Towards A Better Initial Policy Model For Scalable Long-CoT Reinforcement Learning

Bofei Gao[1], Yejie Wang[2], Yibo Miao[3], Ruoyu Wu[1], Feifan Song[1], Longhui Yu[1], Tianyu Liu[1] †, Baobao Chang[1]† ‡

[1] National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University

[2] Beijing University of Posts and Telecommunications [3] Shanghai Jiao Tong University

gaobofei@stu.pku.edu.cn, chbb@pku.edu.cn

Abstract

Long-CoT reasoning combined with reinforcement learning for large language models demonstrates remarkable performance and scalability. However, we observe that the initial policy model could significantly influence the final performance as well as the token efficiency. Additionally, there is a lack of systematic guidelines for obtaining a better initial policy model. To bridge this gap, we initiate a comprehensive investigation by activating the initial model using a variety of datasets with different data volumes and reasoning patterns. Then, we conduct a thorough analysis and comparison of the RL process for different initial models from the perspectives of upper bounds, diversity, and token efficiency, providing a deeper understanding and insight into the long-CoT RL. Based on our empirical results, we propose a systematic guideline and a novel Re-RFT method for constructing a better RL start point. Our experiment results based on the 14B model surpass the DeepSeek-R1-Distill-Qwen-14B by an average of 4.6%, demonstrating our approach's effectiveness and superiority.

1 Introduction

With the rapid advancement of artificial intelligence, the complex reasoning abilities of large language models (DeepSeek-AI et al., 2025; Team et al., 2025; Team, 2024; Yang et al., 2024), such as mathematical reasoning (Yeo et al., 2025; Shen et al., 2025) and code generation (OpenAI et al., 2025), have garnered significant attention. To enhance these capabilities, a powerful technique, *test-time scaling* has been developed, where the model using more tokens during inference has a significant improvement in performance.

There are currently two lines of research to implement test-time scaling. One method, based on the Monte Carlo Tree Search (Zhang et al., 2024; Xie et al., 2024; Luo et al., 2024), guides the model to generate more accurate answers through extensive repeated sampling and the outcome feedback. Another line of research adopts reinforcement learning, and achieves test-time scaling by reinforcing longer model generation and more detailed chain of thought (CoT) (Team et al., 2025; DeepSeek-AI et al., 2025). In this pipeline, the model is first activated with high-quality long-CoT data and then reinforced through simple and effective outcome reward and policy optimization algorithms, enhancing both the model's output length and final performance. This method has proven to be highly efficient, enabling the model to achieve superior performance.

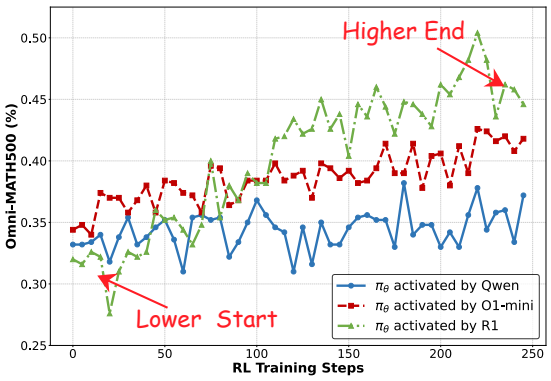


Figure 1: Performance of different initial policy models in reinforcement learning. We use the same problem set but different generated responses from Qwen-2.5-MATH (Qwen et al., 2025), o1-mini (OpenAI, 2024), and R1 (DeepSeek-AI et al., 2025) to cold-start the initial model. All the activation data is correct ensured by an answer rejection sampling.

Despite its effectiveness, we discover that the initial policy model plays a crucial role in the reinforcement learning process. As shown in Figure 1, we activate the Qwen-14B-Instruct (Qwen et al., 2025) model with the same prompt set but different reasoning-pattern CoT generated from

† Project Lead.

‡ Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 7652–7665

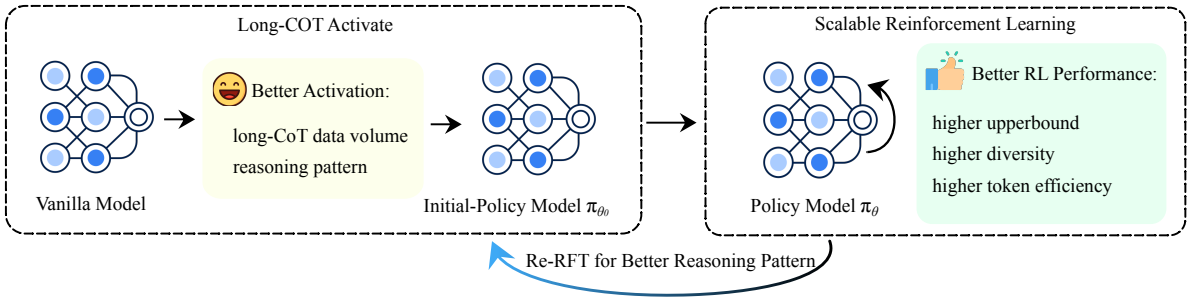


Figure 2: Our work primarily focus on what constitutes a good initial policy model and how to obtain a better initial policy model. We conduct a comprehensive study by isolating the reasoning patterns and the amount of activation data as key variables. Through an in-depth analysis of the RL process from the perspectives of model upper bound, diversity, and token efficiency, we provide a detailed recipe and a novel ReRFT approach for better policy init.

Qwen-2.5-MATH, o1-mini derived from prompt engineering, and R1, resulting in vastly different RL performances. What's more, we find that the performance of the start point does not accurately reflect the eventual gains it will achieve through RL. This raises an essential question: *What is a better initial policy model for scalable RL?*

To answer this question, we investigate the factors influencing the reinforcement learning process from perspectives beyond traditional accuracy, focusing specifically on the model's potential, diversity, and token efficiency. Our analysis reveals that the initial model's pass@k metric provides a more accurate reflection of the final RL performance than accuracy alone. Furthermore, the gap between pass@k and accuracy, along with model diversity, serves as a reliable indicator of training token efficiency during the RL process. Additionally, we find that the choice of the initial model can also significantly impact the length token efficiency throughout the RL training process.

