It Takes Two: Your GRPO Is Secretly DPO

Yihong Wu[*1], **Liheng Ma**[*2,3], **Lei Ding**[4], **Muzhi Li**[5], **Xinyu Wang**[2], **Kejia Chen**[6], **Zhan Su**[1], **Zhanguang Zhang**[7], **Chenyang Huang**[7,8,9], **Yingxue Zhang**[7], **Mark Coates**[2,3], **Jian-Yun Nie**[1]
[1]Université de Montréal [2]McGill University [3]Mila - Quebec AI Institute
[4]University of Manitoba [5]The Chinese University of Hong Kong [6]Zhejiang University
[7]Huawei Noah's Ark Lab [8]University of Alberta [9]Alberta Machine Intelligence Institute (Amii)

Abstract

Group Relative Policy Optimization (GRPO) is a prominent reinforcement learning algorithm for post-training Large Language Models (LLMs). It is commonly believed that GRPO necessitates a large group size to ensure stable training via precise statistical estimation, which incurs substantial computational overhead. In this work, we challenge this assumption by reframing GRPO as a form of contrastive learning, which reveals a fundamental connection to Direct Preference Optimization (DPO). Motivated by DPO's empirical success, we investigate the minimal two-rollout case (2-GRPO)—a configuration previously deemed infeasible. We provide a rigorous theoretical analysis to validate 2-GRPO and demonstrate empirically that it achieves performance on par with 16-GRPO, despite using only $1/8$ of the rollouts and reducing training time by over 70%.

1 Introduction

Reinforcement Learning (RL) is now a central paradigm for post-training Large Language Models (LLMs), aligning preference through RL with Human Feedback (RLHF) (Ouyang et al., 2022) and incentivizing reasoning capability through RL with Verifiable Rewards (RLVR) (Shao et al., 2024; Guo et al., 2025). Among recent advances, *Group Relative Policy Optimization* (GRPO) (Shao et al., 2024; Guo et al., 2025), has emerged as a powerful variant of Proximal Policy Optimization (PPO) (Schulman et al., 2017). Unlike PPO, which relies on value networks to stabilize rewards, GRPO samples multiple responses (rollouts) per prompt and normalizes their rewards within each group. This simple and effective strategy achieves state-of-the-art performance on various tasks while reducing significant computational resources.

Despite GRPO's strong empirical performance, its theory remains largely unexplored. In this work, we revisit GRPO through the lens of contrastive learning (Wang and Isola, 2020; Chen et al., 2020; He et al., 2020; Wu et al., 2024). From this viewpoint, the GRPO objective naturally resembles a contrastive loss: its intra-group normalization implicitly divides responses into positive and negative samples, encouraging positive responses while suppressing negative ones. This perspective reveals a key conceptual link between GRPO and Direct Preference Optimization (DPO) (Rafailov et al., 2023), a prominent alignment algorithm in RLHF. Both approaches optimize policies based on preference signals, though under different settings. Building on this connection, we introduce **2-GRPO**, a DPO-inspired variant of GRPO with a response group size of two. Despite its simplicity, 2-GRPO preserves unbiased gradient estimation while offering greater efficiency.

Conventional viewpoint attributes GRPO's empirical success to its stable group normalization, which relies on large group sizes for accurate statistical estimation. However, generating many rollouts per prompt leads to substantial computational and time costs. Our proposed 2-GRPO algorithm tackles

[*]Equal contribution. yihong.wu@umontreal.ca, liheng.ma@mail.mcgill.ca.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

this inefficiency head-on by reducing the group size to 2. At first glance, this design might violate the principle of GRPO, yet our theoretical analysis and experiments reveal the opposite. Specifically, we show that: (i) 2-GRPO preserves an implicit form of advantage estimation; (ii) the potential increase in gradient variance can be mitigated by a larger batch size; and (iii) 2-GRPO does not have less positive signals compared to its large-group counterpart. Empirically, 2-GRPO achieves performance on par with standard GRPO while reducing computational overhead and training time significantly.

Our findings challenge the prevailing assumption that large group sizes are essential for the performance of GRPO. By demonstrating that 2-GRPO is a competitive and substantially more efficient alternative, we offer a new direction for designing resource-efficient RL algorithms for LLM post-training. Our main contributions are:

- **A Contrastive Reinterpretation of GRPO.** We formalize GRPO as a contrastive objective distinguishing positive from negative rollouts via group-normalized advantages. This reframing clarifies its conceptual connection to preference-based methods like DPO.
- **Theoretical Guarantees for the Pairwise Setting.** In the context of RLVR, we prove that pairwise grouping is sufficient. Our analysis shows that 2-GRPO not only preserves the contrastive optimization behavior of standard GRPO but also provides unbiased gradient estimates, dispelling the notion that large groups are necessary for stable learning.
- **Empirical Validation.** Across multiple language models and reasoning datasets, we show that 2-GRPO matches the performance of standard GRPO while significantly reducing training time and computational resource usage.

The rest of the paper is organized as follows. We begin with a brief review of RL for LLM post-training and summarize commonly used algorithms. Next, we present a theoretical analysis that connects GRPO and DPO through the lens of contrastive learning via gradient analysis. We then analyze the properties of 2-GRPO in depth, demonstrating that it yields unbiased gradients and preserves the key characteristics of standard GRPO despite its reduced group size. Finally, we validate our approach through extensive experiments across diverse datasets and model scales.

2 Preliminary

Our work focuses on RL-based post-training of pre-trained LLMs to improve their reasoning capabilities, with particular emphasis on settings where responses can be automatically verified as correct or incorrect, i.e., the RLVR setting.

Let π_θ denote the policy network, i.e., the LLM parameterized by θ. Given an input prompt q, the model generates a response $o_i = (o_{i,1}, \ldots, o_{i,T})$, where $o_{i,t}$ is the token generated at step $t \in [0, T]$ and $o_{i,<t}$ denotes the sequence of preceding tokens.

We let $\mathcal{Q}$ be the set of prompts, each consisting of a question and any necessary instructions[2]. A trajectory $\tau \in \mathcal{T}$ is defined as a pair consisting of a prompt $q \in \mathcal{Q}$ and its corresponding LLM-generated response sequence o, i.e., $\tau = (q, o)$.

In RL-based post-training, the reward function is typically defined at the trajectory level, i.e., $r : \mathcal{T} \to \mathbb{R}$. The learning objective is to maximize the expected reward over the space of trajectories:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{Q}}\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}[r(\tau)] \, . \tag{1}$$

Vanilla Policy Gradient (VPG) (Williams, 1992): VPG (a.k.a. REINFORCE) aims to maximize the reward with gradient ascent:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{Q}}\mathbb{E}_{o_i\sim\pi_\theta(\cdot|q)}\left[r_i \sum_{t=0}^{|o_i|} \nabla_\theta \pi_\theta(o_{i,t}|o_{i,<t}, q) \right] \, . \tag{2}$$

where r_i is the reward of (q, o_i).

[2]Throughout this paper, we use the terms "prompt" and "question" interchangeably.

