WHY GRPO NEEDS NORMALIZATION: A LOCAL-CURVATURE PERSPECTIVE ON ADAPTIVE GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has become a key driver of language model reasoning. Among RL algorithms, Group Relative Policy Optimization (GRPO) is the de facto standard, avoiding the need for a critic by using per-prompt baselines and variance normalization. Yet why and when this normalization helps remains unclear. In this work, we answer both questions through the lens of local curvature of the sequence-level policy gradient: standard deviation normalization implements an adaptive gradient. Theoretically, we prove that under mild conditions GRPO attains an improved convergence rate over unnormalized REINFORCE, with gains characterized by the average within-prompt reward standard deviation across prompts and iterations. We further introduce IS-GRPO, an importance-sampling variant of GRPO whose expected update remains aligned with the full gradient, and prove a convergence guarantee of the same form as REINFORCE's, which is tighter in practice during real training runs by a factor we measure. Empirically, on GSM8K and MATH we validate the curvature–variance link and measure the quantities that govern our bounds along real training runs. At 1.5B scale, per-prompt normalization outperforms both unnormalized and globally normalized baselines, with gains that emerge in a middle phase of training where per-prompt variances become heterogeneous. At 7B scale, the normalization schemes become statistically indistinguishable while IS-GRPO retains a small lead, in line with the theory's prediction that the gains require heterogeneity.

1 INTRODUCTION

Large language models (LLMs) have recently exhibited striking gains in multi-step reasoning, particularly when a lightweight reinforcement-learning (RL) stage is applied on top of a strong and pretrained base model. Among the many post-training recipes, Group Relative Policy Optimization (GRPO) has emerged as a practical, critic-free alternative that has powered some of the most visible reasoning systems (Shao et al., 2024), where it consistently improves solution accuracy under tight compute budgets.

While Proximal Policy Optimization (PPO) (Schulman et al., 2017) remains a popular default in RLHF pipelines, it couples the policy with a learned value-function critic (and often GAE (Schulman et al., 2015)). This increases memory footprint and implementation complexity, since the environment is a pretrained LLM and rewards arrive only at the end of whole sequences. This has renewed interest in critic-free policy-gradient methods that operate at the sequence level, such as REINFORCE-style methods, including ReMax (Li et al., 2023) and RLOO (Ahmadian et al., 2024), which often match or outperform PPO for LLM alignment while being simpler and lighter-weight.

Classical REINFORCE (Williams, 1992) reduces gradient variance by subtracting a baseline (e.g., a running mean reward), yielding an unbiased estimator with lower variance (Greensmith et al., 2004). This is textbook policy-gradient variance reduction. GRPO goes one step further: for each prompt, it samples multiple responses from the current policy, computes the group-mean reward as a baseline, and normalizes each response's update by the within-prompt reward standard deviation, which effectively uses a per-prompt z-scored advantage (Shao et al., 2024). Empirically, this simple normalization has been repeatedly observed to stabilize optimization and improve sample-efficiency in LLM RL (Lambert, 2025). Yet, the underlying mechanism of this normalization step has not been theoretically clarified. A natural question arises:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Why and when does GRPO benefit from normalization?

In this work, we give a principled answer to this research question. Our key insight is that the reward variance of each question serves as an estimate of the local Lipschitz constant (curvature) of the policy gradient. Standard-deviation normalization therefore implements an adaptive gradient rule, scaling each update by an inverse-curvature proxy and assigning a larger step size to smoother prompts and a smaller step size to sharper prompts. This mechanism improves stability and convergence precisely when (i) curvature is heterogeneous across prompts and evolves over training, and (ii) cross-prompt interference is controlled so that prompt-wise adaptive steps are not overwhelmed by interaction noise.

Moreover, we show that the benefit of normalization is not tied to condition (ii): we introduce IS-GRPO, an importance-sampling variant of GRPO that selects each question in proportion to its current reward standard deviation while keeping the update positively aligned with the full gradient. We prove a strictly tighter overall bound under an explicit trajectory-heterogeneity condition for IS-GRPO that hold without any condition on cross-prompt interference. This perspective explains why GRPO outperforms unnormalized policy gradient methods in practice.

We establish concrete conditions under which normalization yields gains both theoretically and empirically. To summarize, our contributions are:

- **Curvature interpretation.** We provide a new perspective on GRPO as a principled adaptive gradient mechanism that adapts learning to per-prompt curvature.
- **When normalization provably helps.** We show that under reasonable and mild assumptions, GRPO attains a faster convergence rate than unnormalized REINFORCE, with the speedup explicitly governed by the average per-prompt standard deviation over time. We further introduce IS-GRPO, an importance-sampling variant of GRPO, and establish a similar advantage over REINFORCE without any condition on cross-prompt interference.
- **When normalization empirically helps and when it does not**. On GSM8K and MATH, we validate the curvature–variance link and uncover three regimes for normalization: (I) early, near-orthogonal high-variance training where gains can be muted by noisy gradients; (II) a mid-stage with moderate variance and small pairwise gradient cosines where normalization delivers the strongest and most consistent improvements, especially on harder prompts; and (III) a late stage where the gap stops widening, coinciding with growing gradient coupling and, on easier prompts, with the accuracy ceiling.

Our results explain why GRPO needs normalization. It is a principled adaptive gradient mechanism whose benefits emerge under identifiable geometric and statistical conditions. This provides a theoretical foundation for the empirical success of GRPO and offers broader insights into the design of critic-free RL algorithms for LLM training.

Related works. We defer a detailed discussion of related work to Appendix A, including prior work on REINFORCE-style policy gradient methods, GRPO and its variants, theoretical studies of GRPO, and RLVR.

2 Preliminaries and Problem Settings

We begin by introducing the RLVR framework for LLM training and reviewing the GRPO algorithm, including the policy parametrization and update rules.

Notation. For a finite set $\mathcal{X}$, we denote by $\Delta(\mathcal{X})$ the set of probability distributions over $\mathcal{X}$. By default, all vectors are treated as column vectors. $\|\cdot\| = \|\cdot\|_2$ denotes the Euclidean norm for vectors and the spectral norm for matrices. For $v \in \mathbb{R}^m$, we use $\mathrm{diag}(v) \in \mathbb{R}^{m\times m}$ to denote the diagonal matrix with v on its diagonal. We also use the shorthand notation $[m] \coloneqq \{1, \dots, m\}$ and $\mathcal{B}(v, r) \coloneqq \{x \in \mathbb{R}^m \mid \|x - v\|_2 \le r\}$.

We say that a continuously differentiable function $f : \mathbb{R}^m \to \mathbb{R}^n$ is C-Lipschitz continuous if $\|\nabla f(x)\| \le C$ for all $x \in \mathbb{R}^m$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be ρ-weakly convex over $\mathcal{B}(v, r)$ if $f + \frac{\rho}{2}\|\cdot\|^2$ is convex over $\mathcal{B}(v, r)$. Moreover, f is L-smooth over $\mathcal{B}(v, r)$ if for all $x_1, x_2 \in \mathcal{B}(v, r)$, $\|\nabla f(x_1) - \nabla f(x_2)\| \le L\|x_1 - x_2\|$. An L-smooth function over $\mathcal{B}(v, r)$ is automatically L-weakly convex on the same set. Throughout the paper, we do not distinguish between a Lipschitz-smooth function and a function with Lipschitz-continuous gradients, nor do we differentiate among the

smoothness constant, the Lipschitz constant of the gradient, and the curvature. Finally, we use the terms question (answer) and prompt (output) interchangeably.

2.1 Problem Setup

RLVR. We consider a sequence-level RL with verifiable reward (RLVR) framework for training LLMs. RLVR (Lambert et al., 2024; Guo et al., 2025; Team et al., 2025; Wang et al., 2025) has recently emerged as an effective paradigm for improving LLM reasoning performance, particularly in domains with deterministic and automatically verifiable rewards. Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ denote the set of questions and $\mathcal{O} = \{o_1, \ldots, o_K\}$ the set of possible output sequences. A predefined deterministic reward function $r : \mathcal{Q} \times \mathcal{O} \to \{0, 1\}$ evaluates the correctness of a response, where $r(q, o) = 1$ if the o is a correct response to q, and $r(q, o) = 0$ otherwise. Given a question q, the LLM generates a response $o \sim \pi_\theta(q)$ according to a stochastic policy $\pi_\theta : \mathcal{Q} \to \Delta(\mathcal{O})$, parameterized by $\theta \in \Theta$. The objective is to learn a policy that maximizes the expected reward over the question set:

$$J(\theta) := \frac{1}{n} \sum_{i=1}^{n} J_i(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{o \sim \pi_\theta(\cdot|q_i)}[r(o, q_i)], \tag{1}$$

where $J_i(\theta) \coloneqq \mathbb{E}_{o \sim \pi_\theta(\cdot|q_i)}[r(o, q_i)]$ denotes the expect reward of policy π_θ on question q_i.

Remark 2.1. In LLM alignment settings such as RLHF, an additional KL penalty term is often introduced to mitigate over-optimization of the reward model:

$$J_{\mathrm{KL}}(\theta) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{o \sim \pi_\theta(\cdot|q_i)}[r(o, q_i)] - \beta \mathrm{KL}(\pi_\theta \| \pi_{\text{ref}}),$$

where $\beta \geq 0$ is the regularization parameter and π_{ref} denotes a reference policy (Shao et al., 2024). In contrast, for RLVR, recent studies (Chu et al., 2025; Hu et al., 2025) have shown that removing the KL term can yield faster convergence and improved performance. Consequently, throughout this paper we focus on the unregularized objective, where $J_{\mathrm{KL}}(\theta)$ reduces to $J(\theta)$ for $\beta = 0$.

In this paper, we consider an on-policy setting with *exact* gradient and randomly selected questions. At each iteration, a question q_i is sampled *uniformly* from $\mathcal{Q}$, and the corresponding gradient $\nabla J_i(\theta)$ is evaluated exactly.

PPO. PPO (Schulman et al., 2017) has become a standard baseline for LLM alignment due to its stability and empirical effectiveness (Ouyang et al., 2022; Rafailov et al., 2023). By combining policy gradient updates with a learned value-function critic and a clipped surrogate objective, PPO mitigates large policy updates and improves training robustness. Its objective function of PPO is given by:

$$J_{\text{PPO}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{o \sim \pi_{\theta_{\text{old}}}(\cdot|q_i)} \Big[\min \Big(\gamma_i(o) A_i(o), \text{clip}(\gamma_i(o), 1 - \epsilon, 1 + \epsilon) A_i(o) \Big) \Big], \tag{2}$$

where $\epsilon \in (0, 1)$ is the clipping parameter, and $\gamma_i(o) \coloneqq \frac{\pi_\theta(o|q_i)}{\pi_{\theta_{\text{old}}}(o|q_i)}$ is the importance ratio between the current policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$. The advantage $A_i(o)$ is calculated using Generalized Advantage Estimation with a learned critic. However, the reliance on a critic introduces additional computational overhead and implementation complexity. These considerations have motivated growing interest in critic-free alternatives such as REINFORCE.

REINFORCE. Unlike PPO, REINFORCE (Williams, 1992) appears well-suited for LLM alignment as it efficiently estimates the gradient with a single query of the language and reward model. It does not require training a value model, making it computationally more efficient than PPO. However, REINFORCE is known to suffer from high variance in its stochastic gradient estimates (Li et al., 2023). In this paper, we focus on the *exact* setting where the full gradient is computed for randomly selected questions. Under this setting, all critic-free PG methods, including REINFORCE, reduce to the vanilla PG update, as summarized in Algorithm 1.

