FastCuRL: Improving RL Training Efficiency of R1-like Reasoning Models via Curriculum-Guided Iterative Lengthening

Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, Feng Zhang
Tencent Hunyuan
nickmysong@tencent.com

GitHub: https://github.com/nick7nlp/FastCuRL
HuggingFace: https://huggingface.co/Nickyang/FastCuRL-1.5B-Preview

Abstract

Recently, training Large Language Models (LLM) via large-scale Reinforcement Learning (RL) has emerged as an increasingly promising paradigm for addressing complex reasoning tasks.

In this work, we propose a *simple yet efficient* **Cu**rriculum-guided ite**R**ative **L**engthening reinforcement learning approach (**FastCuRL**) to accelerate the RL training process for R1-like models while improving their performance in long CoT reasoning scenarios, particularly on smaller language models (e.g., a 1.5B parameter language model).

Experimental results demonstrate that **FastCuRL-1.5B-Preview** surpasses DeepScaleR-1.5B-Preview on all five datasets (MATH 500, AIME 2024, AMC 2023, Minerva Math, and OlympiadBench) while reducing computational resource consumption by more than two times. In addition, all training stages for **FastCuRL-1.5B-Preview** are completed on 8 GPUs.

This report mainly shares insights from our experiments, analyses, and key observations gathered while reproducing R1-like reasoning models.

Model	Training Steps	Training Stages	Number of GPUs Used in Each Stage
DeepScaleR-1.5B-Preview	$\sim 1,750$	3	8, 16, 32
FastCuRL-1.5B-Preview	~ 860	4	8, 8, 8, 8

Table 1: Training Details. Here, we uniformly set the batch size to 128 for counting training steps, meaning two steps with batch size 64 are counted as one with batch size 128.

Model	MATH 500	AIME 2024	AMC 2023	Minerva Math	OlympiadBench	**Avg.**
Qwen-2.5-Math-7B-Instruct (Yang et al., 2024)	79.8	13.3	50.6	34.6	40.7	43.8
rStar-Math-7B (Guan et al., 2025)	78.4	26.7	47.5	-	47.1	-
Eurus-2-7B-PRIME (Cui et al., 2025)	79.2	26.7	57.8	38.6	42.1	48.9
Qwen2.5-7B-SimpleRL (Zeng et al., 2025)	82.4	26.7	62.5	**39.7**	43.3	50.9
DeepSeek-R1-Distill-Qwen-1.5B	82.8	28.8	62.9	26.5	43.3	48.9
Still-1.5B (STILL-Team, 2025)	84.4	32.5	66.7	29.0	45.4	51.6
DeepScaleR-1.5B-Preview (Luo et al., 2025)	87.8	43.1	73.6	30.2	50.0	57.0
DeepScaleR-1.5B-Preview†	87.3	42.1	73.7	30.9	50.0	56.8
FastCuRL-1.5B-Preview	**88.0**	**43.1**	**74.2**	31.6	**50.4**	**57.5**

Table 2: *Pass*@1 accuracy is reported, averaged over 16 samples for each problem. † indicates the results obtained by re-evaluating using checkpoints provided by this method.

1 Introduction

Training large language models through large-scale reinforcement learning has emerged as a promising paradigm for mastering complex reasoning tasks. Recent breakthroughs, DeepSeek-R1 (DeepSeek-AI, 2025), have demonstrated a remarkable training time scaling phenomenon: as the training computation scales up, both benchmark performance and response length of the trained model consistently and steadily increase without any sign of saturation. Inspired by these achievements, many interesting and valuable works have emerged that attempt to explore and reproduce R1-like reasoning models (e.g., from a distilled model or a pre-trained model), such as DeepScaleR (Luo et al., 2025), Open R1 (Hugging-Face, 2025), Open-Reasoner-Zero Hu et al. (2025), etc.