Proximity Policy Optimization (PPO) (Schulman et al., 2017): VPG might suffer from high variance and instability (Schulman et al., 2015). To reduce the variance and instability, PPO introduces importance sampling, clipping, and a value function for computing advantage:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathop{\mathbb{E}}_{\substack{q\sim\mathcal{Q} \\ o_i\sim\pi_{\theta_{\text{old}}}}} \frac{1}{T}\sum_{t=1}^{T} \min\left[\frac{\pi_\theta(o_{i,t}|o_{i,<t},q)}{\pi_{\theta_{\text{old}}}(o_{i,t}|o_{i,<t},q)}A_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|o_{i,<t},q)}{\pi_{\theta_{\text{old}}}(o_{i,t}|o_{i,<t},q)}, 1-\epsilon, 1+\epsilon\right)A_{i,t}\right], \tag{3}$$

where $\pi_{\theta_{\text{old}}}$ is the policy which generates the sequences, π_θ is the policy to update, ϵ is a hyperparameter for clipping, and $A_{i,t}$ is the advantage, which is computed from r_i by subtracting a value baseline. Here, the baseline is provided by a value function, which is usually parameterized as another LLM in LLM post-training.

Direct Preference Optimization (DPO) (Rafailov et al., 2023): DPO is proposed for RLHF, which is usually trained with offline human-annotated preference data $(q, o_+, o_-) \sim \mathcal{D}_{\text{DPO}}$. The loss function of DPO is

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(q,o_+,o_-)\sim\mathcal{D}_{\text{DPO}}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(o_+|q)}{\pi_{\text{ref}}(o_+|q)} - \beta\log\frac{\pi_\theta(o_-|q)}{\pi_{\text{ref}}(o_-|q)}\right)\right], \tag{4}$$

where o_+ and o_- denote a preferred (positive) and a dispreferred (negative) response, respectively; β is a parameter controlling the deviation from the base reference policy π_{ref}.

Group Relative Policy Optimization (GRPO) (Shao et al., 2024): GRPO – the RL algorithm behind the success of DeepSeek-R1 (Guo et al., 2025) – has become one of the most widely used RL algorithms for LLM post-training. Instead of maintaining a value network like PPO, GRPO generates a group of G trajectories for each prompt (usually referred to as rollouts), and normalizes the corresponding rewards within each group to compute the advantages:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathop{\mathbb{E}}_{\substack{q\sim\mathcal{Q} \\ o_i\sim\pi_{\theta_{\text{old}}}}} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \min\left[\frac{\pi_\theta(o_{i,t}|o_{i,<t},q)}{\pi_{\theta_{\text{old}}}(o_{i,t}|o_{i,<t},q)}A_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|o_{i,<t},q)}{\pi_{\theta_{\text{old}}}(o_{i,t}|o_{i,<t},q)}, 1-\epsilon, 1+\epsilon\right)A_{i,t}\right], \tag{5}$$

where $G \geq 2$, o_i denotes the i-th trajectory, and $A_{i,t}$ denotes the corresponding advantage.

The token-level advantage is given by the intra-group normalization:

$$A_{i,t} = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r}) + \epsilon}, \tag{6}$$

where $r_i \in \mathbf{r}$ is the reward of the rollout, and ϵ is a small constant added to avoid division by zero. In the degenerate cases, where all generated trajectories receive identical rewards (all correct or all incorrect), we have $A_{i,t} = 0$ for all i, t, leading to a zero gradient for the parameter update. In practice, the choice of G is typically relatively large, e.g., 16, in order to perform proper normalization.

3 Bridging GRPO and DPO with Contrastive Learning

In this section, we bridge GRPO and DPO from the perspective of contrastive learning via gradient analysis. This perspective not only clarifies the underlying mechanism of GRPO but also motivates a deeper investigation into how group structures in GRPO can be more effectively designed.

The key insight is that advantage values are inherently signed quantities: they are either positive or negative. This observation naturally leads to a contrastive interpretation: trajectories with positive advantages can be viewed as "positive examples", while those with negative advantages correspond to "negative examples". This mirrors the core principle of contrastive learning, which seeks to increase the likelihood of positive samples (given an anchor) while decreasing that of negative ones.

Although various contrastive loss functions and settings exist—ranging from 1-vs-1 (one positive sample and one negative sample) (Rendle et al., 2009) and 1-vs-n (Oord et al., 2018) to n-vs-n (Frosst et al., 2019)—we aim to unify them under a general framework. To this end, we define a general form of contrastive loss inspired by the analysis of Tao et al. (2022):

Definition 3.1 (General contrastive loss)**.** Let π_θ be a probabilistic model, $\mathcal{D}$ an arbitrary data distribution, $\boldsymbol{x} \sim \mathcal{D}$ be an anchor, and $\mathcal{D}^+(\cdot \mid \boldsymbol{x})$ and $\mathcal{D}^-(\cdot \mid \boldsymbol{x})$ be the positive and negative distributions conditioned on $\boldsymbol{x}$. We call $\boldsymbol{y}^+ \sim \mathcal{D}^+$ the positive sample and $\boldsymbol{y}^- \sim \mathbb{D}^-$ the negative sample w.r.t. $\boldsymbol{x}$. We say a differentiable loss function $\mathcal{L}$ is *contrastive* if its gradient has the following form:

$$\nabla_\theta \mathcal{L} = -\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}^+,\boldsymbol{y}^-}\left[a(\boldsymbol{x},\boldsymbol{y}^+,\mathcal{D}^-)\nabla_\theta \pi_\theta(\boldsymbol{y}^+|\boldsymbol{x}) - b(\boldsymbol{x},\boldsymbol{y}^-,\mathcal{D}^+)\nabla_\theta \pi_\theta(\boldsymbol{y}^-|\boldsymbol{x})\right] , \tag{7}$$

where a, b are arbitrary coefficient functions that weight positive and negative contributions. In practice we only have access to empirical gradients: given groups $\{\boldsymbol{y}_i^+\}$ and $\{\boldsymbol{y}_j^-\}$ we use empirical coefficients $\hat{a}(\boldsymbol{x},\boldsymbol{y}^+,\{\boldsymbol{y}_j^-\})$ and $\hat{b}(\boldsymbol{x},\boldsymbol{y}^-,\{\boldsymbol{y}_i^+\})$ in place of a, b.

Let $q \sim \mathcal{Q}$ be a prompt sample from $\mathcal{Q}$ and $\{o_i\}_{i=1}^G$ be a group of trajectories drawn i.i.d. from the policy $\pi_\theta(\cdot \mid q)$. Let $G > 1$ denote the group size of GRPO (i.e., the number of rollouts/trajectories generated per prompt). Given the prompt q and the policy π_θ, we let G_q^+ and G_q^- denote the numbers of correct and incorrect trajectories, respectively, in the G sampled trajectories. $\hat{p}_{\theta,q} = G_q^+/G$ is the proportion of correct trajectories in the sampled G trajectories, which approximates the probability of correct $p_{\theta,q}$ given the policy $\pi_{\theta,q}$ on the prompt q. In the following discussion, we drop the subscript θ for simplicity.

