# RiskPO: Risk-based Policy Optimization with Verifiable Reward for LLM Post-Training

**Tao Ren*** **Jinyang Jiang*** **Hui Yang** **Wan Tian** **Yijie Peng†**
Peking University
{rtkenny, jinyang.jiang, yanghui6}@stu.pku.edu.cn
wantian61@foxmail.com, pengyijie@pku.edu.cn

## Abstract

Reinforcement Learning with Verifiable Reward has become a central paradigm for post-training Large Language Models (LLMs). Group Relative Policy Optimization (GRPO) with the mean-based objective suffers from limited exploration and reasoning gains. We propose Risk-based Policy Optimization (RiskPO), which leverages risk measures from Operations Research to address these issues. In particular, we introduce a Mixed Value-at-Risk objective and adopt a bundle-wise training scheme that bundles multiple questions to provide stable and informative signals. Numerical results show that RiskPO consistently outperforms GRPO and its variants across multiple mathematical reasoning benchmarks, achieving substantial improvements on both Pass@1 and Pass@k metrics. These results highlight the effectiveness of risk-based optimization in enhancing exploration and expanding the reasoning capabilities of LLMs.

## 1 Introduction

Reinforcement learning with verifiable reward (RLVR) has emerged as a powerful paradigm for large language models (LLMs) post-training, particularly in enhancing their reasoning abilities. Unlike traditional RL from human feedback, RLVR leverages objective and binary reward signals, providing clear optimization feedback. Maximizing the expected average reward is anticipated to improve task performance of LLMs. Within this framework, a series of efficiency-oriented extensions have been developed from the classical policy-based RL method. Among them, Group Relative Policy Optimization (GRPO) achieves substantial efficiency gains by discarding redundant structures originally designed for standard RL tasks, and has become the de facto baseline in this area (Shao et al., 2024). Since then, several variants have been proposed. Yu et al. (2025) integrate a set of practical techniques into GRPO. Liu et al. (2025) remove the length and standard deviation normalization terms. Zhao et al. (2025) adopt the geometric mean of token-level rewards. Zheng et al. (2025) perform sequence-level clipping, rewarding, and optimization. These algorithms have each demonstrated improvements to varying degrees on standard benchmarks.

However, RLVR methods that maximize average performance exhibit a fundamental limitation: entropy collapse. Prior work shows that models trained with RLVR often experience rapid entropy collapse in the early stages of training, leading to premature convergence and a plateau in performance with little subsequent improvement (Cui et al., 2025). Entropy, as emphasized by several studies, is a key indicator of exploration capacity in reinforcement learning (Wang et al., 2025; Cheng et al., 2025; Hou et al., 2025). Once entropy collapses, the model becomes overconfident, reduces exploration prematurely, and fails to acquire new knowledge effectively. As a consequence, LLMs do not truly expand their intrinsic reasoning capacity; the observed improvements often reflect more efficient sampling of known answers rather than genuinely stronger reasoning skills (see, e.g., Yue et al., 2025;

* These authors contributed equally to this work.
† Corresponding author.

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

Xiong et al., 2025). In practice, RLVR tends to improve fluency and proficiency on easier problems but does not substantially advance the asymptotic reasoning capability of the base model.

We posit that a fundamental limitation of classical RL algorithms arises from their mean-based objective, which disproportionately emphasizes common, high-probability trajectories while underweighting rare but informative reasoning paths. Consequently, the optimization often suffers from vanishing gradients on difficult problems, leading to restricted exploration and premature convergence, as it primarily reinforces behaviors the model has already mastered. In contrast, risk-sensitive objectives such as Conditional Value-at-Risk (CVaR) or Range Value-at-Risk (RVaR) (see, e.g., Hu et al., 2025; Shao & Zhang, 2024) place greater emphasis on low-reward cases, thereby mitigating overconfidence and encouraging the discovery of novel reasoning strategies. Inspired by the risk measure from Operations Research (Glynn et al., 2021), we propose Risk-based Policy Optimization (RiskPO), a risk-sensitive RL algorithm that enhances exploration and yields substantial improvements on mathematical reasoning benchmarks.

