Taming Extreme Tokens: Covariance-Aware GRPO with Gaussian-Kernel Advantage Reweighting

Cheng Wang[†] **Qin Liu**[‡] **Wenxuan Zhou**[§] **Muhao Chen**[‡]
[†]National University of Singapore [‡]University of California, Davis
[§]University of Southern California
wangcheng@u.nus.edu

Abstract

Group Relative Policy Optimization (GRPO) has emerged as a promising approach for improving the reasoning capabilities of large language models. However, it struggles to effectively balance the tradeoff between exploration and exploitation during training, often resulting in suboptimal performance. Motivated by the theoretical insight that changes in entropy are governed by the covariance between token probabilities and their corresponding advantages, we propose a hyperparameter-free, *covariance-weighted optimization* method that dynamically down-weights extreme token-level updates via a Gaussian kernel. This approach automatically reduces the instability caused by exploration-exploitation trade-off while preserving informative learning signals. Extensive empirical evaluations show that our approach improves downstream performance across reasoning benchmarks compared with GRPO, and effectively stablizes entropy as training progresses.

1 Introduction

Group Relative Policy Optimization (GRPO) (Shao et al., 2024a) has emerged as a promising approach for enhancing the reasoning capabilities of large language models (LLMs), particularly in complex mathematical and coding tasks. Despite its demonstrated effectiveness, GRPO faces a critical limitation in properly balancing between exploitation and exploration during policy optimization, which can undermine its performance (Wang et al., 2025a,a).

Excessive exploitation can cause the model to become overconfident in its suboptimal solutions, thereby limiting its capabilities to explore novel reasoning strategies and potentially overlook more effective approaches. Conversely, while exploration is necessary for identifying better policies, excessive exploration may result in unstable training dynamics and hinder convergence to a stable, high-performing solution. These opposing risks highlights the importance of a principled mechanism for balancing exploration and exploitation so as to realize a more robust GRPO.

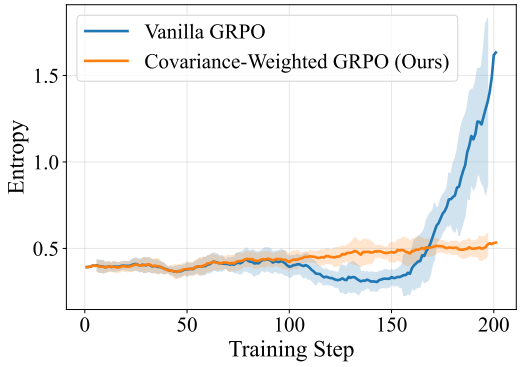

Figure 1: **Policy Entropy During Training.** Vanilla GRPO exhibits entropy instability, while our method keeps entropy at a reasonable level that effectively balances exploration and exploitation.

Specifically, the trade-off in GRPO is fundamentally tied to the policy's entropy dynamics during training. As established by Cui et al. (2025a), entropy changes under the natural policy gradient update are governed by the covariance between token log-probabilities and their corresponding advantage estimates. Based on this theoretical foundation and our empirical observations, we identify that a small fraction of tokens with extreme covariance values disproportionately dominate the policy updates, resulting in entropy instability and degraded training dynamics.

To mitigate this issue, we propose a covariance-aware variant of GRPO that attenuates extreme token-level updates through Gaussian kernel weighting. Specifically, our approach computes the covariance between centered log-probabilities and centered advantages for each token, and applies a smooth down-weighting function to tokens with high-magnitude covariances while preserving the influence of those with moderate covariances. This mechanism effectively regulates the contribu-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 540–546

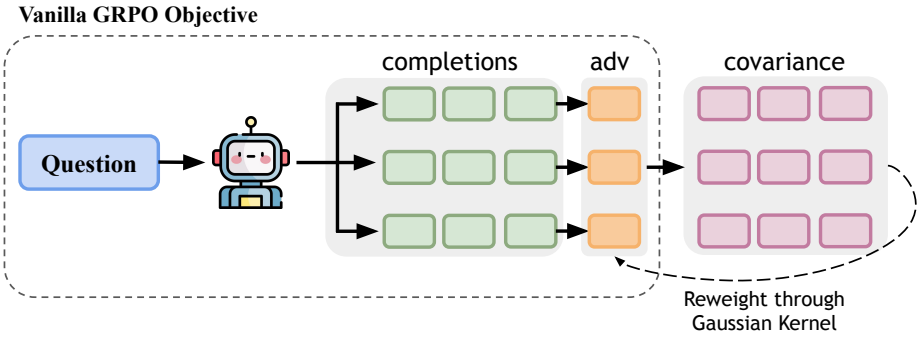

Figure 2: **Illustration of Our Proposed Method.** Compared with vanilla GRPO, our method reweights the advantages based on the covariance between token probabilities and advantages.

tion of outlier tokens to the policy gradient, thereby improving the balance between exploration and exploitation in a hyperparameter-free way. Extensive experiments demonstrate that our approach drastically improves over the vanilla GRPO, achieving better downstream performance and maintaining stable entropy dynamics, as illustrated in Figure 1.