Building on these findings, we further explore *how to obtain a better initial policy model* by considering both data volume and reasoning patterns. In terms of data volume, we find that if the target is higher performance, more high-quality activation data is beneficial. However, as the data volume increases, the training token efficiency tends to decrease, but the length token efficiency LTE shows a slight increase. On the other hand, with smaller data volumes, RL tends to result in rapid growth in both length and performance. Regarding reasoning patterns, we find that the R1 reasoning pattern outperforms the traditional CoT pattern and the o1-mini pattern derived from prompt engineering in terms of a higher pass@k, diversity, and training token efficiency. Additionally, we propose a novel ReRFT method, which can identify more efficient patterns through the policy model itself, thus enabling more effective and scalable RL.

Our experimental results based on the 14B model surpass the DeepSeek-R1-Distill-Qwen-14B by an average of 4.6% across four challenging olympiad-level math benchmarks, demonstrating the effectiveness and superiority of our approach.

2 Preliminary

2.1 Long-Cot Based Reasoning

With the introduction of the O1 series models, long-cot combined with reinforcement learning has seen significant development (DeepSeek-AI et al., 2025; Team et al., 2025; Yeo et al., 2025; Shen et al., 2025). The common practice of long-cot-based post-training can be summarized as follows:

1) Activating the model with high-quality long-cot data $\mathcal{D}_{\text{train}_s} = \{(x_i, y_i)\}_{i=1}^{N}$ using supervised fine-tuning as Equation 1 and then serving it as a starting point π_{θ_0} for reinforcement learning.

$$\mathcal{L}_{\text{SFT}} = E_{[(x,y)\sim\mathcal{D}_{\text{train}_s}]}\left(\sum_{t=1}^{|y|}\log \pi_\theta(y_t \mid y_{<t}, x)\right) \tag{1}$$

2) Optimizing the policy model π_θ with RL algorithms. Formally, given a training dataset $\mathcal{D}_{\text{train}_r} = \{(x_i, a_i)\}_{i=1}^{n}$, where x_i denotes a problem and a_i denotes its corresponding ground truth answer, our goal is to optimize a policy model π_θ. Given a problem x_j of the test set $\mathcal{D}_{\text{test}} = \{(x_j, a_j)\}_{j=1}^{m}$, the policy model can correctly generate $\hat{y}_j = (\hat{z}_j, \hat{a}_j)$ where $\hat{z}_j$ denotes the long and detailed chain-of-thought with specific reasoning pattern and $\hat{a}_j$ is consistent with the gold answer a_j.

2.2 Policy Optimization

There are currently many strategies and variations for policy model optimization including PPO (Schulman et al., 2017), Reinforce++ (Hu, 2025), Policy Mirror Descent (Team et al., 2025), and GRPO (DeepSeek-AI et al., 2025). For simplicity, we directly follow Kimi k1.5 (Team et al., 2025) for our RL policy optimization.

Specifically, for the i-th iteration, given the problem x with the answer a, we let the policy model π_θ explore k times and the gradient of the policy model can be computed as follows:

$$\frac{1}{k}\sum_{i=1}^{k}(\nabla_\theta \log \pi_\theta(y_i|x)\cdot\mathcal{R}(x,y_i,a)-\frac{\tau}{2}\nabla_\theta Z(x,y_i)), \tag{2}$$

where τ is the hyper-parameter controlling the regularization and $Z(x,y_i)$ measures the prediction difference between policy model π_θ and the reference model as Equation 3. The reference model π_{θ_i} can be obtained from the last optimization step of the previous iteration.

$$Z_{\pi_\theta,\pi_{\theta_i}}(x,y_i)=\left(\log\frac{\pi_\theta(y_i|x)}{\pi_{\theta_i}(y_i|x)}\right)^2 \tag{3}$$

Reward Given that the result of the olympiad-level mathematical problem can be complex and difficult to process with a rule-based evaluation (Gao et al., 2024; Team et al., 2025), we adopt the same outcome reward as Team et al. (2025) with a chain-of-thought RM trained from Qwen-2.5-72B-Instruct (Qwen et al., 2025). Chain-of-thought RM can explicitly generate a step-by-step reasoning process before providing a final judgment. Given a question of x with answer a as well as the $\hat{y}$ generated by the policy model, the final reward can be computed as follows:

$$\mathcal{R}(x,\hat{y},a)=(r(x,\hat{y},a)-\overline{r}), \tag{4}$$

where $r(x,\hat{y},a)$ is the 1 when the final judgment of the RM is "True" and otherwise 0 when the judgment is "False" and $\overline{r}$ is the average reward across the k responses per question explored by the policy model.

3 Delving into scalable RL

In this section, we conduct a comprehensive and in-depth analysis of the influence of the initial policy model activated with different datasets during the reinforcement learning process. We argue that beyond the final accuracy, it is essential to consider additional aspects such as *Pass@k* (Section 3.1), *Diversity* (Section 3.2), and *Token Efficiency* (Section 3.3). These metrics can provide more valuable insights of the long-context reinforcement learning process.

Experimental Setup Our objective is to investigate the impact of varying quantities and patterns of data on the model's reinforcement learning process, thus we avoid using excessively large datasets, as they could hinder the ease of our exploration. To enhance the model's reasoning capabilities, we have compiled a activation dataset and RL prompt set comprising level-4 and level-5 problems from MATH (Hendrycks et al., 2021), AIME 1983-2023, and a portion of Omni-MATH (Gao et al., 2024).

For SFT activation, we conduct our study using three different initial reasoning patterns. The first is the traditional CoT generated by Qwen-2.5-MATH (Yang et al., 2024). Additionally, we introduce two long-CoT variants derived from O1-mini and R1. Since O1 does not publicly provide its hidden CoT, we extract its reasoning pattern using prompt engineering, inspired by the Think Claude [*].