In the analysis, we can assume that clipping is not triggered, since the gradient will be zero outside the clipping range. Then we have the following equation for the GRPO objective function:

$$\begin{aligned}
&\mathcal{J}_{\text{GRPO}}(\theta, G) \\
&= \mathop{\mathbb{E}}_{\substack{q\sim\mathcal{Q} \\ o\sim\pi_\theta(\cdot|q)}} \sqrt{\widehat{\text{Var}}_G(q)} \left(\frac{1}{G^+} \sum_{j=1}^{G^+} \frac{1}{|o_j|} \sum_{t=1}^{|o_j|} \pi_\theta(o_{j,t}|o_{j,<t},q) - \frac{1}{G^-} \sum_{k=1}^{G^-} \frac{1}{|o_k|} \sum_{t=1}^{|o_k|} \pi_\theta(o_{k,t}|o_{k,<t},q) \right),
\end{aligned} \tag{8}$$

where $\widehat{\text{Var}}_G(q) = (1-\hat{p}_q)\hat{p}_q$, is the empirical standard deviation of a group of G samples from Bernoulli(p_q), which is the distribution of rewards in the verifiable setting.

Regarding the group of sampled trajectories, the empirical objective equation 8 is approximating the true objective:

$$\begin{aligned}
&\mathcal{J}_{\text{GRPO}}(\theta) \\
&= \mathop{\mathbb{E}}_{q\sim\mathcal{Q}} \sqrt{\text{Var}(q)} \left(\mathop{\mathbb{E}}_{o_j\sim\pi_\theta^+(\cdot|q)} \frac{1}{|o_j|} \sum_{t=1}^{|o_j|} \pi_\theta(o_{j,t}|o_{j,<t},q) - \mathop{\mathbb{E}}_{o_k\sim\pi_\theta^-(\cdot|q)} \frac{1}{|o_k|} \sum_{t=1}^{|o_k|} \pi_\theta(o_{k,t}|o_{k,<t},q) \right).
\end{aligned} \tag{9}$$

where $\text{Var}(q) = (1-p_q)p_q$ is the variance of the Bernoulli(p_q). For simplicity, we use $\pi_\theta^+(\cdot|q)$ and $\pi_\theta^-(\cdot|q)$ to denote the corresponding positive and negative subdistribution, respectively. The detailed derivation is provided in Appendix A.1.

In the theoretical analysis that follows, we center our attention on the true objective equation 9, since the empirical version equation 8 merely serves as an approximation derived from finite samples.

For each prompt q, the GRPO objective equation 9 can be interpreted as an intra-group contrastive loss: it increases the likelihood of positive trajectories while suppressing the likelihood of negative ones. Importantly, each prompt is weighted by the standard deviation of the reward distribution, Bernoulli(p_q), which quantifies the uncertainty of the policy π_θ under that prompt. As a result, equation 9 naturally emphasizes prompts where the policy exhibits higher uncertainty. This observation leads to the following proposition: the GRPO objective is, in essence, a form of contrastive loss.

Proposition 3.2. *The GRPO objective is a contrastive loss.*

Proof of Proposition 3.2. equation 9 has the following derivatives:

$$\nabla_\theta \mathcal{J}_{\text{GRPO}} = \mathop{\mathbb{E}}_{q\sim\mathcal{Q}} \sqrt{\text{Var}(q)} \left(\mathop{\mathbb{E}}_{o_j\sim\pi_\theta^+(\cdot|q)} \nabla_\theta \pi_\theta^{\text{GRPO}}(o_j|q) - \mathop{\mathbb{E}}_{o_k\sim\pi_\theta^-(\cdot|q)} \nabla_\theta \pi_\theta^{\text{GRPO}}(o_k|q) \right), \tag{10}$$

where we denote $\pi_\theta^{\text{GRPO}}(o_i|q) = \frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \pi_\theta(o_{i,t}|o_{i,<t},q)$ (see Appendix A.2 for further discussion). Let $a = b = -\sqrt{\text{Var}(q)}$. Under this choice, the expectation in equation 10 can be factored out of the

parentheses, aligning the expression with the form of equation 7. Therefore, by Definition 3.1, the GRPO objective satisfies the definition of a contrastive loss, which completes the proof. □

Proposition 3.3. *The DPO objective is a contrastive loss.*

Proof of Proposition 3.3. The DPO objective has the following derivatives:

$$\nabla_\theta \mathcal{L}_{\text{DPO}} = -\beta \mathbb{E}_{(q,o^+,o^-)\sim\mathcal{D}_{\text{DPO}}} \left[\sigma'\left(\nabla_\theta \log \pi_\theta(o^+|q) - \nabla_\theta \log \pi_\theta(o^-|q)\right)\right] \tag{11}$$

$$= -\beta \mathbb{E}_{q,o^+,o^-} \left(\frac{\sigma'}{\pi_{\text{ref}}(o^+|q)} \nabla_\theta \pi_\theta(o^+|q) - \frac{\sigma'}{\pi_{\text{ref}}(o^-|q)} \nabla_\theta \pi_\theta(o^-|q)\right), \tag{12}$$

where $\hat{r}_\theta = \beta(x, y) \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ and $\sigma' = \sigma(\hat{r}_\theta(q, o^-) - \hat{r}_\theta(q, o^+))$. Let $a = \frac{\beta\sigma'}{\pi_{\text{ref}}(o^+|q)}$ and $b = \frac{\beta\sigma'}{\pi_{\text{ref}}(o^-|q)}$. equation 11 aligns with the form of equation 7, which indicates the DPO objective is a contrastive loss. □

According to Proposition 3.2 and Proposition 3.3, both the GRPO and DPO objectives can be interpreted as contrastive losses.

4 Rethinking Group Size in GRPO from DPO: Why 2 Is Enough

Building on our contrastive interpretation of GRPO, we are naturally led to 2-GRPO. At first glance, using only two rollouts per prompt may seem insufficient, since one might intuitively expect poor reward normalization and less positive signals. In this section, we analyze it from multiple perspectives and show that large groups are not strictly necessary for effective learning.

In standard GRPO, each training step generates G trajectories per prompt. A reward function partitions these trajectories into positive and negative groups, which are then used to compute gradients for policy updates. Crucially, the generation phase is the dominant computational bottleneck, accounting for up to 70% of total training time (Liu et al., 2025). Reducing the group size G, therefore, offers a direct path to higher training throughput via more frequent updates.

Our gradient analysis in Sec. 3 shows that GRPO estimates expectations over positive and negative trajectories, which closely mirrors the formulation of DPO, where only a single positive–negative pair is used. This connection raises a natural question: if DPO succeeds with just one pair, could GRPO perform well with a minimal group size?

To push the limit, we introduce 2-GRPO, i.e., GRPO with group size $G = 2$, which has unbiased gradient estimation and is substantially more efficient:

$$\mathcal{J}_{\text{2-GRPO}} = \mathbb{E}_q \mathbb{E}_{o^+} \mathbb{E}_{o^-} \frac{1}{2} \left(\pi_\theta^{\text{GRPO}}(o^+|q) - \pi_\theta^{\text{GRPO}}(o^-|q)\right) . \tag{13}$$

This expression is obtained by replacing $\sqrt{\text{Var}(q)}$ in equation 9 with the constant $1/2$.

