Correct, Concise and Complete: Multi-stage Training For Adaptive Reasoning

Anonymous ACL submission

Abstract

The reasoning capabilities of large language models (LLMs) have improved substantially through increased test-time computation, typically in the form of intermediate tokens known as chain-of-thought (CoT). However, CoT can become excessively long, raising computation costs without improving and sometimes even degrading performance, a phenomenon known as "overthinking". We propose a multi-stage efficient reasoning method consisting of supervised fine-tuning—via rejection sampling or reformatting—followed by reinforcement learning with an adaptive length penalty. We design a simple, yet effective reward function that penalizes tokens generated after the first correct answer in the reasoning trace, incentivizing the model to perform self-verification only when necessary. Our experimentation on seven diverse reasoning tasks, showed that our best approach reduces response length by 28% on average for Qwen3-8B and 40% for Qwen3-32B, with a minimal drop in downstream model performance—1.6-points and 2.5 points on average, respectively. Moreover, our approach outperforms state-of-the-art efficient reasoning approaches in terms of area under the Overthinking-Adjusted Accuracy curve (AUC_{OAA}), achieving a score of 76.6—a 5-point improvement over the base model and a 2.5-point gain over the second-best approach.

1 Introduction

Large language models (LLMs) achieve stronger performance on reasoning-heavy tasks such as math and code generation by increasing test-time computation (Snell et al., 2024; OpenAI et al., 2024; DeepSeek-AI et al., 2025; OpenAI, 2025). Accuracy often improves as the model generates longer chains of thought (CoT). However, reasoning traces can also become unnecessarily long and often times repetitive, yielding no further gains–and in some cases even reducing accuracy, a phenomenon known as "overthinking" (Chen et al., 2024; Wu et al., 2025; Yang et al., 2025c).

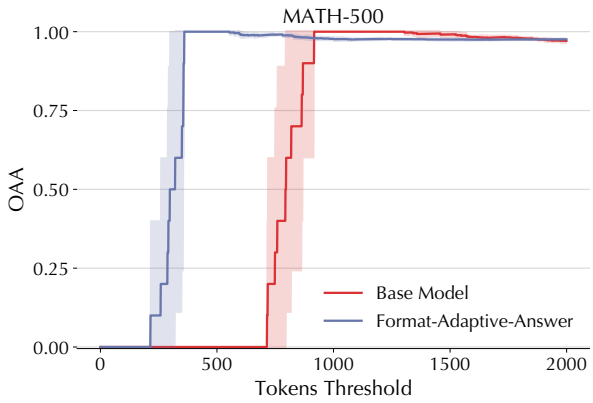

Figure 1: Overthinking-Adjusted Accuracy (OAA) (Aggarwal et al., 2025) as a function of the response length threshold on MATH-500 for Qwen3-8B. Our approach achieves similar accuracy with fewer tokens, leading to a larger area under the curve.

To mitigate this, existing methods often impose a fixed thinking budget set a priori, truncating the trace once the budget is reached (Yang et al., 2025c) or enforcing it as a hard constraint during RL training (Hou et al., 2025). However, such non-adaptive approaches are unable to strike the optimal balance between accuracy and efficiency in terms of response length (Snell et al., 2024; Wu et al., 2025; Yang et al., 2025c).

We introduce a multi-stage efficient reasoning method that adaptively reduces response length while preserving accuracy. Our method consists of supervised fine-tuning (SFT), followed by reinforcement learning (RL) with a length penalty. We construct the training dataset for the SFT stage using two different approaches: rejection sampling—selecting the shortest correct response for each problem—and reformatting of the reasoning traces to omit the summary and directly provide the final answer. Regarding the RL stage, we design a reward function that penalizes tokens generated

after the first correct answer in the trace. This encourages short but complete reasoning traces that lead to the correct answer. It also incentivizes the model to perform self-verification only when necessary as confirmed by our analyses.

We evaluate our methods on models of various sizes from the Qwen3 (Yang et al., 2025a) and DeepSeek (DeepSeek-AI et al., 2025) families, and across diverse datasets spanning math, science, code generation, question answering, and long-context reasoning. Our approaches reduce response length with only minimal accuracy degradation. Moreover, when evaluated with area under the Overthinking-Adjusted Accuracy (OAA) (Aggarwal et al., 2025)—a unified metric illustrated in Figure 1 that accounts for overthinking—our methods consistently outperform prior efficient reasoning approaches. Our main contributions are as follows:

- We propose a multi-stage efficient reasoning method combining SFT through rejection sampling or reformatting, and RL with a length penalty that penalizes tokens generated after the first correct answer. This reduces response length by 28% for Qwen3-8B with only a 1.6-point accuracy drop and 40% for Qwen3-32B with a 2.5-point drop.
- We compare our approach with state-of-the-art efficient reasoning approaches and demonstrate consistent improvements as evaluated by area under the OAA curve, a unified metric that accounts for overthinking.
- We analyze the trade-off between response length and accuracy, and study how the trained models adapt their CoT on problems of varying difficulty.