In large-scale reinforcement learning, the high computational cost is one of the biggest challenges. Luo et al. (2025) indicates that directly replicating DeepSeek-R1's experiments (with context lengths of at least 32K tokens and roughly 8,000 training steps) requires a minimum of 70,000 A100 GPU hours, even for a relatively small 1.5B parameter language model. To mitigate this, DeepScaleR leverages a distilled model (Deepseek-R1-Distilled-Qwen-1.5B[1]) and proposes a novel iterative lengthening scheme for reinforcement learning, reducing the computational requirement to only 3,800 A100 GPU hours, while surpassing the performance of OpenAI's o1-preview using just a 1.5B parameter language model. Specifically, DeepScaleR has three stages for RL training:

- ***Stage-I (8K context)***: Analyzing the initial model's responses, DeepScaleR discovers that lengthy responses often exhibit repetitive patterns that fail to contribute meaningfully to effective long CoT reasoning. Given this insight, DeepScaleR initiated training with an 8K context to lead the trained model to utilize context more effectively and improve the quality of long CoT rationales.
- ***Stage-II (16K context)***: After approximately 1,000 training steps, the model's response length increases again, and then DeepScaleR extends the training context from 8K to 16K to adapt this transition.
- ***Stage-III (24K context)***: After training an additional 500 steps with a 16K context length, the model's performance began to plateau. Therefore, DeepScaleR further extends the output context length to 24K to improve the model's long CoT reasoning capabilities.

In summary, DeepScaleR provides valuable insights into leveraging large-scale RL through an iterative lengthening scheme to train models, enhancing their ability to master complex reasoning tasks. Furthermore, through observing, reproducing, and analyzing DeepScaleR's experiments, we derive ***two key findings*** as follows:

(1) Progressively relaxing π_{ref} regularization: Training in multiple stages versus training only in one stage is not just a difference in parameters (such as the context length); the reference policy π_{ref} is also changing in the GRPO objective (Shao et al., 2024), as illustrated in Eq. (1). In comparison, the multi-stage training strategy gradually relaxes the KL regularization imposed by the reference policy on the model, reducing the penalty from the KL divergence between the trained policy π_θ and the reference policy π_{ref} to the loss, thereby allowing the trained policy to explore more solutions. Therefore, the timing of switching stages needs to be carefully considered.

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)] \\ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t},\right.\right. \\ \left.\left.\text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\}, \quad (1)\end{aligned}$$

[1] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B

(2) Potential impact of 8K context constraint: First, setting an 8k context window helps improve model response quality and accelerate the model's training process. Here, we only discuss the potential negative impacts of the context constraint. As shown in Figure 1, over 40% of samples (may include both lengthy incorrect and correct long CoT reasoning) are truncated at the beginning of training. A critical bottleneck in effectively scaling reinforcement learning for complex reasoning tasks lies in determining the optimal context window size during training. Reasoning-centric tasks inherently demand substantial computational resources, as they typically produce far lengthier outputs compared to standard tasks, thus considerably slowing down trajectory sampling and policy gradient updates. Indeed, increasing the context window length by merely twofold leads directly to at least a twofold computational overhead (Luo et al., 2025). This dynamic introduces a fundamental trade-off: longer contexts grant models expanded cognitive capacity, facilitating deeper reasoning at the expense of substantially longer training cycles, whereas shorter contexts enable more efficient training yet potentially restrict the model's capacity to address complex tasks that necessitate extensive reasoning spans successfully. Consequently, achieving an optimal balance between training efficiency and problem-solving accuracy remains essential.