4.1 Advantage Estimate

The first concern lies in the spurious lack of advantage to stabilize rewards. In 2-GRPO, the advantage computation is straightforward: $A^+ = 1, A^- = -1$ for a positive-negative pair and $A^+ = A^- = 0$ otherwise. It seems 2-GRPO simply shifts the reward from $0/1$ to $-1/1$ and lacks any normalization effect. However, the following proposition exposes that 2-GRPO does implicitly perform the normalization.

Proposition 4.1. *Given a constant $p \in (0, 1)$ and a small positive constant ϵ, we consider two scenarios below:*

- ***Case 1**: Consider $X_1, \cdots, X_{2N} \overset{\text{i.i.d.}}{\sim}$ Bernoulli(p). Let $Y_i = \frac{X_i - \hat{\mu}}{\hat{\sigma} + \epsilon}$, where $\hat{\mu} = \frac{1}{2N} \sum_{i=1}^{2N} X_i$ and $\hat{\sigma} = \sqrt{\frac{1}{2N} \sum_{i=1}^{2N} (X_i - \hat{\mu})^2}$. Then, it follows that*

$$\lim_{\epsilon \to 0} \lim_{N \to \infty} \mathbb{E}[Y_i | X_i = x] = \frac{x - p}{\sqrt{p(1-p)}}. \tag{14}$$

- ***Case 2**: Consider N pairs of $(X_{i,1}, X_{i,2})$ with each $X_{i,j} \overset{i.i.d.}{\sim}$ Bernoulli(p). Let $Y_{i,j} = \frac{X_{i,j} - \hat{\mu}_i}{\hat{\sigma}_i + \epsilon}$, where $\hat{\mu}_i = \frac{1}{2}(X_{i,1} + X_{i,2})$ and $\hat{\sigma}_i = \sqrt{\frac{1}{2}\sum_{j=1}^{2}(X_{i,j} - \hat{\mu}_i)^2}$. Then, it follows that*

$$\lim_{\epsilon \to 0} \mathbb{E}[Y_{i,j} | X_{i,j} = x] = x - p. \tag{15}$$

The $\lim_{\epsilon \to 0} \mathbb{E}[Y_{i,j} | X_{i,j} = x]$ differs from $\lim_{\epsilon \to 0} \lim_{N \to \infty} \mathbb{E}[Y_i | X_i = x]$ by a scaling factor $\frac{1}{\sqrt{p(1-p)}}$.

In Proposition 4.1, **Case 1** corresponds to regular GRPO with sufficiently large group size; in this case $\mathbb{E}[Y_i | X_i = 1]$ and $\mathbb{E}[Y_i | X_i = 0]$ are, respectively, the advantage estimates of positive and negative trajectories given a prompt. A large G will lead to a smaller bias, and thus better estimation of the advantages. **Case 2** corresponds to 2-GRPO, where $\mathbb{E}[Y_{i,j} | X_{i,j} = 1]$ and $\mathbb{E}[Y_{i,j} | X_{i,j} = 0]$ are unbiased advantage estimates. These advantage estimates differ from the ones of regular GRPO merely by a scaling factor. This indicates that, even though the possible advantage values in 2-GRPO are only $-1, 0, 1$, the advantage estimates are still proportional to p across training steps, suggesting the rationale behind the optimization. The proof is in Appendix A.3.

4.2 Gradient Estimate

A second concern is that decreasing the group size increases gradient variance. We first provide a formal definition of gradient variance, followed by a lemma for empirical gradient estimation.

Definition 4.2 (Gradient Variance)**.** Let $\{\boldsymbol{x}_i\}_{i=1}^{B}$ be a training batch of size B, where $\boldsymbol{x}_i$ are sampled from the same distribution $\mathcal{D}$, and let $g_i = \nabla_\theta L_\theta(\boldsymbol{x}_i)$ denote the gradient of $L_\theta(\boldsymbol{x}_i)$ w.r.t. θ. Define the empirical batch gradient $\hat{g}(\xi_B) = \frac{1}{B}\sum_{i=1}^{B} g_i$, where ξ_B denote the randomness from sampling B samples from the distribution and the expectation of gradient $\bar{g} = \mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{D}}[\nabla g_i]$. The variance of the gradient estimate over the batch is then defined as:

$$\mathrm{Var}(\hat{g}) = \mathbb{E}_\xi(\hat{g} - \bar{g})^2.$$

Lemma 4.3. *Let $\{\boldsymbol{x}_i\}_{i=1}^{B_1}, \{\boldsymbol{x}_i\}_{i=1}^{B_2}$ be two training batches of batch size B_1 and B_2, respectively. Assume all data are i.i.d. sampled from the same distribution and the gradient of each data point has the same variance σ_g. Let $\hat{g}_{B_1}, \hat{g}_{B_2}$ denote the average of gradient of batch B_1 and B_2, respectively. If $B_1 < B_2$, then $\mathrm{Var}[\hat{g}_{B_1}] > \mathrm{Var}[\hat{g}_{B_1}]$.*

Proof of Lemma 4.3.

$$\mathrm{Var}(\hat{g}_B) = \mathrm{Var}\left(\frac{1}{B}\sum_{i}^{B} g_i\right) = \frac{1}{B^2}\left(\sum_{i}^{B}\mathrm{Var}(g_i)\right) = \frac{\sigma_g^2}{B}\ , \tag{16}$$

where the second equation is obtained by the fact that the covariance between i.i.d. data is zero. By the above equation, increasing B decreases Var. □

At first glance, decreasing the group size in equation 10 seems to increase the variance of the gradient by Lemma 4.3. However, we have omitted the fact that the actual gradient calculation is obtained across different prompts. The actual calculation is described by the empirical GRPO objective:

$$\widehat{\mathcal{J}}_{\text{GRPO}}(\theta, G, Q) = \frac{1}{QG}\sum_{j=1}^{Q}\sum_{i=1}^{G} A_{ij}\pi_\theta^{\text{GRPO}}(o_{ij}|q_j), \tag{17}$$

where Q is the number of prompts in the mini-batch, and the batch size of training is $B = QG$ rollouts. When we decrease G, we can increase Q to compensate. Since the total number of questions in the dataset is fixed, increasing Q will not affect the overall computational burden.

Note that we are not arguing that we must pursue low variance or that high variance must necessarily lead to poor training outcomes. In fact, there are works showing that moderate variance can benefit the model generalization (Zhou et al., 2020). Therefore, the goal here is to use a reduced group size to improve efficiency while controlling variance by adjusting Q.

4.3 Exploration on Hard Questions

Another common concern with using a small group size (e.g., $G = 2$) is that it may perform poorly on difficult prompts, where multiple attempts are often needed to produce a correct answer. The intuition is that a smaller group provides fewer opportunities to sample a correct response within a single batch, potentially slowing down learning.

However, under a fixed computational budget – where the dominant cost is rollout generation – 2-GRPO and 16-GRPO explore approximately the same total number of rollouts across all training epochs. Consequently, the overall probability of sampling a correct answer under $G = 2$ is comparable to that under $G = 16$.