For the vanilla model, we have selected Qwen-14b-instruct (Qwen et al., 2025) as well as DeepSeek-R1-distill-Qwen-14B (DeepSeek-AI et al., 2025) activated with 800K R1-generated data for further validation.

In constructing the test set, we have chosen four complex mathematical reasoning benchmarks: MATH-500 (Hendrycks et al., 2021), Omni-MATH-500 (Gao et al., 2024), AIME 2024, and 10 problems from AIMO 2024 (AIMO, 2024). Given the limited data available for AIME 2024 and AIMO 2024, we let the model repeat sampling 10 times and take the average result to ensure a robust evaluation. We have performed string-level matching detection to prevent the data leakage. The detailed data statistics are shown in Table 1.

3.1 Pass@K

The pass@k represents the potential for model improvement, as a higher pass@k indicates that the model is capable of generating higher-quality data during training, thereby facilitating further progress. For a given question x with answer a,

[*] https://github.com/richards199999/Thinking-Claude

Usage	Dataset Name	Num.
SFT	MATH (Hendrycks et al., 2021)	9631
	AIME 1983–2023	918
RL	MATH (Hendrycks et al., 2021)	3988
	Omni-MATH (Gao et al., 2024)	2926
Test	AIME 2024	30×8
	MATH-500 (Hendrycks et al., 2021)	500
	Omni-MATH-500 (Gao et al., 2024)	500
	AIMO 2024 (AIMO, 2024)	10×8

Table 1: Statistics of datasets of SFT, RL and Test. The RL dataset only contains the questions and corresponding final answer without CoT.

π_{θ_0}	Acc	pass@K	Δ.	Acc^{test} π_{θ_t}
w/ Qwen	66.0	73.1	8.1	35.1
w/ o1-mini	**66.7**	75.9	9.2	40.6
w/ R1	62.9	**79.9**	**17.0**	**46.2**

Table 2: The RL training accuracy and training pass@K of the π_{θ_0} activated with different reasoning pattern as well as the accuracy of the RL-trained model π_{θ_t} on Omni-MATH-500 test set.

pass@k measures the probability of the model generation $\hat{Y} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_k)$ passing at least once in k samples, as shown in below:

$$Pass@k(x, \hat{Y}, a) = 1 - \prod_{i=1}^{k} \mathbb{I}(g(\hat{y}_i) \neq a), \quad (5)$$

where $g(\cdot)$ denotes the answer extraction function.

The pass@k and accuracy of different initial policy models are shown in Table 2. For models initialized with different activation patterns, we observe that the initial pass@k aligns more perfectly with the final RL performance than initial accuracy. Models initialized with long COT tend to exhibit higher pass@k, indicating greater potential, even though their Pass@1 may be lower than that of Qwen-MATH's traditional COT. Additionally, throughout the reinforcement learning process, we find that models with a larger gap between pass@k and accuracy tend to show a steeper performance improvement as the RL steps progress as shown in Figure 1. In other words, the greater the difference between the model's current accuracy and its potential, the larger the gain at each RL training step. In conclusion, an initial policy model with a higher pass@k is considered a better initial policy model.

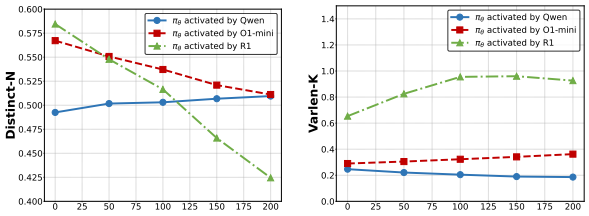
3.2 Diversity

During the reinforcement learning process, particularly in the early stages, another important factor to consider is the diversity of the data rollout by the model (Zintgraf et al., 2021; Norman and Clune, 2024). A greater diversity implies a higher exploration rate of the model to find better solutions, which allows the model to reinforce itself based on these successful explorations. In this section, we proposed two methods for measuring the diversity of the model-generated responses and conducted a detailed analysis of the models activated with the three aforementioned patterns.

Distinct-N Distinct-N (Li et al., 2016) directly measures the diversity of the N-grams in the model's output. For a model generated content s, the distinct-N of s can be computed as follow:

$$\text{Distinct-N}(s) = \frac{1}{N} \sum_{n=1}^{N} \frac{\text{\# unique n-grams}(s)}{\text{\# total n-grams}(s)}. \quad (6)$$

For the same problem set, a higher distinct-N indicates a higher lexical diversity from the model, which proved to be essential for post-training process (Wu et al., 2024b).



(a) Distinct-N of π_{θ_0} with RL training Steps.

(b) Varlen-K of π_{θ_0} with RL training Steps.

Figure 3: Diversity of π_{θ_0} activated by different reasoning pattern during RL training

Figure 3a illustrates the distinct-N values for the three initial-policy models activated by different patterns discussed above. As shown, the data generated by the R1 activation, which is based on long-COT with no prompt restrictions, results in the highest diversity, followed by the fixed-prompt of o1's long-COT, and finally, the model activated by traditional COT. An intriguing finding is that, during the reinforcement learning process, we observe a continuous decline in distinct-N for all models, but the long-COT-based model shows a more significant drop, eventually under-passing the traditional COT. However, despite this decline, the RL performance continues to rise. We believe this is a result of the exploration-exploitation trade-off: At the early stages, the model explores a wide variety of reasoning traces and reinforce itself with the

guidance of the reward. As RL training progresses, the model gradually converges toward more efficient reasoning traces. Consequently, the diversity of its reasoning paths decreases, and its overall performance stabilizes and converges. Therefore, we conclude that lexical diversity in the initial phase is more important, as it implies a larger search space for the model, increasing the likelihood of discovering better reasoning traces, which is also one of the key advantages of long- CoT over traditional CoT.

