Why GRPO Needs Normalization: A Local-Curvature Perspective on Adaptive Gradients

Cheng Ge*
Department of Aeronautics and Astronautics
MIT
gec_mike@mit.edu

Caitlyn Heqi Yin*
Department of Statistics
University of Wisconsin–Madison
hyin66@wisc.edu

Hao Liang†
Department of Informatics
King's College London
hao.liang@kcl.ac.uk

Jiawei Zhang†
Department of Computer Sciences
University of Wisconsin–Madison
jzhang2924@wisc.edu

Abstract

Group Relative Policy Optimization, a critic-free, per-prompt REINFORCE-style method with within-prompt standardization, reliably stabilizes RL for LLM reasoning, but the mechanism is unclear. We show that within-prompt reward variance estimates the local curvature of the sequence-level policy gradient, so standard-deviation normalization implements a prompt-wise adaptive step size. Under a mild orthogonality assumption we prove faster convergence than unnormalized REINFORCE, and validate the effect on synthetic tasks and GSM8K.

1 Introduction

Lightweight, critic-free policy gradients (e.g., GRPO) are widely used to fine-tune large language models for multi-step reasoning because they avoid learning a value critic and are computationally cheap. GRPO samples multiple responses per prompt, subtracts the group mean, and normalizes advantages by the within-prompt standard deviation, which is a simple recipe that empirically improves stability and sample efficiency [17, 6].

What does this normalization actually do? Our key observation is that the reward variance for a prompt serves as a local estimate of the gradient's curvature: high-variance prompts correspond to regions where the policy gradient can be steep or noisy, requiring smaller effective step sizes. Standard-deviation normalization therefore behaves like a prompt-wise, iteration-wise adaptive learning-rate that rescales updates inversely to local curvature, improving both stability and convergence when curvature varies across prompts and over training.

We formalize this view in a sequence-level bandit framework. First, we relate per-prompt variance to a local Lipschitz/smoothness measure of the prompt-specific policy gradient. Second, under a mild *orthogonality* assumption on prompt representations (which decouples cross-prompt gradient interference), we prove that GRPO's variance normalization yields provably faster convergence than a single, global learning rate (REINFORCE). Finally, we corroborate our theory empirically via orthogonality checks, a difficulty-sliced evaluation on GSM8K, and synthetic high-variance tasks, where standard-deviation normalization consistently improves stability and final accuracy.

*Author order is alphabetical denoting equal contributions.
†Co-last authors

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

Contributions.

- We identify within-prompt reward variance as a proxy for local gradient curvature and show normalization implements an adaptive per-prompt step size.
- We prove faster convergence of normalized GRPO vs. unnormalized REINFORCE under a mild orthogonality assumption.
- We empirically validate the theory on synthetic regimes and GSM8K (difficulty-sliced), demonstrating improved stability and convergence under high variance.

2 Preliminaries and problem setting

We present a concise formulation of sequence-level RL with verifiable rewards (RLVR), along with the policy parameterization and update rules for GRPO and REINFORCE. Extra notations and update expressions are deferred to Appendix B.

Notation. For finite $\mathcal{X}$, let $\Delta(\mathcal{X})$ denote distributions over $\mathcal{X}$. All vectors are column vectors, and $\|\cdot\|$ denotes the Euclidean (or spectral) norm. For $v \in \mathbb{R}^m$, $\mathrm{diag}(v) \in \mathbb{R}^{m\times m}$ is the diagonal matrix with entries given by v. We write $[m] = \{1, \ldots, m\}$ and define the Euclidean ball $\mathcal{B}(\mathbf{v}, r) := \{\mathbf{x} \in \mathbb{R}^m : \|\mathbf{x} - \mathbf{v}\|_2 \le r\}$. We adopt standard notions of Lipschitz continuity and Lipschitz smoothness, and use "smoothness constant/curvature" interchangeably.

Problem setup We adopt sequence-level RL with verifiable rewards for LLM training. Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be the set of questions and $\mathcal{O} = \{o_1, \ldots, o_K\}$ be the set of candidate output sequences. A deterministic reward $r : \mathcal{Q} \times \mathcal{O} \to \{0, 1\}$ evaluates whether an output is correct for a specific question. $\pi_\theta : \mathcal{Q} \to \Delta(\mathcal{O})$ represents the LLM generation policy, which induces expected reward

$$J_i(\theta) = \mathbb{E}_{o\sim\pi_\theta(\cdot|q_i)}[r(q_i, o)]$$

for question q_i. The goal of RLVR is to learn a policy that maximizes $J(\theta) := \frac{1}{n}\sum_{i=1}^n J_i(\theta)$.

In this paper, we analyze a simplified on-policy setting with *exact* parameter updates: at step t, a question $i(t)$ is sampled uniformly from $\mathcal{Q}$ and $\nabla J_{i(t)}(\theta)$ can be computed exactly. We also impose the assumption that each question in $\mathcal{Q}$ admits a unique correct answer in $\mathcal{O}$ [9, 13, 14]:

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

Furthermore, we consider the *log-linear policy parameterization* with feature vectors $\mathbf{x}_{i,j} \in \mathbb{R}^d$. The policy and the feature matrices are given by:

$$\pi_\theta(o_j \mid q_i) = \frac{\exp(\mathbf{x}_{i,j}^\top \theta)}{\sum_{l\in[K]} \exp(\mathbf{x}_{i,l}^\top \theta)}, \quad X_i = [\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,K}]^\top.$$

In the exact on-policy setting, all critic-free policy gradient (PG) methods, including REINFORCE, reduce to gradient ascent on $J_{i(t)}$ at step t:

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

Similarly, on-policy GRPO also performs gradient ascent on $J_{i(t)}$ but rescales the per-question gradient by the within-prompt Bernoulli standard deviation:

$$\theta_{t+1} = \theta_t + \eta \frac{\nabla J_{i(t)}(\theta_t)}{\sqrt{\pi^*_{\theta_t}(i(t))\big(1 - \pi^*_{\theta_t}(i(t))\big)}},$$

where $\pi^*_\theta(i)$ represents the probability that the correct answer is generated.