Algorithm 1 Critic-free Policy Gradient Method

Input: learning rate $\eta > 0$, initial parameter θ_0.
for $t = 0$ to $T-1$ **do**
  Select i_t uniformly at random from $\{1, \ldots, n\}$
  $\theta_{t+1} \leftarrow \theta_t + \eta \, \nabla J_{i_t}(\theta_t)$
end for
Return: final policy $\pi_{\theta_{T-1}}$

Algorithm 2 On-policy GRPO

Input: learning rate $\eta > 0$, initial parameter θ_0.
for $t = 0$ to $T-1$ **do**
  Select i_t uniformly at random from $\{1, \ldots, n\}$
  $\theta_{t+1} \leftarrow \theta_t + \eta \, \frac{\nabla J_{i_t}(\theta_t)}{\sqrt{\mathbb{V}[\pi_{\theta_t}(i_t)]}}$
end for
Return: final policy $\pi_{\theta_{T-1}}$

2.2 GRPO

GRPO, introduced in DeepSeek-Math (Shao et al., 2024) and DeepSeek-R1 (Guo et al., 2025), builds on the computational efficiency of REINFORCE by eliminating the learned value-function while substantially improving empirical performance. The GRPO objective closely resembles the PPO objective in Equation (2):

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{o\sim\pi_{\theta_{\text{old}}}(\cdot|q_i)}\Big[\min\Big(\gamma_i(o)A_i(o), \text{clip}(\gamma_i(o), 1-\epsilon, 1+\epsilon)A_i(o)\Big)\Big], \tag{3}$$

differing only in the advantage term $A_i(o)$. Specifically, GRPO defines the advantage as:

$$A_i(o) \coloneqq \frac{r(q_i, o) - \mathbb{E}_{o'\sim\pi_{\theta_{\text{old}}}(\cdot|q_i)}\left[r\left(q_i, o'\right)\right]}{\sqrt{\mathbb{V}_{o'\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[r\left(q_i, o'\right)\right]}},$$

which uses the group mean reward as a baseline, and normalizes by the within-prompt reward standard deviation. For theoretical analysis, we impose the assumption that each question admits a unique correct answer, a condition commonly adopted in the study of RL algorithms (Mei et al., 2020; 2023; Lin et al., 2025a).

Assumption 1 (Unique correct answer)**.** For any $q \in \mathcal{Q}$, there exists a unique $o^*(q) \in \mathcal{O}$ such that $r(q, o^*(q)) = 1$.

Under Assumption 1, let a_i denote the index of the correct answer for question $q_i \in \mathcal{Q}$, so that $r(q_i, o_{a_i}) = 1$. We use $r_i \in \mathbb{R}^K$ to denote the *one-hot* reward vector for question q_i, with $[r_i]_j = r(q_i, o_j), \forall j \in [K]$. We consider the on-policy scenario where $\pi_\theta = \pi_{\theta_{\text{old}}}$. Let $\pi^*_\theta(i) \coloneqq \pi_\theta(o_{a_i} \mid q_i)$ be the success probability of policy π_θ on question q_i, and let $\mathbb{V}[\pi_\theta(i)] = \pi^*_\theta(i)\Big(1 - \pi^*_\theta(i)\Big)$ denote its Bernoulli reward variance. Under this setting, the importance ratio $\gamma_i(o) = 1$, and the GRPO advantage simplifies to

$$A_i(o) = \frac{r(q_i, o) - \text{sg}(\pi^*_\theta(i))}{\text{sg}(\sqrt{\mathbb{V}[\pi_\theta(i)]})}, \tag{4}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator, which treats its argument as a constant during backpropagation. The GRPO objective can be simplified as:

$$J_{\text{GRPO}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} J^i_{\text{GRPO}}(\theta) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{o\sim\pi_\theta}\Big[A_i(o)\Big].$$

We further denote by $\pi_\theta(i) \in \mathbb{R}^K$ the probability vector for π_θ on question i, with $[\pi_\theta(i)]_j \coloneqq \pi_\theta(o_j \mid q_i), \forall j \in [K]$. Since subtracting a constant baseline does not affect the gradient of the objective, we therefore refer to the resulting algorithm as *(on-policy) GRPO*. Our key observation is that the variance normalization in GRPO implicitly implements an adaptive step size. For all $i \in [n]$,

$$\nabla J^i_{\text{GRPO}}(\theta) = \mathbb{E}_{o\sim\pi_\theta}[A_i(o)\nabla \ln \pi_\theta(o \mid q_i)] = \frac{\mathbb{E}_{o\sim\pi_\theta}[r(q_i, o)\nabla \ln \pi_\theta(o \mid q_i)]}{\sqrt{\mathbb{V}[\pi_\theta(i)]}} = \frac{\nabla J_i(\theta)}{\sqrt{\mathbb{V}[\pi_\theta(i)]}}. \tag{5}$$

The first and last equalities follow from the policy gradient theorem (Sutton et al., 1998). The second equality holds because subtracting a constant baseline does not affect the gradient calculation. The pseudo-code for GRPO is provided in Algorithm 2.

Policy parametrization. We focus on the *log-linear policy parametrization*, which has been widely studied in the analysis of policy gradient methods (Agarwal et al., 2021; Yuan et al., 2022). Specifically, for each question-output pair (q_i, o_j), we assume the existence of a fixed feature vector $x_{i,j} \in \mathbb{R}^d$, and define the policy as $\pi_\theta(o_j \mid q_i) := \frac{\exp(x_{i,j}^\top \theta)}{\sum_{l=1}^K \exp(x_{i,l}^\top \theta)}$.

We denote by $X_i \in \mathbb{R}^{K\times d}$ the feature matrix associated with question q_i: $X_i \coloneqq (x_{i,1}, \cdots, x_{i,K})^\top$. The explicit update rules for REINFORCE (Algorithm 1) and GRPO (Algorithm 2) under this parametrization are provided in Appendix B.

3 Theoretical Results

In this section, we present a convergence analysis of REINFORCE and GRPO. We show that GRPO can achieve provably faster convergence than REINFORCE under our assumptions. Because PG with linear function approximation may fail to converge to the optimal policy in general, our analysis focuses on convergence rates toward stationary points.

3.1 Local Smoothness of Objective Function

We begin by relating reward variance to the smoothness of of the per-prompt objective. Let $X_{\max} := \max_{i\in[n]} \|X_i\|$.

Lemma 3.1. *Under Assumption 1, for all $i \in [n]$ and $\theta \in \mathbb{R}^d$, we have $\|\nabla^2 J_i(\theta)\| \le 4X_{\max}^2 \cdot \mathbb{V}[\pi_\theta(i)]$. Moreover, $\|\nabla^2 J_i(\theta)\| \le X_{\max}^2$, and hence $J_i(\theta)$ is $X_{\max}^2$-smooth on $\mathbb{R}^d$.*

The proof is provided in Appendix C.1. This lemma shows that the local smoothness constant of $J_i(\theta)$ is directly controlled by the reward variance of q_i under the current policy π_θ.

For deterministic gradient descent on an L-smooth function, a step size of $1/L$ is a standard choice (Garrigos & Gower, 2023). Lemma 3.1 therefore motivates adapting the step size to the local smoothness constant of each per-prompt objective $J_i(\theta)$, which varies across prompts. GRPO achieves exactly this by normalizing updates with reward variance, which implicitly implements an adaptive step size matched to the local curvature of each prompt.

3.2 Orthogonal Representation

We first impose the following assumption on the bound of within-prompt Bernoulli variance at every step.

Assumption 2 (Bounded variance)**.** For each $i \in [n]$, there exists a positive sequence $\{C_{i,t}\}_{t=1}^\infty$

$$\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]} \le C_{i,t} \le \frac{1}{2}.$$

We further denote $\mu_{k,t} \coloneqq \frac{1}{n}\sum_{i=1}^n C_{i,t}^k$. To extend Lemma 3.1 from the per-prompt objectives $J_i(\theta)$ to the averaged objective $J(\theta) = \frac{1}{n}\sum_{i=1}^n J_i(\theta)$, we must control how gradients of different prompts interact during optimization; in particular, they should not interfere destructively with one another. Empirically, as we show in Section 4, gradients associated with different prompts are nearly orthogonal. Motivated by this observation, we introduce the following assumption.

Assumption 3 (Orthogonal representation)**.** For all $i, j \in [n]$ with $i \ne j$, we have $X_i X_j^\top = 0$.

This assumption ensures that the gradients associated with different prompts are mutually orthogonal, allowing their effects on the optimization dynamics to be analyzed independently for both REINFORCE and GRPO.

3.3 Convergence Result under the Orthogonal Representation Assumption

Before presenting the convergence results for Algorithm 1 and 2, we further characterize the smoothness properties of J_i at each iteration.

Lemma 3.2. *Under Assumption 1, for all $i \in [n]$, we know*

$$\|\nabla J_i(\theta)\| \le 2\mathbb{V}[\pi_\theta(i)]X_{\max}.$$

Corollary 3.1. *Under Assumption 1, for all $i \in [n]$, $J_i(\theta)$ is $\frac{1}{2}X_{\max}$-Lipschitz over $\mathbb{R}^d$.*

Lemma 3.3 (Non-uniform local smoothness). *Under Assumption 1, for all $i \in [n]$ and $\theta \in \mathbb{R}^d$, we have $\|\nabla^2 J_i(\theta')\| \le \frac{5}{2} X_{\max}^2 \cdot \sqrt{\mathbb{V}[\pi_\theta(i)]}$ for every $\theta' \in \mathcal{B}(\theta, \frac{1}{X_{\max}} \cdot \sqrt{\mathbb{V}[\pi_\theta(i)]})$. Consequently, J_i is $\frac{5}{2} X_{\max}^2 \cdot \sqrt{\mathbb{V}[\pi_\theta(i)]}$-smooth within this neighborhood.*

The proofs are provided in Section C.2 and C.3. We now present our two main theorems, which establish the convergence guarantees for REINFORCE and GRPO, respectively. Their detailed proofs are deferred in Appendix D.

Theorem 3.1 (Convergence rate of REINFORCE per question). *Under Assumption 1 and Assumption 3, with step size $\eta = \frac{1}{X_{\max}^2}$, Algorithm 1 satisfies*

$$\mathbb{E}[J_i(\theta_t)] - \mathbb{E}[J_i(\theta_{t-1})] \;\ge\; \frac{1}{2nX_{\max}^2}\, \mathbb{E}[\|\nabla J_i(\theta_{t-1})\|^2].$$

Moreover,

$$\min_{t \in \{0,1,\ldots,T-1\}} \mathbb{E}[\|\nabla J_i(\theta_t)\|^2] \le \frac{2n(1-\pi_{\theta_0}^*(l))X_{\max}^2}{T}.$$

Theorem 3.2 (Convergence rate of GRPO per question). *Under Assumption 1-3 with the step size $\eta = \frac{2}{5X_{\max}^2}$, Algorithm 2 satisfies*

$$\mathbb{E}[J_i(\theta_t)] - \mathbb{E}[J_i(\theta_{t-1})] \;\ge\; \frac{\mathbb{E}[\|\nabla J_i(\theta_{t-1})\|^2]}{5nX_{\max}^2\, C_{i,t}}.$$

Moreover, we have

$$\min_{t \in \{0,1,\ldots,T-1\}} \mathbb{E}[\|\nabla J_i(\theta_t)\|^2] \le \frac{2n(1-\pi_{\theta_0}^*(l))X_{\max}^2}{T} \cdot C(i,T),$$

where $C(i,T) \coloneqq \frac{5}{2T} \sum_{t=0}^{T-1} C_{i,t}$.

Remark 3.3. Comparing Theorems 3.1 and 3.2, we observe that GRPO achieves a strictly better average convergence bound than REINFORCE, provided that the following constant factor satisfies

$$C(T) := \frac{1}{n} \sum_{i=1}^{n} C(i,T) = \frac{5}{2T} \sum_{t=0}^{T-1} \mu_{1,t} < 1. \tag{6}$$

Here, $C(T)$ equals $\frac{5}{2}$ times the average within-prompt Bernoulli standard deviation over prompts and iterations.

In practice, $C(T)$ is typically much smaller than 1 when the question set contains heterogeneous prompts with varying levels of difficulty. We further validate $C(T) < 1$ empirically in Section 4. Intuitively, GRPO automatically assigns smaller effective step sizes to prompts with higher curvature, yielding faster convergence on average. Similar improvements can also be expected in settings where the curvature varies significantly across iterations.

3.4 Removing the Orthogonality Assumption

The preceding results isolate the benefit of variance normalization, but their proofs rely on Assumption 3 to decouple the per-question objectives. We now remove this requirement and analyze the mean objective $J(\theta) = \frac{1}{n} \sum_{i=1}^{n} J_i(\theta)$ directly. To this end, we introduce IS-GRPO (Algorithm 3), a variant of GRPO whose question-selection rule is based on importance sampling, a classical technique for non-uniform sampling in stochastic gradient methods (Needell et al., 2014; Zhao & Zhang, 2015). Instead of the uniform question sampling in Algorithms 1 and 2, IS-GRPO samples each question in proportion to its current reward standard deviation and rescales the corresponding gradient, so that the update remains positively aligned with the full gradient $\nabla J(\theta)$.