Proposition 4.4. *Let $p_i \in [0, 1]$ denote the probability that a single rollout under the policy π_i produces a correct answer. Then:*

1. *The probability of obtaining at least one correct answer in $2m$ independent rollouts with policy π_0 is*
$$P_{2m} = 1 - (1 - p_0)^{2m}. \tag{18}$$

2. *The probability of obtaining at least one correct answer when performing m consecutive trials of 2 independent rollouts each, with the corresponding policy $[\pi_0, \pi_1, \cdots, \pi_{m-1}]$ is*
$$P_{m\times 2} = 1 - \prod_{i=0,\cdots m-1} (1 - p_i)^2 \geq 1 - (1 - p_0)^{2m} = P_{2m} \tag{19}$$
when we have $p_i \geq p_0, \forall i > 0$.

Note that the assumption $p_i \geq p_0, \forall i > 0$ is prevailing, as we assume that the reasoning ability of LLM can be improved by RL post-training.

Proposition 4.4 indicates that for hard questions, 2-GRPO will not breakdown compared to 16-GRPO, given the same total rollouts traversed. It is worth mentioning that, due to its greater number of policy updates, 2-GRPO may have a higher probability of getting a correct output for a difficult question and is more adaptive to capture more nuanced update requirements for different questions.

5 Experiments

5.1 Experiment Details

Tasks and Training Framework Following prior studies, we consider mathematical tasks as representative instances of RLVR to verify our hypothesis, given their demonstrated transferability to a broad range of other tasks (Yu et al., 2025). For training, we adopt the *verl* framework (Sheng et al., 2025) and utilize the built-in implementation of GRPO (Shao et al., 2024) as the baseline algorithm.

Dataset and Baselines Following prior work (Chu et al., 2025), we employ Qwen-2.5-Math-1.5B (Qwen-1.5B) and Qwen-2.5-Math-7B (Qwen-7B) (Yang et al., 2025) as base models. Both models are post-trained via RL on the MATH (Hendrycks et al., 2021a) and DAPO-Math-17k (Yu et al., 2025) datasets, and evaluated on MATH-500 (Hendrycks et al., 2021b), AMC23, Minerva Math (Lewkowycz et al., 2022), AIME-2025, and OlympiadBench (Huang et al., 2024). For DAPO-Math-17k dataset, we randomly sample 7.5k questions from the original data to form a subset for training in order to align with the size of MATH. In addition, we assess the proposed method on DeepSeek-R1-Distill-Qwen-1.5B (DS-1.5B) (DeepSeek-AI, 2025), which is post-trained on MATH. Owing to computational constraints, we do not extend its post-training to DAPO-Math-17k. All 1.5B models are trained on 4 GPUs. Qwen-7B is trained on 8 GPUs. We evaluate model performance using two metrics: Mean@32, the average accuracy across 32 i.i.d. samples, and Pass@32, which measures whether a problem is solved in at least one of those 32 attempts.

Hyper-parameters We mainly follow the default configuration of the *verl* framework. For sampling parameters in training generation, we set temperature to 1, top-p to 1 to encourage exploration, sequence length to 4096 for Qwen-series model and 8192 for DS-1.5B. For sampling parameters in test generation, we set temperature to 0.7, top-p to 0.8, top-k to 20 and sequence length to 4096 for

all models. For optimization, training employs the Adam optimizer (Kingma, 2014) with a constant learning rate and a linear warm-up over the first 10 steps. For GRPO hyper-parameters, All models are trained for 10 epochs. The baseline method, 16-GRPO, is trained with batch sizes of 32 (32 prompts and 16 rollouts per prompt) and a learning rate 1×10^{-6}. As discussed in Sec. 4.2, we trained 2-GRPO with a larger batch size of 256 (256 prompts and 2 rollouts per prompt). Both case will have 512 rollouts in each mini-batch of training. Since we have fewer update steps due to the larger batch size, we adjust the learning rate of 2-GRPO to 8×10^{-6} based on the linear relationship of learning rate and batch size (Goyal et al., 2017).

Goal of Experiment Building on the theoretical justification for 2-GRPO, we seek to empirically assess its validity in RLVR. We anticipated that ***2-GRPO will exhibit better efficiency***—with respect to computational resources and/or wall-clock time—while maintaining the same performance as regular GRPO (16-GRPO).

5.2 Main Experiments

Table 1: 2-GRPO v.s. 16-GRPO: post-trained on MATH/DAPO-Math-Sub and evaluated on five mathematical reasoning benchmarks. M/P@32 stands for Mean@32 and Pass@32. G is the group size. Δ denotes the difference $16 \rightarrow 2$.

M/P@32 ↑	G	Time (h) ↓	MATH-500	AMC 2023	Minerva Math	AIME 2025	Olympiad Bench
Post-training on MATH dataset							
Qwen-1.5B	w/o	-	31.83 / 81.92	34.30 / 79.23	5.33 / 28.91	3.64 / 22.31	15.40 / 37.16
	2	2.05	69.28 / 87.43	49.53 / 81.76	16.25 / 33.26	9.48 / 32.88	22.31 / 37.24
	16	8.53	70.24 / 87.24	51.25 / 83.46	16.84 / 33.46	10.10 / 35.82	23.11 / 37.82
	Δ	-75.96%	-0.96 / +0.19	-1.71 / -1.70	-0.59 / -0.19	-0.62 / -2.94	-0.80 / -0.58
Qwen-7B	w/o	-	47.16 / 85.95	38.36 / 85.29	5.99 / 31.10	5.00 / 25.17	9.83 / 34.30
	2	2.43	75.23 / 89.77	64.60 / 81.53	23.13 / 38.45	12.81 / 38.85	26.39 / 40.20
	16	9.30	75.90 / 88.24	61.79 / 80.77	22.81 / 37.68	13.23 / 34.22	25.99 / 40.11
	Δ	-73.87%	-0.67 / +1.53	+2.81 / +0.76	+0.32 / +0.77	-0.42 / +4.63	+0.40 / 0.09
DS-1.5B	w/o	-	65.11 / 84.90	44.14 / 73.86	14.64 / 32.80	22.40 / 42.79	20.07 / 33.23
	2	7.07	74.36 / 88.85	56.95 / 88.63	21.28 / 38.34	24.89 / 46.79	33.69 / 45.86
	16	38.40	75.98 / 89.16	58.91 / 87.26	21.76 / 38.29	26.97 / 56.36	35.39 / 47.05
	Δ	-81.6%	-1.62 / -0.31	-1.96 / +1.38	-0.48 / -0.05	-2.08 / -9.56	-1.70 / -1.19
Post-training on DAPO-Math-Sub dataset							
Qwen-1.5B	w/o	-	31.83 / 81.92	34.30 / 79.23	5.33 / 28.91	3.64 / 22.31	15.40 / 37.16
	2	2.12	68.81 / 87.36	52.19 / 85.77	16.79 / 33/61	8.13 / 29.33	23.52 / 39.29
	16	13.30	70.66 / 87.04	56.56 / 85.54	18.00 / 34.16	9.58 / 32.31	24.56 / 39.19
	Δ	-84.06%	-1.85 / +0.32	-4.37 / +0.23	-1.21 / +0.71	-2.50 / -2.98	-1.04 / +0.10
Qwen-7B	w/o	-	47.16 / 85.95	38.36 / 85.29	5.99 / 31.10	5.00 / 25.17	9.83 / 34.30
	2	3.63	77.43 / 90.51	64.84 / 91.59	21.95 / 38.05	14.58 / 33.03	29.86 / 45.24
	16	17.68	77.35 / 88.79	69.69 / 87.31	24.45 / 40.04	14.27 / 33.73	28.86 / 39.84
	Δ	-79.47%	+0.08 / 1.72	-4.85 / +4.28	-2.50 / -1.99	+0.31 / -0.70	+1.00 / +5.4