3 Main theoretical results

In this section, we provide the convergence analysis for REINFORCE-style PG methods and GRPO in the *exact* setting, and show that GRPO achieves provably faster convergence. We outline the intuition: per-question reward variance upper-bounds the local Hessian norm of J_i, so variance normalization implements a curvature-matched step size. Under an orthogonality condition on features, this local adaptation yields faster convergence. Full proofs appear in Appendix C and D.

Connection between local curvature and variance.

Lemma 1 (Local Hessian bound; informal)**.** Under Assumption 1,

$$\|\nabla^2 J_i(\theta)\| \le 4X_{\max}^2\, \pi_\theta^*(i)\big(1-\pi_\theta^*(i)\big) = 4X_{\max}^2\, \mathrm{Var}_{\pi_\theta}(r \mid q_i).$$

Thus the local smoothness (curvature) of J_i is proportional to the within-prompt Bernoulli variance.

Cross-question interaction. To ensure that gradients for different prompts do not interfere destructively, we control interference between questions via:

Assumption 2 (Orthogonal representation)**.** For all $i, j \in [n]$ with $i \neq j$, we have $X_i^\top X_j = \mathbf{0}$.

This assumption decouples per-question gradients. Empirical checks are shown in Section F.1.

Convergence rates. Let T be the total updates and n the number of questions. The following theorems establish the convergence guarantee for REINFORCE-style PG methods and GRPO:

Theorem 1 (REINFORCE; informal)**.** Under Assumptions 1 and 2, for REINFORCE-style policy gradient methods with step size $\eta = \Theta(1/X_{\max}^2)$, we have

$$\min_{t\in\{0,\dots,T-1\}} \mathbb{E}\big[\|\nabla J_i(\theta_t)\|^2\big] = \mathcal{O}\big(\tfrac{n}{T}\big), \quad \forall i \in [n].$$

To show the convergence guarantee for GRPO, we further impose the following assumption on the bound of within-prompt Bernoulli variance at every step:

Assumption 3 (Bounded variance)**.** For all $i \in [n]$, there exists a constant sequence $\{C_i(t)\}_{t=1}^{\infty}$

$$\sqrt{\pi_{\theta_t}^*(i)\big(1-\pi_{\theta_t}^*(i)\big)} \le C_i(t) \le \frac{1}{2}$$

Theorem 2 (GRPO; informal)**.** Under Assumptions 1–3, for GRPO with step size $\eta = \Theta(1/(2X_{\max}^2))$, we have

$$\min_{t\in\{0,\dots,T-1\}} \mathbb{E}\big[\|\nabla J_i(\theta_t)\|^2\big] = \mathcal{O}\left(\frac{n}{T}\cdot\frac{1}{T}\sum_{t=0}^{T-1} C_i(t)\right).$$

Note that we could use $\frac{1}{T}\sum_{i=0}^{T-1}\mathbb{E}[\sqrt{\pi_{\theta_t}^*(i)\big(1-\pi_{\theta_t}^*(i)\big)}]$ as an estimation of $\frac{1}{T}\sum_{t=0}^{T-1} C_i(t)$ and is typically much smaller than 1 if the curvature varies across iterations.

Proof sketch. Lemma 1 implies a local Lipschitz constant proportional to variance; scaling the gradient by its square-root normalizes local smoothness and yields a larger allowable step in high-curvature directions. Orthogonality ensures these per-question gains add without destructive cancellation. Formal proofs are in Appendix D.

4 Empirical studies

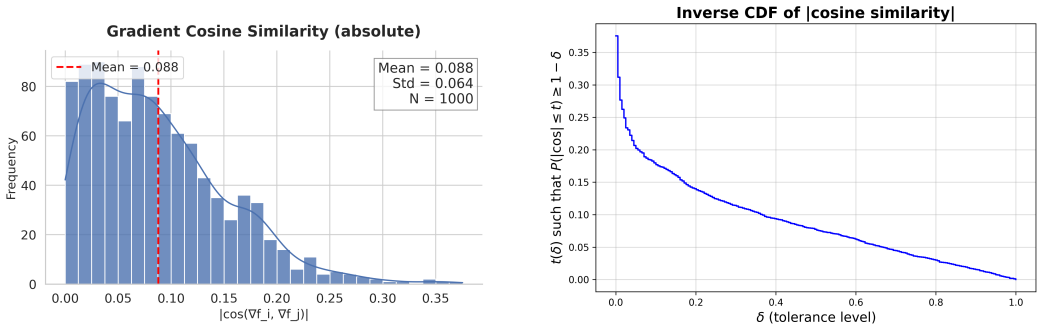
We validate two modeling assumptions central to our theory (near-orthogonality of prompt representations; variance $\leftrightarrow$ local curvature) and compare normalization strategies on GSM8K. The details are provided in Appendix F.

4.1 Orthogonality of prompt representations

We extract sentence-level embeddings (penultimate hidden states) from `Qwen2.5-MATH-1.5B` for 1,000 random question pairs on GSM8K. Absolute cosine similarities concentrate near zero (mean 0.088, std 0.064), and over 90% of pairs have $|\cos| < 0.15$. These statistics (Fig. 1) support the orthogonality assumption.

4.2 Curvature proxy and temporal stability

We use the diagonal Fisher estimate as a practical curvature proxy (see Appendix F for estimator and unbiasedness argument). In Table 1, prompt-level Fisher entries correlate with reward-variance at the same iteration (mean Pearson ≈ 0.34, $p<0.01$) but not across different times, indicating the curvature–variance link is local in time and supports an iteration-wise adaptive step-size.



