

Stable Reinforcement Learning for Efficient Reasoning

Muzhi Dai^{◇*}, Shixuan Liu^{♡*}, Qingyi Si^{◇†},

[◇]Huawei Technologies Co., Ltd. [♡]Australian National University

mzdai666@gmail.com, u6920173@anu.edu.au, siqingyi@huawei.com

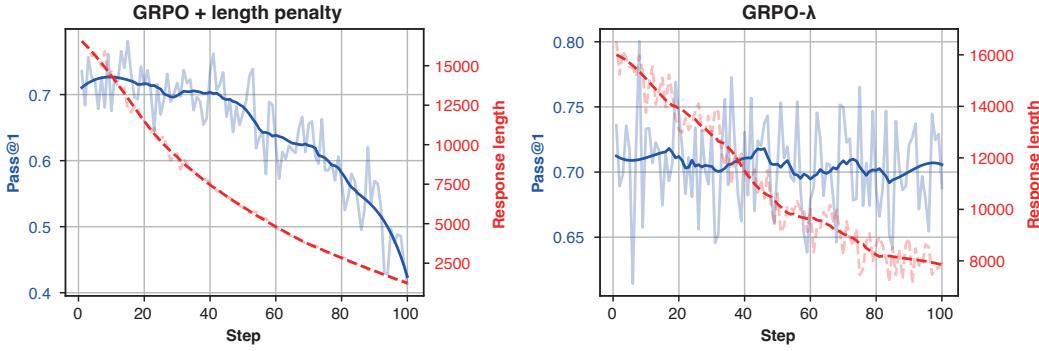


Figure 1: Training process of *GRPO+length penalty* and our *GRPO- λ* .

Abstract

The success of Deepseek-R1 has drawn the LLM community’s attention to reinforcement learning (RL) methods like GRPO. However, such rule-based 0/1 outcome reward methods lack the capability to regulate the intermediate reasoning processes during chain-of-thought (CoT) generation, leading to severe overthinking phenomena. In response, recent studies have designed reward functions to reinforce models’ behaviors in producing shorter yet correct completions. Nevertheless, we observe that these length-penalty reward functions exacerbate RL training instability: as the completion length decreases, model accuracy abruptly collapses, often occurring early in training. To address this issue, we propose a simple yet effective solution **GRPO- λ** , an efficient and stabilized variant of GRPO, which dynamically adjusts the reward strategy by monitoring the correctness ratio among completions within each query-sampled group. A low correctness ratio indicates the need to avoid length penalty that compromises CoT quality, triggering a switch to length-agnostic 0/1 rewards that prioritize reasoning capability. A high ratio maintains length penalties to boost efficiency. Experimental results show that our approach avoids training instability caused by length penalty while maintaining the optimal accuracy-efficiency trade-off. On the GSM8K, GPQA, MATH-500, AMC 2023, and AIME 2024 benchmarks, it improves average accuracy by 1.48% while reducing CoT sequence length by 47.3%.

* Equal Contribution.

† Corresponding Author.

1 Introduction

Recent advances in large language model (LLM) community have been driven by the development of test-time scaling [1], demonstrating a positive correlation between generation length and models’ reasoning capability, which is more effective than model-parameter scaling law [2]. The open-source releases of DeepSeek-R1 [3] and Qwen3 [4] have further stimulated recent research on reinforcement learning (RL) [5–9] for achieving reasoning models [10]. These models typically generate extended chain-of-thought (CoT) [11] sequences containing rich and diverse reasoning paths.

However, recent studies [12, 13] have revealed that reasoning models often suffer from severe overthinking [12, 14] issues, characterized by excessive shallow reasoning steps and frequent thought-switching in prolonged CoTs [15, 14, 16]. This occurs because the rule-based outcome rewards in GRPO [5] cannot effectively regulate intermediate reasoning processes. While longer reasoning chains statistically increase the probability of containing correct reasoning steps (thus improving answer accuracy and rewards during RL training), this GRPO mechanism continuously reinforces the lengthy CoT generation, and results in overthinking problems.

To address this issue, representative reasoning models like Kimi-1.5 [17–19] incorporate length penalty into RL training, constraining the model to generate higher-quality reasoning within shorter sequences, thereby mitigating overthinking while improving inference efficiency. For example, [18] assigns the highest reward to the shortest correct completion within the group. However, as shown in Figure 1 (left), we reveal that introducing length-aware reward or penalty functions leads to premature RL training collapse: although CoT sequence length decreases as intended, model accuracy abruptly plummets, preventing stable RL training for sufficient iterations.