2 Method

2.1 Preliminaries

GRPO (Shao et al., 2024b) extends Proximal Policy Optimization (Schulman et al., 2017) by removing the value network and using group-based rewards to estimate advantages. For each prompt q sampled from $\mathcal{D}$, GRPO samples a group of G responses $\{o_1, o_2, \ldots, o_G\}$ from the current policy π_θ and evaluates them using a reward model r_ϕ, which is usually a rule-based verifier. GRPO computes the advantage for response i as $\hat{A}_i = \frac{r_i - \bar{r}}{\sigma_r}$, where r_i is the reward for response i, and $\bar{r}$ and σ_r are the mean and standard deviation of rewards within the group. The GRPO aims to maximize the following objective:

$$J_{GRPO}(\theta) = \mathbb{E}_{q\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}\hat{A}_i\right] - \beta\mathbb{E}_{q\sim\mathcal{D}}\left[D_{KL}[\pi_\theta(\cdot|q)\|\pi_{ref}(\cdot|q)]\right],$$

where $\pi_{\theta_{old}}$ is the policy from the previous iteration, π_{ref} is the reference policy, and β is the KL penalty coefficient.

2.2 Motivation

To measure the exploration-exploitation trade-off in GRPO, we can use policy entropy as an indicator, which is defined as:

$$\mathcal{H}(\pi_\theta) = -\mathbb{E}_{q\sim\mathcal{D}}\left[\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}[\log\pi_\theta(o|q)]\right] = -\frac{1}{|\mathcal{D}|}\sum_{q\in\mathcal{D}}\frac{1}{|o|}\sum_{t=1}^{|o|}\mathbb{E}_{o_t\sim\pi_\theta}\left[\log\pi_\theta\left(o_t \mid q, o_{<t}\right)\right].$$

Cui et al. (2025a) show that, under Natural Policy Gradient (Kakade, 2001), the change in policy entropy is governed by the covariance between token log-probabilities and advantages:

$$\Delta\mathcal{H} \approx -\eta \cdot \mathrm{Cov}_t(\log\pi_\theta(o_t|q, o_{<t}), \hat{A}_i).$$

This relationship reveals that the covariance between log-probabilities and advantages directly drives entropy dynamics during training. As a preliminary experiment, we use GRPO to fine-tune DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025) and track token-level covariance throughout 50 training steps. As shown in Figure 3 and Table 1, we observe that a small fraction of tokens possess large-magnitude covariance values, which disproportionately dominate the overall covariance and precipitate unstable entropy dynamics. These extreme values push the policy away from ideal exploration-exploitation balance, leading to suboptimal performance. This observation directly leads to our method, which moderates extreme covariance updates through covariance-aware advantage reweighting, as suppressing these extreme updates is crucial for maintaining stable entropy.

2.3 Covariance-Aware Advantage Reweighting

To address the issue of extreme covariance values destabilizing training so as to balance the exploration-exploration trade-off, we propose a covariance-weighted GRPO variant (CW-GRPO)

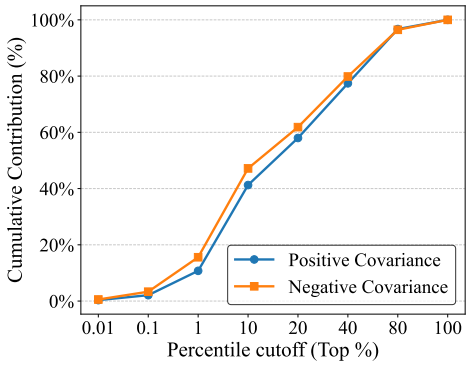

Figure 3: **Cumulative Contribution of Covariance Values**. A small fraction of tokens with extreme covariance values disproportionately dominate policy updates,

Percentile	Positive Covariance	Negative Covariance
0.01%	11.52	-13.62
1.00%	3.32	-3.34
20.00%	0.58	-0.36
40.00%	0.33	-0.22
100.00%	0.06	-0.04

Table 1: **Covariance Distribution Statistics.** The table presents numerical values of covariance magnitudes at specific percentile thresholds.

that automatically down-weights tokens with large-magnitude covariances while preserving learning signals from moderate-covariance tokens.