2 Related Work

Test-Time Scaling. Large Language Models perform better on reasoning-heavy tasks such as math, problem-solving, and coding by increasing test-time computation (Wei et al., 2022; Wang et al., 2022; Snell et al., 2024). Models generate intermediate tokens in parallel—by sampling multiple traces (Wang et al., 2022)—or sequentially, by verifying and correcting their own outputs (Madaan et al., 2023; Kumar et al., 2024). Recent works use reinforcement learning with verifiable rewards to further enhance reasoning capabilities, leading to longer CoT as well as self-verification and self-correction behaviors (OpenAI et al., 2024; DeepSeek-AI et al., 2025; OpenAI, 2025; Yang et al., 2025a).

Efficient Reasoning. While reinforcement learning improves reasoning ability, it often comes at the cost of efficiency. In some cases, reasoning traces become excessively long, increasing computation without improving accuracy—and sometimes even harming it, a phenomenon known as "overthinking" (Chen et al., 2024; Yang et al., 2025c; Wu et al., 2025). Several methods have been proposed to address this issue. The most direct approach, budget forcing (Muennighoff et al., 2025; Yang et al., 2025a), interrupts generation once a predefined threshold is exceeded. Other methods (Yang et al., 2025c; Xia et al., 2025; Cui et al., 2025) construct synthetic datasets by shortening model-generated reasoning traces (via rejection sampling or pruning) and then perform supervised fine-tuning. A different line of work, to which our method belongs, uses reinforcement learning with a length penalty in addition to the correctness reward. The length constraint can be either hard, applied once the CoT length exceeds a fixed threshold (Hou et al., 2025), or soft, where the penalty increases gradually as the trace length approaches the threshold (Yu et al., 2025; Aggarwal and Welleck, 2025). Instead of applying penalties independently per example, some approaches (Team et al., 2025; Yang et al., 2025b) define them relative to the length and correctness of other traces within the same GRPO group.

Unlike prior methods, which rely on a manually fixed "thinking budget" shared across all inputs, we train models to produce short yet complete reasoning traces while preserving accuracy. Our approach incentivizes the model to adaptively infer an input-dependent budget.

3 Methodology

We propose a multi-stage training method to shorten the reasoning traces of LLMs: supervised fine-tuning followed by reinforcement learning with an adaptive length penalty. This is analogous to the paradigm originally used to train these reasoning LLMs (DeepSeek-AI et al., 2025).

Supervised Fine-Tuning This first stage serves as an initialization for the RL training and helps improve its convergence. We construct supervised training datasets using two different approaches:

- **Rejection sampling**: For each problem, we

generate multiple continuations and select the shortest correct one. While rejection sampling has been explored in prior work as a baseline or stand-alone method (Yang et al., 2025c), we use it as the initial stage of training to bias the model towards concise reasoning traces.

- **Reformatting**: The second approach consists of modifying the format of the traces generated by the models. Reasoning models typically generate a structured trace in which the intermediate reasoning (often enclosed within <think></think>) is followed by a summary and then the final answer. We construct the dataset by removing the summary and retain only the final answer. This encourages the model to produce direct solutions without redundant reformulations.

RL with Adaptive Length Penalty After SFT, we further improve efficiency through RL with adaptive length penalty. More precisely, we design a verifiable reward and use group relative policy optimization (GRPO) (Shao et al., 2024) for model training. In addition to the standard correctness reward, we apply a length penalty that incentivizes shorter, input-dependent reasoning traces. Specifically, we penalize tokens after the first correct answer in the trace. Prior methods truncate or prune traces at the token or sentence level (Cui et al., 2025; Xia et al., 2025), which may disrupt the reasoning flow. Instead, our reward function incentivizes complete, concise and correct responses. The penalty is the proportion of excess tokens beyond the first correct answer relative to the entire trace. Formally, let y denote the full sequence of generated tokens, y_{first} the subsequence up to the first correct answer (empty if no correct answer is produced), and L a function that returns the number of tokens in a sequence. The length penalty is:

$$R_L(y) = \begin{cases} \frac{L(y)-L(y_{\text{first}})}{L(y)} & \text{if correct} \\ 0 & \text{otherwise} \end{cases}$$

If we denote by R_C the correctness and format reward function, the overall reward is:

$$R(y) = R_C(y) - \lambda R_L(y)$$

where λ is a hyperparameter controlling the influence of the length penalty.

This formulation encourages the model to minimize unnecessary self-verification while still allowing self-correction: if the model initially produces an incorrect answer but later correctly revises it, no penalty is applied.

We refer to the method that applies rejection sampling during SFT as ***Adaptive-Answer***, and to the method that applies reformatting during SFT as ***Format-Adaptive-Answer***.

4 Experimental Setup

Models. We use Qwen3-8B (Yang et al., 2025a) as the main model in our experiments. We further validate our method on Qwen3-1.7B, Qwen3-32B and DeepSeek-R1-Qwen-7B-distilled (DeepSeek-AI et al., 2025). Qwen3-32B was directly fine-tuned with reinforcement learning with verifiable reward, whereas the other models were trained through supervised fine-tuning on reasoning traces generated by a bigger model.