## 2 Rethinking RLVR from a Distributional Perspective

We formalize the post-training problem of RLVR as follows. Given an input problem $x$ sampled from a dataset $\mathcal{D}$, an LLM parameterized as $\pi_\theta$ generates a response $y \sim \pi_\theta(\cdot|x)$. A rule-based verifier $R(\cdot)$ then evaluates the correctness of the response, returning one if $y$ is correct and zero otherwise. Notably, no intermediate process-level feedback is provided. The standard objective in this setting is to maximize the expected reward: $\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}[R(y)]$. With a score-function method, the gradient can be given by $\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}[R(y)\nabla_\theta \ln \pi_\theta(y|x)]$, resulting in a standard RL framework, where a baseline or so-called value model is used for variance reduction.

As a widely adopted baseline for RLVR, GRPO (Shao et al., 2024) replaces the value model with sequence-level standardized rewards computed within a group of responses. We denote by $y_{<t}$ the partial response consisting of the first $t$ tokens, i.e., $\pi_\theta(y|x) = \prod_t \tilde{\pi}_\theta(y_t|x, y_{<t})$. Specifically, given a query $x$ and a group of $G$ responses $\{y_i\}_{i=1}^G$ sampled from a reference model $\pi_{\theta'}(\cdot|x)$, the GRPO objective is defined as

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{x\sim\mathcal{D} \\ \{y_i\}_{i=1}^G\sim\pi_{\theta'}(\cdot|x)}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\Big(w_{i,t}(\theta)\,\hat{A}_i,\ \text{clip}(w_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_i\Big)\right],$$

where $\hat{A}_i = \frac{R(y_i) - \frac{1}{G}\sum_{j=1}^G R(y_j)}{\text{Std}(\{R(y_j)\}_{j=1}^G)}$ denotes the standardized feedback, while $w_{i,t}(\theta) = \frac{\tilde{\pi}_\theta(y_{i,t}|x, y_{i,<t})}{\tilde{\pi}_{\theta'}(y_{i,t}|x, y_{i,<t})}$ is the importance sampling ratio that enables multiple parameter updates per group of generated data. Despite these modifications, GRPO remains fundamentally a method for optimizing the *mean performance* of LLMs. Since the reward provided by the verifier is an indirect objective, we argue it may not be the best practice for RLVR to optimize its expectation. Instead, we propose to adopt a *distributional perspective*. The most challenging problems correspond to the left tail of the reward distribution. These samples represent the questions that the model has not yet mastered. Such hard cases often lead to gradient vanishing in GRPO. For example, when all responses are incorrect, the computed advantage collapses to zero, which provides no meaningful training signal. As a result, the model fails to improve on its weakest regions of the distribution.

Therefore, beyond optimizing the expectation, we claim that it is more beneficial to consider the distributional structure of performance, particularly the lower tail. Incorporating risk measures, such as CVaR or RVaR, into the training objective emphasizes hard problems in the tail of the distribution and provides a finer-grained and more robust learning signal for RLVR.

## 3 Mastering the Uncertainty with Risk-based Policy Optimization

Denote the RLVR reward signal distribution by $F_\theta(\cdot)$, where the parameter $\theta$ reflects the stochasticity induced by the LLM $\pi_\theta(\cdot|x)$. RVaR is defined to capture the average performance within a specified quantile interval of the distribution. Let $F_\theta^{-1}(\alpha)$ be the $\alpha$-level quantile of $R(y)$. Then, for $0 \le \alpha < \beta \le 1$, RVaR on the interval $[\alpha, \beta]$ is written as

$$\mathcal{J}_{\text{RVaR}_{\alpha:\beta}}(\theta) := \mathbb{E}\big[R(y)|R(y) \in [F_\theta^{-1}(\alpha), F_\theta^{-1}(\beta)]\big], \tag{1}$$

that is, the conditional expectation of $R(y)$ given that it falls between its $\alpha$- and $\beta$-quantiles. To optimize the RVaR through gradient descent algorithms, we first derive the gradient of RVaR as shown in Theorem 1.