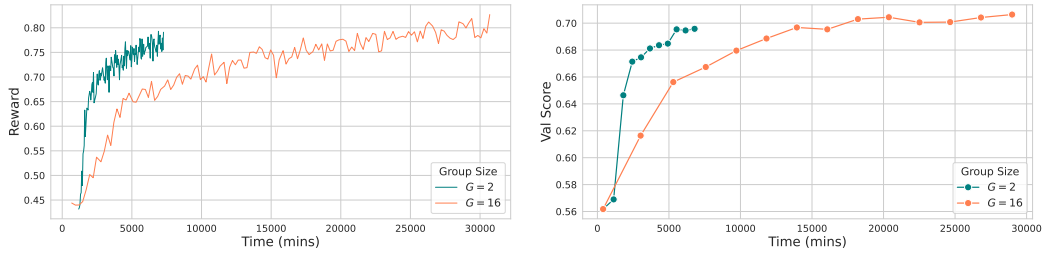
As shown in Table 1, 2-GRPO requires at least 70% less wall-clock time than 16-GRPO while achieving comparable performance. The models are post-trained on the MATH and DAPO-Math-Sub datasets and evaluated on five widely-used mathematical reasoning benchmarks, representing an out-of-distribution evaluation. This setting imposes stringent requirements on the generalization ability of the post-trained models. Notably, 2-GRPO is optimized with only 0.15 million generated rollouts — just 12.5% of the 1.2 million rollouts utilized by 16-GRPO. [3] These results provide strong corroboration of our theoretical finding that reducing group size preserves performance while substantially improving efficiency. To further support this statement, we conduct ablation study on various k-GRPO ($k = 4, 8$) in Appendix B.2.

[3] Appendix B.1 discusses the relationship between the total number of rollouts and computational cost.

5.3 Visualization

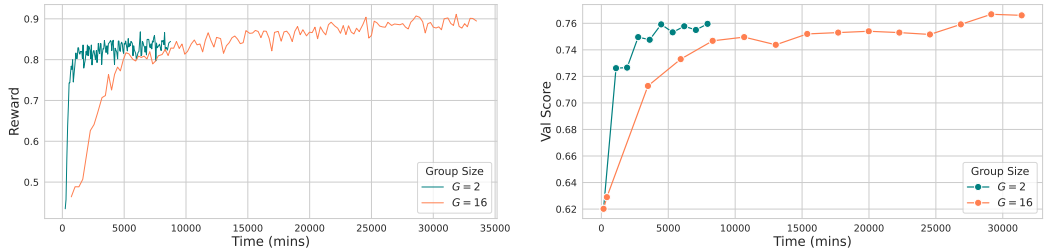
In Sec. 5.2, we present empirical results comparing 2-GRPO and 16-GRPO. However, the out-of-distribution evaluation setting may not fully reflect the post-training with 2-GRPO, as the distribution shift could obscure the underlying performance differences. Therefore, in this section, we visualize the reward and evaluation scores on the MATH dataset to demonstrate the in-distribution generalization of the post-trained models using 2-GRPO in comparison to 16-GRPO. [4]

The figures presented in Fig. 1 and Fig. 2 illustrate the performance of Qwen-2.5-Math-1.5B and Qwen-2.5-Math-7B, respectively. As depicted, the reward and evaluation scores for 2-GRPO are comparable to those of 16-GRPO, indicating that the in-distribution generalization of the post-trained models using 2-GRPO is on par with that of 16-GRPO.



(a) Curve of rewards. (b) Curve of eval-scores.

Figure 1: Qwen-1.5B: Visualization of reward and evaluation scores on the MATH dataset.



(a) Curve of rewards. (b) Curve of eval-scores.

Figure 2: Qwen-7B: Visualization of reward and evaluation scores on the MATH dataset.

6 Discussion

Stronger Efficiency There remains potential for further enhancements of 2-GRPO in efficiency. In 2-GRPO, many rollouts generated are ultimately assigned zero advantage, which actually do not demand the computation of gradients. Consequently, a more advanced implementation could optimize these computations during the training phase. It is important to note that, as discussed in Sec. 4.1, these zero-advantage rollouts are still necessary for accurate advantage estimation. Therefore, we must simulate the contributions of these zero-advantage rollouts during the training phase rather than simply discarding them after the inference phase.

2-GRPO is a Quantization of GRPO An alternative perspective on 2-GRPO is that it serves as a quantization of standard GRPO, wherein the candidate values for advantages are discretized to $-1, 0, 1$. Nevertheless, due to the stochastic nature of neural network optimization, 2-GRPO is capable of approximating continuous advantage values effectively, provided that a sufficiently large number of training steps are employed.

Data Efficiency The quantized nature of 2-GRPO inherently results in the rejection of a number of generated rollouts. While this characteristic enhances computational efficiency, it may concurrently compromise data efficiency – numerous rollouts are discarded when the policy exhibits either exceptionally poor or exceptionally strong performance. This limitation in data efficiency could

[4] The DAPO dataset does not provide a test set.

impede the ability of the policy post-trained by 2-GRPO to attain near-optimal performance. This observation motivates the design of adaptive adjustments to the group size, aiming to strike a balance between computational and data efficiency, where we leave this direction to future exploration.

Conclusion In this work, we present a theoretical analysis of GRPO from a contrastive learning perspective, establishing a key conceptual connection between GRPO and DPO and offering a new lens for understanding GRPO. Building on this insight, we propose 2-GRPO, a DPO-inspired variant with a group size of two, which achieves significant efficiency gains while maintaining comparable performance.

References

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Chu, X., Huang, H., Zhang, X., Wei, F., and Wang, Y. (2025). Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*.

DeepSeek-AI (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Frosst, N., Papernot, N., and Hinton, G. (2019). Analyzing and improving representations with the soft nearest neighbor loss. In *International conference on machine learning*, pages 2012–2020. PMLR.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

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

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

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

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

Huang, Z., Wang, Z., Xia, S., Li, X., Zou, H., Xu, R., Fan, R.-Z., Ye, L., Chern, E., Ye, Y., et al. (2024). Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai. In *Advances in Neural Information Processing Systems*.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Liu, L., Yao, F., Zhang, D., Dong, C., Shang, J., and Gao, J. (2025). Flashrl: 8bit rollouts, full power rl.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Pang, L. and Jin, R. (2025). On the theory and practice of grpo: A trajectory-corrected approach with fast convergence. *arXiv preprint arXiv:2508.02833*.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

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

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

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. (2025). Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Tao, C., Wang, H., Zhu, X., Dong, J., Song, S., Huang, G., and Dai, J. (2022). Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14431–14440.