Intuitively, reasoning models require distinct training priorities at different competency stages: when reasoning capability is underdeveloped, reinforcement should prioritize accuracy, whereas efficiency optimization (via length penalty) should only be introduced once the model demonstrates sufficient reasoning capability. Current methods [19, 18] overlook this progression, indiscriminately shortening CoT sequences for all samples during RL training, ultimately degrading the model’s inherent reasoning capacity and causing RL training to collapse. Motivated by these insights, we propose a simple yet effective modification to GRPO, namely **GRPO- λ** , that sustainably improves reasoning efficiency without compromising reasoning accuracy, thereby preventing RL training collapse and ensuring sufficient training iterations, as shown in Figure 1(right). Specifically, we sample a set of completions per query following standard GRPO method, then evaluate the group-wise correctness rate, and dynamically switches between optimization modes: applying length penalties once correctness is adequately high (indicating mature reasoning capability to prioritize efficiency) or defaulting to standard GRPO’s 0/1 outcome rewards (to reinforce accuracy fundamentals when below threshold). In this way, our method enables the joint optimization of reasoning efficiency and accuracy while ensuring training stability.

Experimental results on GSM8k [20], GPQA [21], AIME 2024 [22], AMC 2023 [23], and MATH-500 [24] benchmarks demonstrate that GRPO- λ approach achieves the dual benefit: (1) enhanced training stability (enabling at least 2.5x more viable iterations) and (2) optimal performance-length tradeoffs, with a remarkable 47.3% reduction in sequence length while improving accuracy by 1.48%.

2 Related Work

The success of OpenAI-o1 [25, 26] reveals that post-training through reinforcement learning serves as a mainstream paradigm for unlocking advanced reasoning capabilities in LLMs. Following the pioneering work of Deepseek-R1 [3] and Qwen3 [4], rule-based outcome reward RL methods [3, 17, 27–31] like GRPO [5] are widely adopted in post-training, encouraging models to produce long CoT outputs, at the cost of inducing overthinking issues [12–14].

To solve it, recent studies [19, 18, 17] have independently proposed various length penalty mechanisms in reward function design. While these approaches share the common objective of promoting shorter responses and penalizing longer responses among correct ones, they implement distinct strategies. Specifically, Kimi 1.5 [12] first normalizes the length of sampled responses. For all responses exceeding 0.5 of the normalized length threshold, it assigns negative rewards, whereas those below receive positive rewards. Incorrect responses are restricted to a maximum reward of 0. Similarly, [18] employs a soft-clip sigmoid function to standardize and smooth length deviations

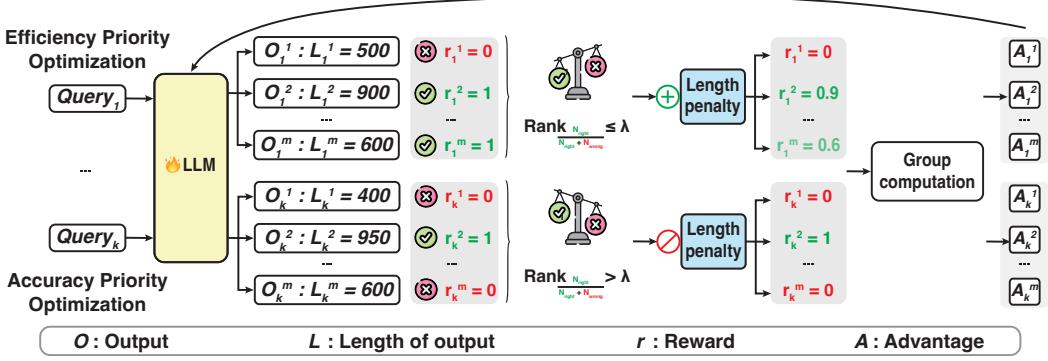


Figure 2: Framework of GRPO- λ .

from the group distribution. This maps rewards to the interval (0,1), where shorter and correct responses receive values closer to 1, while incorrect responses are assigned zero reward. S-GRPO [19] adopts a dual-rollout strategy, performing early-exit interventions at different positions within the first rollout response to construct a serial group, and allocating exponentially decaying rewards based on positional precedence, with zero reward for incorrect ones.