**Theorem 1.** *Assume $F_\theta(r)$ is continuously differentiable with respect to both the parameter $\theta$ and the variable $r$; the density is positive at the quantiles, i.e., $f_\theta(F_\theta^{-1}(\alpha)) > 0$ and $f_\theta(F_\theta^{-1}(\beta)) > 0$; and that the differentiation under the integral sign is justified. Then the gradient of RVaR is given by*

$$\nabla_\theta \mathcal{J}_{\mathrm{RVaR}_{\alpha:\beta}}(\theta) = \frac{1}{\beta-\alpha}\mathbb{E}\big[g\big(R(y), F_\theta^{-1}(\alpha), F_\theta^{-1}(\beta)\big)\nabla_\theta \ln \pi_\theta(y|x)\big],$$

*where $g(z,a,b) = (z-a)^+ - (z-b)^+ + a - b$, and $(z)^+ = \max\{z, 0\}$.*

When $\alpha = 0$, $\mathcal{J}_{\mathrm{RVaR}_{0:\beta}}(\theta)$ corresponds to the $\mathrm{CVaR}_\beta$, whose gradient from Theorem 1 reduces to $\nabla_\theta \mathcal{J}_{\mathrm{CVaR}_\beta}(\theta) = \beta^{-1}\mathbb{E}\big[-(F_\theta^{-1}(\beta) - R(y))^+ \nabla_\theta \ln \pi_\theta(y|x)\big]$. Since RVaR effectively places a window for control on the reward distribution, it provides a natural opportunity to combine multiple RVaRs in order to better control the overall distributional shape. Therefore, we introduce a new objective into RLVR, Mixed Value-at-Risk (MVaR), which integrates RVaR and CVaR as follows:

$$\mathcal{J}_{\mathrm{MVaR}^{\omega}_{\alpha:\beta}}(\theta) = (1+\omega)\mathcal{J}_{\mathrm{CVaR}_\alpha}(\theta) + (1-\omega)\mathcal{J}_{\mathrm{RVaR}_{\alpha:\beta}}(\theta),$$

where $\omega \in [-1, 1]$ controls the emphasis placed on tail samples during optimization, and high-performance samples are excluded from the current training process.

However, the distributional information for a single question $x$ is limited, as the feedback takes binary values. Consider the advantage for GRPO, generating either all correct or wrong answers will lead to zero advantage. When the LLM fails to answer a question $x$, the gradient on it essentially becomes zero, which indicates that the model receives no gradient signal on initially unsolved problems, preventing progress on its weakest areas. We propose to group several questions as a bundle, i.e., $X := \{x_i\}_{i=1}^b \sim \mathcal{D}^{\otimes b}$, and calculate the advantage according to the score of the bundle, i.e., the sum of the questions' scores in the bundle. We focus on optimizing the MVaR of the bundle's score:

$$\mathbb{E}_{X\sim\mathcal{D}^{\otimes b}, \{y^i\sim\pi_\theta(\cdot|x_i)\}_{i=1}^b}\left[R_b\big((1+\omega)\mathbf{1}_{\{R_b \le F_\theta^{-1}(\alpha)\}} + (1-\omega)\mathbf{1}_{\{F_\theta^{-1}(\alpha) < R_b \le F_\theta^{-1}(\beta)\}}\big)\right],$$

where $R_b = \sum_{i=1}^b R(y^i)$ denotes the bundle score. For each $i \in \{1, \ldots, b\}$, we sample $Y_i := \{y_j^i\}_{j=1}^G$ with $y_j^i \sim \pi_\theta(\cdot|x_i)$ i.i.d., and define $Y := \{Y_i\}_{i=1}^b$. We can sample $G$ bundles without overlaps from the $G \times b$ responses of $b$ questions. The gradient can be derived as

$$\mathbb{E}_{X\sim\mathcal{D}^{\otimes b}, \{y_j^i\}_{j=1}^G\sim\pi_\theta(\cdot|x_i), \xi_i\sim\mathrm{Unif}(\mathfrak{S}_G)}\left[\frac{1}{G}\sum_{j=1}^G A^{(j)} \frac{1}{b}\sum_{i=1}^b \nabla_\theta \ln \pi_\theta\big(y_{\xi_i(j)}^i \,\big|\, x_i\big)\right],$$