We adopt the standard GRPO setup in which a policy π_θ produces G responses o_i per prompt q. The vanilla GRPO objective is

$$\mathcal{L}_{\text{GRPO}} = \mathbb{E}_{q,\{o_i\}}\Big[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\ell_{i,t}\Big],$$

where $\ell_{i,t}$ is the token-level loss defined as:

$$\ell_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_i - \beta\text{KL}_{i,t}.$$

Based on the motivation from Section 2.2, we propose to reweight the advantage signal based on the covariance between log-probabilities and advantages. We compute the token-level covariance between centered log-probabilities and centered advantages:

$$c_{i,t} = \left(\log\pi_\theta(o_{i,t}|q,o_{i,<t}) - \overline{\log\pi}\right)\cdot\left(\hat{A}_i - \overline{\hat{A}}\right),$$

where $\overline{\log\pi}$ and $\overline{\hat{A}}$ are the means of log-probabilities and advantages computed over responses.

We then apply a Gaussian kernel that exponentially suppresses only the magnitudes that exceed typical variability by setting the bandwidth to the empirical standard deviation σ of the covariances:

$$w_{i,t} = \exp\left(-\frac{c_{i,t}^2}{2\sigma^2}\right),$$

where σ is the standard deviation of $\{c_{i,t}\}$. The Gaussian kernel softly filters out the handful of extreme covariance tokens that would otherwise destabilize entropy, yet leaves informative moderate-covariance tokens intact.

To maintain the expected loss, we normalize the weights:

$$\tilde{w}_{i,t} = w_{i,t}\cdot\frac{N}{\sum_{j=1}^{G}\sum_{k=1}^{|o_j|}w_{j,k}},$$

where $N = \sum_{j=1}^{G}|o_j|$ is the total number of tokens across all responses.

The covariance-weighted advantage reweighting modifies the token-level loss as:

$$\tilde{\ell}_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\cdot(\tilde{w}_{i,t}\hat{A}_i) - \beta\text{KL}_{i,t}.$$

Our advantage reweighting approach automatically maintains the policy entropy at a reasonable level by adaptively filtering extreme covariance tokens that would otherwise cause performance issues.

3 Experiments

3.1 Experimental Setup

Models. We consider two different scales of models. For 1.5B models, we use DeepSeek-R1-Distill-Qwen-1.5B, which is Qwen2.5-Math-1.5B (Yang et al., 2024) fine-tuned on reasoning data from DeepSeek-R1 (Guo and DeepSeek-AI, 2025). For 7B models, we use Qwen2.5-Math-7B-Instruct (Yang et al., 2024).

Datasets. For training dataset, we use Open-RS (Dang and Ngo, 2025), which consists of 7k high quality math questions. For evaluation datasets, we use five broadly used math benchmarks. More details about the dataset information can be found in Appendix A.

Experimental Setup. Following Dang and Ngo (2025), we set the sampling temperature to 0.7 during training. The specific prompt template used for training is detailed in Appendix B, and comprehensive implementation details are provided in Appendix C.

Model	Fine-tuning Samples	AIME24	MATH-500	AMC23	Minerva	OlympiadBench	Avg.
General Models							
Llama-3.1-70B-Instruct		16.7	64.6	30.1	35.3	31.9	35.7
o1-preview		44.6	85.5	–	–	–	–
1.5B Models							
Still-3-1.5B-Preview	30,000	32.5	84.4	66.7	29.0	45.4	51.6
DeepScaleR-1.5B-Preview	40,000	43.1	87.8	73.6	30.2	50.0	57.0
Open-RS1	18,615	30.0	83.8	70.0	29.0	52.4	53.0
1.5B Model Experiments							
DeepSeek-R1-Distill-Qwen-1.5B	***Base Model***	28.8	82.8	62.9	26.5	43.3	48.9
GRPO	7000	33.3	85.0	67.5	27.2	49.9	52.6 (+3.7)
Clip-Cov (Cui et al., 2025b)	7000	33.3	85.5	70.0	29.0	50.0	53.6 (+4.7)
GRPO + Gaussian Reweight (ours)	7000	30.0	87.0	77.5	29.8	52.0	55.3 (+6.4)
7B Models							
rStar-Math-7B		26.7	78.4	47.5	–	47.1	–
Eurus-2-7B-PRIME		26.7	79.2	57.8	38.6	42.1	48.9
Qwen2.5-7B-SimpleRL		26.7	82.4	62.5	39.7	43.3	50.9
7B Model Experiments							
Qwen-2.5-Math-7B-Instruct	***Base Model***	3.3	82.6	47.5	33.1	40.4	41.4
GRPO	7000	10.0	82.2	55.0	33.1	40.3	44.1 (+2.7)
Clip-Cov (Cui et al., 2025b)	7000	10.0 (3/30)	82.4 (412/500)	57.5 (23/40)	32.4 (88/272)	41.3 (279/675)	44.7 (+3.3)
GRPO + Gaussian Reweight (ours)	7000	13.3	82.8	62.5	32.0	42.7	46.7 (+4.3)