Varlen-K Besides the lexical diversity, in the context of Long-COT, we believe that generating responses of varying lengths for the same question signifies greater diversity in the reasoning patterns. Specifically, different approaches or thinking patterns generated by the model are reflected in outputs of varying lengths. Therefore, we propose a diversity metric named *Varlen-K*. Varlen-K defines the normalized standard deviation of the lengths of the k responses $\hat{Y} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_k)$ generated by the model π_θ for a given question x:

$$\text{Varlen-K}(\hat{Y}) = \sqrt{\frac{\sum_{i=1}^{k}(\frac{|y_i|}{avg(|\hat{Y}|)} - \mu)^2}{k}}, \quad (7)$$

where μ is the average value of the $\frac{|y_i|}{\text{avg}(|\hat{Y}|)}$.

Figure 3b illustrates the Varlen-K for three different initial policy models. The model activated by R1, which has no additional prompt constraints, exhibits significantly higher length diversity compared to the model activated by o1 with prompt restrictions, as well as the traditional CoT model.

Additionally, we observed that, the length diversity of the π_θ traditional CoT and prompt engineered o1-mini CoT remains consistently low during RL training. The model fails to learn how to utilize a larger budget to solve more difficult tasks, which limits its ability or efficiency to scale up in length and performance.

3.3 Token Efficiency

In the long-CoT reinforcement learning process, token efficiency plays a critical role in determining the upper bounds of RL performance. This is primarily due to the limited context windows in most existing models; for instance, the Qwen (Qwen et al., 2025) series model has a maximum context length of 128K tokens.

Moreover, the improvement achieved within certain training tokens of RL is also crucial. Based on our previous findings in Section 3.2, the model's distribution gradually converges during the RL process, leading to a slower performance gain over time. Therefore, performance improvements within a fixed training tokens indicate that the model can effectively leverage rollout data to enhance itself without being constrained by locally optimal distributions.

Throughout the RL process, we observed that the model's token efficiency exhibited different trends depending on the activation data of different patterns. To provide a clearer definition of token efficiency, we outline three key metrics:

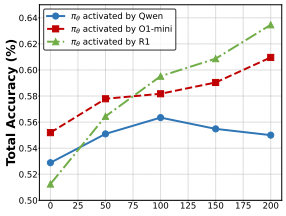
Delta Length Token Efficiency (LTE) LTE measures the relationship between performance improvements and the increase in average token length throughout the RL process. A higher LTE signifies that, within a fixed length budget increase, the model is capable of achieving greater performance gains. LTE is influenced by many factors, such as the efficiency of the RL algorithm and the suitability of the RL prompt set. However, our paper specifically focuses on the impact of different initial policy models on LTE.

$$LTE(\pi_\theta) = \frac{acc(x, \hat{y}_{\pi_\theta}, a) - acc(x, \hat{y}_{\pi_{\theta_0}}, a)}{len(\hat{y}_{\pi_\theta}) - len(\hat{y}_{\pi_{\theta_0}})} \quad (8)$$

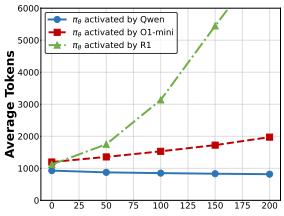
Training Token Efficiency (TTE) Given that online-RL requires the policy model rollout and real-time interactions with the environment, its computational cost is significantly higher than that of supervised fine-tuning. What's more, performance improvements within a fixed number of training tokens indicate that the model can effectively utilize rollout data to enhance itself without being constrained by locally optimal distributions. Therefore, the training token efficiency is an important metric to monitor. A better TTE indicates an efficient RL training.

$$TTE(\pi_\theta) = \frac{acc(x, \hat{y}_{\pi_\theta}, a) - acc(x, \hat{y}_{\pi_{\theta_0}}, a)}{\#\, train\_iters} \quad (9)$$

Figure 4 and Figure 5 illustrate the growth in the length and token efficiency of policy models activated by different patterns during the reinforcement learning process. As shown in Figure 4b, under the same hyperparameters, the initial policy



(a) Overall Acc of π_θ with RL training steps.



(b) Total average tokens of π_θ with RL training steps.

Figure 4: Experimental results of different pattern activated π_θ in the total test set (without averaging AIME and AIMO multiple times).

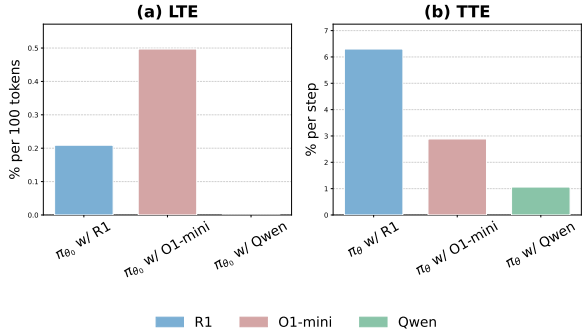


Figure 5: The token efficiency of π_θ activated by different reasoning patterns.

models activated by different patterns exhibit significant differences in length growth during RL. The traditional-CoT pattern does not achieve test-time scaling during RL and even performs a slight length reduction. In contrast, the model activated by the long-cot pattern shows a steady increase in length, although the growth rate varies significantly. When the policy model is activated by o1-mini by a fixed prompt, it performs a gradual increase in length during RL. However, the policy model activated by R1, sees a rapid increase in length, accompanied by a corresponding fast improvement in performance.

Figure 5 shows the token efficiency of different models. The R1-activated model demonstrates the highest training token efficiency among the three models, indicating more efficient RL returns. The o1-mini-activated model, on the other hand, exhibits a higher length token efficiency, suggesting a greater likelihood of solving tasks within the same length and showing a higher upper bound for RL improvement. However, although the traditional-cot model has a slightly higher initial token efficiency than the other two long-cot-activated models, its length does not increase proportionally with RL growth, resulting in a lack of length token efficiency. Additionally, its train token efficiency is significantly lower than that of the long-CoT models.