(a) Absolute cosine similarities (embeddings).

(b) Inverse CDF of absolute cosine similarity ($|\cos|$).

Figure 1: Empirical validation of near-orthogonality assumption. (a) Histogram of absolute cosine similarities between question pairs. (b) Inverse CDF showing tail behavior.

Time Lag	**Mean Correlation**	**Significant** ($p < 0.05$)
Same time ($\Delta t = 0$)	0.342	Yes (0.008)
Different times ($\Delta t \neq 0$)	-0.028	No (0.18)

Table 1: Temporal Independence of Fisher Information and Reward Variance

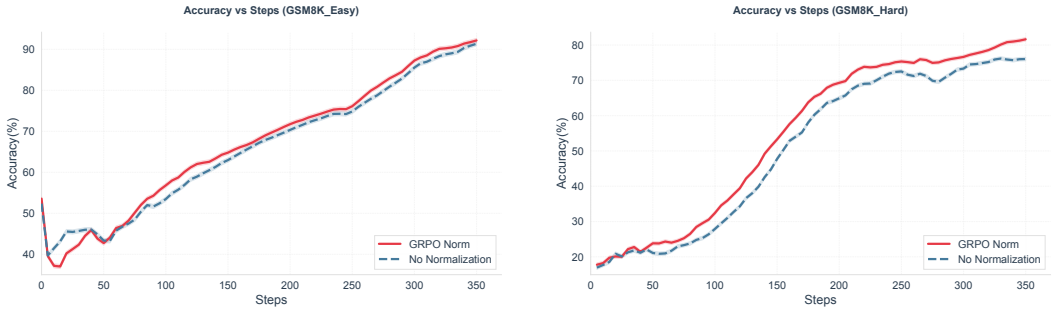
4.3 Normalization comparison on GSM8K

Setup. Base model: `Qwen2.5-MATH-1.5B` finetuned with LoRA. We split GSM8K into *Easy* (4,695) and *Hard* (1,909) by solution complexity (evaluator: `Qwen2-7B-Instruct`). We compare:

- **GRPO-Std**: per-question z-score (mean-subtract then divide by std).
- **No-Std**: mean-centering only (no variance scaling).

Training hyperparams and LoRA ranks appear in Appendix F.

Results. Figure 2 shows training accuracy trajectories (smoothed). On *Easy* both methods converge quickly with minor gap (final $\approx 92\%$ vs 91%). On *Hard* GRPO-Std yields a clear advantage (final $\approx 81\%$ vs 76%) and noticeably more stable learning. The benefit is smallest near the $\approx 50\%$ region (maximal Bernoulli variance) and grows as training moves away from that regime.



(a) GSM8K Easy split

(b) GSM8K Hard split

Figure 2: Smoothed GSM8K training accuracy on Easy/Hard: GRPO-Std (red, solid) vs No-Std (blue, dashed). GRPO yields 5% gain and more stable learning on Hard; little gap on Easy.

5 Conclusion

We reinterpret GRPO's within-prompt standardization as a curvature-matched adaptive gradient: within-prompt reward variance proxies local smoothness and rescales step sizes accordingly. Under a mild orthogonality assumption we prove improved convergence constants compared to unnormalized REINFORCE. Experiments on synthetic tasks and GSM8K corroborate greater stability and up to $\sim$5% gains on harder problems. This connection motivates adaptive, critic-free updates for efficient LLM fine-tuning.

References

[1] 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.

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

[3] 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.

[4] 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.

[5] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

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

[7] 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.

[8] 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.

[9] 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*, 2025.

[10] 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.

[11] Zhen Liu, Yuxuan Wang, Ziyang Chen, Han Wang, Jie Zhou, Maosong Sun, and et al. Formalmath: A large-scale formal benchmark for verifiable mathematical reasoning in lean4. *arXiv preprint arXiv:2505.02735*, 2025.

[12] 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.

[13] 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*, pages 6820–6829. PMLR, 2020.

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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

[21] 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.

[22] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[23] 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*, 2025.

[24] Jiayi Zhang, Yuzhuo Tang, Haotian Li, Shengding Huang, Maosong Sun, Jie Zhou, and et al. Omni-math: A universal olympiad-level benchmark for mathematical reasoning. *arXiv preprint arXiv:2410.07985*, 2024.

A Related works

REINFORCE-style PG methods. ReMax proposes a simple sequence-level REINFORCE objective for LLM alignment with strong performance and minimal complexity [8]. RLOO extends this by sampling multiple responses per prompt and using a leave-one-out baseline to further reduce variance [1]. REINFORCE++ continues this line, emphasizing simplicity and efficiency at scale [5].

GRPO and its variants. GRPO has become the default in state-of-the-art reasoning systems, combining a per-prompt baseline with within-prompt standard-deviation normalization [17]. Large-scale systems work (e.g., DAPO) has consolidated GRPO-style training across diverse tasks and compute regimes [23]. Related analyses examine design choices in normalization and sampling [12].

Emerging theory for GRPO. Recent studies analyze what GRPO optimizes and how it behaves in on- and off-policy regimes [15], its implicit alignment objective [20], and trajectory-corrected variants with convergence guarantees [16]. Other work highlights a trade-off between normalization and calibration, showing that removing the std term can improve probability calibration at the cost of optimization speed [2]. We contribute a new perspective: interpreting the std term as an adaptive gradient mechanism tied to local curvature, thereby unifying disparate empirical observations.

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 such as correctness checks [7, 4, 19, 21]. This avoids reward-model bias and simplifies training, while scaling effectively with compute and dataset size. Strong results have been reported on GSM8K, MATH, Omni-MATH, and FormalMATH [24, 11]. In this paper, we study GRPO in the RLVR setting, where deterministic rewards enable sharper theoretical analysis of normalization and its role in adaptive gradient updates.