Training Dataset. We train models on a sample of 13K problems from DeepScaleR (Luo et al., 2025), a collection of math datasets with problems drawn from AIME 1983-2023, AMC, Omni-Math (Gao et al., 2024), and STILL (Min et al., 2024).

Implementation Details. For rejection sampling, we generate 8 continuations for each problem. During the SFT stage, we train for 2 epochs with a batch size of 256 and a learning rate of 1e-5. Regarding the RL stage, we use GRPO (Shao et al., 2024) as implemented by Verl framework (Sheng et al., 2024). We fine-tune the models with a group size of 8 and a global batch size of 256 for 50 iterations. We use the Adam optimizer with a learning rate of 1e-6, KL regularization with $\beta = 0.001$. For all experiments including the baselines, we set the maximum number of output tokens to 16,384.

Evaluation. We evaluate the models on a wide range of datasets spanning math, coding, question answering and long context reasoning: AIME 24, AIME 25 (AIME, 2025), MATH-500 (Lightman et al., 2023), LiveCodeBenchv6 (Jain et al., 2024), GPQA-Diamond (Rein et al., 2024), LongBenchv2 (Bai et al., 2024), CommonsenseQA (Talmor et al., 2018). We provide more details about the datasets in Appendix A. We use the following decoding hyperparameters as recommended in Yang et al. (2025a) for the Qwen3 models: temperature = 0.7, top-p = 0.8, top-k = 20, and presence penalty = 1.5. We set the maximum number of output tokens to 32,768 except for MATH-500, AIME 24, AIME 25 where we set it to 40,000. For each question, we sample N times and report the average accuracy

as the final score. We use $N = 64$ for AIME 24, AIME 25 and $N = 10$ for the rest.

Metrics. We report both the accuracy and the response length or number of generated tokens to characterize the performance-efficiency trade-off. Not all points can be directly compared using these two metrics. Therefore. we also report the area under the Overthinking-Adjusted Accuracy curve (AUC_{OAA}) (Aggarwal et al., 2025). OAA_t measures the accuracy of the model when using fewer than t tokens:

$$\text{OAA}_t = \frac{1}{n}\sum_{i=1}^{n}(\text{Accuracy}_i \cdot \mathbb{I}(\#\text{Tokens}_i < t)$$

AUC_{OAA} is the area under the OAA_t curve, where the x-axis represents the threshold of number of tokens t and the y-axis represents the corresponding OAA_t score as illustrated in Figure 1.

$$\text{AUC}_{\text{OAA}} = \int_0^{t_{\text{max}}} \frac{\text{OAA}_t}{t_{\text{max}}} dt \approx \sum_0^{t_{\text{max}}} \frac{\text{OAA}_t}{t_{\text{max}}}$$

where t_{max} is a predefined maximum number of tokens. Setting t_{max} to a very high value is equivalent to using regular accuracy that does not account for the shorter traces. Therefore, for each dataset, we set t_{max} to the mean number of generated tokens of the original base model.

Baselines. We compare our methods with existing state-of-the-art efficient reasoning approaches. We chose a representative set of methods to cover a wide variety of techniques:

- **Supervised Fine-tuning (SFT)**: For each problem in the training dataset, we generate 8 continuations and retain the shortest correct answer. We then fine-tune the model on the resulting dataset.
- **RL with Hard Length Penalty** (Hou et al., 2025): Traces are truncated if they exceed a pre-defined maximum length. We set this threshold to 16k tokens, the maximum used in all our experiments, and 8k tokens, the average response length on the training set. We also report a curriculum variant that first trains with an 8k cutoff before lowering the threshold to 4k.
- **RL with Soft Length Penalty** (Yu et al., 2025): In addition to a hard cutoff L_{max}, a second threshold L_{cache} introduces a gradually increasing penalty once the response length exceeds it. We set $L_{\text{max}} = 10\text{k}$ and $L_{\text{cache}} = 8\text{k}$.
- **RL with Normalized Length Penalty** (Team et al., 2025): The length penalty is normalized using the minimum and maximum response lengths sampled within the same GRPO group.
- **RL with TWYN** (Yang et al., 2025b): Think When You Need (TWYN) is an adaptive method where rewards are based on pairwise comparisons: shorter correct responses receive higher rewards, while all incorrect responses receive equally low rewards.

5 Experimental Results

Response Length Reduction. Figure 2 shows accuracy as a function of response length across datasets when applying different efficient reasoning methods to Qwen3-8B (see Table 4 in the Appendix B for absolute numbers). The green region corresponds to the set of points dominated by *Adaptive-Answer*, while the orange region corresponds to the set of points that dominate *Adaptive-Answer*.

