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

Nathanaël Carraz Rakotonirina* ◇ **Ren Pang** ♣ **Neha Anna John** ♣
Michael Bohlke-Schneider ♣ **Momchil Hardalov** ♠
◇Universitat Pompeu Fabra ♣AWS AI Labs ♠Amazon AGI
nathanael.rakotonirina@upf.edu
{renpang, nehajohn, bohlkem, momchilh}@amazon.com

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 often becomes unnecessarily long, increasing computation costs without improving accuracy and sometimes even degrading performance, a phenomenon known as "*overthinking*". We propose a multi-stage efficient reasoning method that combines supervised fine-tuning—via rejection sampling or reasoning trace reformatting—with reinforcement learning using an adaptive length penalty. We introduce a lightweight reward function that penalizes tokens generated after the first correct answer, encouraging the model to perform self-verification only when beneficial. We conduct a holistic evaluation across seven diverse reasoning tasks, analyzing the accuracy–response length trade-off. Our approach reduces response length by an average of 28% for 8B models and 40% for 32B models, while incurring only minor performance drops of 1.6 and 2.5 points, respectively. Despite its conceptual simplicity, it achieves a better trade-off than more complex state-of-the-art efficient reasoning methods, scoring 76.6 on the area under the Overthinking-Adjusted Accuracy curve ($\mathrm{AUC}_{\mathrm{OAA}}$)—5 points above the base model and 2.5 points above the second-best approach.

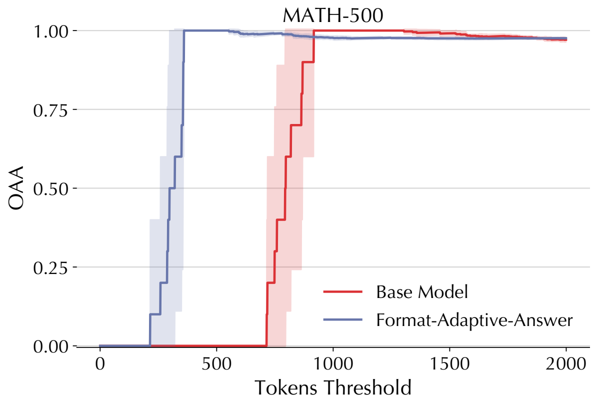

Figure 1: Overthinking-Adjusted Accuracy (OAA) (Aggarwal et al., 2025) as a function of the response length threshold on MATH-500 for Qwen3-8B. Our approach achieves similar accuracy with fewer tokens, leading to a larger area under the curve.

1 Introduction

Large language models (LLMs) achieve stronger performance on reasoning-intensive tasks, such as math and code generation, by increasing test-time computation (Snell et al., 2025; OpenAI, 2024; DeepSeek-AI, 2025; OpenAI, 2025; Amazon AGI, 2025). Accuracy often improves as the model generates longer chains of thought (CoT). However, reasoning traces can also become unnecessarily long and repetitive, yielding no additional gains and, in some cases, even reducing accuracy, a phenomenon known as "*overthinking*" (Chen et al., 2025; Yang et al., 2025c; Wu et al., 2026).

To mitigate this, existing methods often impose a predefined thinking budget, truncating the reasoning trace once the budget is reached (Yang et al., 2025c) or enforcing it as a hard constraint during reinforcement learning (RL) training (Hou et al., 2025). However, such non-adaptive methods cannot optimally balance accuracy and efficiency, as measured by response length (Snell et al., 2025; Yang et al., 2025c; Wu et al., 2026).

We introduce a multi-stage efficient reasoning framework that adaptively shortens response length while maintaining the base models' accuracy. Our method consists of supervised fine-tuning (SFT), followed by reinforcement learning with a length penalty. We construct the training dataset for the SFT stage using two approaches: rejection sampling, selecting the shortest correct response for each problem, and reformatting reasoning traces to omit additional summaries and provide the fi-

*Work conducted during an internship at AWS AI Labs.

Findings of the Association for Computational Linguistics: ACL 2026, pages 12798–12810

nal answer. For the RL stage, we design a reward function that penalizes tokens generated after the first correct answer in the trace, encouraging concise yet complete reasoning traces that lead to the correct answer. It also incentivizes the model to perform self-verification only when necessary, as we show in our analysis.

We evaluate our methods on models of different sizes from the Qwen3 and DeepSeek families, using a wide range of reasoning benchmarks, including mathematics, science, code generation, question answering, and long-context tasks. Our approach significantly reduces response length while maintaining high accuracy. Furthermore, when measured using the area under the Overthinking-Adjusted Accuracy curve (OAA; Aggarwal et al. (2025)), a unified metric that accounts for overthinking (see Figure 1), our methods consistently improve over both the base models and state-of-the-art efficient reasoning approaches. Our main contributions are as follows:

- We propose a multi-stage efficient reasoning framework that combines SFT via rejection sampling or trace reformatting with RL using a reward 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 by 40% for Qwen3-32B with a 2.5-point drop.