For the sampled question q_{i_t}, the update direction g_t defined in Algorithm 3 coincides with the normalized GRPO gradient in (5) up to a constant scalar of 2. Thus, importance sampling changes only how questions are selected, while each selected question is still updated with GRPO's normalized gradient. The convergence rates of the two algorithms are governed by the trajectory coefficients

$$D(T) \coloneqq \frac{2}{T} \sum_{t=0}^{T-1} \mu_{2,t}\, \mu_{4,t} \qquad \text{and} \qquad D_{\mathrm{IS}}(T) \coloneqq \frac{\sum_{t=0}^{T-1} \mu_{4,t}}{\sum_{t=0}^{T-1} \frac{1}{\mu_{1,t}}},$$

associated with Algorithm 1 and Algorithm 3, respectively. Our first theorem provides the baseline guarantee for Algorithm 1 on the mean objective.

Algorithm 3 **I**mportance **S**ampling **GRPO**

Input: learning rate $\eta > 0$, initial parameter θ_0.
for $t = 0$ to $T - 1$ **do**
  Form $p_{t,i} = \frac{\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}}{\sum_{j=1}^{n} \sqrt{\mathbb{V}[\pi_{\theta_t}(j)]}}$ for $i \in [n]$
  Sample $i_t \sim p_t$ and compute $g_t := \frac{\nabla J_{i_t}(\theta_t)}{2\sqrt{\mathbb{V}[\pi_{\theta_t}(i_t)]}}$
  $\theta_{t+1} \leftarrow \theta_t + \eta\, g_t$
end for
Return: final policy $\pi_{\theta_{T-1}}$

Theorem 3.4 (Convergence of REINFORCE without orthogonality)**.** *Suppose Assumption 1 and Assumption 2 hold. For every step size* $0 < \eta \le \frac{1}{2X_{\max}^2}$*, Algorithm 1 satisfies*

$$\min_{t\in\{0,\dots,T-1\}} \mathbb{E}\|\nabla J(\theta_t)\|^2 \;\le\; \frac{1 - J(\theta_0)}{\eta T} + 8\,\eta\, X_{\max}^4\, D(T). \tag{7}$$

Theorem 3.4 extends the per-question guarantee of Theorem 3.1 to the mean objective without Assumption 3. Our second theorem shows that Algorithm 3 satisfies a guarantee of a similar form, with $D(T)$ replaced by $D_{\mathrm{IS}}(T)$.

Theorem 3.5 (Convergence of IS-GRPO without orthogonality)**.** *Suppose Assumption 1 and Assumption 2 hold. For every step size* $0 < \eta \le \frac{1}{2X_{\max}^2}$*, Algorithm 3 satisfies*

$$\min_{t\in\{0,\dots,T-1\}} \mathbb{E}\|\nabla J(\theta_t)\|^2 \;\le\; \frac{1 - J(\theta_0)}{\eta T} \cdot \frac{4C(T)}{5} + 8\,\eta\, X_{\max}^4\, D_{\mathrm{IS}}(T), \tag{8}$$

where $C(T)$ *is defined in* (6)*.*

All proofs for this subsection are deferred to Section E.

Remark 3.6. Since both theorems share the same step-size range, the comparison rests on the factor $\frac{4}{5}C(T)$ and on $D_{\mathrm{IS}}(T)$ versus $D(T)$. For the first term, we already showed in Section 3.3 that $C(T) < 1$, which leads to $\frac{4}{5}C(T) < \frac{4}{5}$. Thus, the first term of Theorem 3.5 is strictly smaller than that of Theorem 3.4.

For the second term, $D_{\mathrm{IS}}(T) < D(T)$ is satisfied when $C_{i,t}$ vary substantially across iterations and some of the $\mu_{1,t}$ are close to zero, since such iterations contribute terms of order $1/\mu_{1,t}$ to the left-hand side. We validate $D_{\mathrm{IS}}(T) < D(T)$ empirically in Section 4.

Finally, with the step size $\eta \asymp 1/\sqrt{T}$, the two algorithms converge at the same rate $O(1/\sqrt{T})$. However, IS-GRPO attains this rate with strictly smaller constants.

4 Empirical Studies

The analysis of Section 3 is stated for a log-linear policy updated by plain gradient ascent, with exact per-prompt gradients and exactly known reward variances. Practical GRPO differs on every one of these points: gradients are estimated from $G = 8$ rollouts per prompt, the optimizer is AdamW, and each prompt's standard deviation is itself estimated from the same eight samples. We therefore do not expect the constants in Theorems 3.1, 3.2, 3.4 and 3.5 to transfer literally. What the experiments test is (i) whether the assumptions and conditions on which the theory rests are met by real training runs (Sections 4.1 and 4.3), and (ii) whether the qualitative predictions of the theory hold: how the methods should be ordered, and under which conditions the gaps between them should appear or vanish (Section 4.2).

4.1 Validation of Local Curvature–Variance Connection

We use the diagonal Fisher information as a curvature proxy (Liu et al., 2023), with estimation details provided in Appendix F.3. As shown in Table 1, curvature and reward variance are positively associated within the same iteration, but *not* across different iterations. This finding is consistent with Lemma 3.1. Since Lemma 3.1 bounds rather than equates the two, we test only for a positive association.

Table 1: Fisher–variance correlation.

Lag	r	ρ	Sig.
Same iter.	.342	.41	Yes
Diff. iter.	$-.028$	$-.015$	No

Same iter.: curvature and variance measured at the same checkpoint.
Diff. iter.: curvature at step t, variance at step $t' \neq t$.

Does the orthogonality assumption hold? Assumption 3 requires that per-prompt gradients are orthogonal (Lemma D.1). Tracking pairwise gradient cosine similarities on GSM8K, we observe that they are near-orthogonal at initialization but grow increasingly coupled during training, so the gradients are no longer nearly orthogonal by the end (Figure 1, Section F.2). This observation motivates IS-GRPO, as Theorems 3.4 and 3.5 require no orthogonality assumption and remain valid once interference grows.

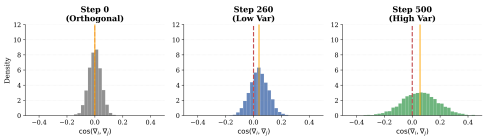


Figure 1: Pairwise gradient cosine similarity during GRPO training (1.5B, GSM8K, 4,950 pairs). Red dashed: $\cos = 0$; orange: mean.

4.2 Comparisons on LLM Reasoning Tasks

Methods. We compare four update rules that share the same rollouts, the same per-prompt mean baseline and the same optimizer: (1) **per-prompt std** (standard GRPO), which divides by the within-prompt reward std; (2) **global std**, which divides by the mini-batch-wide std, so that all prompts in a batch share one scaling factor; (3) **REINFORCE**, with mean subtraction only; and, at 7B scale, (4) **IS-GRPO**, which combines per-prompt normalization with the question-selection rule of Algorithm 3. In practice, IS-GRPO replaces $\sqrt{\mathbb{V}[\pi_\theta(i)]}$ in Algorithm 3 by a running per-prompt estimate s_i and normalizes advantages by the within-group std as in GRPO (Section F.1).

What the theory predicts. Theorems 3.2 and 3.5 guarantee that, in the exact-gradient model, per-prompt GRPO (under Assumption 3) and IS-GRPO (without it) have smaller convergence constants than REINFORCE once $C(T) < 1$. We expect per-prompt normalization to help while active prompts are heterogeneous, and to bring little gain when few, similar prompts remain active. IS-GRPO's selection rule only needs to distinguish zero from non-zero variance and is not subject to this limitation.

4.2.1 Small Scale Experiments: Accuracy from Low to Relatively High

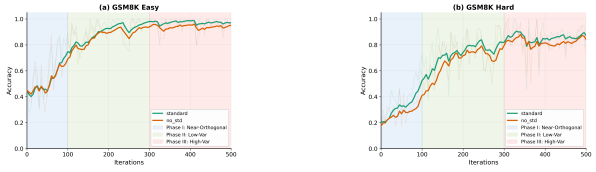


Figure 2: Training accuracy on GSM8K (1.5B) with phase annotations; left: Easy, right: Hard. *standard*: per-prompt normalization; *no_std*: unnormalized (global std omitted for readability; see Table 3).

Figure 2 shows training accuracy on GSM8K with Qwen2.5-Math-1.5B; Table 3 reports final numbers. Per-prompt std consistently outperforms both baselines.

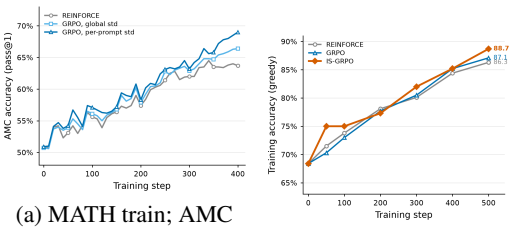
Phase analysis. The dynamics fall into three phases. In **Phase I (0–100)** all methods overlap: with $G = 8$ rollouts many groups are all-correct or all-incorrect and receive zero advantage, while the active prompts have similar success probabilities, so per-prompt and global normalization coincide. In **Phase II (100–300)**, as accuracies rise heterogeneously, c_t falls and the scaling factors of active prompts spread out; per-prompt std opens a widening gap, reaching ≈ 3 pp (Easy) and ≈ 7 pp (Hard) over REINFORCE by iteration 300. In **Phase III (300–500)** the gap narrows but persists (Table 3). This coincides with a

rise of the gradient-cosine std and, on Easy, with accuracies approaching the ceiling, which by itself limits the room for any gap.

4.2.2 Larger Scale Experiments: 7B Model

We run Qwen2.5-Math-7B in two regimes that differ in how much of the training set is already solved (Section F.5).

Harder training set. Trained on MATH Level 3–5 and evaluated on AMC (Figure 3a), the schemes follow the 1.5B pattern: they overlap early, after which per-prompt std pulls ahead of global std, which stays ahead of REINFORCE. On a second, harder held-out benchmark (AIME 2024), we observe a slightly positive result for normalization (Section F.5).



(a) MATH train; AMC test

(b) MATH train

Figure 3: 7B results (Qwen2.5-Math-7B). (a) Consistent with normalization helping when many prompts remain unsolved (AMC held-out). (b) When most prompts are solved, the normalization schemes coincide and IS-GRPO leads.

Largely solved training set. Trained on MATH Level 1–5 and evaluated on the training prompts, the proxy of $J(\theta)$ (Figure 3b, three seeds), the three normalization schemes end within one across-seed standard deviation, while IS-GRPO leads REINFORCE, most visibly early in training. This setting resembles Phase III from the start: most groups carry no gradient and the remaining prompts are similar. Among the features that make per-prompt normalization coincide with global std in this run, resolution and optimizer invariance ((i), (ii) in Section F.5) hold equally at 1.5B, where per-prompt normalization does help; what distinguishes this run is (iii), the high base accuracy that leaves few, similar active prompts. IS-GRPO is affected by neither: its guarantee does not require orthogonality, and its selection rule stops drawing solved prompts.

4.3 Trajectory Coefficients in Practice

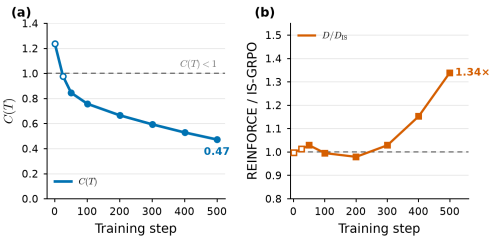


Figure 4: Trajectory coefficients along the 7B IS-GRPO run. (a) Running average $C(T)$; the dashed line is the speed-up condition (6). (b) Ratio $D(T)/D_{\mathrm{IS}}(T)$.

Along the 7B IS-GRPO run, C(T) falls below 1 early in training and keeps decreasing, ending at 0.47 (Figure 4a), so condition (6) holds for essentially the whole run. $D(T)$ and $D_{\mathrm{IS}}(T)$ remain comparable, so the advantage of (8) over (7) comes from the first-term factor $\frac{4}{5}C(T)$; at the optimal step size the IS-GRPO guarantee is tighter throughout, by a factor that grows as prompts are solved. The heterogeneity across prompts that develops is dominated by the contrast between solved and active prompts, which per-prompt normalization cannot exploit and IS-GRPO's selection rule can, consistent with Figure 3(b).