Table 2: **Main Experimental Results.** This table presents zero-shot pass@1 performance across mathematical reasoning benchmarks. Values in parentheses indicate the improvement over the base model.

3.2 Results

Accuracy Results. We present the experimental results in Table 2. Our covariance-weighted GRPO consistently outperforms vanilla GRPO across both model scales and all evaluation benchmarks. For the 1.5B DeepSeek-R1-Distill-Qwen model, our method achieves an average improvement of +2.7 points over vanilla GRPO, with particularly notable gains on AMC23 (Li et al., 2024) and Minerva (Lewkowycz et al., 2022). Similarly, for the 7B Qwen2.5-Math model, our approach delivers a +2.4 point improvement on average, with the most substantial gain observed on AMC23 and OlympiadBench (He et al., 2024). These consistent improvements across different model architectures and mathematical reasoning benchmarks demonstrate the effectiveness of our covariance-aware advantage reweighting in enhancing reasoning performance.

Policy Entropy Results. Using DeepSeek-R1-Distill-Qwen-1.5B, we plot the dynamics of entropy during training as introduced in Section 2.2. As shown in Figure 1, our proposed method maintains entropy at a reasonable level, effectively balancing exploration and exploitation, while vanilla GRPO exhibits significant entropy instability. To demonstrate that stable entropy correlates with better reasoning performance, we evaluate model performance at different training checkpoints on MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023) and OlympiadBench (He et al., 2024) to ensure more reliable statistical measurements, as shown in Table 3.

Method	Training Step	MATH-500	OlympiadBench
GRPO	100	85.0	49.9
	150 (entropy low)	82.0	49.9
	200 (entropy high)	79.8	47.8
CW-GRPO	100	87.0	52.0
	150 (entropy low)	86.2	53.9
	200 (entropy high)	86.4	53.5

Table 3: **Performance Comparison at Different Checkpoints.** Vanilla GRPO training exhibits entropy instability, resulting in degraded performance.

The results confirm our hypothesis: vanilla GRPO's entropy instability leads to performance degradation (from 85.0% to 79.8% on MATH-500), while our method maintains consistent performance (86.2%-87.0%) across all checkpoints, validating the importance of controlling extreme covariance values.

4 Conclusion

We present a covariance-aware extension to GRPO that uses Gaussian kernel weighting to moderate extreme token-level updates, automatically improving the exploration-exploitation trade-off without additional hyperparameters. Experimental results across multiple reasoning benchmarks demonstrate consistent improvements over vanilla GRPO, validating our principled approach to enhancing reasoning performance through more balanced gradient updates.

Limitations

Our experiments are conducted on models up to 7B parameters, and while the results demonstrate consistent improvements, further validation on larger-scale models would strengthen the evidence for the method's broad applicability. Additionally, our evaluation focuses primarily on mathematical reasoning tasks, which provide clear correctness criteria ideal for testing our approach. Exploring the method's effectiveness across more diverse tasks would offer broader insights into its general utility.

Acknowledgments

We appreciate the reviewers for their insightful comments and suggestions. Qin Liu and Muhao Chen were supported by an Amazon Nova Trusted AI Prize, grants OAC 2531126 and ITE 2333736 from the United States National Science Foundation.

References

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. 2025a. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. 2025b. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, et al. 2025c. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Quy-Anh Dang and Chris Ngo. 2025. Reinforcement learning for reasoning in small llms: What works and what doesn't.

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

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models.

Daya Guo and DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. 2025. Rethinking entropy interventions in rlvr: An entropy change perspective. *arXiv preprint arXiv:2510.10150*.

Andre He, Daniel Fried, and Sean Welleck. 2025. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems.