We can see that our methods substantially reduce response length while maintaining comparable accuracy on most datasets. The degree of reduction varies across tasks, however we can clearly see that even our the less aggressive length reduction variant (*Adaptive-Answer*) dominates other alternatives (i.e., there are almost no points in the orange area in the figures). More precisely, *Adaptive-Answer* dominates most methods on MATH-500, AIME 24 and CommonsenseQA. While it gets dominated only in two cases: *(i)* by *Hard-Length* and *Soft-Length* on LiveCodeBench, and *(ii)* by *Hard-Length* on LongBenchv2. *Format-Adaptive-Answer* dominates almost all other methods on the math and QA datasets, but it is dominated on LiveCodeBench and LongBenchv2. The reduction in response length across all efficient reasoning approaches is smaller on these two datasets compared to the others, possibly due to training exclusively on math datasets.

In relative terms, the largest reductions—without any performance degradation—are observed on MATH-500 (36% for *Adaptive-Answer* and 50% for *Format-Adaptive-Answer*) and CommonsenseQA (30% and 45%, respectively). On GPQA Diamond, AIME 24, and AIME 25, re-

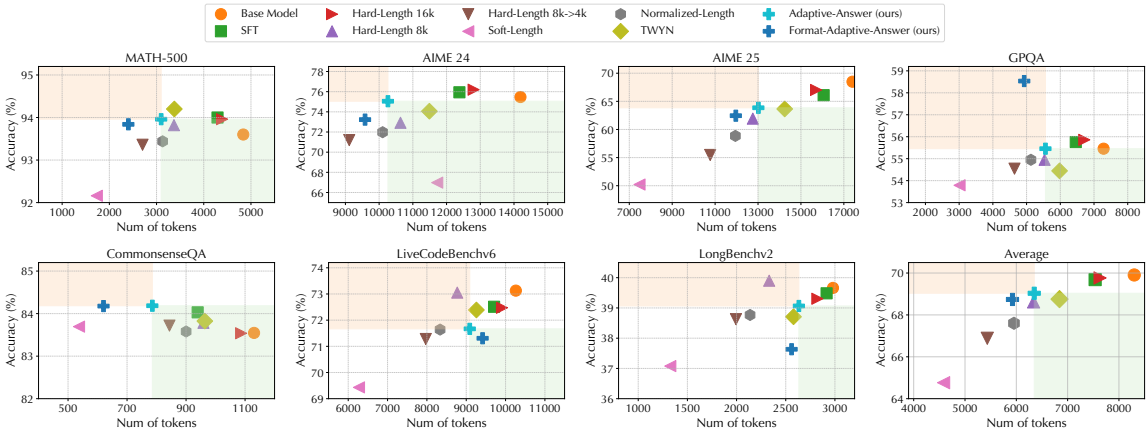

Figure 2: Average accuracy and number of tokens of each method using the Qwen3-8B model. Points in the green region are dominated by *Adaptive-Answer* while points in the orange area dominate *Adaptive-Answer*.

	MATH-500	AIME 24	AIME 25	GPQA Diamond	Common-sense QA	LiveCode-Bench	Long-Benchv2	Average
Base model	81.7	68.6	75.8	69.4	72.1	**94.3**	39.4	71.6
SFT	87.9	73.7	76.9	68.7	76.8	93.9	40.2	74.0
Hard-Length 16k	83.4	70.7	76.3	68.9	72.5	94.1	40.9	72.3
Hard-Length 8k	87.9	73.3	77.2	63.5	73.3	93.1	40.7	72.7
Hard-Length 8k → 4k	89.1	75.1	69.8	61.0	76.5	92.7	38.9	71.8
Soft-Length	87.9	72.4	72.9	62.9	73.3	93.0	39.4	71.6
Normalized-Length	**91.9**	77.4	61.0	58.3	78.6	90.0	36.6	70.5
TWYN	89.6	74.5	79.6	67.1	74.8	93.9	38.8	74.0
Adaptive-Answer (ours)	90.3	75.8	**80.0**	68.8	**81.4**	93.1	39.4	75.5
Format-Adaptive-Answer (ours)	91.2	**81.8**	**80.0**	**71.6**	**81.3**	93.6	37.4	**76.6**

Table 1: AUC_{OAA} of all approaches applied to Qwen3-8B across datasets. On average, *Format-Adaptive-Answer* achieves the best performance, followed by *Adaptive-Answer*.

sponse lengths decrease by about 25% for *Adaptive-Answer* and 32% for *Format-Adaptive-Answer*. For LiveCodeBench and LongBenchv2, both methods show only minor accuracy drops (less than two points), accompanied by smaller length reductions—11% and 12% for *Adaptive-Answer*, and 8% and 14% for *Format-Adaptive-Answer*. On average, *Adaptive-Answer* shortens responses by 28% and *Format-Adaptive-Answer* by 33%, with just a one-point decrease in accuracy. Even though the training is performed only on math datasets, our methods also shorten responses on science, code, QA, and long-context reasoning datasets.