5 Conclusion

GRPO's normalization implements a curvature-adaptive gradient mechanism. This perspective explains when normalization helps and why GRPO can outperform REINFORCE. Theoretically, we establish faster convergence guarantees for GRPO and IS-GRPO, a newly introduced algorithm, under mild assumptions. Empirically, experiments on GSM8K and MATH validate the curvature–variance connection, show that per-prompt normalization improves convergence when per-prompt variances are heterogeneous and brings no gain when they are not, and confirm that the conditions of our bounds $C(T) < 1$ and a tighter guarantee for IS-GRPO, hold along real training runs.

Limitation. Our analysis focuses on on-policy GRPO in the RLVR and log-linear policy parametrization. Extending the framework to more general RLHF settings, off-policy training, larger-scale models, and more challenging reasoning datasets remains an important direction. Our three-seed 7B experiments measure accuracy on the training prompts, the quantity our bounds concern; held-out evaluation at this scale is limited to a single-seed run.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing. We did not use generative AI tools to formulate or refine mathematical claims, draft derivations, to design the research methodology or experiments, to implement methods, or to interpret results. The remaining tasks with required disclosure, including generating synthetic datasets, translation, dataset cleaning and reformatting, and qualitative or thematic data analysis, are not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure: drafting and editing parts of the paper to improve readability, identifying and summarizing relevant literature, and formatting references. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

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

Michael Bereket and Jure Leskovec. Uncalibrated reasoning: Grpo induces overconfidence for stochastic outcomes. *arXiv preprint arXiv:2508.11800*, 2025.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*, 2024.

Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

Roger Girgis, Rodrigue de Schaetzen, Luke Rowe, Azalée Robitaille, Christopher Pal, and Liam Paull. Constrained group relative policy optimization. *arXiv preprint arXiv:2602.05863*, 2026.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

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

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

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

Samyak Jain, Ayush Agrawal, and Navin Goyal. Towards understanding the optimization landscape of grpo and its variants. In *First Workshop on Foundations of Reasoning in Language Models*, 2025.

Yuxin Jiang, Yufei Wang, Qiyuan Zhang, Xingshan Zeng, Liangyou Li, Jierun Chen, Chaofan Tao, Haoli Bai, and Lifeng Shang. From verifiable dot to reward chain: Harnessing verifiable reference-based rewards for reinforcement learning of open-ended generation. *arXiv preprint arXiv:2601.18533*, 2026.

Nathan Lambert. Reinforcement learning from human feedback. *arXiv preprint arXiv:2504.12501*, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Yingru Li, Jiawei Xu, Ziniu Li, Jiacai Liu, Wei Liu, Yuxuan Tong, Longtao Zheng, Zhenghai Xue, Yaxiang Zhang, Tianle Cai, et al. The optimal token baseline: Variance reduction for long-horizon llm-rl. *arXiv preprint arXiv:2602.07078*, 2026.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

Max Qiushi Lin, Jincheng Mei, Matin Aghaei, Michael Lu, Bo Dai, Alekh Agarwal, Dale Schuurmans, Csaba Szepesvari, and Sharan Vaswani. Rethinking the global convergence of softmax policy gradient with linear function approximation. *arXiv preprint arXiv:2505.03155*, 2025a.

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

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023.

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

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International conference on machine learning*, pp. 6820–6829. PMLR, 2020.

Jincheng Mei, Zixin Zhong, Bo Dai, Alekh Agarwal, Csaba Szepesvari, and Dale Schuurmans. Stochastic gradient succeeds for bandits. In *International Conference on Machine Learning*, pp. 24325–24360. PMLR, 2023.

Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Greenewald, Jiri Navratil, Jerret Ross, and Jesus Rios. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. *arXiv preprint arXiv:2505.22257*, 2025.

Deanna Needell, Rachel Ward, and Nathan Srebro. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

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

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

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

Marco Simoni, Aleksandar Fontana, Giulio Rossolini, Andrea Saracino, and Paolo Mori. Gtpo: Stabilizing group relative policy optimization via gradient and entropy control. *arXiv preprint arXiv:2508.03772*, 2025.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

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

Milan Vojnovic and Se-Young Yun. What is the alignment objective of grpo? *arXiv preprint arXiv:2502.18548*, 2025.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

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

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

Zhouliang Yu, Ruotian Peng, Keyi Ding, Yizhe Li, Zhongyuan Peng, Minghao Liu, Yifan Zhang, Zheng Yuan, Huajian Xin, Wenhao Huang, et al. Formalmath: Benchmarking formal mathematical reasoning of large language models. *arXiv preprint arXiv:2505.02735*, 2025b.

Huining Yuan, Zelai Xu, Huaijie Wang, Xiangmin Yi, Jiaxuan Gao, Xiao-Ping Zhang, Yu Wang, Chao Yu, and Yi Wu. Verifiable process rewards for agentic reasoning. *arXiv preprint arXiv:2605.10325*, 2026.

Rui Yuan, Simon Shaolei Du, Robert Mansel Gower, Alessandro Lazaric, and Lin Xiao. Linear convergence of natural policy gradient methods with log-linear policies. *ArXiv*, abs/2210.01400, 2022. URL `https://api.semanticscholar.org/CorpusID:252693021`.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *International Conference on Machine Learning*, pp. 1–9, 2015.

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

Hongyi Zhou, Kai Ye, Erhan Xu, Jin Zhu, Ying Yang, Shijin Gong, and Chengchun Shi. Demystifying group relative policy optimization: Its policy gradient is a u-statistic. *arXiv preprint arXiv:2603.01162*, 2026.

A Related Work

REINFORCE-style PG methods. Li et al. (2023) proposes ReMax, a sequence-level REINFORCE-style algorithm for LLM alignment. Ahmadian et al. (2024) propose RLOO, which samples multiple responses per prompt and uses a leave-one-out baseline to further reduce variance. Li et al. (2026) derives an optimal token-level baseline for variance reduction in long-horizon LLM RL, continuing this critic-free line of work.

GRPO and its variants. GRPO (Shao et al., 2024) and related methods (Simoni et al., 2025; Liu et al., 2025; Chu et al., 2025; Lin et al., 2025b) have gained widespread adoption due to their simplicity and scalability, and are now commonly used in post-training pipelines for reasoning-oriented models. Several extensions improve stability in long-CoT RL. DAPO (Yu et al., 2025a) introduces several techniques, including higher clipping thresholds and sequence-level losses, to make the training more stable. GSPO (Zheng et al., 2025) defines the importance ratio at the sequence level from sequence likelihood, avoiding high-variance token-level weights. CISPO (Chen et al., 2025) further revises the clipping strategy to avoid low-probability tokens from being clipped out after the first on-policy update. Constrained GRPO Girgis et al. (2026) extends GRPO to constrained settings and standardizes reward and cost advantages separately, so that normalization does not distort the intended trade-off.

Emerging theory for GRPO. Recent studies analyze what GRPO optimizes and how it behaves in on- and off-policy regimes (Mroueh et al., 2025), its implicit alignment objective (Vojnovic & Yun, 2025), and trajectory-corrected variants with convergence guarantees (Pang & Jin, 2025). Zhou et al. (2026) characterize the GRPO gradient as a U-statistic, which yields finite-sample error bounds and guidance on the group size.Bereket & Leskovec (2025) highlights a trade-off between normalization and calibration, showing that removing the std term can improve probability calibration at the cost of optimization speed. Jain et al. (2025) interprets GRPO's objective as a weighted combination of likelihood maximization for correct rollouts and likelihood minimization for incorrect ones, while noting that std normalization lacks a clear theoretical motivation. We contribute a new perspective, interpreting the std term as an adaptive gradient mechanism that reflects local curvature, which helps unify a range of previously disparate empirical findings.

RLVR. Reinforcement learning with verifiable rewards (RLVR) has emerged as an effective paradigm for reasoning-intensive domains. Unlike RLHF, which relies on a learned reward model, RLVR uses deterministic, verifiable rewards (Lambert et al., 2024; Guo et al., 2025; Team et al., 2025; Wang et al., 2025). This mitigates reward-model bias and hacking and simplifies training, while scaling effectively with compute and dataset size. Strong results have been reported on the well-recognized GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), Omni-MATH (Gao et al., 2024), and FormalMATH (Yu et al., 2025b). Recent work extends the paradigm with verifiable process rewards for multi-step agentic reasoning (Yuan et al., 2026) and reference-based verifiable rewards for open-ended generation (Jiang et al., 2026). In this paper, we focus on developing a theoretical understanding of GRPO in the RLVR setting.

B Gradient Update Rule of REINFORCE and GRPO

In this paper, we focus on the *log-linear policy parametrization* (Agarwal et al., 2021; Yuan et al., 2022). Specifically, we assume that for each question-output pair (q_i, o_j), there exists a constant feature vector $x_{i,j} \in \mathbb{R}^d$ and the policy is given by:

$$\pi_\theta(o_j \mid q_i) := \frac{\exp(x_{i,j}^\top \theta)}{\sum_{l=1}^K \exp(x_{i,l}^\top \theta)}. \tag{9}$$

For ease of notation, we simply drop t from i_t whenever it clear in context. The update of Algorithm 1 takes the following form (Lin et al., 2025a):

$$\theta_t \leftarrow \theta_{t-1} + \eta X_i^\top \left(\operatorname{diag}\left(\pi_{\theta_{t-1}}(i)\right) - \pi_{\theta_{t-1}}(i)\pi_{\theta_{t-1}}^\top(i)\right) r_i. \tag{10}$$

Table 2: List of notations.

Notations	Definition
$\mathcal{Q}$	Set of questions / prompts
$\mathcal{O}$	Set of possible output sequences
$o^*(q)$	Unique correct output for prompt q (Assumption 1)
a_i	Index of the correct output for prompt q_i
r_i	Reward vector for prompt q_i
π_θ	Sequence-level policy of the LLM parameterized by θ
$\pi_\theta(i)$	Probability vector for prompt q_i under policy π_θ
$\pi^*_\theta(i)$	Success probability on prompt q_i under policy π_θ
$J_i(\theta)$	Objective function for prompt q_i under policy π_θ
$J(\theta)$	Global objective under policy π_θ
$J_{\mathrm{GRPO}}(\theta)$	GRPO objective under policy π_θ
$x_{i,j}$	Feature vector for prompt-output pair (q_i, o_j)
X_i	Feature matrix for q_i with rows $x_{i,1}, \ldots, x_{i,K}$
$X_{\max}$	Maximal feature norm, $X_{\max} = \max_{i\in[n]} \|X_i\|$
$H(\pi_\theta(i))$	Covariance matrix $\mathrm{diag}(\pi_\theta(i)) - \pi_\theta(i)\pi_\theta(i)^\top$
$\mathbb{V}(\pi_\theta(i))$	Variance of reward for prompt q_i under policy π_θ
$\|\cdot\|$	Euclidean norm (vectors) / spectral norm (matrices)
$\mathrm{diag}(v)$	Diagonal matrix with vector v on its diagonal
$\mathcal{B}(v, r)$	Euclidean ball $\{x \in \mathbb{R}^m : \|x - v\|_2 \le r\}$
$C_{i,t}$	Prompt-wise variance bound introduced in Assumption 2

Under Assumption 1, the update of REINFORCE (Algorithm 1) can be simplified as:

$$\theta_t \leftarrow \theta_{t-1} + \eta\Big(\pi^*_{\theta_{t-1}}(i)(1-\pi^*_{\theta_{t-1}}(i)x_{i,a_i} - \pi^*_{\theta_{t-1}}(i)\sum_{j\neq a_i}[\pi_{\theta_{t-1}}(i)]_j \cdot x_{i,j}\Big). \tag{11}$$

Similarly, the update of GRPO (Algorithm 2) can be simplified as:

$$\theta_t \leftarrow \theta_{t-1} + \eta\Big(\sqrt{\pi^*_{\theta_{t-1}}(i)(1-\pi^*_{\theta_{t-1}}(i))}x_{i,a_i} - \sqrt{\frac{\pi^*_{\theta_{t-1}}(i)}{1-\pi^*_{\theta_{t-1}}(i)}}\sum_{j\neq a_i}[\pi_{\theta_{t-1}}(i)]_j \cdot x_{i,j}\Big). \tag{12}$$