B Extra Notations and Update Formulae

In this section, we show extra notations that are used in the proofs, deductions that are omitted in the problem setup, and update details when applying REINFORCE-style PG methods and GRPO. Under Assumption 1, we use a_i to denote the index of correct answer for question $q_i \in \mathcal{Q}$:

$$r(q_i, o_j) = \begin{cases} 1, & \text{if} \quad j = a_i \\ 0, & \text{if} \quad j \neq a_i, \end{cases} \tag{1}$$

and use $\mathbf{r}_i \in \mathbb{R}^K$ to denote the reward vector for question q_i: $[\mathbf{r}_i]_j = r(q_i, o_j) \quad \forall j \in [K]$. We consider the on-policy scenario where $\pi_\theta = \pi_{\theta_{\text{old}}}$, and the reward function has a unique correct answer. Therefore, the importance ratio remains to be $\gamma_i(o) = 1$, and

$$A_i(o) = \frac{r(q_i, o) - \pi_\theta^*(i)}{\sqrt{\pi_\theta^*(i)\Big(1 - \pi_\theta^*(i)\Big)}}. \tag{2}$$

where $\pi_\theta^*(i) \coloneqq \pi_\theta(o_{a_i} \mid q_i)$ denotes the success probability of policy π_θ on question q_i. The GRPO objective can be further simplified as:

$$J_{\text{GRPO}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} J^i_{\text{GRPO}}(\theta) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{o\sim\pi_\theta}\Big[A_i(o)\Big] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{o\sim\pi_\theta}\left[\frac{r(q_i, o) - \pi_\theta^*(i)}{\sqrt{\pi_\theta^*(i)\Big(1 - \pi_\theta^*(i)\Big)}}\right].$$

We also denote $\pi_\theta(i) \in \mathbb{R}^K$ as the probability vector for π_θ in question i, that is, $[\pi_\theta(i)]_j \coloneqq \pi_\theta(o_j \mid q_i), \forall j \in [K]$.

Under the *exact* parameter updates setting, the gradient of GRPO does not change when removing the baseline. We term such an algorithm as on-policy GRPO. Our key observation is that the variance normalization in (on-policy) GRPO implicitly implements an adaptive step size. In particular,

$$\begin{aligned} \nabla J^i_{\text{GRPO}}(\theta) &= \mathbb{E}_{o\sim\pi_\theta}[A_i(o)\nabla \ln \pi_\theta(o \mid q_i)] = \mathbb{E}_{o\sim\pi_\theta}\left[\frac{r(q_i, o)}{\sqrt{\pi_\theta^*(i)\Big(1 - \pi_\theta^*(i)\Big)}}\nabla \ln \pi_\theta(o \mid q_i)\right] \\ &= \frac{\mathbb{E}_{o\sim\pi_\theta}[r(q_i, o)\nabla \ln \pi_\theta(o \mid q_i)]}{\sqrt{\pi_\theta^*(i)\Big(1 - \pi_\theta^*(i)\Big)}} = \frac{\nabla J_i(\theta)}{\sqrt{\pi_\theta^*(i)\big(1 - \pi_\theta^*(i)\big)}}. \end{aligned} \tag{3}$$

for all $i \in [n]$. The first and last equalities follow from the policy gradient theorem [18]. The second equality holds because subtracting a constant baseline does not affect the gradient calculation. The third equality follows from the fact that $A_i(o)$ is treated as constant in the gradient propagation.

For ease of notation, we simply drop t from $i(t)$ whenever it clear in context. The update of REINFORCE takes the following form [9]:

$$\theta_t \leftarrow \theta_{t-1} + \eta X_i^\top \Big(\text{diag}\left(\pi_{\theta_{t-1}}(i)\right) - \pi_{\theta_{t-1}}(i)\pi_{\theta_{t-1}}^\top(i)\Big)\,\mathbf{r}_i. \tag{4}$$

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

$$\theta_t \leftarrow \theta_{t-1} + \eta\Big(\pi^*_{\theta_{t-1}}(i)(1 - \pi^*_{\theta_{t-1}}(i)\mathbf{x}_{i,a_i} - \pi^*_{\theta_{t-1}}(i)\sum_{j\neq a_i}[\pi_{\theta_{t-1}}(i)]_j \cdot \mathbf{x}_{i,j}\Big). \tag{5}$$

Similarly, the update of GRPO can be simplified as:

$$\theta_t \leftarrow \theta_{t-1} + \eta\Big(\sqrt{\pi^*_{\theta_{t-1}}(i)(1-\pi^*_{\theta_{t-1}}(i))}\mathbf{x}_{i,a_i} - \sqrt{\frac{\pi^*_{\theta_{t-1}}(i)}{1-\pi^*_{\theta_{t-1}}(i)}}\sum_{j\neq a_i}[\pi_{\theta_{t-1}}(i)]_j \cdot \mathbf{x}_{i,j}\Big). \tag{6}$$

C Analysis of the Local Smoothness Constant

In this section, we provide the full description of the lemmas related to the local smoothness constant and provide their proofs.