Empirical observations reveal that such methods consistently induce premature collapse during RL training. This stems from their unilateral emphasis on length penalization without assessing potential compromises to the model’s reasoning capability. In essence, when the model demonstrates strong pass@1 performance, length optimization should take priority for enhanced efficiency. Conversely, when sampled responses within the group fail to achieve satisfactory accuracy (weak pass@1), the focus should shift to reinforcing reasoning abilities rather than pursuing reasoning efficiently. Our method GRPO- λ addresses this limitation by adaptively balancing these objectives, enabling more stable and prolonged RL training that ultimately achieves superior performance-efficiency trade-offs.

3 Methods

We introduce GRPO- λ , a stabilized and efficient variant of GRPO designed to address training instability caused by length-penalty reward. GRPO- λ uses batch-wise dynamic adjustment of reward strategies, which selectively applies efficiency-prioritized or accuracy-prioritized optimization for different subsets of groups within a batch. This design ensures a controlled reduction in reasoning sequence length while maintaining accuracy, thereby preventing abrupt training collapse. Below, we detail the components and workflow of GRPO- λ .

Query-Sampled Group Generation. For each training query Q_k in the batch, the model generates m candidate completions $\{O_k^1, O_k^2, \dots, O_k^m\}$ using standard sampling techniques. Each completion O_k^i is associated with: (1) Length L_k^i , indicating the number of tokens in the completion, and (2) Outcome Reward r_k^i , a binary 0/1 reward indicating whether O_k^i is correct ($r_k^i = 1$) or incorrect ($r_k^i = 0$).

Batch-Wise Top- λ Selection. For each batch of queries, we evaluate the correctness of each query-completion group and compute its correctness ratio. GRPO- λ selects the top- λ fraction of query-completion groups in terms of correctness ratio within the batch for efficiency-prioritized optimization. Specifically, the groups are ranked based on their correctness ratio within the batch. The top- λ fraction (e.g., the top 20%) is selected for efficiency-prioritized optimization, as shown in Figure 2 (Upper), as these groups demonstrate sufficient reasoning capability to focus on length reduction. The remaining groups in the batch are assigned to accuracy-prioritized optimization to ensure that the model continues to improve its reasoning capability.

Dynamic Reward Strategy Adjustment. Based on the batch-wise top- λ selection, GRPO- λ applies two distinct reward strategies:

Table 1: Experimental results on Qwen3-8B. "LP" indicates length penalty. * indicates results trained with identical step counts to GRPO- λ , having undergone training collapse. "Acc" denotes accuracy, "Tok" denotes token count, and "CR" denotes compression rate. The top-2 best results are in bold.

Method	GSM8K			GPQA			MATH-500			AMC 2023			AIME 2024			Overall	
	Acc↑	Tok↓	CR↓	Acc↑	Tok↓	CR↓	Acc↑	Tok↓	CR↓	Acc↓	Tok↓	CR↓	Acc↑	Tok↓	CR↓	Acc↑	CR↓
Qwen3-8B																	
Vanilla	95.4	2,370	100%	55.6	8,741	100%	93.4	5,577	100%	91.3	9,452	100%	74.1	15,326	100%	81.90	100%
+GRPO	95.8	2,355	99.4%	55.8	8,819	100.9%	94.4	5,440	97.5%	92.8	8,983	95.0%	72.7	15,154	98.9%	82.30	98.34%
+LP	95.4	1,323	55.8%	55.4	4,930	56.4%	94.2	2,874	51.5%	92.8	4,933	52.2%	71.9	9,266	60.5%	81.94	55.28%
+LP*	94.6	250	10.5%	53.8	732	8.4%	86.0	507	9.1%	75.9	874	9.2%	32.1	2,037	13.3%	68.48	10.1%
+GRPO- λ	95.5	1,114	47.0%	56.8	4,872	55.7%	96.0	2,990	53.6%	94.4	4,751	50.3%	74.4	8,714	56.9%	83.42	52.7%

- Efficiency Priority Optimization (with Length Penalty): For the top- λ fraction of query-completion groups (those with higher correctness ratio), a length-penalty reward is applied to encourage shorter reasoning sequences:

$$r_k^i = \begin{cases} 1 - \alpha \cdot \sigma\left(\frac{L_k^i - \text{mean}(L_k)_{\text{correct}}}{\text{std}(L_k)_{\text{correct}}}\right) & \text{if } O_k^i \text{ is correct} \\ 0 & \text{if } O_k^i \text{ is wrong} \end{cases} \quad (1)$$

where α is the length penalty coefficient. $\text{mean}(L_k)_{\text{correct}}$ and $\text{std}(L_k)_{\text{correct}}$ are mean and standard deviation of completion lengths whose answers are correct, respectively. Incorrect completions ($r_k^i = 0$) receive no reward. This strategy prioritizes reasoning efficiency for groups that already demonstrate sufficient accuracy.