We must highlight that not all methods are directly comparable using accuracy and response length. For example, on all datasets except AIME 24, *Soft-Length* neither dominates nor is dominated by other methods. Therefore, we also compare the AUC_{OAA} of all approaches applied to Qwen3-8B across datasets (see Table 1). This also confirms that the effectiveness of our methods—on average, *Format-Adaptive-Answer* outperforms all methods followed by *Adaptive-Answer*. *Format-Adaptive-Answer* obtains the highest score on all math and question answering datasets except for MATH-500 where it is the second-best. Interestingly, on LiveCodeBench and LongBenchv2, simple baselines such as *SFT* and *Hard-Length 16k*, which is equivalent to RL without length penalty, outperform all efficient reasoning alternatives.

Component ablations. To evaluate the contribution of each component of our approach, we evaluate each component of our approaches. Table 2 reports the average accuracy, response length, and AUC_{OAA} for several configurations: the original model, rejection sampling–based SFT, format-based SFT, RL with adaptive length penalty (without SFT), rejection sampling–based SFT followed by RL (*Adaptive-Answer*), and format-based SFT followed by RL (*Format-Adaptive-Answer*).

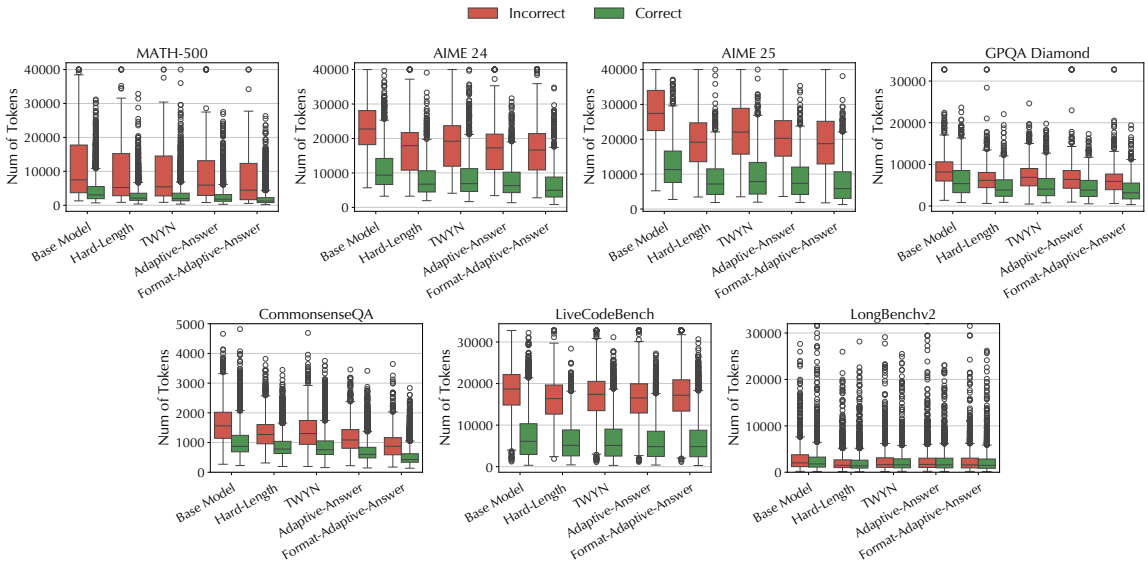

Figure 3: Response length distributions of some representative efficient reasoning methods applied to Qwen3-8B. We separate the correct and incorrect responses.

Adding the rejection sampling–based SFT (*SFT (Rejection Sampling)*) stage does not yield clear improvements when we examine the accuracy or the response length alone—*Adaptive-Answer* achieves slightly higher accuracy than *RL (no SFT)* but produces longer responses. Hereby, we argue that the AUC_{OAA} is a more suitable metric for ranking the models, that clearly highlights the benefit of the SFT phase becomes evident when considering that metric. We see a sizable improvement of 3 AUC points over the base model.

Format-based SFT (*SFT (Formatting)*) reduces response length by about 10% on average without any loss in accuracy. This indicates that the summary generated before the final answer does not contribute to performance, as the model already arrives at the correct answer by the end of the thinking content. Adding the RL stage further improves AUC_{OAA} by 3 points and shortens responses by 18%, though with a minor accuracy drop. However, combining rejection sampling and formatting during SFT did not yield additional improvements over formatting alone.

Finally, we can see that SFT is a vital stage for the RL performance, even though *RL (no SFT)* has the second shortest reasoning length, it comes with a performance penalty, and thus a lower AUC_{OAA} compared to the full approaches (*Adaptive-Answer* and *Format-Adaptive-Answer*).

Efficiency and Model Size. We ablate how our approaches scale with model size and training regimes. For this set of experiments we chose Qwen3-1.7B, Qwen3-8B, Qwen3-32B, and DeepSeek-R1-Qwen-7B-Distilled. Across all models, the performance drop after applying our methods is less than 2.5 points—and negligible on DeepSeek-R1-Qwen-7B-Distilled (see Table 3). The reduction in the number of generated tokens grows with model size: *Adaptive-Answer* reduces response length by 22% on Qwen3-1.7B, 23% on Qwen3-8B, and 40% on Qwen3-32B. AUC_{OAA} does not always align with accuracy–response length trade-offs, particularly for smaller models. For example, the fine-tuned DeepSeek-R1-Qwen-7B-Distilled dominates the base model but achieves a lower AUC_{OAA} score. This results clearly show that our methods are invariant to the base model, and hold effective across different variants.