Lemma 2 (Formal Version of Lemma 1)**.** Under Assumption 1, for all $i \in [n]$ and $\theta \in \mathbb{R}^d$,

$$\|\nabla^2 J_i(\theta)\| \leq 4X^2_{\max} \cdot \pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big) = 4X^2_{\max} \cdot \mathrm{Var}(\pi_\theta(i)). \tag{7}$$

Proof. According to Lemma 17 in [9], for any $y \in \mathbb{R}^d$, we have

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

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

$$\begin{aligned}
|\mathbf{y}^\top \nabla^2 J_i(\theta)\mathbf{y}| &\leq |\left(H\left(\pi_\theta(i)\right) r_i\right)^\top \left(X_i\mathbf{y} \odot X_i\mathbf{y}\right)| + 2|\left(H\left(\pi_\theta(i)\right) r_i\right)^\top \left(X_i\mathbf{y}\right)| \cdot |\left(\pi_\theta^\top(i) X_i\mathbf{y}\right)| \\
&\leq \|\left(H\left(\pi_\theta(i)\right) r_i\right)\|_\infty \|X_i\mathbf{y} \odot X_i\mathbf{y}\| + 2\|H\left(\pi_\theta(i)\right) r_i\| \cdot \|X_i\mathbf{y}\| \cdot \|\pi_\theta(i)\| \cdot \|X_i\mathbf{y}\| \\
&= \|\left(H\left(\pi_\theta(i)\right) r_i\right)\|_\infty \|X_i\mathbf{y}\|^2 + 2\|H\left(\pi_\theta(i)\right) r_i\| \cdot \|\pi_\theta(i)\| \cdot \|X_i\mathbf{y}\|^2 \\
&\leq \|\left(H\left(\pi_\theta(i)\right) r_i\right)\|_\infty \|X_i\mathbf{y}\|^2 + 2\|H\left(\pi_\theta(i)\right) r_i\| \cdot \|X_i\mathbf{y}\|^2.
\end{aligned} \tag{8}$$

The last inequality follows because $\|\pi_\theta(i)\| \leq \|\pi_\theta(i)\|_1 = 1$. 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

$$\|H(\pi_\theta(i))r_i\|_\infty = \pi^*_\theta(i)(1-\pi^*_\theta(i)), \tag{9}$$

and

$$\begin{aligned}
\|H(\pi_\theta(i))r_i\| &= \pi^*_\theta(i)\sqrt{(1-\pi^*_\theta(i))^2 + \sum_{j\neq a_i}[\pi_\theta(i)]_j^2} \\
&\leq \pi^*_\theta(i)\sqrt{(1-\pi^*_\theta(i))^2 + \sum_{j\neq a_i}[\pi_\theta(i)]_j(1-\pi^*_\theta(i))} \\
&= \sqrt{2}\pi^*_\theta(i)(1-\pi^*_\theta(i)).
\end{aligned} \tag{10}$$

Combining (9) and (10) with (8), we get

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

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

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

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

Corollary 1. Under Assumption 1, for all $i \in [n]$ and $\theta \in \mathbb{R}^d$,

$$\|\nabla^2 J_i(\theta)\| \leq X_{\max}^2, \tag{11}$$

so that $J_i(\theta)$ is $X_{\max}^2$-smooth on $\mathbb{R}^d$.

Lemma 3. Under Assumption 1, for all $i \in [n]$, $J_i(\theta)$ is $\frac{1}{2}X_{\max}-$Lipschitz over $\mathbb{R}^d$.

Proof. According to (5), the gradient of $J_i(\theta)$ takes the following form:

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

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

$$\begin{aligned}
\|\nabla J_i(\theta)\| &\leq \|\mathbf{x}_{a_i}\|(1 - \pi_{\theta_t}^*(i))\pi_{\theta_t}^*(i) + \sum_{j \neq a_i} \|\mathbf{x}_j\|\pi_{\theta_t}(i)_j \cdot \pi_{\theta_t}^*(i) \\
&\leq \|X_i\|(1 - \pi_{\theta_t}^*(i))\pi_{\theta_t}^*(i) + \sum_{j \neq a_i} \|X_i\|\pi_{\theta_t}(i)_j \cdot \pi_{\theta_t}^*(i) \\
&= 2\|X_i\|(1 - \pi_{\theta_t}^*(i))\pi_{\theta_t}^*(i) \\
&\leq \frac{1}{2}X_{\max}
\end{aligned}$$

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

Lemma 4 (Non-uniform local smoothness)**.** Under Assumption 1, for all $i \in [n]$ and $\theta \in \mathbb{R}^d$, $J_i(\theta)$ is $\frac{5}{2}X_{\max}^2 \cdot \sqrt{\pi_\theta^*(i)(1 - \pi_\theta^*(i))}-$smooth over $\mathcal{B}(\theta, \frac{1}{X_{\max}} \cdot \sqrt{\pi_\theta^*(i)(1 - \pi_\theta^*(i))})$.

Proof. By Assumption 1, the objective $J_i(\theta)$ is same as $\pi_\theta^*(i)$. From Lemma 3, $J_i(\theta)$ 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 2,

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

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 4 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

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

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

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

which implies that

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

$f(a)$ 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 of the Main Result

D.1 Auxiliary Lemma

Lemma 5. Under Assumption 1 and 2, for any $i, j \in [n], i \neq j$ and $\theta \in \mathbb{R}^d$, we have

$$\nabla J_i(\theta)^\top \nabla J_j(\theta) = 0 \tag{12}$$

Proof. According to (4), we get

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

where the second step is by Assumption 2. □

Theorem 3 (Convergence rate of REINFORCE, formal version of Theorem 1)**.** Under Assumption 1 and Assumption 2, with the step size $\eta = \frac{1}{X_{\max}^2}$, the sequence $\{\theta_t\}_{t=0}^{T-1}$ generated by REINFORCE satisfies:

$$\min_{t \in \{0,1,\ldots,T-1\}} \mathbb{E}[\|\nabla J_i(\theta_t)\|^2] \leq \frac{2n(1 - \pi_{\theta_0}^*(i))X_{\max}^2}{T} \tag{13}$$

for any $i \in [n]$.