- Accuracy Priority Optimization (0/1 Outcome Reward): For the remaining groups in the batch (those not in the top- λ subset), the reward defaults to the standard GRPO 0/1 outcome reward:

$$r_k^i = \begin{cases} 1 & \text{if } O_k^i \text{ is correct} \\ 0 & \text{if } O_k^i \text{ is wrong} \end{cases} \quad (2)$$

This strategy ensures that the model focuses on improving reasoning accuracy for completions with lower correctness scores.

This reward strategy prevents the imbalanced emphasis on efficiency over accuracy that can arise from directly using length penalty for all groups [17, 32]. This ensures a controlled transition between accuracy and efficiency priorities, effectively curbing the risk of a sharp decline in accuracy.

Advantage Computation and Parameter Update After obtaining the decaying rewards, like GRPO, GRPO- λ calculates the advantage for each sample based on the group rewards. Specifically, the mean and standard deviation (std) of the rewards within the group are computed, and the advantage for each sample is calculated using the formula: $\hat{A}_i = \frac{r_i - \text{mean}(r_i)}{\text{std}(r_i)}$. Subsequently, the computed advantage for each sample is broadcast to all corresponding response tokens. Finally, parameter updates are performed based on the advantage values of each sample.

4 Experiments

4.1 Benchmarks and Settings.

We conducted comprehensive evaluations of our method on several mainstream reasoning benchmarks, including mathematical tasks (GSM8K [20], MATH-500 [24], and the more challenging AMC 2023 [23] and AIME 2024 [22]) as well as the scientific reasoning benchmark GPQA [21].

We choose Qwen3-8B [4] as the base model for experiments. For training data, we select queries from DeepMath-103K [33]. Specifically, we sample 8 times for each query using Qwen3-8B, and select queries that can be answered correctly 2-6 times. During training, we use a learning rate of 1×10^{-6} and randomly sample 16 times for each query. The generation batch size and training batch size are both set to 128×16 . For the length penalty, we set the scalar parameter α to 0.2. For GRPO- λ , we set λ equal to 20%. Across all experiments, we employ Adam [34] as the standard optimizer.

4.2 Experimental Results

As shown in Table 1, our method achieves the optimal trade-off between accuracy and efficiency. Compared to the conventional *GRPO+length penalty* approach, GRPO- λ further improves the average accuracy by 1.48% while achieving more significant sequence length compression on five benchmarks. Notably, for more challenging mathematical tasks (e.g., AIME 2024, AMC 2023), the benefits of our method become even more pronounced, as the relatively simpler mathematical and scientific tasks (e.g., GSM8K, GPQA datasets) are less sensitive to length variations.

The results of *GRPO+length penalty** confirm that incorporating length penalty into the reward function leads to training collapse. Specifically, when trained for the same number of steps as GRPO- λ , *GRPO+length penalty** achieves more significant sequence length compression of 89.9% but suffers a substantial accuracy drop of 13.42%. This phenomenon should be avoided in post-training optimization, as length compression without preserving accuracy becomes meaningless. Furthermore, as shown in Figure 1, the accuracy of *GRPO+length penalty* begins to decline after 40 steps, whereas our method maintains stable performance even at 100 steps, extending effective training steps by at least 2.5 \times . This demonstrates that our approach provides stable reinforcement learning for efficient reasoning.

4.3 Discussion.

Figure 3 presents the relationship between CoT length and accuracy for *GRPO+length penalty* and GRPO- λ , where our method’s curve consistently occupies the Pareto-superior region to the left and above *GRPO+length penalty*’s curve. Specifically, when *GRPO+length penalty* attains similar lengths to our approach, we observe a significant accuracy gap in our favor; conversely, when matching our accuracy levels, *GRPO+length penalty* requires substantially longer reasoning chains (e.g., ~ 7000 vs. ~ 5000 tokens at accuracy ≈ 0.94).