Beyond these conclusions, **we also observed that the TTE, the gap between pass@K and acc, and the diversity of the initial policy model are positively correlated**. One plausible explanation is that a model with greater diversity is more likely to discover high-quality reasoning traces within its K attempts, thereby yielding a higher pass@k that exceeds its single-attempt accuracy. Furthermore, as the model continuously reinforces its learning of these superior reasoning traces through reward mechanisms, it ultimately achieves a higher TTE.

Therefore, we can conclude that the long-cot activation pattern is crucial for effective RL training.

4 Towards a Better Initial Policy Model

In the previous section, we analyzed the impact of different pattern-activated initial policy models from various perspectives. We also explored what is a good initial policy model for reinforcement learning. Specifically, the one with higher pass@k, greater diversity at the early training stage, and better token efficiency during the RL training. Based on the observations and conclusions, we try to explore the optimal activation strategy by two key aspects: the quantity of activation data and the reasoning pattern for the policy model.

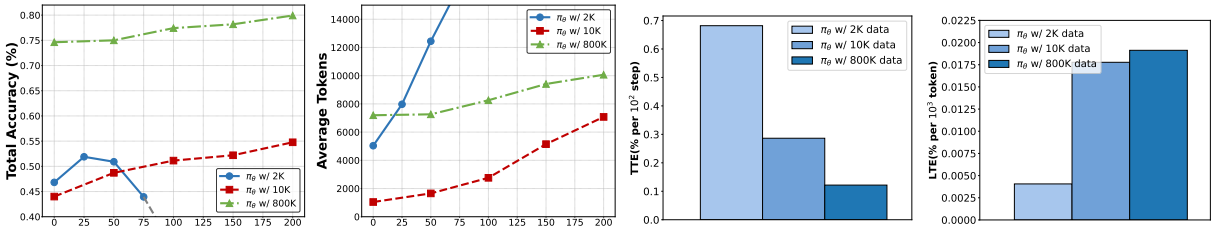
4.1 The Quantity of Activation Data

In RL activation, an important question is how much data should be used to activate a vanilla model. Existing studies have shown that activating a model with a small amount of high-quality data can yield strong reasoning capabilities (Muennighoff et al., 2025; Ye et al., 2025). *However, is a small amount of activation data sufficient to obtain superior reasoning ability?* Based on our experimental findings, our conclusion is that **if the goal is to maximize the final performance, we should use as many high-quality long-CoT data as possible to train the initial policy model. If the goal is to achieve rapid performance improvement in reinforcement learning with limited training data and training steps, a small amount of activated data is more effective**.

To prove this, we sample 2K data from a total of 10K R1-pattern data shown in Table 1 to activate the Qwen-14b-Instruct, in comparison with the total 10K activation data. We also include an extreme situation: DeepSeek-distill-Qwen-14B trained with

Model	AIME2024	AIMO2024	MATH-500	OmniMATH-500	Total
OpenAI o1-12-17 (OpenAI, 2024)	79.2	-	96.4	-	-
Kimi-k1.5 (Team et al., 2025)	77.5	-	96.2	-	-
DeepSeek-R1 (DeepSeek-AI et al., 2025)	**79.8**	-	**97.3**	-	-
QwQ-32B-Preview (Team, 2024)	50.0	36.2	90.6	46.2	63.1
Satori-Qwen-7B (Shen et al., 2025)	23.3	-	83.6	-	-
s1-32B (Muennighoff et al., 2025)	56.7	-	93.0	-	-
DeepSeek-R1-Distill-Qwen-14B (DeepSeek-AI et al., 2025)	68.8	**48.8**	94.9	**65.2**	**76.1**
Qwen-14B-Custom-Activated w/ 2K♠	27.5	13.8	85.4	36.2	51.9
Qwen-14B-Custom-Activated w/ 10K♠	32.5	**22.5**	87.6	42.8	55.5
Qwen-14B-Custom-Activated w/ 10K♠ + ReRFT♠	**33.8**	18.8	**89.2**	**49.0**	**59.6**
DeepSeek-R1-Distill-Qwen-14B♠	75.4	60.0	95.4	69.2	79.9
DeepSeek-R1-Distill-Qwen-14B♠ + ReRFT♠	**75.8**	**61.3**	**97.6**	**69.2**	**80.7**

Table 3: Main results of the models across all mathematical reasoning benchmarks. Models marked with ♠ indicate the performance after reinforcement learning. For example, DeepSeek-R1-Distill-Qwen-14B♠ means we apply reinforcement learning using DeepSeek-R1-Distill-Qwen-14B as initial policy model.



(a) Total acc of π_θ activated with different data volume. (b) Length of π_θ activated with different data volume. (c) The TTE of π_θ activated with different data volume. (d) The LTE of π_θ activated with different data volume.

Figure 6: The performance of π_θ activated with different data volume on total test set.

800K R1-generated data. The experimental results are shown in Figure 6a and the token efficiency is shown in Figure 6c and Figure 6d.

As a result, during the early stages of RL training, the model initialized with minimal data exhibits a higher initial length and accuracy as well as a rapid increase in both length and performance. This could be attributed to the fact that the minimal data allowed the model to learn the long-CoT format without introducing excessive human inductive bias, resulting in the highest training token efficiency. However, we observed that this rapid performance and length increase is accompanied by a lower LTE. The model is soon constrained by the 64K sequence length window, limiting further performance improvements. The higher ratio of truncated rollout data even made the model collapse, as seen in 6a.

In contrast, activating the model with full SFT data, resulted in slower length growth and a higher LTE and, consequently, better performance within the same 64K context window. This trend can also be observed in the 800K dataset activation of the DeepSeek-distill-Qwen-14B model in contrast with 2K and 10K activation data, which achieved the highest LTE and the lowest TTE. However, due to the large volume of high-quality data introduced in the supervised fine-tuning stage, the excessive prior knowledge could anchor the model's reasoning from more exploration, thus result in lower TTE. Even though, the model achieved the highest initial total accuracy and pass@k through its superior capability, ultimately resulting in the best overall performance, as shown in Figure 6a.