Proof. We consider a specific question q_l. Combining Lemma 5 with *log-linear policy parameterization* in our setting, if question $q_{i(t)}$ is selected on iteration t in REINFORCE, we get

$$\begin{aligned}
J_j(\theta_t) &= J_j(\theta_{t-1} + \eta \nabla J_i(\theta_{t-1})) \\
&= J_j(\theta_{t-1})
\end{aligned} \tag{14}$$

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 REINFORCE, we have

$$\begin{aligned}
J_l(\theta_t) - J_l(\theta_{t-1}) &\geq \langle \theta_t - \theta_{t-1}, \nabla J_l(\theta_{t-1}) \rangle - \frac{X_{\max}^2}{2} \|\theta_t - \theta_{t-1}\|^2 \\
&= (\eta - \frac{X_{\max}^2}{2}\eta^2) \|\nabla J_l(\theta_{t-1})\|^2 \\
&= \frac{1}{2X_{\max}^2} \|\nabla J_l(\theta_{t-1})\|^2
\end{aligned} \tag{15}$$

where the first step is by Corollary 1, which also indicate that $J_i(\theta)$ is $X_{\max}^2$-weakly convex. Taking expectation of (15) on $i(t)$, we get

$$\mathbb{E}[J_l(\theta_t)] - \mathbb{E}[J_l(\theta_{t-1})] \geq \frac{1}{2nX_{\max}^2} \|\nabla J_l(\theta_{t-1})\|^2. \tag{16}$$

Summing up (16) for $t = 1, \ldots, T$, we get

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

This directly leads to

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

□

Theorem 4 (Convergence rate of GRPO, formal version of Theorem 2)**.** Under Assumption 1–3, with the step size $\eta = \frac{1}{2X^2_{\max}}$, the sequence $\{\theta_t\}_{t=0}^{T-1}$ generated by GRPO satisfies:

$$\min_{t\in\{0,1,\ldots,T-1\}} \mathbb{E}[\|\nabla J_i(\theta_t)\|^2] \le \frac{2n(1-\pi^*_{\theta_0}(i))X^2_{\max}}{T}\frac{8\sum_{t=0}^{T-1} C_i(t)}{3T} \tag{17}$$

for any $i \in [n]$.

Proof. Similar to (14) in the proof of Theorem 3, the gradient update based on question q_i does not affect the objective for question q_l if $i \neq l$. That is,

$$J_l(\theta_t) = \begin{cases} J_l(\theta_{t-1}), & \text{if } i(t) \neq l \\ J_l(\theta_t), & \text{if } i(t) = l. \end{cases} \tag{18}$$

Consider the case where $i(t) = l$, from the parameter update rule in GRPO, we get

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

Also, by setting $\eta = \frac{1}{2X^2_{\max}}$, we have

$$\begin{aligned}
&\|\eta\Big(\sqrt{\pi^*_{\theta_{t-1}}(l)(1-\pi^*_{\theta_{t-1}}(l))}\mathbf{x}_{l,a_l} - \sqrt{\frac{\pi^*_{\theta_{t-1}}(l)}{1-\pi^*_{\theta_{t-1}}(l)}}\sum_{j\neq a_l}[\pi_{\theta_{t-1}}(l)]_j \cdot \mathbf{x}_{l,j}\Big)\| \\
=&\frac{1}{2X^2_{\max}}\|\Big(\sqrt{\pi^*_{\theta_{t-1}}(l)(1-\pi^*_{\theta_{t-1}}(l))}\mathbf{x}_{l,a_l} - \sqrt{\frac{\pi^*_{\theta_{t-1}}(l)}{1-\pi^*_{\theta_{t-1}}(l)}}\sum_{j\neq a_l}[\pi_{\theta_{t-1}}(l)]_j \cdot \mathbf{x}_{l,j}\Big)\| \\
\le&\frac{1}{2X^2_{\max}}\Big(\sqrt{\pi^*_{\theta_{t-1}}(l)(1-\pi^*_{\theta_{t-1}}(l))}\|\mathbf{x}_{l,a_l}\| + \sqrt{\frac{\pi^*_{\theta_{t-1}}(l)}{1-\pi^*_{\theta_{t-1}}(l)}}\sum_{j\neq a_l}[\pi_{\theta_{t-1}}(l)]_j \cdot \|\mathbf{x}_{l,j}\|\Big) \\
\le&\frac{1}{2X^2_{\max}}(2\sqrt{\frac{\pi^*_{\theta_{t-1}}(l)}{1-\pi^*_{\theta_{t-1}}(l)}}X_{max}) \\
=&\frac{1}{X_{\max}}\cdot\sqrt{\pi^*_\theta(l)(1-\pi^*_\theta(l))}.
\end{aligned}$$

This implies that $\theta_t \in \mathcal{B}(\theta, \frac{1}{X_{\max}}\cdot\sqrt{\pi^*_\theta(l)(1-\pi^*_\theta(l))})$. According to Lemma 4, we obtain

$$\begin{aligned}
J_l(\theta_t) &\ge J_l(\theta_{t-1}) + \langle\theta_t - \theta_{t-1}, \nabla J_l(\theta_{t-1})\rangle - \frac{5}{4}X^2_{\max}\cdot\sqrt{\pi^*_\theta(l)(1-\pi^*_\theta(l))}\|\theta_t - \theta_{t-1}\|^2 \\
&= J_l(\theta_{t-1}) + \frac{3}{16X^2_{\max}\sqrt{\pi^*_\theta(l)(1-\pi^*_\theta(l))}}\|\nabla J_l(\theta_{t-1})\|^2 \\
&\ge J_l(\theta_{t-1}) + \frac{3}{16X^2_{\max}C_l(t-1)}\|\nabla J_l(\theta_{t-1})\|^2
\end{aligned} \tag{19}$$

where the last step is by Assumption 3. Taking expectation of (19) on $i(t)$, we have

$$\mathbb{E}[J_l(\theta_t)] \ge \mathbb{E}[J_l(\theta_{t-1})] + \frac{3}{16nX^2_{\max}C_l(t-1)}\mathbb{E}[\|\nabla J(\theta_{t-1})\|^2]. \tag{20}$$

because the objective J_l remains unchanged if $i(t) \neq l$ according to (18). Summing up (20) for $t = 1, \ldots, T$, we get

$$\mathbb{E}[J_l(\theta_T)] \ge J_l(\theta_0) + \sum_{t=0}^{T-1}\frac{3}{16nX^2_{\max}C_l(t-1)}\mathbb{E}[\|\nabla J(\theta_{t-1})\|^2]. \tag{21}$$