Wang, T. and Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Wu, Y., Zhang, L., Mo, F., Zhu, T., Ma, W., and Nie, J.-Y. (2024). Unifying graph convolution and contrastive learning in collaborative filtering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3425–3436.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. (2025). Qwen2.5 Technical Report.

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

Zhao, Y., Liu, Y., Liu, J., Chen, J., Wu, X., Hao, Y., Lv, T., Huang, S., Cui, L., Ye, Q., et al. (2025). Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*.

Zheng, C., Liu, S., Li, M., Chen, X.-H., Yu, B., Gao, C., Dang, K., Liu, Y., Men, R., Yang, A., et al. (2025). Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Zhou, P., Feng, J., Ma, C., Xiong, C., Hoi, S. C. H., et al. (2020). Towards theoretically understanding why sgd generalizes better than adam in deep learning. *Advances in Neural Information Processing Systems*, 33:21285–21296.

Appendix

A Theorems

A.1 Reveal GRPO as Contrastive

Details of Sec. 3. In the RLVR setting, rewards are binary, which leads to binary advantages given a prompt. Let A_q^+, A_q^- denote the positive and negative advantage, respectively. From equation 6, we can have

$$\begin{aligned} A_q^+ &= \frac{1-\hat{p}_q}{\sqrt{\hat{p}_q(1-\hat{p}_q)}} = \sqrt{\frac{1-\hat{p}_q}{\hat{p}_q}}\ , \\ A_q^- &= \frac{0-\hat{p}_q}{\sqrt{\hat{p}_q(1-\hat{p}_q)}} = -\sqrt{\frac{\hat{p}_q}{1-\hat{p}_q}}\ . \end{aligned} \tag{20}$$

In equation 5, the clipping function can be considered as applying an indicator function to the token, which does not affect trajectory-level behavior. The omission of the clipping function does not affect the analysis, as the out of range will lead to zero gradient.

The key derivation is as follows:

$$\begin{aligned} &\mathcal{J}_{\text{GRPO}}(\theta) \\ &= \mathbb{E}_{\substack{q\sim\mathcal{Q} \\ o_i\sim\pi_{\theta_{\text{old}}}(\cdot|q)}} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\frac{\pi_\theta(o_{i,t}|o_{i,<t},q)}{\pi_{\theta_{\text{old}}}(o_{i,t}|o_{i,<t},q)}A_{i,t}\ , \\ &= \mathbb{E}_{\substack{q\sim\mathcal{Q} \\ o_i\sim\pi_\theta(\cdot|q)}} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\pi_\theta(o_{i,t}|o_{i,<t},q)A_{i,t}\ , \\ &= \mathbb{E}_{\substack{q\sim\mathcal{Q} \\ o_j\sim\pi_\theta^+(\cdot|q) \\ o_k\sim\pi_\theta^-(\cdot|q)}} \frac{1}{G}\left(\sum_{j=1}^{G^+}\frac{1}{|o_j|}\sum_{t=1}^{|o_j|}A_j^+\pi_\theta(o_{j,t}|o_{j,<t},q) + \sum_{k=1}^{G^-}\frac{1}{|o_k|}\sum_{t=1}^{|o_k|}A_k^-\pi_\theta(o_{k,t}|o_{k,<t},q)\right)\ , \\ &= \mathbb{E}_{\substack{q\sim\mathcal{Q} \\ o_j\sim\pi_\theta^+(\cdot|q) \\ o_k\sim\pi_\theta^-(\cdot|q)}} A_q^+\frac{G^+}{G}\frac{1}{G^+}\sum_{j=1}^{G^+}\frac{1}{|o_j|}\sum_{t=1}^{|o_j|}\pi_\theta(o_{j,t}|o_{j,<t},q) + A_q^-\frac{G^-}{G}\frac{1}{G^-}\sum_{k=1}^{G^-}\frac{1}{|o_k|}\sum_{t=1}^{|o_k|}\pi_\theta(o_{k,t}|o_{k,<t},q)\ , \\ &= \mathbb{E}_{\substack{q\sim\mathcal{Q} \\ o_j\sim\pi_\theta^+(\cdot|q) \\ o_k\sim\pi_\theta^-(\cdot|q)}} \sqrt{\widehat{\text{Var}}_G(q)}\left(\frac{1}{G^+}\sum_{j=1}^{G^+}\frac{1}{|o_j|}\sum_{t=1}^{|o_j|}\pi_\theta(o_{j,t}|o_{j,<t},q) - \frac{1}{G^-}\sum_{k=1}^{G^-}\frac{1}{|o_k|}\sum_{t=1}^{|o_k|}\pi_\theta(o_{k,t}|o_{k,<t},q)\right)\ . \end{aligned} \tag{21}$$

The first equation is obtained by omitting the clipping function. The second equation is obtained by the fact of important sampling that $\mathbb{E}_q[\frac{p(x)}{q(x)}f(x)] = \mathbb{E}_p[f(x)]$. The third equation is obtained by dividing the trajectories into two groups: positive and negative. The fourth equation is obtained by the fact that all positive advantages are the same and that all negative advantages are the same. Since $A^+\frac{G^+}{G} = \sqrt{\frac{1-\hat{p}}{\hat{p}}}\hat{p} = \sqrt{(1-\hat{p})\hat{p}}$ and $A^-\frac{G^-}{G} = -\sqrt{(1-\hat{p})\hat{p}}$, we obtain equation 8. When $G\to\infty$,

we have the following facts:

$$\begin{aligned}
&\lim_{G\to\infty} G^+ = \infty \,, \\
&\lim_{G\to\infty} G^- = \infty \\
&\lim_{G\to\infty} \sqrt{(1-\hat{p})\hat{p}} = \sqrt{(1-p)p} \,, \\
&\lim_{G^+\to\infty} \frac{1}{G^+} \sum_{j=1}^{G^+} f(o_j) = \mathbb{E}_{o_j\sim O_\theta^+} f(o_j) \,, \\
&\lim_{G^-\to\infty} \frac{1}{G^-} \sum_{k=1}^{G^-} f(o_k) = \mathbb{E}_{o_k\sim O_\theta^-} f(o_k) \,.
\end{aligned} \tag{22}$$

Based on the above facts, it is easy to derive equation 9. □

A.2 Justification of Proposition 3.2

Most of autoregressive LLMs adopt causal probability modelling that $\sum_t \log \pi_\theta(o_t|o_{<t}, q) = \log \pi_\theta(o|q)$. Then we have the following equation to describe the gradient of trajectory probability and the gradient of token probabilities:

$$\nabla_\theta \pi_\theta(o|q) = \pi_\theta(o|q) \sum_t \frac{1}{\pi_\theta(o_t|o_{<t}, q)} \pi_\theta(o_t|o_{<t}, q)$$

However, the original GRPO objective does not hold this property. Or one can consider GRPO using the mean-field assumption for probability modelling. Some papers believe that GRPO should be corrected by sequence level importance sampling (Zheng et al., 2025; Zhao et al., 2025; Pang and Jin, 2025). It is still an open question for the choice of important sampling for GRPO. To avoid overhead discussion, we keep the assumption implicit adopted by the original GRPO and denote $\pi_\theta^{\text{GRPO}} = \sum_t \pi_\theta(o_t|o_{<t}, q)$.