	Accuracy	**#Tokens**	**AUC_{OAA}**
Base Model	69.9	8295	71.6
SFT (Rejection Sampling)	69.7	7536	74.0
SFT (Formatting)	70.1	7475	74.7
RL (no SFT)	68.8	6303	73.2
Adaptive-Answer	69.0	6344	75.5
Format-Adaptive-Answer	68.7	5918	76.6

Table 2: Average accuracy, response length, and AUC_{OAA} of the original model, rejection sampling–based SFT, format-based SFT, RL with adaptive length penalty (without SFT), rejection sampling–based SFT followed by RL (*Adaptive-Answer*), and format-based SFT followed by RL (*Format-Adaptive-Answer*).

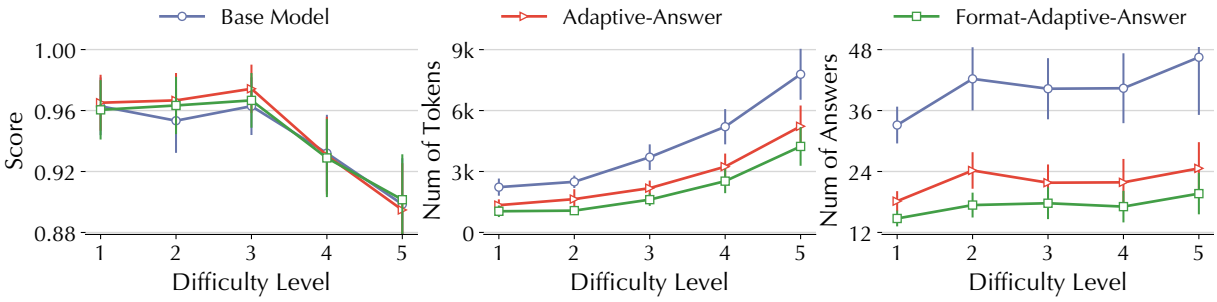

Figure 4: Accuracy, response length, and number of intermediate correct answers as a function of difficulty level on MATH-500.

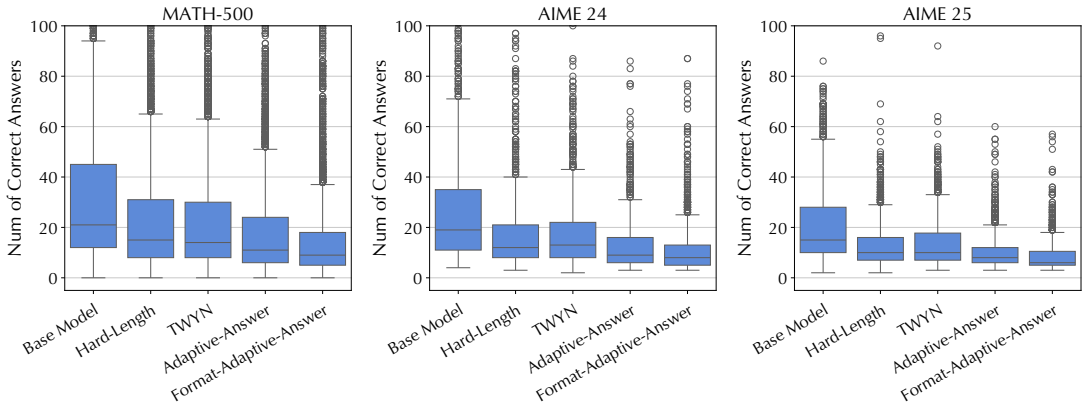

Figure 5: Distributions of the number of correct answers in the traces of some representative efficient reasoning methods applied to Qwen3-8B for MATH-500, AIME 24 and AIME 25.

	Accuracy	#Tokens	AUC_{OAA}
Qwen3-1.4B			
Base Model	50.9	7619	65.4
Adaptive-Answer	49.1	5884 (-22%)	62.1
Format-Adaptive-Answer	48.3	5918 (-22%)	62.1
Qwen3-8B			
Base Model	69.9	8298	71.6
Adaptive-Answer	69.0	6344 (-23%)	75.5
Format-Adaptive-Answer	68.7	5918 (-28%)	76.6
Qwen3-32B			
Base Model	74.8	7294	69.2
Adaptive-Answer	72.1	4280 (-41%)	72.5
Format-Adaptive-Answer	72.2	4372 (-40%)	72.3
DeepSeek-R1-Qwen-7B-Distill			
Base Model	50.3	6272	62.1
Adaptive-Answer	50.4	5133 (18%)	59.6
Format-Adaptive-Answer	50.7	4612 (26%)	59.7

Table 3: Average accuracy, response length and AUC_{OAA} of our methods applied to Qwen3.1.7B, Qwen3-8B, Qwen-32B and DeepSeek-R1-Qwen-7B-distilled.