According to the Cauchy-Schwarz inequality, we obtain

$$\min_{t\in\{0,1,\dots,T-1\}} \mathbb{E}[\|\nabla J_l(\theta_t)\|^2] \le \frac{2n(1-\pi^*_{\theta_0}(l))X^2_{\max}}{T}\frac{8\sum_{t=0}^{T-1} C_l(t)}{3T}.$$

□

E Discussion on $C(n,T)$

We are interested in the meaningful small-constant regime where $C(n,T) = o(1)$. Let $\varepsilon_{i,j} := 1-\pi_{\theta_j}(i) \in [0,1]$. Then

$$C(n,T) = \frac{8}{3nT}\sum_{i=1}^{n}\sum_{j=0}^{T-1}\sqrt{\pi_{\theta_j}(i)\big(1-\pi_{\theta_j}(i)\big)} \le \frac{8}{3n}\sum_{i=1}^{n}\underbrace{\frac{1}{T}\sum_{j=0}^{T-1}\min\big\{\tfrac{1}{2},\sqrt{\varepsilon_{i,j}}\big\}}_{=:A_i(T)}. \tag{22}$$

Hence $C(n,T)=o(1)$ whenever each prompt's Cesàro mean $A_i(T)\to 0$. A convenient pointwise bound is

$$0 \le \sqrt{\pi(1-\pi)} \le \min\Big\{\tfrac{1}{2},\sqrt{1-\pi}\Big\}.$$

A sufficient (and essentially necessary) condition is that, for every fixed $\delta>0$,

$$\frac{1}{T}\Big|\{\, j<T:\ \varepsilon_{i,j}\ge\delta\,\}\Big| \xrightarrow[T\to\infty]{} 0 \qquad \text{for all } i\in[n].$$

We provide improvement regimes under which $A_i(T)\to 0$ below.

(i) Exponential improvement. If $\varepsilon_{i,j}\le c_i\rho_i^j$ with $\rho_i\in(0,1)$, then

$$\frac{1}{T}\sum_{j<T}\sqrt{\varepsilon_{i,j}} \le \frac{\sqrt{c_i}}{T}\sum_{j<T}\rho_i^{j/2} = O\big(\tfrac{1}{T}\big),$$

so $C(n,T)=O(1/T)=o(1)$.

(ii) Polynomial improvement. If $\varepsilon_{i,j}\le c_i\, j^{-\alpha_i}$ for some $\alpha_i>0$, then

$$\frac{1}{T}\sum_{j<T}\sqrt{\varepsilon_{i,j}} \le \frac{\sqrt{c_i}}{T}\sum_{j<T} j^{-\alpha_i/2} = \begin{cases} O\big(T^{-\alpha_i/2}\big), & 0<\alpha_i<2,\\ O\big((\log T)/T\big), & \alpha_i=2,\\ O\big(1/T\big), & \alpha_i>2,\end{cases}$$

hence $C(n,T)=o(1)$ for any $\alpha_i>0$. A notable special case is the *harmonic* regime $\varepsilon_{i,j}=\Theta(1/j)$, which yields

$$C(n,T) = O\Big(\sqrt{\tfrac{\log T}{T}}\Big) = o(1).$$

Note. This refines the example following Eq. (14): the intended assumption is $1-\pi_{\theta_j}(i)=\Theta(1/j)$ (not $\Theta(1/T)$).

Observe that

$$\sum_{j=0}^{T-1}\sqrt{\pi^*_{\theta_j}(i)(1-\pi^*_{\theta_j}(i))} \le \sqrt{T\cdot\sum_{j=0}^{T-1}\pi^*_{\theta_j}(i)(1-\pi^*_{\theta_j}(i))} \le \sqrt{T\cdot\sum_{j=0}^{T-1}(1-\pi^*_{\theta_j}(i))}.$$

For instance, if $(1-\pi^*_{\theta_j}(i))=\mathcal{O}(1/T)$, then $C(n,T)=\mathcal{O}(\sqrt{\log T/T})$.

(iii) Log-slow improvement. If $\varepsilon_{i,j}\asymp 1/\log(j+e)$, then

$$\frac{1}{T}\sum_{j<T}\sqrt{\varepsilon_{i,j}} \asymp \frac{1}{\sqrt{\log T}}, \qquad \Rightarrow \qquad C(n,T)=O\big(1/\sqrt{\log T}\big)=o(1).$$

(iv) Persistent hard prompts (plateau). If for some i there exists $\varepsilon_0 > 0$ such that $\varepsilon_{i,j} \geq \varepsilon_0$ on a non-vanishing fraction of iterations, then $A_i(T)$ is bounded away from 0, and $C(n,T) \not\to 0$. Thus, *for fixed* n, every prompt must become (asymptotically) easy in Cesàro mean in order to have $C(n,T) = o(1)$.