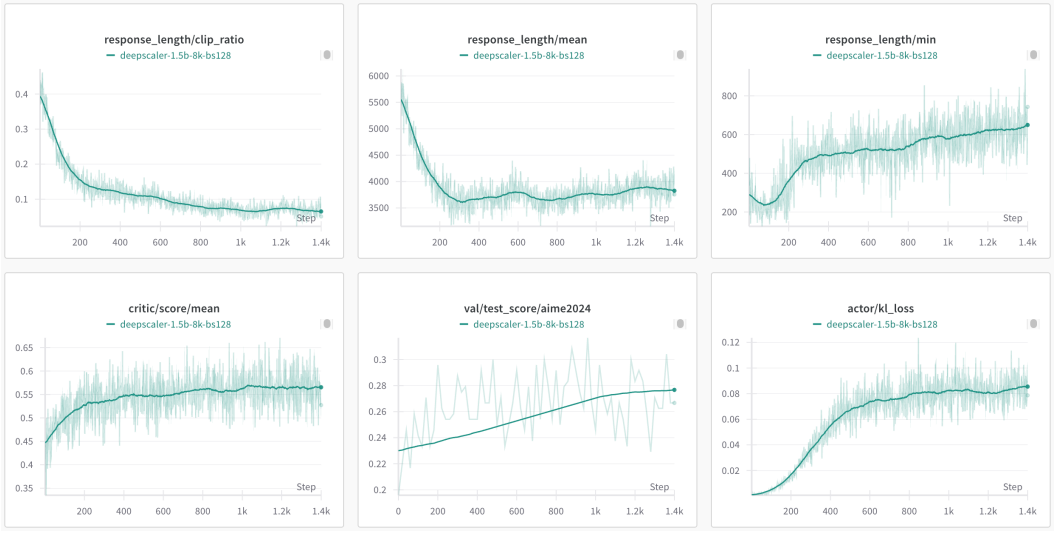

Figure 1: ***Stage-I (8K context)*** training records of DeepScaleR (reproduced by us).

Meanwhile, by observing the long CoT reasoning responses generated by R1-like models when solving complex reasoning questions, we observe that there could be two kind of questions for the model's excessively lengthy responses. Specifically, one is the question's difficulty, which requires continuous thinking to find a solution. The other situation is that the question's contains numerous conditions. Such conditions cause the model to verify each constraint repeatedly during the task solution (may cause repetitive thinking patterns), thus ultimately resulting in excessively long reasoning CoT responses. Both situations mentioned above may slow down the model's training efficiency during the 8K context training stage in iterative lengthening training scheme. Therefore, if the training data can be segmented according to specific features (e.g., response length), this segmentation approach may help reduce truncation caused by context limitations, thereby accelerating training.

Inspired by these findings and observations, a question arises:

Can simple data segmentation and multi-stage training further accelerate the RL training?

In this report, we first propose a condition-sensitive training data segmentation strategy that separates the original dataset into different levels. Next, we introduce a curriculum-guided iterative lengthening approach designed to accelerate the RL training process of R1-like reasoning models, helping them efficiently master complex reasoning tasks.

2 FastCuRL's Recipe

> Simplicity is the ultimate sophistication.
>
> —— Leonardo da Vinci

This report shares several interesting phenomena discovered by reproducing the R1-like reasoning models. To better observe and analyze these findings, except for training strategies, the original training dataset, reward designs, and other experimental settings remain the same as DeepScaleR's. For detailed information, please refer to Luo et al. (2025). Next, we detail the proposed condition-sensitive training data segmentation and curriculum-guided iterative lengthening training approaches.

2.1 Condition-Sensitive Training Data Segmentation

In this report, data segmentation mainly aims to distinguish between data samples that may require long CoT reasoning rationales and data samples with short CoT reasoning rationales, minimizing the impact caused by truncation in training with a shorter context window. An intuitive idea is to infer all the training data once by the reference model and then split the training data based on the length of the responses. However, this approach indirectly introduces restrictions from the reference policy into the training policy, affecting it. At the same time, we expect to retain uncertainty during data segmentation, that is, to mix long CoT reasoning and short CoT reasoning data in a particular proportion. This means the dataset labeled as short CoT reasoning data may contain a small portion of long CoT reasoning data, and the dataset labeled as long CoT reasoning data may include some short CoT reasoning data.