C Analysis of the Local Smoothness Constant

C.1 Proof of Lemma 3.1

According to Lemma 17 in Lin et al. (2025a), for any $y \in \mathbb{R}^d$, we have

$$y^\top \nabla^2 J_i(\theta) y = \left(H\left(\pi_\theta(i)\right) r_i\right)^\top \left(X_i y \odot X_i y\right) - 2\left(H\left(\pi_\theta(i)\right) r_i\right)^\top \left(X_i y\right)\left(\pi_\theta^\top(i) X_i y\right)$$

where $H(\pi_\theta(i))$ is defined as $H\left(\pi_\theta\right) := \mathrm{diag}\left(\pi_\theta(i)\right) - \pi_\theta(i)\pi_\theta^\top(i) \in \mathbb{R}^{K\times K}$ and $\odot$ denotes the Hadamard (component-wise) product. Using the triangle inequality and Cauchy-Schwarz inequality, we get

$$\begin{aligned}
|y^\top \nabla^2 J_i(\theta) y| &\le |\left(H\left(\pi_\theta(i)\right) r_i\right)^\top \left(X_i y \odot X_i y\right)| + 2|\left(H\left(\pi_\theta(i)\right) r_i\right)^\top \left(X_i y\right)| \cdot |\left(\pi_\theta^\top(i) X_i y\right)| \\
&\le \|\left(H\left(\pi_\theta(i)\right) r_i\right)\|_\infty \|X_i y \odot X_i y\|_1 + 2\|H\left(\pi_\theta(i)\right) r_i\| \cdot \|X_i y\| \cdot \|\pi_\theta(i)\| \cdot \|X_i y\| \\
&= \|\left(H\left(\pi_\theta(i)\right) r_i\right)\|_\infty \|X_i y\|^2 + 2\|H\left(\pi_\theta(i)\right) r_i\| \cdot \|\pi_\theta(i)\| \cdot \|X_i y\|^2 \\
&\le \|\left(H\left(\pi_\theta(i)\right) r_i\right)\|_\infty \|X_i y\|^2 + 2\|H\left(\pi_\theta(i)\right) r_i\| \cdot \|X_i y\|^2.
\end{aligned} \tag{13}$$

The last inequality follows because $\|\pi_\theta(i)\| \le \|\pi_\theta(i)\|_1 = 1$. According to Assumption 1, we have

$$[H(\pi_\theta(i)) r_i]_j = \begin{cases} \pi^*_\theta(i)(1-\pi^*_\theta(i)), & \text{if } j = a_i \\ -\pi^*_\theta(i)[\pi_\theta(i)]_j, & \text{if } j \neq a_i \end{cases}$$

With this expression, we get

$$\|H(\pi_\theta(i))r_i\|_\infty = \pi^*_\theta(i)(1-\pi^*_\theta(i)), \tag{14}$$

and

$$\begin{aligned}\|H(\pi_\theta(i))r_i\| &= \pi^*_\theta(i)\sqrt{(1-\pi^*_\theta(i))^2 + \sum_{j\neq a_i}[\pi_\theta(i)]_j^2} \\ &\leq \pi^*_\theta(i)\sqrt{(1-\pi^*_\theta(i))^2 + \sum_{j\neq a_i}[\pi_\theta(i)]_j(1-\pi^*_\theta(i))} \\ &= \sqrt{2}\pi^*_\theta(i)(1-\pi^*_\theta(i)).\end{aligned} \tag{15}$$

Combining (14) and (15) with (13), we get

$$\begin{aligned}|y^\top \nabla^2 J_i(\theta) y| &\leq \|\left(H\left(\pi_\theta(i)\right) r_i\right)\|_\infty \|X_i y\|^2 + 2\|H\left(\pi_\theta(i)\right) r_i\| \cdot \|X_i y\|^2 \\ &\leq (2\sqrt{2}+1)\pi^*_\theta(i)(1-\pi^*_\theta(i))\|X_i y\|^2 \\ &\leq (2\sqrt{2}+1)\pi^*_\theta(i)(1-\pi^*_\theta(i))\|X_i\|^2\|y\|^2 \\ &\leq 4\pi^*_\theta(i)(1-\pi^*_\theta(i))X^2_{\max}\|y\|^2\end{aligned} \tag{16}$$

where the third inequality is due to the definition of operator norm, and the last inequality is by definition of $X_{\max}$. Note that

$$\|\nabla^2 J_i(\theta)\| = \max_y \frac{|y^\top \nabla^2 J_i(\theta) y|}{\|y\|^2}$$

for symmetric Hessian matrix $\nabla^2 J_i(\theta)$, which completes the proof.

C.2 Proof of Lemma 3.2

According to (11), the gradient of $J_i(\theta)$ takes the following form:

$$\nabla J_i(\theta) = x_{a_i}^\top (1-\pi^*_{\theta_t}(i))\pi^*_{\theta_t}(i) - \sum_{j\neq a_i} x_j^\top \pi_{\theta_t}(i)_j \cdot \pi^*_{\theta_t}(i).$$

Note that a matrix's operator norm is larger than the norm of any of its row vector, we get

$$\begin{aligned}\|\nabla J_i(\theta)\| &\leq \|x_{a_i}\|(1-\pi^*_{\theta_t}(i))\pi^*_{\theta_t}(i) + \sum_{j\neq a_i}\|x_j\|\pi_{\theta_t}(i)_j \cdot \pi^*_{\theta_t}(i) \\ &\leq \|X_i\|(1-\pi^*_{\theta_t}(i))\pi^*_{\theta_t}(i) + \sum_{j\neq a_i}\|X_i\|\pi_{\theta_t}(i)_j \cdot \pi^*_{\theta_t}(i) \\ &= 2\|X_i\|(1-\pi^*_{\theta_t}(i))\pi^*_{\theta_t}(i) \\ &\leq 2\mathbb{V}[\pi_\theta(i)]X_{\max},\end{aligned}$$

where the last inequality is due to the definition of $X_{\max}$, finishing the proof.

C.3 Proof of Lemma 3.3

By Assumption 1, the objective $J_i(\theta)$ is same as $\pi^*_\theta(i)$. From Lemma 3.2, $J_i(\theta)$ is $\frac{1}{2}X_{\max}$-Lipschitz. Consequently, for any $\theta' \in \mathcal{B}\Big(\theta, \frac{1}{X_{\max}}\sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)}\Big)$, we have

$$\big|\pi^*_{\theta'}(i) - \pi^*_\theta(i)\big| \leq \tfrac{1}{2}X_{\max} \cdot \tfrac{1}{X_{\max}} \cdot \sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)} = \tfrac{1}{2}\sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)}.$$

Combining with Lemma 3.1,

$$\|\nabla^2 J_i(\theta')\| \leq \max_l 4X^2_{\max} \cdot l(1-l)$$

over $\mathcal{B}\Big(\theta, \frac{1}{X_{\max}}\sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)}\Big)$, where l satisfies

$$\big|\, l - \pi^*_\theta(i)\,\big| \leq \tfrac{1}{2}\sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)}.$$

We denote $\pi_\theta^*(i)$ as a. Thus, proving Lemma 3.3 is equivalent as proving

$$f(a) \coloneqq \max_{l \in [a - \frac{\sqrt{a(1-a)}}{2}, a + \frac{\sqrt{a(1-a)}}{2}]} \frac{4l(1-l)}{\sqrt{a(1-a)}} \leq \frac{5}{2}.$$

WLOG, we assume $a \in [0, \frac{1}{2}]$ and consider two cases.

Case 1: When $a \in [\frac{1}{2} - \frac{\sqrt{5}}{10}, \frac{1}{2}]$, we know that

$$\frac{1}{2} \in [a - \frac{\sqrt{a(1-a)}}{2}, a + \frac{\sqrt{a(1-a)}}{2}],$$

which implies that

$$f(a) = \frac{1}{\sqrt{a(1-a)}} \leq f(\frac{1}{2} - \frac{\sqrt{5}}{10}) = \sqrt{5} \leq \frac{5}{2}.$$

Case 2: When $a \in [0, \frac{1}{2} - \frac{\sqrt{5}}{10}]$, we know that

$$\frac{1}{2} \notin [a - \frac{\sqrt{a(1-a)}}{2}, a + \frac{\sqrt{a(1-a)}}{2}].$$

Because $g(l) = l(1-l)$ is monotonically increasing in $[0, \frac{1}{2}]$, we can obtain

$$f(a) = \frac{4(a + \frac{\sqrt{a(1-a)}}{2})(1 - a - \frac{\sqrt{a(1-a)}}{2})}{\sqrt{a(1-a)}} = 3\sqrt{a(1-a)} + (2 - 4a).$$

$f(a)$ takes its maximum when $a = \frac{1}{10}$ and $f(a) = \frac{5}{2}$.

Combining the above two cases, we conclude the lemma.

D Convergence Analysis under Orthogonal Representation Assumption

D.1 Auxiliary Lemma

Lemma D.1. *Under Assumption 1 and 3, for any $i, j \in [n], i \neq j$ and $\theta \in \mathbb{R}^d$, we have*

$$\nabla J_i(\theta)^\top \nabla J_j(\theta) = 0 \tag{17}$$

Proof. According to (10), we get

$$\begin{aligned}
\nabla J_i(\theta)^\top \nabla J_j(\theta) &= r_i^\top \left(\operatorname{diag}\left(\pi_\theta(i)\right) - \pi_\theta(i)\pi_\theta^\top(i)\right) X_i X_j^\top \left(\operatorname{diag}\left(\pi_\theta(j)\right) - \pi_\theta(j)\pi_\theta^\top(j)\right) r_j \\
&= r_i^\top \left(\operatorname{diag}\left(\pi_\theta(i)\right) - \pi_\theta(i)\pi_\theta^\top(i)\right) 0 \left(\operatorname{diag}\left(\pi_\theta(j)\right) - \pi_\theta(j)\pi_\theta^\top(j)\right) r_j \\
&= 0,
\end{aligned}$$

where the second step is by Assumption 3. □

D.2 Proof of Theorem 3.1

We consider a specific question q_l. Combining Lemma D.1 with *log-linear policy parameterization* in our setting, if question q_{i_t} is selected on iteration t in Algorithm 1, we get

$$\begin{aligned}
J_j(\theta_t) &= J_j(\theta_{t-1} + \eta \nabla J_i(\theta_{t-1})) \\
&= J_j(\theta_{t-1})
\end{aligned} \tag{18}$$

for any $i_t \neq l$. That is, the parameter update on question q_{i_t} will not affect the expected reward on other questions.

If question $i_t = l$ is selected on iteration t in Algorithm 1, we have

$$\begin{aligned} J_l(\theta_t) - J_l(\theta_{t-1}) &\geq \langle \theta_t - \theta_{t-1}, \nabla J_l(\theta_{t-1}) \rangle - \frac{X_{\max}^2}{2} \|\theta_t - \theta_{t-1}\|^2 \\ &= (\eta - \frac{X_{\max}^2}{2}\eta^2) \|\nabla J_l(\theta_{t-1})\|^2 \\ &= \frac{1}{2X_{\max}^2} \|\nabla J_l(\theta_{t-1})\|^2 \end{aligned} \tag{19}$$

where the first step is by Lemma 3.1, which also indicate that $J_i(\theta)$ is $X_{\max}^2$-weakly convex. Combining (18) and (19) and taking the expectation over i_t, we get

$$\mathbb{E}[J_l(\theta_t)] - \mathbb{E}[J_l(\theta_{t-1})] \geq \frac{1}{2nX_{\max}^2} \mathbb{E}\|\nabla J_l(\theta_{t-1})\|^2. \tag{20}$$

Summing up (20) for $t = 1, \ldots, T$, we have

$$\frac{1}{2nX_{\max}^2} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla J_l(\theta_{t-1})\|^2] \leq \mathbb{E}[J_l(\theta_T)] - J_l(\theta_0) \leq 1 - \pi_{\theta_0}^*(l).$$

This directly leads to

$$\min_{t \in \{0,1,\ldots,T-1\}} \mathbb{E}[\|\nabla J_l(\theta_t)\|^2] \leq \frac{2n(1 - \pi_{\theta_0}^*(l)) X_{\max}^2}{T}.$$

D.3 Proof of Theorem 3.2

Similar to (18) in the proof of Theorem 3.1, the gradient update based on question q_i does not affect the objective for question q_l if $i \neq l$. That is,

$$J_l(\theta_t) = \begin{cases} J_l(\theta_{t-1}), & \text{if } i_t \neq l \\ J_l(\theta_t), & \text{if } i_t = l. \end{cases} \tag{21}$$

Consider the case where $i_t = l$, from the parameter update rule in Algorithm 2, we get

$$\theta_t = \theta_{t-1} + \eta \Big(\sqrt{\pi_{\theta_{t-1}}^*(l)(1 - \pi_{\theta_{t-1}}^*(l))} x_{l,a_l} - \sqrt{\frac{\pi_{\theta_{t-1}}^*(l)}{1 - \pi_{\theta_{t-1}}^*(l)}} \sum_{j \neq a_l} [\pi_{\theta_{t-1}}(l)]_j \cdot x_{l,j} \Big).$$