In summary, to achieve superior final performance, it is essential to collect as much high-quality long-CoT data as possible during the SFT phase. However, as the volume of long-CoT data increases, the training token efficiency tends to decrease. In resource-constrained scenarios, for example, lack of high-quality supervised fine-tuning data, long-CoT RL offers a highly efficient approach to enhancing model performance within fewer training steps.

4.2 ReRFT for Better Reasoning Pattern

From Section 3, it is evident that an effective reasoning pattern has a significant impact on reinforcement learning. Especially when the activation data is heavy, our objective is to identify a more efficient

reasoning pattern that enables the model to achieve a higher TTE, thereby rapidly enhancing the final model performance.

The insights from Section 3 suggest that a superior pattern might be uncovered directly from the policy model itself. During the RL process, we observed that while the diversity of outputs declines, the model's Pass@K steadily increases. This indicates that the model gradually converges toward a narrower distribution that is more effective at improving the rewards. We hypothesize that the patterns produced by the model's rollouts at this stage are more efficient. However, after multiple iterations, the model π_{θ_t} tends to favor exploitation, resulting in a lower TTE. To address this, we propose leveraging data generated by π_{θ_t} to re-activate the initial model π_{θ_0}, thereby achieving more efficient pattern activation as well as a higher diversity compared to π_{θ_t}.

Formally, for the initial policy model π_{θ_0}, we first conduct RL training until the model's performance converges, yielding π_{θ_t}. At this convergence point, we use π_{θ_t} to re-rollout SFT activation data and, after outcome rejection sampling, obtain a new dataset $D_{\pi_{\theta_t}}$. Subsequently, we combine the initial SFT dataset D_0 and the data from π_{θ_t} to fine-tune the model π_{θ_0}, resulting in a new initial policy model $\pi_{\theta'_0}$. At this moment, $\pi_{\theta'_0}$ have a more efficient reasoning pattern as well as a higher diversity.

As demonstrated in Table 3, our experimental results validate the effectiveness of this approach. We applied ReRFT to both a custom-trained model and the DeepSeek-R1-Distill-Qwen-14B. Our findings indicate that, compared to the baseline, the proposed ReRFT can further enhance the performance of the RL model through obtaining efficient reasoning patterns and reactivating them, thereby increasing diversity beyond π_{θ_t}.

5 Related Work

Olympiad-Level Reasoning for LLM With the release of OpenAI O1 series models (OpenAI, 2024), which demonstrated the ability of large language models to handle Olympiad-level reasoning tasks, there has been a significant interest in Olympiad-level reasoning in recent studies. Currently, there are two main technical approaches. The first involves verifier and search (Wang et al., 2024; Qi et al., 2024; Guan et al., 2025), where the model repeatedly samples during inference and relies on a verifier to select the highest-quality final result or reasoning steps, thereby enabling test-time scaling for better performance. The second approach utilizes long-CoT combined with RL. Previous studies have focused on short-CoT (Yang et al., 2024), but with the emergence of long-CoT models such as Kimi K1.5 (Team et al., 2025) and R1 (DeepSeek-AI et al., 2025), increasing research has highlighted the advantages of long-CoT in tackling complex reasoning tasks (Yeo et al., 2025; Shen et al., 2025). Moreover, by incorporating RL, the number of completion tokens of long-CoT can be further expanded, enhancing the model's overall capabilities. However, current research has not fully explored why long-CoT outperforms short-CoT. In Section 3, we compare the initial policy models activated with different CoT patterns and analyze them from different perspectives, which provide more valuable insight into this question for future studies.

Reinforcement Learning for LLM Since the introduction of RLHF (Ouyang et al., 2022), the combination of LLMs and reinforcement learning has proven to be highly effective, particularly in tasks such as aligning with human preferences, mathematical reasoning, and code generation. Recently, long-CoT combined with RL (DeepSeek-AI et al., 2025; Shen et al., 2025; Team et al., 2025; Hu, 2025; Yeo et al., 2025; Muennighoff et al., 2025) has demonstrated powerful capabilities in solving complex problems. The primary pipeline involves first activating the initial policy model through long-CoT activation, followed by policy optimization methods such as GRPO (DeepSeek-AI et al., 2025) or REINFORCE (Team et al., 2025; Hu, 2025), combined with rule-based rewards. However, there is currently a lack of systematic research on Long-CoT RL and the initial policy model.

6 Conclusion

Our work presents the first systematic investigation into the critical role of the initial policy model in long-CoT reinforcement learning process. We offer an in-depth analysis of long-cot RL using metrics beyond performance, providing insights into what constitutes a good initial policy model. Additionally, we investigate a recipe for a better initial policy model in terms of data volume. We also introduce a simple yet effective Re-RFT method to obtain more efficient reasoning patterns. We hope our work provides valuable insights into scalable reinforcement learning for future work.

7 Limitation

We systematically analyze the RL process from multiple perspectives in this paper and propose a more effective initial policy model activation recipe. However, due to cost constraints, we are unable to conduct experiments on larger models, such as those at the 72B scale. We believe that different model sizes may also exhibit a scaling taw for length token efficiency and training token efficiency, We leave this exploration for our future work.

8 Acknowledgment

We thank all reviewers for their valuable advice. This work is supported by the National Science Foundation of China under Grant No.61876004 and 61936012.

References

AIMO. 2024. Aimo prize official website. Accessed: 2023-10-10.

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, Zhuoshu 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, Fuli 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, Wanjia 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, Xiangyue 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. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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. *Preprint*, arXiv:2410.07985.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *Preprint*, arXiv:2501.04519.

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. *Preprint*, arXiv:2103.03874.

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

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024. Improve mathematical reasoning in language models by automated process supervision. *Preprint*, arXiv:2406.06592.

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. *Preprint*, arXiv:2501.19393.

Ben Norman and Jeff Clune. 2024. First-explore, then exploit: Meta-learning to solve hard exploration-exploitation trade-offs. *Preprint*, arXiv:2307.02276.