6 Analysis

Response Length Distribtion. In Figure 3, we present the response length distributions of a set of representative efficient reasoning methods applied to Qwen3-8B. The plots demonstrate that the number of generated tokens is higher for incorrect answers than for correct ones on all three datasets. Notably, our methods effectively shifts the response length distribution not only for correct answers but also for incorrect ones, indicating that it is adapting all traces regardless of the answer.

Intermediate Answers. The RL stage of our approaches encourages the model to produce as few self-verifications as needed. To quantify this, we report the number of correct answers in the traces for the math datasets MATH-500, AIME 24 and AIME 25 in Figure 5. This metric is only a proxy of the number of self-verifications, as the answer might appear multiple times during a verification. Both *Adaptive-Answer* and *Format-Adaptive-Answer* shift its distribution, indicating reduced redundancy in reasoning.

Difficulty Analysis. We examine the relationship between problem difficulty and accuracy, response length, and the number of intermediate correct answers. Each problem in MATH-500 is assigned a difficulty level ranging from 1 to 5. As shown in

Base Model

<think>Okay, so Jen is playing this lottery game ... So yes, $m = 1$ and $n = 115$, so $m + n =$ **116**. But wait, that seems low ... Then $m + n = 1 + 115 =$ **116**. Hmm. But I feel like this might not be correct. ... Therefore, $m + n = 1 + 115 =$ **116**. But I need to make sure that this is correct. Let me think again ... Therefore, $m + n = 1 + 115 =$ **116**. But wait, intuitively, if the probability of winning the grand prize given ... Therefore, the answer is **116**. But wait, let me check again ... Therefore, answer is $1 + 115 =$ **116**. But let me check if there's an alternative approach... so the answer is $1/115$, so $m + n = 1 + 115 =$ **116**. But just to be thorough, let me check if ... Final Answer: **116** <think> To solve this problem, we are asked to find the **conditional probability** ... and their sum is:$m + n = 1 + 115 =$ **116**

Adaptive-Answer

<think>Okay, so Jen is entering a lottery where she has to pick 4 distinct numbers from the set S ... So m + n is 1 + 115 = **116**. Wait, but before I conclude, let me verify once more because sometimes with probability ... Therefore, the answer is 1 + 115 = **116**. But just to make sure, let me think of another way ... I think the answer is 116.**Final Answer** **116** </think> To solve the problem, we are asked to find the probability ... These are relatively prime.$m + n = 1 + 115 =$ **116**

Format-Adaptive-Answer

<think>Okay, let me try to figure out this probability problem. So, Jen is entering a lottery where ... **Final Answer** **116** </think> **Final Answer** **116**.

Figure 6: Reasoning traces of Qwen3-8B on AIME 24 Problem 10 before and after fine-tuning. The base model performs seven self-verifications after arriving at the correct answer, whereas *Adaptive-Answer* performs only two and *Format-Adaptive-Answer* performs none.

Figure 4, our approaches maintain the base model's accuracy across all difficulty levels. We also observe that our approach is truly adpative such that the number of reasoning tokens increases with increase in difficulty suggesting the need for more thinking. Although response length and the number of intermediate correct answers does increase with problem difficulty, both remain lower than those of the base model.

Qualitative Analysis. In Figure 5, we show a comparison between the reasoning traces produces by: (*i*) the base model (Qwen3-8B), on a math problem from AIME24. (*ii*) *Adaptive-Answer*, and (*iii*) *Format-Adaptive-Answer* (long responses are trimmed). We can see that the base model , needlessly, performs 7 self-verifications after producing the first correct answer (*116*). In contrast, *Adaptive-Answer* reduces them in half, to only 3, while the *Format-Adaptive-Answer* has an optimal reasoning trace, that does not perform any self-verification, and directly generates the final answer without summarizing the thinking content.

7 Conclusion

Large Language Models (LLMs) tend to perform better on reasoning-intensive tasks such as math and code generation by producing longer chains of thought. However, these CoT can often be unnecessarily long, increasing inference costs without improving accuracy. We propose a multi-stage efficient reasoning method consisting of supervised fine-tuning—via rejection sampling or reformatting—followed by reinforcement learning with an adaptive length penalty. Our best approach effectively shortens response length (by 28% for Qwen3-8B and 40% for Qwen3-32B)with only a minor performance drop across models of different sizes (up to 2.5 points), and when measuring with the unified metric AUC_{OAA}, they outperform existing methods by 2.5 points.

Limitations

Although we evaluate our methods on datasets from multiple domains, our training is performed exclusively on math datasets. Extending training to a more diverse set of tasks could yield a better accuracy–response length trade-off. Moreover, the efficient reasoning methods we propose are post hoc interventions; we do not explore incorporating the adaptive length penalty directly during the initial RL training. Additionally, due to resourcing constraints, we focused our experiment scope to models of different sizes within the Qwen family based on a dense architecture. Future explorations can consider extending our approach to more model families and other architecturea such as Mixture-of-Experts. Finally, we focus exclusively on the model performance on reasoning tasks, and we do measure the change in performance on other task groups that are less dependent on the CoT quality.