Also, by setting $\eta = \frac{2}{5X_{\max}^2}$, we have

$$\begin{aligned} &\|\eta \Big(\sqrt{\pi_{\theta_{t-1}}^*(l)(1 - \pi_{\theta_{t-1}}^*(l))} x_{l,a_l} - \sqrt{\frac{\pi_{\theta_{t-1}}^*(l)}{1 - \pi_{\theta_{t-1}}^*(l)}} \sum_{j \neq a_l} [\pi_{\theta_{t-1}}(l)]_j \cdot x_{l,j} \Big) \| \\ =& \frac{2}{5X_{\max}^2} \| \Big(\sqrt{\pi_{\theta_{t-1}}^*(l)(1 - \pi_{\theta_{t-1}}^*(l))} x_{l,a_l} - \sqrt{\frac{\pi_{\theta_{t-1}}^*(l)}{1 - \pi_{\theta_{t-1}}^*(l)}} \sum_{j \neq a_l} [\pi_{\theta_{t-1}}(l)]_j \cdot x_{l,j} \Big) \| \\ \leq& \frac{1}{2X_{\max}^2} \Big(\sqrt{\pi_{\theta_{t-1}}^*(l)(1 - \pi_{\theta_{t-1}}^*(l))} \|x_{l,a_l}\| + \sqrt{\frac{\pi_{\theta_{t-1}}^*(l)}{1 - \pi_{\theta_{t-1}}^*(l)}} \sum_{j \neq a_l} [\pi_{\theta_{t-1}}(l)]_j \cdot \|x_{l,j}\| \Big) \\ \leq& \frac{1}{2X_{\max}^2} (2\sqrt{\pi_{\theta_{t-1}}^*(l)(1 - \pi_{\theta_{t-1}}^*(l))} X_{\max}) \\ =& \frac{1}{X_{\max}} \cdot \sqrt{\pi_{\theta_{t-1}}^*(l)(1 - \pi_{\theta_{t-1}}^*(l))}. \end{aligned}$$

This implies that $\theta_t \in \mathcal{B}(\theta, \frac{1}{X_{\max}} \cdot \sqrt{\pi^*_{\theta_{t-1}}(l)(1-\pi^*_{\theta_{t-1}}(l))})$. According to Lemma 3.3, we could obtain

$$\begin{aligned} J_l(\theta_t) &\geq J_l(\theta_{t-1}) + \langle \theta_t - \theta_{t-1}, \nabla J_l(\theta_{t-1}) \rangle - \frac{5}{4} X_{\max}^2 \cdot \sqrt{\pi^*_{\theta_{t-1}}(l)(1-\pi^*_{\theta_{t-1}}(l))} \|\theta_t - \theta_{t-1}\|^2 \\ &= J_l(\theta_{t-1}) + \frac{1}{5X_{\max}^2 \sqrt{\pi^*_\theta(l)(1-\pi^*_\theta(l))}} \|\nabla J_l(\theta_{t-1})\|^2 \\ &\geq J_l(\theta_{t-1}) + \frac{1}{5X_{\max}^2 C_{l,t-1}} \|\nabla J_l(\theta_{t-1})\|^2 \end{aligned} \tag{22}$$

where the last step is by Assumption 2. Combining (21) and (22) and taking the expectation over i_t, we obtain

$$\mathbb{E}[J_l(\theta_t)] \geq \mathbb{E}[J_l(\theta_{t-1})] + \frac{1}{5nX_{\max}^2 C_{l,t-1}} \mathbb{E}[\|\nabla J_l(\theta_{t-1})\|^2]. \tag{23}$$

Summing up (23) for $t = 1, \ldots, T$, we get

$$\mathbb{E}[J_l(\theta_T)] \geq J_l(\theta_0) + \sum_{t=0}^{T-1} \frac{1}{5nX_{\max}^2 C_{l,t}} \mathbb{E}[\|\nabla J_l(\theta_{t-1})\|^2]. \tag{24}$$

According to the Cauchy-Schwarz inequality, we have

$$\min_{t \in \{0,1,\ldots,T-1\}} \mathbb{E}[\|\nabla J_l(\theta_t)\|^2] \leq \frac{2n(1-\pi^*_{\theta_0}(l))X_{\max}^2}{T} \frac{5\sum_{t=0}^{T-1} C_{l,t}}{2T}.$$

E Analysis of Importance Sampling GRPO

This appendix proves the results in Section 3.4.

E.1 Auxiliary Lemma

Lemma E.1. *Under Assumption 1, for all $i \in [n]$ and $\theta, \theta' \in \mathbb{R}^d$, with $\delta := X_{\max}\|\theta - \theta'\|$,*

$$e^{-2\delta}\, \mathbb{V}[\pi_\theta(i)] \;\leq\; \mathbb{V}[\pi_{\theta'}(i)] \;\leq\; e^{2\delta}\, \mathbb{V}[\pi_\theta(i)].$$

Proof. According to the definition of $\mathbb{V}[\pi_\theta(i)]$, we know

$$\mathbb{V}[\pi_\theta(i)] = J_i(\theta)\big(1 - J_i(\theta)\big).$$

Since $J(\theta) \in (0,1)$, $\log \mathbb{V}[\pi_\theta(i)]$ is differentiable with

$$\nabla \log \mathbb{V}[\pi_\theta(i)] = \frac{\nabla \mathbb{V}[\pi_\theta(i)]}{\mathbb{V}[\pi_\theta(i)]} = \frac{\big(1 - 2J_i(\theta)\big)\, \nabla J_i(\theta)}{\mathbb{V}[\pi_\theta(i)]}.$$

By Lemma 3.2 and $|1 - 2J(\theta)| \leq 1$, for all $\theta \in \mathbb{R}^d$,

$$\|\nabla \log \mathbb{V}[\pi_\theta(i)]\| \leq \frac{|1 - 2J(\theta)| \cdot 2\, \mathbb{V}[\pi_\theta(i)]\, X_{\max}}{\mathbb{V}[\pi_\theta(i)]} \leq 2X_{\max}. \tag{25}$$

Thus, $\log \mathbb{V}[\pi_\theta(i)]$ is $2X_{\max}-$Lipschitz over $\mathbb{R}^d$, which directly leads to

$$|\log \mathbb{V}[\pi_{\theta'}(i)] - \log \mathbb{V}[\pi_\theta(i)]| \leq 2X_{\max}\|\theta' - \theta\| = 2\delta.$$

Taking exponent on both sides gives $e^{-2\delta}\mathbb{V}[\pi_\theta(i)] \leq \mathbb{V}[\pi_{\theta'}(i)] \leq e^{2\delta}\mathbb{V}[\pi_\theta(i)]$. □

Lemma E.2. *Suppose Assumption 1 holds. Then, for every $t \geq 0$, we have*

$$\mathbb{E}\big[g_t\big] = \frac{n}{2\sum_{i=1}^n \sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}} \nabla J(\theta_t), \quad \|g_t\| \leq X_{\max}/2.$$

Proof. We fix $t \geq 0$. As in the proof of Lemma E.3, the index i_t is drawn according to the positive probabilities $p_{t,i}$. For the first statement, we know

$$\mathbb{E}[g_t] = \sum_{i=1}^{n} p_{t,i} \frac{\nabla J_i(\theta_t)}{2\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}} = \frac{n}{2\sum_{i=1}^{n}\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}} \cdot \frac{1}{n}\sum_{i=1}^{n} \nabla J_i(\theta_t) = \frac{n}{2\sum_{i=1}^{n}\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}} \nabla J(\theta_t).$$

For the second statement, the definition of p_{t,i_t} gives

$$\|g_t\| = \frac{\|\nabla J_{i_t}(\theta_t)\|}{2\sqrt{\mathbb{V}[\pi_{\theta_t}(i_t)]}}.$$

By Lemma 3.2, we know $\|\nabla J_{i_t}(\theta_t)\| \leq 2X_{\max}\,\mathbb{V}[\pi_{\theta_t}(i_t)]$. Therefore,

$$\|g_t\| \leq X_{\max}\sqrt{\mathbb{V}[\pi_{\theta_t}(i_t)]} \leq \frac{X_{\max}}{2},$$

which completes the proof. □

Lemma E.3. *Suppose Assumption 1 holds. Then, for every $t \geq 0$, we have*

$$\mathbb{E}\big[\|g_t\|^2\big] \;\leq\; X_{\max}^2 \frac{\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)]^{\frac{3}{2}}}{\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)]^{\frac{1}{2}}}.$$

Proof. We fix $t \geq 0$. According to Algorithm 3, the index i_t is drawn according to the probabilities

$$p_{t,i} = \frac{\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}}{\sum_{j=1}^{n}\sqrt{\mathbb{V}[\pi_{\theta_t}(j)]}}$$

for $i \in [n]$. Since $\mathbb{V}[\pi_{\theta_t}(i)] \in (0, \frac{1}{4}]$, so every $p_{t,i} > 0$. Consequently,

$$\mathbb{E}\big[\|g_t\|^2\big] = \sum_{i=1}^{n} p_{t,i}\frac{\|\nabla J_i(\theta_t)\|^2}{4\mathbb{V}[\pi_{\theta_t}(i)]} = \frac{1}{\sum_{j=1}^{n}\sqrt{\mathbb{V}[\pi_{\theta_t}(j)]}} \cdot \sum_{i=1}^{n}\frac{\|\nabla J_i(\theta_t)\|^2}{4\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}}.$$

According to Lemma 3.2, we know $\|\nabla J_i(\theta_t)\|^2 \leq 4X_{\max}^2\,\mathbb{V}[\pi_{\theta_t}(i)]^2$, which directly leads to

$$\frac{\|\nabla J_i(\theta_t)\|^2}{\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}} \leq 4X_{\max}^2\,\mathbb{V}[\pi_{\theta_t}(i)]^{3/2}.$$

Thus, we obtain

$$\mathbb{E}\big[\|g_t\|^2\big] \leq X_{\max}^2 \frac{\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)]^{\frac{3}{2}}}{\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)]^{\frac{1}{2}}},$$

which completes the proof. □

Lemma E.4. *For any fixed $t \geq 0$ and positive constants $\{l_i\}_{i=1}^n$, we have*

$$\big(\sum_i l_i^2\big)\big(\sum_i l_i^3\big) \;\leq\; \big(\sum_i l_i\big)\big(\sum_i l_i^4\big),$$

where the inequality is strict unless $l_1 = \cdots = l_n$.