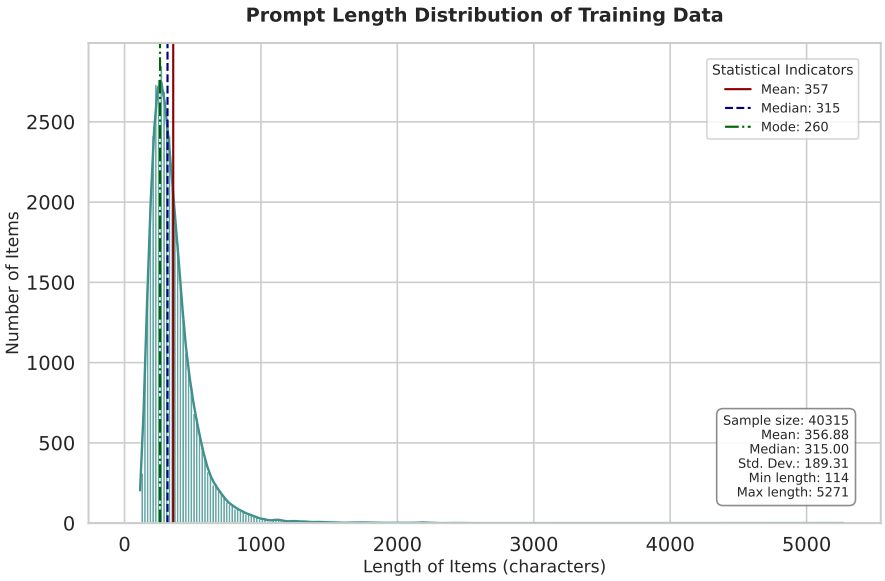

Figure 2: Prompt Length Distribution.

Moreover, a straightforward way is to utilize perplexity, loss functions, or reward models to segment the training samples. However, the experiments within this report aim to distinguish training data likely to result in either long or short CoT reasoning responses, separating these two types for use in multi-stage reinforcement learning to accelerate training speed. Therefore, we intend to split the training data using simpler and more robust rules. In earlier sections, we mentioned that R1-like reasoning models tend to trigger substantial reflection and validation processes when encountering problems with many constraints. Inspired by this observation, we hypothesize that longer inputs often correlate with longer outputs. Based on this assumption, we first analyze the DeepScaleR training

dataset statistically, as shown in Figure 2. Then, we split the original training dataset (named O-CoT) into two parts according to the mean input length, resulting in one subset representing a short CoT reasoning dataset (named S-CoT) and another subset forming a long CoT reasoning dataset (named L-CoT). To validate our hypothesis, we train models on O-CoT, S-CoT, and L-CoT under an 8K context window as seen from Figure 3, whether the results meet expectations in clipping ratio, response length, and reward scores. Therefore, we split the training data based on the average input prompt length.

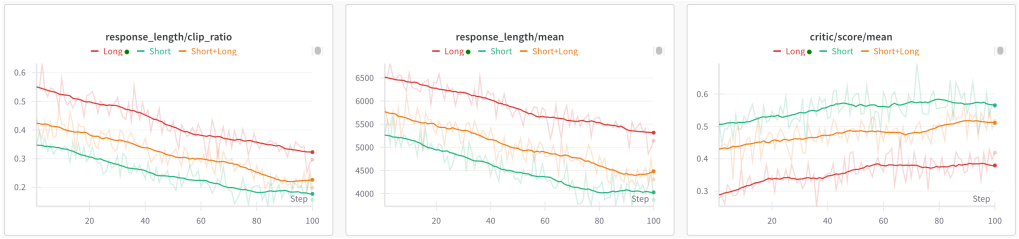

Figure 3: Training records on three different datasets.