where $A^{(j)} = \frac{1+\omega}{-\alpha}(F_\theta^{-1}(\alpha) - R_b(j))^+ + \frac{1-\omega}{\beta-\alpha} g(R_b(j), F_\theta^{-1}(\alpha), F_\theta^{-1}(\beta))$ is the bundle-wise advantage value, $R_b(j) = \sum_{i=1}^b R(y_{\xi_i(j)}^i)$ is the bundle-wise score, $\xi$ is a permutation of $\{1, \ldots, G\}$ that independently draw $\xi_i \sim \mathrm{Unif}(\mathfrak{S}_G)$ for every $i$, $\mathfrak{S}_G$ is the symmetric group on $G$ element, and $\xi_i(j)$ is the $j$-th elements in the permutation. This construction yields $G$ disjoint *bundles*: the $j$-th bundle uses $\{y_{\xi_i(j)}^i\}_{i=1}^b$, so that for each fixed $i$, $\{y_{\xi_i(1)}^i, \ldots, y_{\xi_i(G)}^i\}$ is a permutation of $\{y_1^i, \ldots, y_G^i\}$, i.e., every answer is used only once (without replacement).

To ensure stable improvement (Schulman et al., 2017, 2015) with multiple updates per bundle-wise MVaR objective evaluation, we adopt a trust-region style update with clipping and *sequence-level* importance sampling (Zheng et al., 2025). Since the reward in RLVR is only available at the sequence level, i.e., $y^i$. it is natural to define importance weights also at the sequence (response) level and then aggregate them into the bundle objective. Formally, given $b$ problems $X = \{x_i\}_{i=1}^b$ and $G$ responses per problem $Y_i = \{y_j^i\}_{j=1}^G$, we independently draw $\xi_i \sim \mathrm{Unif}(\mathfrak{S}_G)$ for each $i$, yielding $G$ *bundles*: $\mathcal{P}^{(j)} = \{y_{\xi_i(j)}^i\}_{i=1}^b, j = 1, \ldots, G$, where every responses is used without replication. We then define the clipped MVaR objective at the bundle level as

$$J_{\mathrm{MVaR}}^{\mathrm{clip}}(\theta) = \mathbb{E}_{X,Y,\{\xi_i\}}\left[\frac{1}{G}\sum_{j=1}^G \frac{1}{b}\sum_{i=1}^b \min\Big(s_j^i(\theta)\, A^{(j)},\ \mathrm{clip}(s_j^i(\theta), 1-\epsilon, 1+\epsilon)\, A^{(j)}\Big)\right], \quad (2)$$

where $s_j^i(\theta) = \left(\frac{\pi_\theta(y^i_{\xi_i(j)}|x_i)}{\pi_{\theta_{\text{old}}}(y^i_{\xi_i(j)}|x_i)}\right)^{1/|y^i_{\xi_i(j)}|}$ is the sequence-wise importance sampling ratio.

Every token within the same bundle shares the same MVaR-based advantage $A^{(j)}$, ensuring that optimization is aligned with the unit of reward (the bundle score) and directs training toward the left tail of the performance distribution. We track $F_\theta^{-1}(\alpha)$ and $F_\theta^{-1}(\beta)$ in an online manner. After substituting the tracked quantiles into the advantage and deriving the gradient, we update the model parameters accordingly. Therefore, RiskPO can be implemented as a two-timescale stochastic approximation algorithm. The pseudocode of the proposed algorithm is provided in the appendix.

## 4 Experiments

Please refer to the appendix for detailed experimental settings. Table 1 reports Pass@1 accuracy across six hard-level mathematical reasoning benchmarks. We observe that RiskPO consistently achieves the best performance among all methods, outperforming both the base models and recent GRPO variants. In particular, RiskPO attains an average score of $47.15$, representing a $+4.4$ absolute improvement over the strongest baseline DAPO ($42.75$) and a $+7.9$ improvement over vanilla GRPO ($39.30$). The gains are especially pronounced on the most challenging AIME datasets, where RiskPO surpasses DAPO by nearly $+10$ points ($33.3$ vs. $23.3$). These results demonstrate that emphasizing distributional risk through our MVaR objective substantially improves reasoning ability, not only enhancing performance on easier datasets like AMC and MATH500 but also pushing the frontier on the hardest Olympiad-style tasks.