Proof. We know

$$\begin{aligned}
\big(\sum_i l_i\big)\big(\sum_i l_i^4\big) - \big(\sum_i l_i^2\big)\big(\sum_i l_i^3\big) &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} l_i l_j \big(l_i - l_j\big)\big(l_i^2 - l_j^2\big) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} l_i l_j \big(l_i - l_j\big)^2\big(l_i + l_j\big) \\
&\geq 0,
\end{aligned}$$

where the last step is because $l_i l_j \big(l_i - l_j\big)^2\big(l_i + l_j\big)$ is nonnegative, and it is strictly positive whenever $l_i \neq l_j$. Hence

$$\big(\sum_i l_i^2\big)\big(\sum_i l_i^3\big) \;\leq\; \big(\sum_i l_i\big)\big(\sum_i l_i^4\big),$$

and the inequality is strict unless $l_1 = \cdots = l_n$, which completes the proof. □

E.2 Proof of Theorem 3.4

We fix $t \geq 0$. In Algorithm 1, the index i_t is drawn uniformly from $[n]$ and

$$\theta_t = \theta_{t-1} + \eta \, \nabla J_{i_t}(\theta_{t-1}).$$

Acoording to Lemma 3.2 and $\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]} \leq \frac{1}{2}$, the update is bounded by

$$\|\nabla J_{i_t}(\theta_t)\| \leq 2X_{\max} \, \mathbb{V}[\pi_{\theta_t}(i_t)] \leq \frac{X_{\max}}{2}.$$

Combining Lemma 3.2 with Assumption 2, the expected second moment of update satisfies

$$\mathbb{E}\big[\|\nabla J_{i_t}(\theta_t)\|^2\big] = \frac{1}{n}\sum_{i=1}^{n} \|\nabla J_i(\theta_t)\|^2 \leq \frac{4X_{\max}^2}{n} \sum_{i=1}^{n} \mathbb{V}[\pi_{\theta_t}(i)]^2 \leq 4X_{\max}^2 \, \mu_{4,t}. \tag{26}$$

Since $0 < \eta \leq \frac{1}{2X_{\max}^2}$, we know

$$\|\theta_t - \theta_{t-1}\| = \eta \|\nabla J_{i_t}(\theta_{t-1})\| \leq \eta X_{\max}/2 \leq \frac{1}{4X_{\max}}.$$

For every θ' on the segment $[\theta_t, \theta_{t+1}]$ satisfies

$$\delta := X_{\max}\|\theta' - \theta_{t-1}\| \leq \eta X_{\max}^2/2 \leq \frac{1}{4}.$$

Combining with Lemma 3.1 and Lemma E.1, we know

$$\begin{aligned}
\|\nabla^2 J(\theta')\| &\leq \frac{1}{n}\sum_{i=1}^{n} \|\nabla^2 J_i(\theta')\| \\
&\leq \frac{4X_{\max}^2}{n} \sum_{i=1}^{n} \mathbb{V}[\pi_{\theta'}(i)] \\
&\leq \frac{4e^{2\delta} X_{\max}^2}{n} \sum_{i=1}^{n} \mathbb{V}[\pi_{\theta_t}(i)] \\
&\leq 8X_{\max}^2 \, \mu_{2,t}.
\end{aligned}$$

Applying Taylor's theorem, we could obtain

$$J(\theta_{t+1}) \geq J(\theta_t) + \eta \, \langle \nabla J(\theta_t), \nabla J_{i_t}(\theta_t) \rangle - 4X_{\max}^2 \mu_{2,t} \, \eta^2 \, \|\nabla J_{i_t}(\theta_t)\|^2.$$

Combining (26) and taking expectation, we have

$$\mathbb{E}[J(\theta_{t+1})] \geq J(\theta_t) + \eta \, \|\nabla J(\theta_t)\|^2 - 16\eta^2 X_{\max}^4 \, \mu_{2,t} \, \mu_{4,t}.$$

Summing over $t = 1, \ldots, T$, and using $J \leq 1$ together with the definition of $D(T)$,

$$\eta \sum_{t=0}^{T-1} \mathbb{E}\|\nabla J(\theta_t)\|^2 \leq 1 - J(\theta_0) + 8\eta^2 X_{\max}^4 \, T \, D(T).$$

Dividing by ηT gives (7), which completes the proof.

E.3 Proof of Theorem 3.5

We fix $t \geq 0$. In Algorithm 3, the update is

$$\theta_{t+1} = \theta_t + \eta \, g_t.$$

According to Lemma E.2, $\|\theta_{t+1} - \theta_t\| = \eta\|g_t\| \leq \eta X_{\max}/2 \leq \frac{1}{4X_{\max}}$, so every θ' on the segment $[\theta_t, \theta_{t+1}]$ again satisfies

$$\delta := X_{\max}\|\theta' - \theta_t\| \leq \frac{1}{4}.$$

Combining with Lemma 3.1 and Lemma E.1, we know

$$\begin{aligned}\|\nabla^2 J(\theta')\| &\le \frac{1}{n}\sum_{i=1}^{n}\|\nabla^2 J_i(\theta')\| \\ &\le \frac{4X_{\max}^2}{n}\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta'}(i)] \\ &\le \frac{4e^{2\delta}X_{\max}^2}{n}\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)] \\ &\le 8X_{\max}^2\,\frac{\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)]}{n}.\end{aligned}$$

Applying Taylor's theorem, we obtain

$$J(\theta_{t+1}) \ge J(\theta_t) + \eta\,\langle\nabla J(\theta_t), g_t\rangle - 4X_{\max}^2\,\frac{\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)]}{n}\cdot\eta^2\,\|g_t\|^2.$$

Combining with Lemmas E.2 and E.3, we could obtain

$$\begin{aligned}\mathbb{E}[J(\theta_{t+1})] &\ge J(\theta_t) + \frac{n\cdot\eta}{2\sum_{i=1}^{n}\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}}\,\|\nabla J(\theta_t)\|^2 - 4\eta^2X_{\max}^4\,\frac{(\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)]^{\frac{3}{2}})(\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)])}{n(\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)]^{\frac{1}{2}})} \\ &\ge J(\theta_t) + \frac{n\cdot\eta}{2\sum_{i=1}^{n}\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}}\,\|\nabla J(\theta_t)\|^2 - 4\eta^2X_{\max}^4\,\frac{\sum_{i=1}^{n}\mathbb{V}[\pi_{\theta_t}(i)]^2}{n}\end{aligned}$$

We sum over $t = 0, \ldots, T-1$ and obtain

$$\sum_{t=0}^{T-1}\frac{n\cdot\eta}{2\sum_{i=1}^{n}\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}}\mathbb{E}\|\nabla J(\theta_t)\|^2 \le 1 - J(\theta_0) + 4\eta^2X_{\max}^4\sum_{t=0}^{T-1}\mu_{4,t}.$$

Applying $\frac{n\cdot\eta}{2\sum_{i=1}^{n}\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}} \ge \frac{\eta}{2\mu_{1,t}}$, we obtain (8), which completes the proof.

F Additional Experimental Results

F.1 Experimental Setup Details

Normalization strategies. All schemes use the per-prompt mean as baseline and differ in the scaling factor:

- **Per-prompt std** (standard GRPO): $\hat{A}_i(o) = \frac{r_i - \text{mean}_i(r)}{\text{std}_i(r)}$, with mean and std computed within each prompt's group of G rollouts.
- **Global std**: $\hat{A}_i(o) = \frac{r_i - \text{mean}_i(r)}{\text{std}_{\text{global}}(r)}$, with the std computed over all rollouts of the mini-batch and shared across prompts. Both per-prompt and global std rescale by a standard deviation; only the former adapts the factor to each prompt.
- **REINFORCE** (unnormalized): $\hat{A}_i(o) = r_i - \text{mean}_i(r)$.

IS-GRPO in practice. Algorithm 3 samples prompt i with probability $\propto \sqrt{\mathbb{V}[\pi_{\theta_t}(i)]}$ and updates it with $\nabla J_i/(2\sqrt{\mathbb{V}[\pi_{\theta_t}(i)]})$. The true standard deviations are unknown, and estimating them for all prompts at every iteration would require rolling out the whole training set. We therefore keep a tracker s_i for each of the training prompts:

- *Initialization.* $s_i = 1/2$, the largest possible value, so that every prompt is likely to be drawn at least once.
- *Update.* Whenever prompt i is rolled out, $s_i \leftarrow 0.9\,s_i + 0.1\sqrt{\hat{p}_i(1-\hat{p}_i)}$, with $\hat{p}_i$ the fraction of correct rollouts; the moving average smooths the noise of an estimate from $G = 8$ samples.

Each batch of 16 prompts is drawn with probability $\propto s_i$, so prompts that are nearly always solved or nearly always failed, which carry little gradient, are rarely drawn. Each drawn prompt's advantage is divided by its within-group std, which implements the factor $1/\sqrt{\mathbb{V}}$ of Algorithm 3 exactly as in GRPO; the constant $1/2$ is absorbed into the learning rate. Since s_i is a noisy and lagged estimate of the true standard deviation, this is an approximation of Algorithm 3 (see also Section F.6).

Model scales.

- **1.5B:** Qwen2.5-Math-1.5B with LoRA on all linear layers ($r = 64$, $\alpha = 128$), 3 seeds, trained on GSM8K.
- **7B:** Qwen2.5-Math-7B with LoRA on all linear layers ($r = 32$, $\alpha = 32$), in two runs: MATH Level 3–5 with held-out evaluation on AMC (1 seed, 400 iterations), and MATH Level 1–5 with evaluation on the training prompts (3 seeds, 500 iterations, including IS-GRPO); see Section F.5.

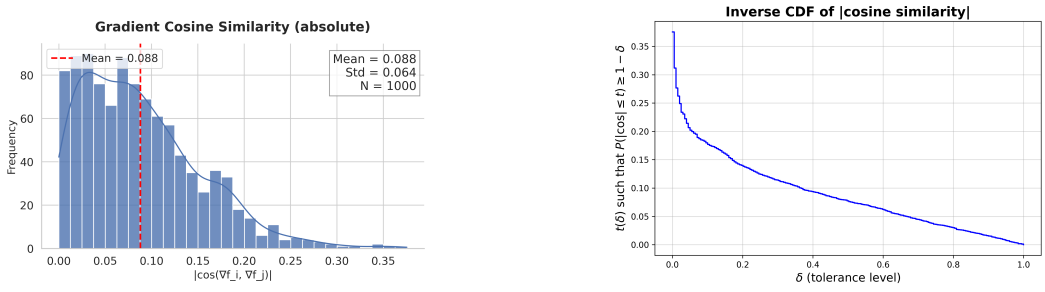
Learning rate control. Each strategy was swept over $\eta \in \{5\times10^{-7}, 1\times10^{-6}, 5\times10^{-6}, 1\times10^{-5}, 5\times10^{-5}\}$ at 1.5B scale. All methods are reported at the REINFORCE baseline's optimal learning rate.

F.2 Validation of the Orthogonal Assumption

Assumption 3 posits that representations of different prompts are nearly orthogonal. We check this at two levels with Qwen2.5-Math-1.5B on GSM8K: the gradient level during training is presented in Section 4.1; here we add a representation-level check at initialization.

Representation level. We sample 1,000 question pairs, extract penultimate-layer hidden states and compute absolute cosine similarities. Similarities concentrate near zero (mean ≈ 0.088, std ≈ 0.064), and over 90% of pairs fall below 0.15 (Figure 5).

Relation to the theory. Gradient orthogonality is a necessary consequence of Assumption 3 under the log-linear parameterization, so these statistics test a necessary condition. Together they support Assumption 3 as an accurate description early in training and document its gradual erosion. Theorem 3.2 relies on exact orthogonality; Theorems 3.4 and 3.5 do not. We use the spread of the cosine distribution as a proxy for cross-prompt interference; it marks the transition from Phase II to Phase III in Section 4.2.1; its growth is consistent with the 1.5B gaps no longer widening late in training.



(a) Histogram of $|\cos|$

(b) Inverse CDF of $|\cos|$

Figure 5: Validation of Assumption 3 (1.5B, GSM8K, 1,000 question pairs).

F.3 Fisher Information Estimation Details

Given a mini-batch of prompts $\{q_i\}_{i=1}^B$ at iteration t, we estimate the diagonal Fisher information as follows:

1. Sample responses $\hat{o}_i \sim \pi_{\theta_t}(\cdot|q_i)$ for each prompt.
2. Compute the mini-batch log-likelihood gradient $\nabla\hat{\mathcal{L}}_B(\theta_t) = \frac{1}{B}\sum_{i=1}^B \nabla \log \pi_{\theta_t}(\hat{o}_i|q_i)$.

3. Estimate the diagonal Fisher via the outer-product estimator $h(\theta_t) := B \cdot \nabla\hat{\mathcal{L}}_B(\theta_t) \odot \nabla\hat{\mathcal{L}}_B(\theta_t)$, where $\odot$ is element-wise multiplication; $\mathbb{E}_{\hat{o}}[h(\theta_t)] = \mathbb{E}_{\hat{o}}[\mathrm{diag}(\hat{F}(\theta_t))]$.

The per-prompt curvature proxy is $\|h_i(\theta_t)\|_1$, the ℓ_1 norm of the prompt-level diagonal Fisher entries. The Fisher information of the log-likelihood is related to, but not identical with, the Hessian of J_i bounded in Lemma 3.1; we use it because it is cheap to estimate at scale. Table 1 correlates this proxy with the per-prompt reward variance: the association is positive and significant when both are measured at the same checkpoint, and absent when they are measured at different checkpoints, as expected if variance tracks the curvature at the current parameters.

F.4 Detailed Phase-by-Phase Analysis (1.5B)

This section expands on Section 4.2.1. Table 3 gives the final accuracies with across-seed standard deviations.

Table 3: 1.5B: final accuracy (%, mean ± std over 3 seeds) at iteration 500.