As the sequence length progressively decreases, the accuracy of *GRPO+length penalty* exhibits a consistent decline, whereas our method maintains robust stability in performance. Crucially, recent studies [1, 35] reveal that excessive length reduction inevitably compromises the model’s reasoning capability. GRPO- λ adaptively optimizes sequence length within an appropriate range without sacrificing accuracy. Notably, the dense clustering of data points around the length of 5000 suggests this represents the minimal length preserving model accuracy, which serves as a critical threshold that our method automatically converges to.

Figure 4 presents case samples that reveal three distinct behaviors: Qwen3-8b, while generating the longest response, provides incorrect answers due to its overthinking issue; *GRPO+length penalty* successfully reduces sequence length but at the cost of impairing the model’s reasoning capability, resulting in erroneous responses; in contrast, our method achieves correct answers while operating at the shortest sequence length.

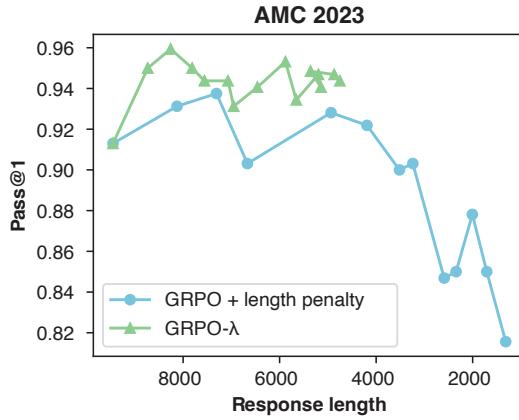


Figure 3: Relationship between performance and response length of GRPO + length penalty and GRPO- λ on AMC 2023 benchmark as training progresses.

5 Conclusion and Future Work

This paper presents the first systematic study on how length-penalty reward design impacts RL training stability in post-training and proposes GRPO- λ , a simple yet effective method. Through extensive experiments, we reveal critical insights for balancing efficiency and accuracy. Specifically, the CoT length reduction rate must be carefully controlled, as excessively rapid shortening inevitably

Q: If Sam and Harry have 100 feet of fence between them, and they agree to split it with Harry getting 60 feet more than Sam, how much is left over for Sam?
Please reason step by step, and put your final answer within \boxed{ }.

Qwen3-8b

Okay, let's see. So the problem is about Sam and Harry having 100 feet of fence between them. They decide to split it, but Harry gets 60 feet more than Sam. The question is asking how much is left

<omitted>

I think the problem is either misworded or there is a translation error. Given the strict interpretation, the answer is 0. So I will go with \boxed{0}.
Final Answer**\boxed{0}\n</think>\n\nTo solve the problem, we need to determine how much fence is left over for Sam after splitting a total of 100 feet of fence between Sam and Harry, with the condition that "Harry gets 60 feet more than Sam".\n\n---\n## Step 1: Define Variables\n\n

<omitted>

split between them.\n\nHowever, since the total amount of fence is exactly "100 feet", and the entire amount is distributed between Sam and Harry (20 feet to Sam, 80 feet to Harry), "there is no fence left over".\n\nTherefore, the amount of fence left over for Sam is:\n\n\boxed{0}

✗ Wrong

Whole token number: 6091

GRPO + Length penalty

Okay, let's see. Sam and Harry have 100 feet of fence together, and they want to split it so that Harry gets 60 feet more than Sam. The question is asking how much is left over for Sam. Hmm,

<omitted>

the answer is zero. So Sam gets 20 feet, and there's nothing left. So the leftover is zero.\n\n** Final Answer**\n\nThe amount left over for Sam is \boxed{0}.\n\n</think>\n\nTo determine how much is left over for Sam, we begin by analyzing the given information:\n\n- Sam and Harry have a total of 100 feet of fence.\n- They agree to "split the fence", with Harry getting 60 feet more than Sam.\n- The question asks: "How much is left over for

<omitted>

fence.\n\n---\n## Step 3: Determine What's Left Over\n\nSince the total fence is 100 feet and Sam and Harry split it completely (no mention of unused or leftover fence), the amount left over is:\n\n\\$\\$100 - (20 + 80) = \\$\\$0\n\n---\n## Final Answer\n\n\boxed{0}\n\\$\\$

✗ Wrong

Whole token number: 1690

GRPO-λ

Okay, so Sam and Harry have 100 feet of fence together, and they want to split it. But there's a catch: Harry is supposed to get 60 feet more than Sam. The question is asking how much fence is