Figure 1 presents the curves of Pass@1 and Pass@16 on AMC and MATH500 across training steps. RiskPO achieves faster convergence and higher final accuracy on both metrics, highlighting more efficient learning. Importantly, the gains go beyond merely improving sampling efficiency on problems the model can already solve (e.g., turning "one success in sixteen attempts" into "one-shot success"). RiskPO also enables the model to acquire genuinely new reasoning capabilities: on previously unsolved problems where GRPO fails even after 16 attempts, RiskPO succeeds in generating correct solutions within the same budget. This demonstrates that RiskPO not only improves sample efficiency but also expands the reasoning frontier, equipping the model with novel solution strategies that were unattainable under mean-based objectives.

Table 1: Comparison of Pass@1 performance across hard-level mathematical reasoning benchmarks.

| Model | AIME25 | AIME24 | AMC | MATH500 | Minerva | Oly. | Avg. |
|---|---|---|---|---|---|---|---|
| Qwen2.5-Math-1.5B | 6.6 | 10.0 | 43.4 | 61.8 | 15.1 | 28.4 | 27.55 |
| Qwen2.5-Math-1.5B-Instruct | 10.0 | 10.0 | 48.2 | 64.2 | 26.5 | 35.2 | 32.35 |
| DeepSeek-R1-Distill-Qwen-1.5B | 13.3 | 13.3 | 32.5 | 59.8 | 20.3 | 30.5 | 28.28 |
| Dr.GRPO-1.5B (Liu et al., 2025) | 20.0 | 16.6 | 54.7 | 77.4 | 26.3 | 38.1 | 38.85 |
| GRPO-1.5B (Shao et al., 2024) | 16.6 | 20.0 | 56.6 | 79.2 | 25.7 | 37.6 | 39.30 |
| GPG-1.5B (Chu et al., 2025) | 16.6 | 20.0 | 55.7 | 74.5 | 28.8 | 37.6 | 38.90 |
| DAPO-1.5B (Yu et al., 2025) | 23.3 | 26.6 | 58.6 | 78.2 | 30.2 | 39.6 | 42.75 |
| GMPO-1.5B (Zhao et al., 2025) | 23.3 | 23.3 | 54.2 | 76.2 | 30.2 | 36.2 | 40.57 |
| RiskPO-1.5B (Ours) | **33.3** | **33.3** | **60.8** | **81.8** | **32.5** | **41.2** | **47.15** |

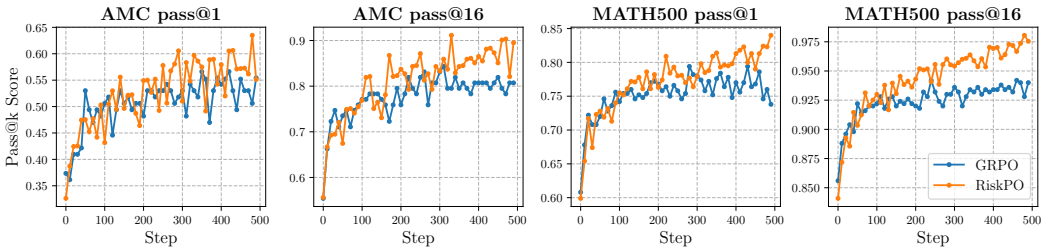


Figure 1: The pass@1 and pass@16 learning curves on the AMC and MATH500 datasets.

## References


Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective on reinforcement learning for llms, 2025. URL https://arxiv.org/abs/2506.14758.

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.

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

Michael C Fu, L Jeff Hong, and Jian-Qiang Hu. Conditional monte carlo estimation of quantile sensitivities. *Management Science*, 55(12):2019–2027, 2009.

Peter W Glynn, Yajun Liu, Chang-Han Rhee, and Rayadurgam Srikant. Computing sensitivities for distortion risk measures. *INFORMS Journal on Computing*, 33(4):1520–1532, 2021.