OpenAI, :, Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, Rhythm Garg, Rui Shu, Szymon Sidor, Vineet Kosaraju, and Wenda Zhou. 2025. Competitive programming with large reasoning models. *Preprint*, arXiv:2502.06807.

OpenAI. 2024. Learning to reason with large language models. Accessed: 2024-12-18.

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. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *Preprint*, arXiv:2408.06195.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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

Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gregory Wornell, Subhro Das, David Cox, and Chuang Gan. 2025. Satori: Reinforcement learning with chain-of-action-thought enhances llm reasoning via autoregressive search. *Preprint*, arXiv:2502.02508.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning 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. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown. Accessed: 2024-11-28.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *Preprint*, arXiv:2312.08935.

Siwei Wu, Zhongyuan Peng, Xinrun Du, Tuney Zheng, Minghao Liu, Jialong Wu, Jiachen Ma, Yizhi Li, Jian Yang, Wangchunshu Zhou, Qunshu Lin, Junbo Zhao, Zhaoxiang Zhang, Wenhao Huang, Ge Zhang, Chenghua Lin, and J. H. Liu. 2024a. A comparative study on reasoning patterns of openai's o1 model. *Preprint*, arXiv:2410.13639.

Ting Wu, Xuefeng Li, and Pengfei Liu. 2024b. Progress or regress? self-improvement reversal in post-training. *Preprint*, arXiv:2407.05013.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. *Preprint*, arXiv:2405.00451.

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. *Preprint*, arXiv:2409.12122.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *Preprint*, arXiv:2502.03387.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. *Preprint*, arXiv:2502.03373.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: Llm

self-training via process reward guided tree search. *Preprint*, arXiv:2406.03816.

Luisa Zintgraf, Leo Feng, Cong Lu, Maximilian Igl, Kristian Hartikainen, Katja Hofmann, and Shimon Whiteson. 2021. Exploration in approximate hyper-state space for meta reinforcement learning. *Preprint*, arXiv:2010.01062.

A Towards a More Efficient Reasoning Pattern with Mannual Ablation

Drawing on the conclusions of Wu et al. (2024a), who have identified several typical patterns in the responses of the o1 model, such as *System Analysis*, *Verification Reflection*, and *Human Reasoning Style*.

System Analysis refers to the model tends to do a systematically analysis before solving a problem. *Verification* indicates the model's tendency to use an extended chain of thought to validate its conclusions. *Reflection* describes the model's iterative self-reflection process, while *Human Reasoning Style* refers to the model adopting a human-like reasoning tone.

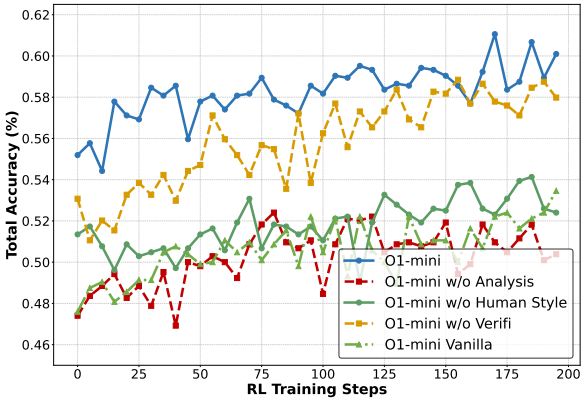


Figure 7: Manual ablation on the reasoning pattern of o1-mini.

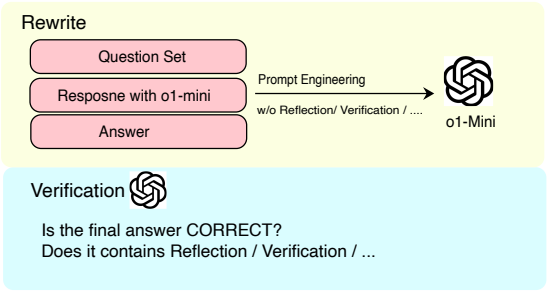


Figure 8: The Rewrite-then-Verify framework for reliable pattern ablation.

To assess the impact of these patterns on model scaling performance, we conducted an ablation study by manually removing them from the training set. Our approach follows a rewrite-then-verify strategy as shown in Figure 8: given a fully developed CoT generated via prompt engineering, we instruct the model to rewrite it while removing specific features. The final results are shown in Figure 7.

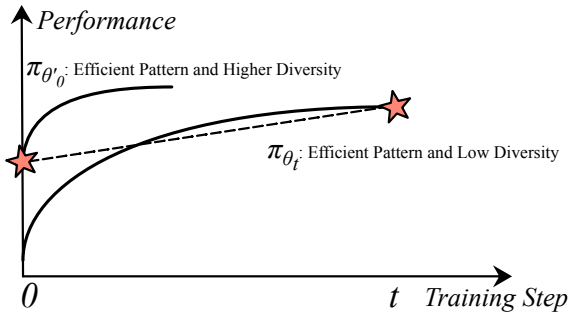


Figure 9: Illustration and explanation of ReRFT method.

Our ablation study reveals that nearly all identified features have a significant impact on model performance. Moreover, manually ablating features to discover reasoning patterns with higher LTE proves to be challenging. Therefore, we shift our focus to analyzing the metrics of the initial policy model during training (as discussed in the main text). This approach reduces human inductive bias and effectively enhances model performance

B Acc and Pass@K Improvements

We present the Train Acc and Pass@K results for different amount of activation data in Figure **??**. It is evident that in a good RL process, both Pass@K and Acc improve over time, although Pass@K grows more slowly than Acc. This indicates that RL progressively transforms the model's latent potential into actual performance. Furthermore, as training progresses, the model's potential continues to increase through reinforcement learning.

