q, o_j^t)\big|\, \big\|\nabla_\theta \log \pi_\theta(o_j^t \mid q)\big\| \\ &\le S_{\max} \cdot \frac{1}{G}\sum_{j=1}^{G} \big(\mu^{(t)}(q) - r_j^t(q)\big) = S_{\max} \cdot \bar{\Delta}_{\text{neg}}(q).\end{aligned}$$

Corollary 2 (Non-vanishing replay-gradient pull). *Suppose additionally that the importance ratio is lower-bounded after clipping,* $\rho(q, x) \ge \rho_{\text{lo}} > 0$ *for every* $x \in \mathcal{R}(q)$. *Define the replay-gradient component*

$$g_{\text{rep}}(q) \;:=\; \frac{1}{K}\sum_{x \in \mathcal{R}(q)} \rho(q, x)\, A^{\text{HRPO}}(q, x)\, \nabla_\theta \log \pi_\theta(x \mid q), \tag{30}$$

where $K = |\mathcal{R}(q)|$. *Then for each replayed sample* $x \in \mathcal{R}(q)$*, the per-sample multiplier acting on its score direction* $\nabla_\theta \log \pi_\theta(x \mid q)$ *is uniformly lower-bounded:*

$$\rho(q, x)\, A^{\text{HRPO}}(q, x) \;\ge\; \rho_{\text{lo}}\, \Delta_{\text{pos}}^{\min}(q) \;>\; 0. \tag{31}$$

Consequently, projecting $g_{\text{rep}}(q)$ *onto the average replay-score direction* $\bar{s}_{\text{rep}}(q) := \frac{1}{K}\sum_{x \in \mathcal{R}(q)} \nabla_\theta \log \pi_\theta(x \mid q)$ *gives a positive lower bound,*

$$\big\langle g_{\text{rep}}(q),\, \bar{s}_{\text{rep}}(q) \big\rangle \;\ge\; \rho_{\text{lo}}\, \Delta_{\text{pos}}^{\min}(q) \cdot \big\|\bar{s}_{\text{rep}}(q)\big\|_{\text{Cov}}^2, \tag{32}$$

where $\|\bar{s}_{\text{rep}}(q)\|_{\text{Cov}}^2 := \frac{1}{K^2}\sum_{x,y \in \mathcal{R}(q)} \langle \nabla_\theta \log \pi_\theta(x \mid q), \nabla_\theta \log \pi_\theta(y \mid q) \rangle \ge 0$.

Proof. Eq. (31) is immediate from $A^{\text{HRPO}}(q, x) \ge \Delta_{\text{pos}}^{\min}(q)$ (Theorem 4(b)) and $\rho(q, x) \ge \rho_{\text{lo}}$. Substituting into Eq. (30) and taking the inner product with $\bar{s}_{\text{rep}}(q)$,

$$\langle g_{\text{rep}}(q), \bar{s}_{\text{rep}}(q) \rangle = \frac{1}{K^2} \sum_{x,y \in \mathcal{R}(q)} \rho(q, x)\, A^{\text{HRPO}}(q, x)\, \langle \nabla \log \pi_\theta(x \mid q), \nabla \log \pi_\theta(y \mid q) \rangle.$$

Since the per-sample multiplier $\rho(q, x) A^{\text{HRPO}}(q, x) \ge \rho_{\text{lo}} \Delta_{\text{pos}}^{\min}(q) > 0$ uniformly, factoring out the lower bound and observing that the resulting kernel $\langle \nabla \log \pi_\theta(x \mid q), \nabla \log \pi_\theta(y \mid q) \rangle$ is the Gram matrix of the replay scores yields Eq. (32). The Gram matrix is positive semi-definite, so $\|\bar{s}_{\text{rep}}\|_{\text{Cov}}^2 \ge 0$.