A.3 Proof of Proposition 4.1

Proof. **Case 1.** Notice that $\hat{\sigma} = \sqrt{\frac{1}{2N}\sum_{k=1}^{2N}(X_k - \hat{\mu})^2} = \sqrt{\hat{\mu}(1-\hat{\mu})}$ and $\hat{\mu} = \frac{1}{2N}\sum_{k=1}^{2N} X_k$. Fix an index i and condition on the event $\{X_i = x\}$ with $x \in \{0, 1\}$. In this case, by the strong law of large numbers and the continuous mapping theorem, we have $\hat{\mu} \overset{a.s.}{\to} p$ and $\hat{\sigma} \overset{a.s.}{\to} \sqrt{p(1-p)}$. Thus, it follows that

$$\lim_{\epsilon\to 0} \lim_{N\to\infty} \mathbb{E}[Y_i \mid X_i = x] = \frac{x-p}{\sqrt{p(1-p)}}.$$

Case 2. When $X_{i,1} = X_{i,2}$, we have $X_{i,j} = \hat{\mu}_i$ and $Y_{i,j} = 0$ for any $j \in \{1, 2\}$. When $X_{i,1} \neq X_{i,2}$, we have $\hat{\mu}_i = 0.5$, $\hat{\sigma}_i = 0.5$, and $Y_{i,j} = \frac{2X_{i,j}-1}{1+2\epsilon}$. By the law of total expectation, it follows that

$$\mathbb{E}\left[Y_{i,j} \mid X_{i,j} = 1\right] = \frac{1-p}{1+2\epsilon}, \qquad \mathbb{E}\left[Y_{i,j} \mid X_{i,j} = 0\right] = \frac{-p}{1+2\epsilon}.$$

Thus, we have

$$\lim_{\epsilon\to 0} \mathbb{E}[Y_{i,j} \mid X_{i,j} = x] = x - p.$$

□

B Experiments

B.1 The Connection Between Training Rollouts and Computational Cost

In Sec. 5.2, the total number of rollouts generated and utilized during training is adopted as a metric for comparing the computational cost of different methods.

The rationale for this choice is as follows. A principled measure of computational cost in the context of RL post-training is the number of floating-point operations (FLOPs) performed. Unlike wall-clock time, which is susceptible to variations arising from software implementation details (e.g., optimization of training libraries) and hardware characteristics (e.g., GPU/CPU architecture, I/O throughput), FLOPs provide a more direct and stable measure of computational effort.

For a fixed base model and the same type of RL algorithm (GRPO in our case), the FLOPs required for a single forward or backward pass with one input prompt can be considered constant, for both the generation and training phases. Accordingly, the total number of rollouts executed during training is directly proportional to the FLOPs executed, thereby serving as a theoretically justified and consistent proxy for computational cost.

B.2 Ablation Study on the Group Size

We conducted an ablation study on the effect of group size. In this experiment, the batch size was fixed at 32 and the learning rate at 1×10^{-6}, following the configuration of the standard GRPO (16-GRPO). [5] Only the group size was varied in order to isolate and evaluate its impact.

Table 2: Ablation study on group size G: post-trained on MATH and DAPO, respectively, and evaluated on five mathematical reasoning benchmarks. M/P@32 stands for Mean@32 and Pass@32.

M/P@32 ↑	G	Time (h) ↓	MATH-500	AMC 2023	Minerva Math	AIME 2025	Olympiad Bench
Post-training on MATH dataset							
Qwen-1.5B	w/o	-	31.83 / 81.92	34.30 / 79.23	5.33 / 28.91	3.64 / 22.31	15.40 / 37.16
	2	2.05	67.73 / 87.85	53.28 / 86.21	14.15 / 34.02	6.15 / 29.54	23.11 / 37.82
	4	2.78	69.05 / 87.49	52.50 / 92.01	15.29 / 33.57	8.33 / 27.13	23.08 / 38.99
	8	4.67	69.34 / 86.05	51.64 / 83.96	14.60 / 32.63	7.18 / 32.24	22.77 / 36.69
	16	8.53	70.24 / 87.24	51.25 / 83.46	16.84 / 33.46	10.10 / 35.82	22.30 / 38.33
Qwen-7B	w/o	-	47.16 / 85.95	38.36 / 85.29	5.99 / 31.10	5.00 / 25.17	9.83 / 34.30
	2	2.43	74.41 / 89.25	63.83 / 89.58	21.53 / 37.72	11.67 / 33.05	26.04 / 41.34
	4	3.48	76.24 / 88.16	63.51 / 84.97	23.09 / 41.03	10.83 / 32.42	26.25 / 40.78
	8	5.48	75.12 / 89.53	64.38 / 88.63	22.24 / 35.94	12.71 / 35.85	26.25 / 40.52
	16	9.30	75.90 / 88.24	61.79 / 80.77	22.81 / 37.68	13.23 / 34.22	25.99 / 40.11
Post-training on DAPO-Math-Sub dataset							
Qwen-1.5B	w/o	-	31.83 / 81.92	34.30 / 79.23	5.33 / 28.91	3.64 / 22.31	15.40 / 37.16
	2	3.63	67.71 / 87.68	53.82 / 88.35	16.85 / 34.83	8.12 / 32.99	23.21 / 39.26
	4	4.90	69.14 / 87.78	54.69 / 86.88	17.53 / 35.74	8.43 / 36.18	23.30 / 39.00
	8	8.62	70.25 / 86.84	57.57 / 81.19	17.80 / 35.08	8.54 / 29.42	24.23 / 39.95
	16	13.30	70.66 / 87.03	56.56 / 85.53	18.00 / 34.16	9.58 / 32.31	24.55 / 39.19
Qwen-7B	w/o	-	47.16 / 85.95	38.36 / 85.29	5.99 / 31.10	5.00 / 25.17	9.83 / 34.30
	2	3.43	74.41 / 89.25	63.83 / 89.58	21.53 / 37.72	11.67 / 33.05	26.04 / 41.34
	4	5.39	76.24 / 88.16	63.51 / 84.97	23.09 / 41.03	10.83 / 32.42	26.25 / 40.78
	8	9.18	75.12 / 89.53	64.38 / 88.63	22.24 / 35.94	12.71 / 35.85	26.25 / 40.52
	16	17.68	75.90 / 88.24	61.79 / 80.77	22.81 / 37.68	13.23 / 34.22	25.99 / 40.11

C The Use of Large Language Models (LLMs)

We used LLMs to polish the writing.

[5] It is worth noting that the batch size and learning rate used for 2-GRPO in this ablation differ from those employed in the main experiment.