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

Sham M Kakade. 2001. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath. `https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf`.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

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

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for llm reasoning.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Yongkang Li, Yu Wu, and Daya Guo. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters.

Chen Wang, Lai Wei, Yanzhi Zhang, Chenyang Shao, Zedong Dan, Weiran Huang, Yuzhi Zhang, and Yue Wang. 2025a. Eframe: Deeper reasoning via exploration-filter-replay reinforcement learning framework.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, and Simon S. Du. 2025b. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. 2025. Genprm: Scaling test-time compute of process reward models via generative reasoning.

A Datasets Information

We use Open-RS dataset as the training set, which is curated by Dang and Ngo (2025), totaling 7,000 samples: 3,000 from the Open-s1 dataset (Dang and Ngo, 2025) (filtered mathematical problems from diverse sources like NuminaMATH (LI et al., 2024) and AIME), 3,000 from the Open-DeepScaleR (Dang and Ngo, 2025) dataset (mathematics problems from AIME, AMC, and Omni-MATH (Gao et al., 2024)), and 1,000 easier problems from the DeepScaleR (Guo and DeepSeek-AI, 2025) dataset. Both models are trained on the Open-RS dataset. For evaluation, we select five datasets: AIME24, MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023), AMC23, Minerva (Lewkowycz et al., 2022) and OlympiadBench (He et al., 2024). More information is presented in Table 4.

Name	Huggingface Path	Size
AIME24	HuggingFaceH4/aime_2024	30
AMC23	knoveleng/AMC-23	40
MATH-500	HuggingFaceH4/MATH-500	500
Minerva	knoveleng/Minerva-Math	272
OlympiadBench	knoveleng/OlympiadBench	675

Table 4: **Datasets Information.**

B Training Prompt

The prompt used during training is presented in Figure 4, in which we instruct the model to use English only, as we have observed some language mixture issues.

Prompt Used for Training

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer, and put your final answer within boxed . The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. Note that respond by English, NOT use other languages.

Figure 4: prompt

C Implementation Details

We conduct all training and evaluation using four NVIDIA H200 GPUs. Our reward function combines accuracy and format metrics. For accuracy, we compare the parsed model output against the

ground truth using a verification function implemented in LaTeX. The function assigns a reward of 1.0 for exact matches and 0.0 otherwise. For format compliance, we award a reward of 1.0 when the output contains properly matched `<think>` tags.

Training is performed using the HuggingFace `trl` framework[1], while evaluation utilizes the HuggingFace `lighteval` framework[2]. The specific hyperparameters used in our experiments are detailed in Table 5.

Parameter	Value
Learning Rate	1.0×10^{-6}
Batch Size	12
Gradient Accumulation Steps	4
Training Steps	100
Warmup Ratio	0.1
Max Prompt Length	512
Max Completion Length	4096
Temperature	0.7
Number of Generations	12

Table 5: **Hyperparameters Configuration for Experiments.**

D Related Work

Test-time Scaling for Reasoning Tasks. Test-time scaling has emerged as a promising paradigm for improving LLM performance by allocating additional computational resources during inference. (Snell et al., 2024) demonstrated that scaling test-time compute optimally can be more effective than scaling model parameters, showing over 4× efficiency gains through compute-optimal strategies. (Muennighoff et al., 2025) introduced a simplified approach using "budget forcing" to control inference compute by appending "Wait" tokens, achieving strong reasoning with minimal training data. (Zhao et al., 2025) advanced the field with GenPRM, a generative process reward model that scales test-time compute through explicit Chain-of-Thought reasoning. (Setlur et al., 2024) proposed that effective process rewards should measure progress by evaluating likelihood changes before and after each reasoning step.

Reinforcement Learning for Reasoning Tasks. Reinforcement Learning with Verifiable Rewards (RLVR) has rapidly become the dominant route for eliciting step-by-step reasoning in LLMs. Shao et al. (2024b) first showed that Group Relative Policy Optimization (GRPO) can improve while dispensing with a value network, and the recipe was later scaled in DeepSeek-R1 (DeepSeek-AI, 2025). Subsequent work diagnoses exploration bottlenecks: unlikeliness reward boosts low-probability but correct trajectories (He et al., 2025), whereas covariance-based clipping traces early saturation to entropy collapse (Cui et al., 2025c; Hao et al., 2025). Efficiency studies reveal that even a *single* worked example can unlock large gains through 1-shot RLVR (Wang et al., 2025b).

[1] https://github.com/huggingface/trl
[2] https://github.com/huggingface/lighteval