Method	GSM8K-Easy	GSM8K-Hard	MATH Lv. 2
Per-prompt std	**97.2±0.4**	**87.1±1.2**	**58.3±0.9**
Global std	96.1±0.5	85.3±1.3	55.7±1.1
REINFORCE	95.1±0.6	84.2±1.5	54.1±1.3

Convergence factor $C(T)$. Figure 6 tracks $c_t = \frac{5}{2n}\sum_i \sqrt{\pi^*_{\theta_t}(i)(1-\pi^*_{\theta_t}(i))}$ and its running average $C(T)$. On both splits $C(T)$ stays below 1 throughout training, so the speed-up condition (6) of Theorem 3.2 holds, and it ends at 0.28 on Easy and 0.45 on Hard. It falls faster on Easy, where more prompts saturate early. On Hard, c_t first rises (from 0.42 to 0.56) as prompts leave the all-incorrect regime and become active, then falls as they are solved. Since $C(T) < 1$ already holds in Phase I, where the methods still overlap, the condition is necessary rather than sufficient for a visible gap: the gap additionally requires the scaling factors of active prompts to differ.

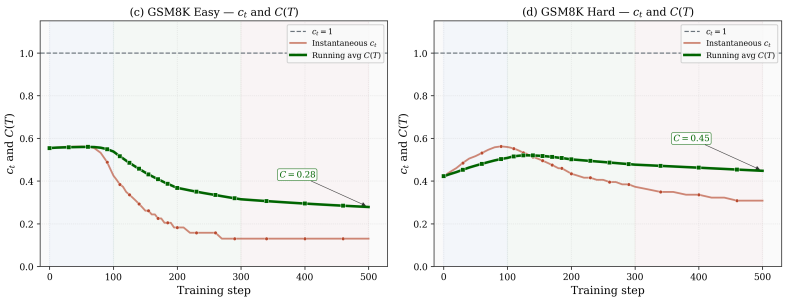


Figure 6: c_t and $C(T)$ on GSM8K Easy and Hard (1.5B). $C(T)$ stays below 1 throughout training, ending at 0.28 (Easy) and 0.45 (Hard).

Phase I (iterations 0–100). All three schemes overlap. On GSM8K-Easy all methods improve from ≈ 0.45 to ≈ 0.70; on GSM8K-Hard from ≈ 0.20 to ≈ 0.40. With $G = 8$ rollouts a large fraction of groups are all-correct or all-incorrect and receive zero advantage under every scheme; the prompts that do receive a gradient have similar success probabilities, so their scaling factors are nearly uniform and per-prompt and global normalization coincide.

Phase II (iterations 100–300). As accuracies spread out, c_t falls and per-prompt std establishes a widening advantage. On GSM8K-Easy it reaches ≈ 0.95 vs. ≈ 0.93 (global std) and ≈ 0.92 (REINFORCE); on GSM8K-Hard the separation is larger, ≈ 0.85 vs. ≈ 0.80 and ≈ 0.78. Pairwise gradient cosines remain small in this phase (Section 4.1).

Phase III (iterations 300–500). Per-prompt std keeps its advantage but the gap no longer widens. On GSM8K-Easy all methods stabilize near 0.96–0.98 with a $\sim$2 pp lead; on GSM8K-Hard they fluctuate around 0.80–0.88 with a $\sim$3 pp lead. The growing spread of gradient cosines (std ≈ 0.130) coincides with the end of the widening; on Easy, the approach to the accuracy ceiling is a simpler explanation for it.

Final gaps. Per-prompt std outperforms global std by 1.1 pp (Easy), 1.8 pp (Hard) and 2.6 pp (MATH Lv. 2), and REINFORCE by 2.1 pp, 2.9 pp and 4.2 pp respectively (Table 3).

F.5 7B Experiments: Details

Held-out run (MATH, AMC). One seed per scheme, 400 iterations, learning rate 5×10^{-6}; AMC accuracy is measured with greedy decoding every 10 iterations. With a single seed and a small benchmark, individual checkpoints are noisy (Figure 3a); we rely on the ordering that persists over the last 150 iterations rather than on any single gap. At iteration 400, per-prompt std leads global std by $\approx$ 2 pp and REINFORCE by $\approx$ 6 pp.

Second held-out benchmark (AIME). On the same MATH Level 3–5 run we additionally evaluate on AIME 2024 (Section F.5), a harder, out-of-distribution benchmark. With a single seed and only 30 AIME problems, we do not read a remarkable result but at least we can see some trends shows a good tendency by using normalization from this run.

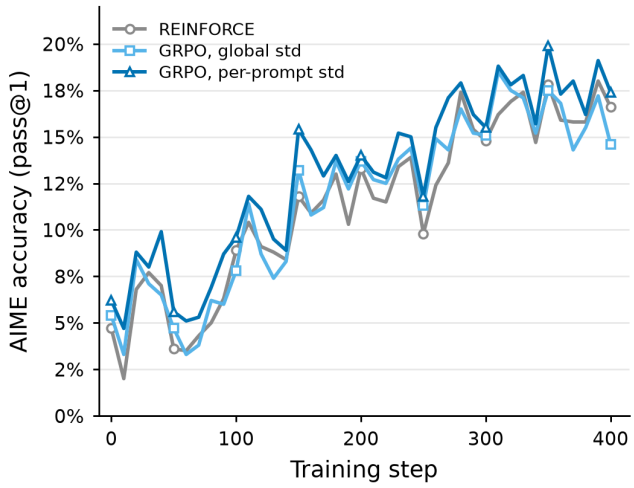


Figure 7: 7B held-out accuracy (pass@1) on AIME 2024, MATH Level 3–5 training run, one seed.

Training-prompt run (MATH Level 1–5). Three seeds per method, 500 iterations, 16 prompts and $G = 8$ rollouts per iteration; greedy accuracy on the 256 training prompts at iterations 0, 50, 100, 200, 300, 400, 450 and 500. The model solves 68% of the training prompts at iteration 0. Table 4 reports the mean of the last three evaluations, whose across-seed standard deviation is 0.6 pp, against 1.8 pp for a single late checkpoint and 2.3 pp for the early ones.

Why the normalization schemes coincide in the training-prompt run. Three features of this run reduce the leverage of per-prompt normalization. (i) *Resolution.* With binary rewards and $G = 8$, a non-degenerate group has $\hat{p} \in \{1/8, \ldots, 7/8\}$, so the estimated factor $1/\sqrt{\hat{p}(1-\hat{p})}$ lies in $[2.0, 3.0]$: per-prompt normalization can reweight two prompts of a batch by at most $1.5\times$, whereas the factor $1/\sqrt{\mathbb{V}}$ of the analysis is unbounded as $\mathbb{V} \to 0$. (ii) *Optimizer.* AdamW is largely invariant to rescaling the gradient by a constant, so the factor shared by per-prompt and global std is absorbed and only the relative reweighting of (i) remains; a factor that varies across iterations, such as global std's, is not fully absorbed. (iii) *Base accuracy.* With 68% of prompts solved at iteration 0, the prompts that receive a gradient are a minority with similar success probabilities. The

Table 4: 7B, MATH Level 1–5: greedy accuracy (%) on the training prompts, 3 seeds. Late window: mean over iterations 400/450/500 ($\pm$ across-seed std). Step 500: single checkpoint (across-seed std ≈ 1.8 pp).

Method	Late window	Step 500
IS-GRPO	$\mathbf{87.1 \pm 0.6}$	88.7
REINFORCE	86.1 ± 0.6	86.3
Per-prompt std	85.9 ± 0.6	87.1
Global std	85.4 ± 0.6	85.9

held-out run starts from a harder training set, where active prompts are more numerous and more heterogeneous.

Early lead of IS-GRPO. At iteration 50 IS-GRPO leads REINFORCE by 3.5 pp, against an across-seed standard deviation of 2.3 pp. This coincides with that every training prompt has been rolled out at least once: once solved prompts are identified they are no longer drawn, so each batch consists of prompts that carry a gradient.

F.6 Trajectory Coefficients: Measurement Details

Estimator. For the 7B IS-GRPO run (seed 1, Section 4.3) we take $s_{i,t}$ from the tracker of Section F.1, i.e. $s_{i,t} = \sqrt{\hat{p}_i(1-\hat{p}_i)}$ with population variance, smoothed by the EMA; it is bounded by $1/2$ as Assumption 2 requires. Prompts with $s_{i,t} = 0$ are kept in the averages $\mu_{k,t} = \frac{1}{n}\sum_i s_{i,t}^k$. From these we compute $c_t = \frac{5}{2}\mu_{1,t}$, $C(T)$, $D(T) = \frac{2}{T}\sum_{t<T}\mu_{2,t}\mu_{4,t}$ and $D_{\mathrm{IS}}(T) = \sum_{t<T}\mu_{4,t} \big/ \sum_{t<T}\mu_{1,t}^{-1}$ exactly as defined in Sections 3.3 and 3.4. All coefficients are measured along the IS-GRPO trajectory, whereas Theorem 3.4 refers to $D(T)$ along REINFORCE's own trajectory; the comparison is therefore a same-trajectory approximation.

Biases. Three biases affect $s_{i,t}$ and do not have a common direction: sampling noise of $\hat{p}_i$ from eight rollouts spreads prompts with equal true p_i apart and inflates the measured heterogeneity; $\sqrt{\hat{p}(1-\hat{p})}$ from eight samples is biased low; and the tracker of a prompt lags the current policy until that prompt is drawn again. Recomputing $\mu_{k,t}$ from many rollouts per prompt at saved checkpoints would remove all three.

Two kinds of heterogeneity. The heterogeneity that develops along the run is dominated by the contrast between prompts whose last group was all-correct or all-incorrect ($s_{i,t} = 0$) and active prompts. Per-prompt normalization cannot exploit this contrast, since a zero-variance group receives zero advantage under every scheme and normalization only reweights active prompts, whose estimated factors differ by at most $1.5\times$ (Section F.5). The selection rule of IS-GRPO acts on exactly this contrast by not drawing prompts with $s_{i,t} = 0$. This accounts for the pattern of Figure 3b: the normalization schemes coincide while IS-GRPO separates.

Why heterogeneity lowers $C(T)$. By concavity of $p \mapsto \sqrt{p(1-p)}$ and Jensen's inequality, the average of $\sqrt{\pi_i(1-\pi_i)}$ over prompts is smaller than $\sqrt{\bar{\pi}(1-\bar{\pi})}$ at the mean accuracy whenever per-prompt accuracies are heterogeneous. At any stage of training a substantial fraction of prompts sit near $\pi_i \approx 0$ or $\pi_i \approx 1$ and contribute little to the sum, which is why $C(T)$ falls well below the value the mean accuracy alone would suggest.

G Details on Datasets and Training

G.1 Datasets

GSM8K (Cobbe et al., 2021): grade-school math, partitioned into Easy ($N = 4{,}695$) and Hard ($N = 1{,}909$) using `Qwen2-7B-Instruct` as evaluator; used for 1.5B training, with GSM8K validation and MATH Level 1–2 for evaluation. **MATH** (Hendrycks et al., 2021): Level 3–5 and

Level 1–5 (256 problems) for the two 7B runs; Level 1–2 for 1.5B evaluation. **AMC, AIME**: held-out evaluation of the 7B Level 3–5 run.

G.2 Evaluation

1.5B: greedy pass@1 via regex extraction of the final answer. 7B: greedy decoding ($n = 1$, temperature $= 0$), on AMC for the Level 3–5 run and on the training prompts for the Level 1–5 run (Section F.5); Table 4 reports the results.

G.3 Training Configuration

Table 5: Hyperparameters. For 7B, entries with "/" give the Level 1–5 run / the Level 3–5 run.

	1.5B	**7B**
Base model	Qwen2.5-Math-1.5B	Qwen2.5-Math-7B
Training data	GSM8K Easy/Hard	MATH Lv. 1–5 / Lv. 3–5
Evaluation	GSM8K val, MATH Lv. 1–2	training prompts / AMC
LoRA r / α	64 / 128	32 / 32
LoRA target	all linear	all linear
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$)	
Learning rate	1×10^{-5}	5×10^{-6}
LR sweep (1.5B)	$\{5\times10^{-7}, 1\times10^{-6}, 5\times10^{-6}, 1\times10^{-5}, 5\times10^{-5}\}$	
Batch (prompts)	64	16
PPO mini-batch	32	16
Rollouts G	8	8
Iterations	500	500 / 400
Seeds	3	3 / 1
Hardware	H200 140GB	
Framework	verl + vLLM	

LR sweep. For 1.5B we ran 5 learning rates $\times$ 3 strategies $\times$ 3 seeds $= 45$ runs on GSM8K. Main-text figures use REINFORCE's optimal learning rate for all methods; per-method optimal results are qualitatively similar.