(v) Mixed populations (curriculum/heterogeneity). Suppose the prompts split into $\mathcal{E}$ (easy) with $\varepsilon_{i,j} \to 0$ sufficiently fast (any of (i)–(iii)), and $\mathcal{H}$ (hard) with $\limsup_T A_i(T) \geq c > 0$. Then

$$C(n,T) \;\leq\; \frac{8}{3}\Big(\frac{|\mathcal{E}|}{n}\cdot o(1) \;+\; \frac{|\mathcal{H}|}{n}\cdot \Theta(1)\Big).$$

Therefore $C(n,T) = o(1)$ iff $|\mathcal{H}| = 0$ (for fixed n); if n grows with T, one additionally needs $|\mathcal{H}|/n \to 0$.

A universal upper bound. Since $\sqrt{\pi(1-\pi)} \leq \frac{1}{2}$, we always have

$$C(n,T) \leq \frac{8}{3nT}\cdot n \cdot \frac{T}{2} = \frac{4}{3}.$$

Thus the multiplicative factor in Eq. (14) is at worst a constant; the benefit over the unnormalized baseline is most pronounced when $C(n,T) = o(1)$, i.e., when success probabilities approach 1 on (almost) all prompts.

In summary, any training dynamic in which $\pi_{\theta_j}(i) \to 1$ for every prompt, no matter how slowly (even logarithmically), drives $C(n,T) \to 0$. Faster per-prompt improvement directly tightens Eq. (14), quantifying how GRPO's normalization converts heterogeneous per-prompt "curvature" into a vanishing multiplicative constant in the convergence bound

F Detailed empirical studies

F.1 Validation of orthogonality assumption

Formally, for two distinct questions $i \neq j$, we expect $\frac{\langle v_i, v_j\rangle}{\|v_i\|\|v_j\|} \approx 0$, where v_i denotes the representation vector (e.g., penultimate-layer hidden state) of question i. This assumption simplifies the analysis by ensuring cross-question interference is negligible. We validate Assumption 2 on GSM8K [3] using `Qwen2.5-MATH-1.5B` [22]. For 1,000 random pairs of distinct questions, we extracted penultimate hidden states, pooled them into sentence-level embeddings, and measured absolute cosine similarities. As shown in Figure 1a, similarities are sharply concentrated near zero (mean ≈ 0.088, std ≈ 0.064). The inverse CDF in Figure **??** further shows that over 90% of pairs have similarity below 0.15, supporting the orthogonality assumption.

F.2 Validation of Local Curvature-Variance Connection

In our implementation, we compute the Fisher Information matrix following the efficient estimator proposed by [10]. Given a batch of prompts $\{q_i\}_{i=1}^B$ at iteration t, we: 1. Sample responses $\hat{o}_i \sim \pi_{\theta_t}(\cdot|q_i)$ for each prompt q_i 2. Compute the mini-batch 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 Information using the efficient estimator: $\mathbf{h}(\theta_t) = \text{diag}(\hat{F}_{\text{eff}}(\theta_t)) = B\cdot\nabla\hat{\mathcal{L}}_B(\theta_t)\odot\nabla\hat{\mathcal{L}}_B(\theta_t)$, where this estimator remains unbiased: $\mathbb{E}_{\hat{o}}[\text{diag}(\hat{F}_{\text{eff}}(\theta))] = \mathbb{E}_{\hat{o}}[\text{diag}(\hat{F}(\theta))]$(the expectation is taken over the sampled responses).

The resulting Fisher Information $\mathbf{h}(\theta_t)$ serves as our curvature proxy, capturing the local smoothness of the loss landscape. This aligns with our theoretical framework where higher Fisher Information (larger curvature) corresponds to regions requiring smaller step sizes, justifying GRPO's variance-based normalization strategy.

F.3 Comparisons on LLM Reasoning Task

Building upon the theoretical foundations established earlier, we conduct empirical evaluations to validate the effectiveness of different *advantage normalization* strategies in GRPO. Our experiments compare two normalization approaches across varying dataset difficulties on the GSM8K mathematical reasoning benchmark.

Experimental setup. We employ the `Qwen2.5-Math-1.5B` model as our base model, enhanced with Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning.

To study the effect of task difficulty, we partition the GSM8K training set by solution complexity into two splits: *Easy* (4,695 examples), and *Hard* (1,909 examples). We employ `Qwen2-7B-Instruct` as an evaluator to partition the dataset into distinct difficulty levels, thereby enabling a controlled study of how normalization behaves under varying difficulty regimes.

Normalization strategies. We evaluate three group-level (per-question) normalization approaches:

- Standard GRPO ($\mathcal{N}_{\text{std}}$): per-question z-score normalization: $\hat{A}_{i,t} = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$
- No-Std ($\mathcal{N}_{\text{no-std}}$): mean-centering without variance scaling: $\hat{A}_{i,t} = r_i - \text{mean}(\mathbf{r})$.

Evaluation metrics. We report complementary metrics: *sample accuracy*: fraction of correct solutions among all generations.

Results and Discussion. Across difficulties shown in Figure 2, we observe a clear variance-dependent pattern consistent with our theory:

- *Easy (low variance).* Both methods converge rapidly and achieve high accuracy. GRPO Norm shows a slight advantage, reaching a final accuracy of $\approx 92\%$, while No Normalization achieves $\approx 91\%$. The performance gap remains small throughout training, with both curves following similar trajectories after the initial steps.
- *Hard (high variance).* GRPO Norm's benefits become increasingly apparent in harder questions. It significantly outperforms No Normalization in the final stages, achieving $\approx 81\%$ accuracy compared to $\approx 76\%$. GRPO Norm not only reaches higher final accuracy but also demonstrates more stable learning, entering the 70–80% accuracy band earlier and maintaining a consistent advantage of approximately 5 percentage points during mid-to-late training phases.

In general, the impact of normalization becomes clearer overall, but around the 50% accuracy region, where Bernoulli reward variance is maximal, the advantage of GRPO-Norm is comparatively small and the improvement is not obvious.