Train **Acc.**	π_{θ_0}	$\pi_{\theta'}$	Δ.
Qwen-14b w/ 10K R1-pattern	43.9%	53.1%	9.2%
DeepSeek-R1-distill-Qwen-14b	80.1%	86.9%	6.8%
Train **Pass@k**	π_{θ_0}	$\pi_{\theta'}$	Δ.
Qwen-14b w/ 10K R1-pattern	79.9%	90.5%	10.6%
DeepSeek-R1-distill-Qwen-14b	95.5%	96.5%	1%

Table 4: Comparison of model performance in terms of Accuracy (Acc) and pass@k metrics.

C Detailed Performance

The detailed model performance of using different quantitty of R1-pattern data activate π_θ is shown in Table 5.

Table 5: The detailed model performance

Model	# R1 data	ckpt	Performance					Other Metric			
			AIME	AIMO	MATH500	OmniMATH	Total	Pass@K	Distinct-N	Varlen-K	TTE
Qwen-14B-Custom-Activated	2K	π_{θ_0}	16.67	11.25	80.8	33.0	46.82	74.9	60.45	94.76	-
		$\pi_{\theta'}$	27.5	13.75	85.4	36.2	51.89	77.3	47.01	89.14	0.68
Qwen-14B-Custom-Activated	10K	π_{θ_0}	17.1	12.5	74.6	32.0	44.2	74.0	58.58	65.37	-
		$\pi_{\theta'}$	32.5	22.5	87.6	42.8	55.45	79.4	50.01	63.29	0.29
DeepSeek-R1-Distill-Qwen-14B	800K	π_{θ_0}	63.75	52.5	94.2	63.8	74.62	95.51	48.42	29.29	-
		$\pi_{\theta'}$	75.42	60.0	95.4	69.2	79.92	97.27	45.67	32.31	0.12

D Detailed Training Setup

We use the data from Tabl 1 for RL training. For the SFT data, to obtain CoT samples with diverse reasoning patterns, we first generate responses using Qwen-2.5-MATH, O1, and R1.

Note that after generating responses for the training set, we apply a strict answer filtering process, retaining only correctly answered questions. To maximize problem coverage, we perform multiple sampling when the model's initial answer is incorrect until a correct response is obtained. This minimizes the impact of answer quality inconsistencies on the model's training.

For RL training, we ensure that all reasoning patterns and activation data use an identical RL setup for the initial model.

E More Explanations on ReRFT

As shown in Figure 9, during the reinforcement learning process, π_{θ_t} gradually converges to a locally optimal distribution, characterized by lower diversity and more efficient reasoning patterns. To counteract this, we reactivate π_{θ_0} using the rollout data from π_{θ_t}, aiming to transfer the more efficient patterns learned in π_{θ_t} to improve π_{θ_0}'s Pass@K. Since π_{θ_0} has not undergone RL training, it naturally retains higher diversity. This reactivation process results in π'_{θ_0}, which exhibits both a higher RL upper bound and improved efficiency.

F Comparative Analysis of Reasoning Patterns in Different Initial Policy Models

We present a comparative analysis of the reasoning patterns in three different initial policy models (R1, Qwen, and O1-mini) based on seven reasoning patterns: *System Analysis*, *Verification*, *Reflection*, *Decomposition*, *Language Style Context Emphasis*, and *Method Reuse*. Among these, the first three are defined in Appendix A, while *Decomposition* refers to breaking down the original problem into subproblems for resolution, *Language Style* refers to the frequent use of conversational markers, *Context Emphasis* refers to whether the model highlights contextual background and constraints related to the problem, and *Method Reuse* refers to the reuse of common algorithms or models, potentially omitting certain steps.

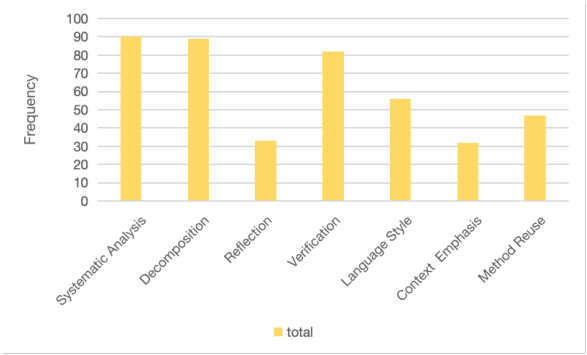


Figure 10: The frequency of reasoning patterns.

We randomly selected 30 samples from mathematical problems and analyzed the frequency of different reasoning patterns across the three models, as shown in Figure 10. In solving mathematical problems, *System Analysis*, *Verification*, and *Decomposition* play the most critical roles.

Figure 11 illustrates the differences in reasoning patterns among the three models. *Reflection* and *Verification* are likely the key factors driving the performance differences between the models. R1 places greater emphasis on reviewing and verifying conclusions, making it more stable in multi-step reasoning tasks. O1-mini shows slightly lower frequencies of *Reflection* and *Verification* compared to R1, while Qwen rarely verifies its answers, which may be a key reason for its lower performance. Furthermore, R1 demonstrates the highest level of *Context Emphasis*, consistently paying attention to contextual constraints, whereas Qwen almost never uses conversational markers. In contrast, both R1 and O1-mini employ conversational markers to guide further thinking or verification. Over-

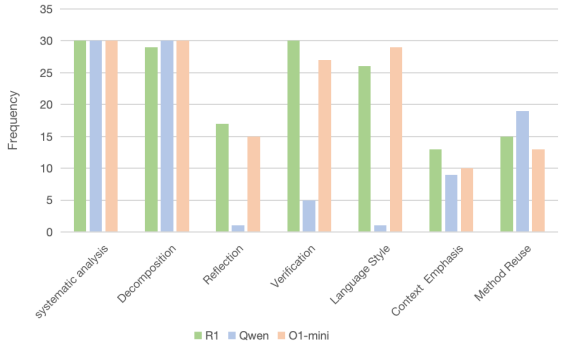


Figure 11: The frequency of reasoning patterns in different models.

all, the combination of rigorous verification, contextual awareness, and interactive language style contributes to R1's superior performance, while Qwen's lack of verification and contextual emphasis limits its effectiveness.