- We compare our approach with state-of-the-art efficient reasoning methods and demonstrate consistent improvements using 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 chains of thought (CoT) for problems of varying difficulty.

2 Methodology

To obtain optimal LLM reasoning traces, we propose a multi-stage training framework based on supervised fine-tuning followed by reinforcement learning with an adaptive length penalty. This approach follows the paradigm originally used to train reasoning LLMs (DeepSeek-AI, 2025).

Supervised Fine-Tuning This first stage serves as a warm-up for RL training and also improves convergence. We construct our supervised training datasets using the following approaches:

1. **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), in contrast, we use it as the initial stage to bias the model toward concise reasoning traces.

2. **Reformatting**: This approach modifies the format of model-generated reasoning traces. Reasoning models typically produce 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 retaining only the final answer, encouraging the model to generate direct solutions without redundant reformulations.

RL with Adaptive Length Penalty After SFT, we further improve efficiency through RL with an adaptive length penalty. Specifically, we design a verifiable reward function and use group relative policy optimization (GRPO; (Shao et al., 2024)) for training. In addition to the standard correctness reward, we apply a length penalty to encourage shorter, input-dependent reasoning traces, penalizing tokens generated after the first correct answer. Prior methods truncate or prune traces at the token or sentence level (Cui et al., 2025; Xia et al., 2025), which can disrupt the reasoning flow. In contrast, our reward function promotes responses that are concise, complete, and correct.

The penalty is defined as the proportion of *tokens after the first correct answer* relative to the full trace. Formally, let y denote the generated token sequence, y_{first} the subsequence up to the first correct answer (empty if none is produced), and L a function returning the number of tokens in a sequence. The length penalty is:

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

where y_{first} denotes the prefix ending at the first correct answer. Let $R_C(y)$ denote the correctness and format reward. The overall reward is

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

where λ controls the trade-off between correctness and reasoning efficiency; in our experiments, we set $\lambda = 1$.

We locate the first correct answer using normalized matching. If no correct answer is produced, $y_{\text{first}} = \emptyset$, yielding zero length penalty. This discourages redundant self-verification while allowing self-correction: if the model initially produces an incorrect answer but later revises it correctly, no penalty is applied.

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

3 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, 2025). Qwen3-32B was directly fine-tuned with reinforcement learning with verifiable reward, while the other models were trained via supervised fine-tuning on reasoning traces generated by a larger model.

Training Dataset. We train 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., 2025), and STILL (Min et al., 2024).

Although we train exclusively on math datasets, we evaluate our models across diverse domains, including science QA, commonsense reasoning, code generation, and long-context tasks. Our results show that the effects of our adaptive length penalty—reducing redundant self-verification and avoiding unnecessary continuation once correctness is reached—are domain-agnostic properties of reasoning traces. We observe consistent reductions in response length with minimal accuracy loss across non-math tasks outside our training domain.

Evaluation. We evaluate on the following datasets, which cover diverse domains including math, science, coding, question answering, and long-context reasoning:

- **MATH-500** (Lightman et al., 2024): A representative subset of 500 problems from the MATH benchmark. Each problem is assigned a difficulty level ranging 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 without relying on easily found internet answers.

- **CommonsenseQA** (Talmor et al., 2019): A dataset consisting of 1,221 multiple-choice questions that require commonsense knowledge to identify the correct answer. Each question has one correct answer and four distractors.

- **LiveCodeBench** (Jain et al., 2025): A holistic, contamination-free benchmark for evaluating the coding capabilities of LLMs. We use the sixth version of the dataset, which contains 55 problems.

- **LongBenchv2** (Bai et al., 2025): A dataset of 503 challenging multiple-choice questions, with contexts ranging from 8k to 2M words. It includes the following categories: single-document QA, multi-document QA, long in-context learning, long-dialogue history understanding, code repository understanding, and long structured data understanding.

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. For the RL stage, we use GRPO (Shao et al., 2024) as implemented in the Verl framework (Sheng et al., 2025). 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 and KL regularization with $\beta = 0.001$. For all experiments, including the baselines, we set the maximum number of output tokens to 16,384.

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. The maximum number of output tokens is set to 32,768, except for MATH-500, AIME 24, and AIME 25, where it is set to 40,000. For each question, we sample N times and report the average accuracy as the final score, using $N = 64$ for AIME 24 and AIME 25, and $N = 10$ for the remaining datasets.

Metrics. We report both accuracy and response length (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 token threshold t and the y-axis represents the corresponding OAA_t score, as illustrated in Figure 1.

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

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

Baselines. We compare our methods with existing state-of-the-art efficient reasoning approaches. We select a representative set of methods to cover a broad range of techniques:

- **No Thinking**: We disable thinking following the original Qwen3 paper (Yang et al., 2025a).
- **First Answer Truncation**: We truncate the reasoning trace at the first occurrence of the correct answer, then allow the model to summarize the truncated trace before producing the final answer. If the model does not generate the correct answer, the trace remains untruncated. This is a strong baseline, as it requires access to the ground truth answer. We only report this baseline for math datasets.
- **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 predefined 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.