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.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. T1: Advancing language model reasoning through reinforcement learning and inference scaling, 2025. URL https://arxiv.org/abs/2501.11651.

Jiaqiao Hu, Meichen Song, and Michael C Fu. Quantile optimization via multiple-timescale local search for black-box functions. *Operations Research*, 73(3):1535–1557, 2025.

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

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

MAA. American mathematics contest 12 (amc 12), november 2023, 11 2023. URL https://artofproblemsolving.com/wiki/index.php/AMC_12_Problems_and_Solutions.

MAA. American invitational mathematics examination (aime), february 2024, 2024. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

MAA. American invitational mathematics examination (aime), february 2025, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Hui Shao and Zhe George Zhang. Extreme-case distortion risk measures: a unification and generalization of closed-form solutions. *Mathematics of Operations Research*, 49(4):2341–2355, 2024.

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.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning, 2025. URL `https://arxiv.org/abs/2506.01939`.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

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

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.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. Geometric-mean policy optimization, 2025. URL `https://arxiv.org/abs/2507.20673`.

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.

# A Theoretical Details

*Proof of Theorem 1.* Recall the definition of the RVaR functional:

$$\mathcal{J}_{\text{RVaR}_{\alpha:\beta}}(\theta) = \mathbb{E}\big[R(y)|R(y) \in [F_\theta^{-1}(\alpha), F_\theta^{-1}(\beta)]\big] = \frac{1}{\beta-\alpha}\int_{F_\theta^{-1}(\alpha)}^{F_\theta^{-1}(\beta)} r f_\theta(r)dr.$$

To compute the RVaR gradient, we apply Leibniz's rule for differentiation, yielding

$$\nabla_\theta \mathcal{J}_{\text{RVaR}_{\alpha:\beta}}(\theta) = \frac{1}{\beta-\alpha}\bigg(\int_{F_\theta^{-1}(\alpha)}^{F_\theta^{-1}(\beta)} r\nabla_\theta f_\theta(r)dr + F_\theta^{-1}(z)f_\theta(F_\theta^{-1}(z))\nabla_\theta F_\theta^{-1}(z)\bigg|_\alpha^\beta\bigg).$$

Note that, by the implicit function theorem, the quantile gradient can be expressed as (see, e.g., Fu et al., 2009) $\nabla_\theta F_\theta^{-1}(z) = -\nabla_\theta F_\theta(F_{\bar\theta}^{-1}(z))\big|_{\bar\theta=\theta} / f_\theta(F_\theta^{-1}(z))$. Substituting this identity into our previous expression, we can obtain

$$\nabla_\theta \mathcal{J}_{\text{RVaR}_{\alpha:\beta}}(\theta) = \frac{1}{\beta-\alpha}\bigg(\int_{F_\theta^{-1}(\alpha)}^{F_\theta^{-1}(\beta)} r\nabla_\theta f_\theta(r)dr - F_\theta^{-1}(z)\nabla_\theta F_\theta(F_{\bar\theta}^{-1}(z))\big|_{\bar\theta=\theta}\bigg|_\alpha^\beta\bigg).$$

By the definition of CDF, we have $\nabla_\theta F_\theta(r) = \nabla_\theta \mathbb{E}\big[\mathbf{1}_{\{R(y)\le r\}}\big] = \mathbb{E}\big[\mathbf{1}_{\{R(y)\le r\}}\nabla_\theta \ln f_\theta(R(y))\big]$. Thus, with the score-function method, we rewrite the RVaR gradient in expectation form as

$$\nabla_\theta \mathcal{J}_{\text{RVaR}_{\alpha:\beta}}(\theta) = \mathbb{E}\bigg[\bigg(R(y)\mathbf{1}_{\{R(y)\in[F_\theta^{-1}(\alpha),F_\theta^{-1}(\beta)]\}} - F_\theta^{-1}(z)\mathbf{1}_{\{R(y)\le F_\theta^{-1}(z)\}}\bigg|_\alpha^\beta\bigg)\frac{\nabla_\theta \ln f_\theta(R(y))}{\beta-\alpha}\bigg].$$