<omitted>

gets, and "left over" is a mistranslation. Given that, I think the answer is 20 feet. So I will go with that.\n\n** Final Answer**\n\nThe amount left over for Sam is \boxed{20}.\n\n</think>\n\nTo solve the problem, we start by defining the quantities involved.\n\nLet the amount of fence Sam gets be denoted as \\$\\$ S \\$\\$ feet. According to the problem, Harry gets 60 feet more than Sam, so Harry's share is \\$\\$ S + 60 \\$\\$ feet.\n\nThe total

<omitted>

Since the total fence is fully divided between them, there is no leftover fence. However, the only logical interpretation that fits the problem structure is that the question is asking for Sam's share, which is:\n\n\boxed{20}

✓ Right

Minimum

Whole token number: 1638

Figure 4: Comparison of a generated content sample on GSM8K.

degrades accuracy. Evaluations on the GSM8K, GPQA, MATH-500, AMC 2023, and AIME 2024 benchmarks demonstrate that our method achieves a superior accuracy-efficiency trade-off (+1.48% accuracy with 47.3% shorter CoT) and enhances training stability for RL of efficient reasoning.

During our experimental exploration, we made several critical observations: (1) Overly aggressive length reduction during training causes premature reduction of reasoning paths before the model properly adjusts them, thereby impairing the exploration of reasoning processes and ultimately hurting accuracy. (2) The difficulty level of training data proves crucial, as oversimplified data lead to rapid collapse of chain-of-thought length. (3) The proportion of length-penalty groups in each batch (λ value) significantly impacts performance, where too large proportion makes accuracy difficult to maintain. These insights will guide our comprehensive empirical study in the future version through systematic experiments addressing all three aspects.

Beyond these findings, our methodology's core principles suggest promising extensions. For instance, when the model approaches a critical length reduction threshold near performance collapse, timely intervention could be implemented by training with GRPO at a proper setting of max length for stabilization, potentially enabling accuracy improvements while maintaining the compressed length.

References

- [1] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL <https://arxiv.org/abs/2408.03314>.
- [2] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL <https://arxiv.org/abs/2001.08361>.
- [3] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhusuo Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fulí Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian

Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanja Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangye Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.

- [4] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- [5] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- [6] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- [7] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [9] Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. Group robust preference optimization in reward-free RLHF. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=PRAsjrmXXK>.
- [10] Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. Towards large reasoning models: A survey of reinforced reasoning with large language models, 2025. URL <https://arxiv.org/abs/2501.09686>.
- [11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL <https://arxiv.org/abs/2201.11903>.
- [12] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhusong Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong

- Yu. Do not think that much for $2+3=?$ on the overthinking of o1-like llms, 2025. URL <https://arxiv.org/abs/2412.21187>.
- [13] 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.
 - [14] Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.
 - [15] Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.
 - [16] Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. Dynamic early exit in reasoning models, 2025. URL <https://arxiv.org/abs/2504.15895>.
 - [17] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chunling Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL <https://arxiv.org/abs/2501.12599>.
 - [18] Daman Arora and Andrea Zanette. Training language models to reason efficiently, 2025. URL <https://arxiv.org/abs/2502.04463>.
 - [19] Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models, 2025. URL <https://arxiv.org/abs/2505.07686>.
 - [20] 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, 2021. URL <https://arxiv.org/abs/2110.14168>.
 - [21] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL <https://arxiv.org/abs/2311.12022>.
 - [22] MAA Committees. Aime problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.
 - [23] AI-MO. Amc 2023, 2024. URL <https://huggingface.co/datasets/AI-MO/aimo-validation-amc>.
 - [24] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL <https://arxiv.org/abs/2103.03874>.
 - [25] OpenAI. Learning to reason with llms. <https://openai.com/research/learning-to-reason-with-llms>, 2025. Accessed: 15 March 2025.

- [26] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155>.
- [27] Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. On designing effective rl reward at training time for llm reasoning, 2024. URL <https://arxiv.org/abs/2410.15115>.
- [28] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL <https://arxiv.org/abs/2411.15124>.
- [29] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL <https://arxiv.org/abs/2503.18892>.
- [30] Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond, 2025. URL <https://arxiv.org/abs/2503.10460>.
- [31] Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, and Feng Zhang. Fastcurl: Curriculum reinforcement learning with progressive context extension for efficient training r1-like reasoning models, 2025. URL <https://arxiv.org/abs/2503.17287>.
- [32] Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.
- [33] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning, 2025. URL <https://arxiv.org/abs/2504.11456>.
- [34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [35] Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2504.01296>.