2.2 Curriculum-Guided Iterative Lengthening Training

After obtaining the new segmented datasets (O-CoT, S-CoT, and L-CoT), we introduce how to accelerate model training by incorporating curriculum reinforcement learning and iterative lengthening. Continuing to follow the principle of "simplicity above all," we let the model learn step by step: first, the simple things (S-CoT), then the mixed ones (O-CoT), then only the difficult ones (L-CoT), and finally, reviewing everything as a whole once more (O-CoT). Among them, each stage trains at most one iteration. After completing the four stages, this is equivalent to having trained for three iterations on the original training data. Specifically, the four progressive stages are as follows:

♢ ***Stage-I (8K context, ∼160 steps)***: As observed by Luo et al. (2025), longer responses do not necessarily imply more accurate results. Therefore, immediately training with long context windows might be inefficient, as many tokens could end up being effectively wasted. To improve training efficiency, during the first stage, we utilize the S-CoT dataset and set an 8K context window to optimize the model for more concise reasoning rationales.

Figure 4: Training records of ***Stage-I (8K context, ∼160 steps)***.

♠ ***Stage-II (16K context, ∼590 steps)***: Figure 4 shows that during Stage-I at approximately step 160, the clipping ratio of the model responses drops to around 10%. Subsequently, we set the context window to 16K and continue training the model using the O-CoT dataset. As shown in Figure 5, the model's response length and reward have stabilized and are gradually improving. Surprisingly, at step 590, the model's *Pass*@1 accuracy on AIME 2024 exceeded 0.4. Thus, one iteration of training for this stage has been completed.

Figure 5: Training records of ***Stage-II (16K context, ∼590 steps)***.

♡ ***Stage-III (24K context, ∼230 steps)***: As previously described, the third stage continues training using the L-CoT dataset and exhibits phenomena in the training process similar to those in the second stage (see Figure 6).

Figure 6: Training records of ***Stage-III (24K contex, ∼230 stepst)***.

♣ ***Stage-IV (24K context, ∼580 steps)***: In the fourth stage, close to completing one iteration, the model achieved its best performance on AIME 2024 throughout the entire training process, as shown in Figure 7. With this, the training across all four stages was completed.

Overall, we find that during the whole training process, the steps chosen for stage transitions mainly occurred toward the end of each stage, further highlighting the efficiency of the proposed FastCuRL approach. Furthermore, we are exploring various ways to combine the

Figure 7: Training records of ***Stage-IV (24K context, ∼580steps)***.

S-CoT, O-CoT, and L-CoT datasets. There might be even faster combination methods, but we are currently in the exploratory phase.

3 Conclusions

In this report, we propose FastCuRL, a simple yet efficient curriculum-guided iterative lengthening reinforcement learning framework designed to significantly accelerate and improve the training of R1-like models, especially small language models around 1.5B parameters, in long-chain-of-thought reasoning tasks. Experimental results demonstrate that **FastCuRL-1.5B-Preview** achieves superior performance over DeepScaleR-1.5B-Preview across all five tested benchmarks (MATH 500, AIME 2024, AMC 2023, Minerva Math, and OlympiadBench) while simultaneously reducing computational resource usage by more than twofold, with all training phases efficiently completed using only 8 GPUs.

For future work, dynamic context window length control and dynamic KL regularization are more interesting and valuable research topics.

References

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025. URL https://arxiv.org/abs/2502.01456.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking, 2025. URL https://arxiv.org/abs/2501.04519.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, and Heung-Yeung Shum Xiangyu Zhang. Open-reasoner-zero: An open source approach to scaling reinforcement learning on the base model. https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero, 2025.

Hugging-Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2, 2025. Notion Blog.

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.

RUCAIBox STILL-Team. Still-3-1.5b-preview: Enhancing slow thinking abilities of small models through reinforcement learning. 2025. URL https://github.com/RUCAIBox/Slow_Thinking_with_LLMs.

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. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122, 2024. doi: 10.48550/ARXIV.2409.12122. URL https://doi.org/10.48550/arXiv.2409.12122.

Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. https://hkust-nlp.notion.site/simplerl-reason, 2025. Notion Blog.