Finally, since the distribution of $R(y)$ is induced by the LLM $\pi_\theta(\cdot|\cdot)$, we can apply the score-function transformation to yield the final expression

$$\nabla_\theta \mathcal{J}_{\text{RVaR}_{\alpha:\beta}}(\theta) = \frac{1}{\beta-\alpha}\mathbb{E}\big[g\big(R(y), F_\theta^{-1}(\alpha), F_\theta^{-1}(\beta)\big)\nabla_\theta \ln \pi_\theta(y|x)\big],$$

which completes the proof. □

# B Experiment Details

## B.1 Pseudocode

**Algorithm 1** Risk-based Policy Optimization

**Input**: quantile levels $\alpha, \beta$, policy $\pi_\cdot$, learning rates $\{\gamma_k\}$, $\{\eta_k\}$, and # of iterations $K$
**Initialize**: policy parameter $\theta_0$, and quantile trackers $q_0^\alpha$, $q_0^\beta$
**for** $k = 1, \cdots, K$ **do**
  Sample $b$ questions, $X = \{x_i\}_{i=1}^b$, from the dataset $\mathcal{D}$
  Generate $G$ responses for each question, $\{y_j^i\}_{j=1}^G \sim \pi(\cdot|x_i)$ and evaluate the reward $R(y_j^i)$
  Sample from the symmetric group for $b$ times, $\xi_i \sim \text{Unif}(\mathfrak{S}_G)$, yielding $G$ bundles
  Track quantiles with batched bundles' scores: $q_{k+1}^\alpha = q_k^\alpha + \gamma_k(\alpha - \frac{1}{G}\sum_{j=1}^G \mathbf{1}\{R_b(j) < q_k^\alpha\})$, $q_{k+1}^\beta = q_k^\beta + \gamma_k(\beta - \frac{1}{G}\sum_{j=1}^G \mathbf{1}\{R_b(j) < q_k^\beta\})$
  Evaluate the clip MVaR objective (2) and its gradient $\nabla_\theta J_{\text{MVaR}}^{\text{clip}}(\theta_k)$ via auto-differentiation
  Update policy parameter: $\theta_{k+1} = \theta_k + \eta_k \nabla_\theta J_{\text{MVaR}}^{\text{clip}}(\theta_k)$
**end for**
**Output**: Final policy parameter $\theta_K$

## B.2 Training configuration

**Model.** We focus on mathematics reasoning tasks. We use DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) as our baseline model to evaluate different algorithms. We also compare our performance with Qwen2.5-Math-1.5B and Qwen2.5-Math-1.5B-Instruct (Yang et al., 2024).

**Training.** For the mathematics reasoning tasks. We use the DAPO-math-17k as the training set. We set the clipping threshold $\epsilon = 0.2$. KL penalty and entropy regularization are omitted from the loss objective. We use vLLM as the inference backend and FSDP as the training backend. We set the temperature to $0.8$ top_p to $1.0$, and maximum output length as $3072$. We generate $10$ responses for each problem. The batch size is $512$, the mini-batch size is set to $128$. For quantile level, we set $\alpha$ to $0.1$ and $\beta$ to $0.9$ correspondingly, and $\omega = 0.5$. The bundle size $b$ is set to 5. All the training is conducted on a machine with 8 NVIDIA H20 GPUs. Each GPU has 96 GB of memory.

**Evaluation.** We evaluate on six math reasoning datasets: AIME24 (MAA, 2024) and AIME25 (MAA, 2025) with $30$ problems from the American Invitational Mathematics Examination, both targeting advanced pre-collegiate reasoning; AMC23 (MAA, 2023) with $83$ problems from the American Mathematics Competitions, testing creative algebraic, geometric, and number-theoretic skills; MATH-500 (Lightman et al., 2023) with 500 graduate-level problems from the original MATH dataset covering algebra, geometry, and number theory; Minerva Math (Lewkowycz et al., 2022) with $272$ undergraduate-level quantitative reasoning problems; and OlympiadBench (He et al., 2024) with $675$ Olympiad-style problems.