Remark 1: when does the lower bound Eq. (32) bite? The Gram-matrix term $\|\bar{s}_{\text{rep}}\|_{\text{Cov}}^2$ is strictly positive whenever the replay scores are not collectively orthogonal to a common direction. In practice, replayed samples for the same prompt are similar high-reward responses (TopK from $\mathcal{B}(q)$), and their score vectors typically share a large common component pointing toward "increase the likelihood of high-reward responses on q," so the inner product is bounded well away from 0. The vanishing case $\bar{s}_{\text{rep}} \equiv 0$ would require the replay scores to be antisymmetric, which has no incentive to occur after the buffer self-pruning. Empirically, ODHR's measured trigger-step improvement scale matches this prediction: the replay term consistently contributes a non-trivial gradient even when the on-policy term has effectively vanished.

Remark 2: contrast with vanilla on-policy. Combining Corollary 1 and Corollary 2 isolates the qualitative gap that motivates ODHR. The average on-policy multiplier $\frac{1}{G}\sum_j |A^{\text{HRPO}}(q, o_j^t)|$ shrinks linearly with $\bar{\Delta}_{\text{neg}}$, which can be arbitrarily small under marginal starvation (e.g., the group mean μ^t just ϵ below $\mu^{(t)}$ yields an average multiplier of ϵ). In contrast, the replay multiplier $\rho_{\text{lo}} \Delta_{\text{pos}}^{\min}$

is determined entirely by the gap between the buffered samples and the anchor, which is bounded away from zero by the buffer invariant and unaffected by how close μ^t is to $\mu^{(t)}$. Hence ODHR is the only mechanism within the HRPO framework that delivers a strictly non-vanishing positive-side gradient signal in the starvation regime.

Remark 3: PPO-style clipping. The IS ratio in Eq. (9) is wrapped in a $\min(\rho, \text{clip}(\rho, 1-\epsilon, 1+\epsilon))$ operator. Whenever the surrogate gradient is in the "increase x's likelihood" direction (the side aligned with $A^{\text{HRPO}}(q, x) > 0$), the active branch is ρ when $\rho < 1 + \epsilon$, hence $\rho_{\text{lo}} = 1 - \epsilon_{\text{off}}$ in the worst case where the off-policy ratio sits at the lower clip boundary. This guarantees $\rho_{\text{lo}} > 0$ whenever $\epsilon_{\text{off}} < 1$, which is satisfied by all standard PPO/HRPO configurations.

Remark 4: complementarity with Theorem 3 and the deception/starvation duality. Theorems 3 and 4 jointly cover the two failure modes that the relative deception analysis exposes. Theorem 3 quantifies the *negative-side* signal ("what to avoid") by lower-bounding the magnitude of the negative expected advantage when the policy regresses; Theorem 4 quantifies the *positive-side* signal ("what to pursue") by lower-bounding the strictly positive replay advantage when on-policy responses fail. Together, they certify that HRPO+ODHR provides directional gradient information on both sides of the anchor, exactly the property that vanilla GRPO lacks (Proposition 1 of Sec. D.3; vanishing on-policy magnitude in Corollary 1 above).

Takeaway. ODHR converts the qualitative argument "on-policy alone can be starved" into a quantitative bifurcation: the on-policy gradient magnitude vanishes proportionally to $\bar{\Delta}_{\text{neg}}$, while the replay gradient contributes a per-sample multiplier $\geq \rho_{\text{lo}} \Delta^{\min}_{\text{pos}} > 0$ that is structurally bounded away from 0 by the buffer invariant. This non-vanishing positive-side guarantee is the formal counterpart of the closed-loop recovery argument in Sec. 3.3, and it sharpens the practical claim that ODHR is not a heuristic patch but a principled mechanism for recovering directional gradient information when the on-policy signal degrades.

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 the corresponding part.

   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 the corresponding part.

   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: See the corresponding part.

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 the corresponding part.

   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: [N/A]

   Justification: Code will be released after the peer-review stage.

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 the corresponding 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: Yes, see experiments.

   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 experiment sections.

   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:

   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: [Yes]

    Justification: We have procided it.

    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 work does not involve the release of models and data. This paper merely presents a new algorithm.

    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 requiring 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:

    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: [N/A]

    Justification:

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:

    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:

    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: [Yes]

    Justification: We have provided it.

    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.