Ethics Statement

One of our methods removes the summary that the model produces at the end of its thinking content. Although, this results in shorter responses, this might also reduce the legibility of the reasoning traces.

References

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. Optimalthinkingbench: Evaluating over and underthinking in llms. *arXiv preprint arXiv:2508.13141*.

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.

AIME. 2025. AIME problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, and 1 others. 2025. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2502.13260*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, and 1 others. 2024. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.

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

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. *Notion Blog*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, and 1 others. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413*.

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

OpenAI. 2025. Openai o3 and o4-mini system card. https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf.

OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Junjie Yang, Ke Lin, and Xing Yu. 2025b. Think when you need: Self-adaptive chain-of-thought learning. *arXiv preprint arXiv:2504.03234*.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. 2025c. Towards thinking-optimal scaling of test-time compute for llm reasoning. *arXiv preprint arXiv:2502.18080*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

A Datasets

We evaluate on the following datasets that come from diverse domains including math, science, coding, question answering and long context reasoning:

- **MATH-500** (Lightman et al., 2023): A representative subset of 500 problems from the MATH benchmark. Each problem is assigned a difficulty level rangin from 1 to 5.
- **AIME 24** (AIME, 2025): 30 math problems from the 2024 edition of the American Invitational Mathematics Examination, a prestigious high school mathematics competition known for its challenging mathematical problems.
- **AIME 25** (AIME, 2025): 30 math problems from the 2025 edition of the American Invitational Mathematics Examination, a prestigious high school mathematics competition known for its challenging mathematical problems.
- **GPQA Diamond** (Rein et al., 2024): a subset of 198 expert-written, graduate-level questions in biology, physics, and chemistry, designed to test the true reasoning abilities of Large Language Models (LLMs) without reliance on easily found internet answers
- **CommonsenseQA** (Talmor et al., 2018): a dataset consisting of 1221 multiple-choice questions that require commonsense knowledge to predict the correct answers . Each question has one correct answer and four distractor answers.
- **LiveCodeBench** (Jain et al., 2024): A holistic and contamination-free benchmark to evaluate the coding capabilities of LLMs. We use the sixth version of the dataset which contains 055 problems.
- **LongBenchv2** (Bai et al., 2024): a dataset of 503 challenging multiple-choice questions, with contexts ranging from 8k to 2M words. It contains the following categories: single-document QA, multi-document QA, long in-context learning, long-dialogue history understanding, code repo understanding, and long structured data understanding.

B Accuracy-Response Length Trade-off

We present the accuracy and response length of the different efficient reasoning methods applied to Qwen3-8B in Table 4.

AI use disclosure: we used AI for assistance in code writing and in manuscript typesetting.

Model	MATH-500		AIME 24		AIME 25		GPQA Diamond		Common-senseQA		LiveCode-Bench		Long-Benchv2		Average	
	Acc.	#Tokens	Acc.	#Tokens	Acc.	#Tokens	Acc.	#Tokens	Acc.	#Tokens	Acc.	#Tokens	Acc.	#Tokens	Acc.	#Tokens
Original	93.6	4837	75.5	14191	68.5	17402	55.5	7284	83.5	1130	73.1	10261	39.7	2980	69.9	8298
SFT	94.0	4292	75.9	12381	66.1	16058	55.8	6459	84.0	939	72.5	9708	39.5	2915	69.7	7536
Hard-Length 16k	94.0	4388	76.2	12803	67.0	15707	55.9	6712	83.5	1087	72.5	9909	39.3	2822	69.8	7633
Hard-Length 8k	93.8	3369	72.9	10633	61.9	12758	54.9	5535	83.8	959	73.1	8772	39.9	2332	68.6	6337
Hard-Length 8k $\rightarrow$ 4k	93.4	2703	71.2	9114	55.5	10770	54.5	4642	83.7	844	71.3	7972	38.6	1994	66.9	5434
Soft-Length	93.4	3129	72.0	10105	58.9	11944	54.9	5128	83.6	900	71.6	8335	38.8	2137	67.6	5954
Normalized-Length	92.2	1734	67.0	11723	50.2	7475	53.8	3011	83.7	537	69.4	6273	37.1	1326	64.8	4583
TWYN	94.2	3377	74.1	11491	63.6	14243	54.4	5978	83.8	964	72.4	9259	38.7	2579	68.8	6841
Adaptive-Answer (ours)	94.0	3098	75.1	10261	63.9	13017	55.5	5560	84.2	786	71.7	9089	39.1	2634	69.0	6349
Format-Adaptive-Answer (ours)	93.8	2403	73.2	9583	62.4	11965	58.5	4931	84.2	620	71.3	9416	37.6	2559	68.7	5925

Table 4: Accuracy and response length (#Tokens) of all approaches applied to Qwen3-8B across datasets. Each cell shows Accuracy / Response Length.