4 Experimental Results

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

We observe that our methods substantially reduce response length while maintaining accuracy on most datasets. The degree of reduction varies across tasks; however, even the less aggressive length-reduction variant, *Adaptive-Answer*, dominates other alternatives (i.e., there are almost no points in the orange area of the figure). More precisely, *Adaptive-Answer* dominates most methods on MATH-500, AIME 24, and CommonsenseQA, and is dominated in only 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 is dominated on LiveCodeBench and LongBenchv2. We attribute the smaller reduction in response length

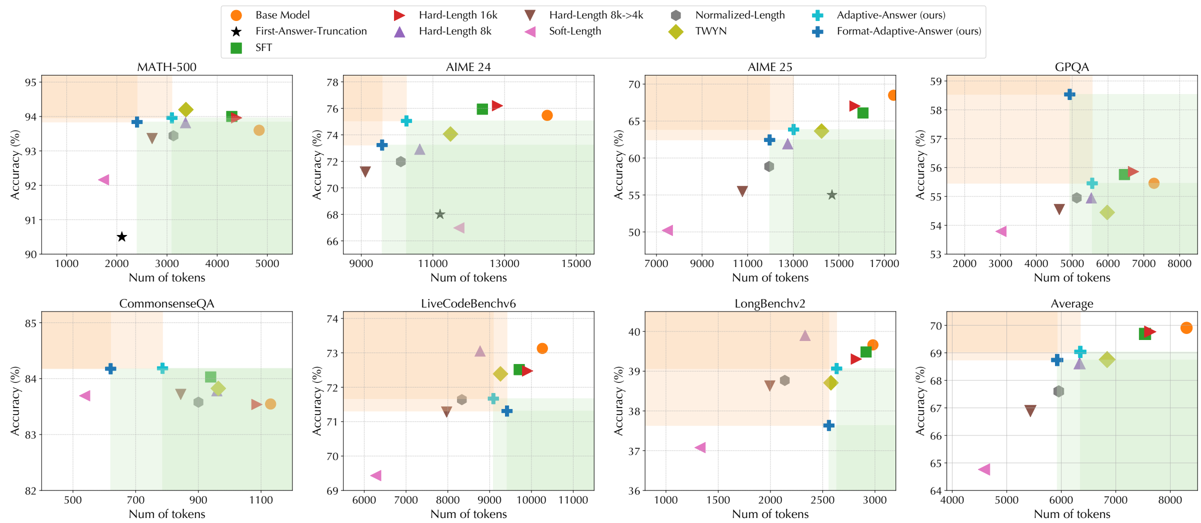

Figure 2: Average accuracy versus number of tokens for each method using Qwen3-8B. Points in the green region are dominated by *Adaptive-Answer* or *Format-Adaptive-Answer*, while points in the orange region dominate them (higher accuracy, fewer tokens).

	MATH-500	**AIME 24**	**AIME 25**	**GPQA Diamond**	**Common-sense QA**	**LiveCode-Bench**	**Long-Benchv2**	**Avg.**
Base model	81.7	68.6	75.8	<u>69.4</u>	72.1	**94.3**	39.4	71.6
No-Thinking	81.3	21.9	16.0	40.8	48.0	41.0	30.0	39.8
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	<u>94.1</u>	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 $\rightarrow$ 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	<u>78.6</u>	90.0	36.6	70.5
TWYN	89.6	74.5	<u>79.6</u>	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	<u>75.5</u>
Format-Adaptive-Answer (ours)	<u>91.2</u>	**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*.

across all efficient reasoning approaches on these two datasets to training primarily on math datasets. LiveCodeBench is the only dataset on which the base model outperforms all efficient reasoning methods. Although the performance gap is small, this suggests that fine-tuning exclusively on relatively short math problems may hurt performance on tasks that require processing long contexts.

The *First-answer-truncation* baseline is dominated by our methods on AIME 24 and AIME 25. This suggests that RL fine-tuning does more than simply remove unnecessary self-verification; it effectively shortens the model's reasoning traces.

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, response 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) with 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 only a one-point decrease in accuracy. Notably, even though training is performed only on math datasets, our methods also shorten responses on science, coding, QA, and long-context reasoning datasets.

	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*).

	Accuracy	#Tokens	AUC_{OAA}
Qwen3-1.7B			
Base Model	50.9	7,619	65.4
Adaptive-Answer	49.1	5,884 (-22%)	62.1
Format-Adaptive-Answer	48.3	5,918 (-22%)	62.1
Qwen3-8B			
Base Model	69.9	8,298	71.6
Adaptive-Answer	69.0	6,344 (-23%)	75.5
Format-Adaptive-Answer	68.7	5,918 (-28%)	76.6
Qwen3-32B			
Base Model	74.8	7,294	69.2
Adaptive-Answer	72.1	4,280 (-41%)	72.5
Format-Adaptive-Answer	72.2	4,372 (-40%)	72.3
DeepSeek-R1-Qwen-7B-Distill			
Base Model	50.3	6,272	62.1
Adaptive-Answer	50.4	5133 (18%)	59.6
Format-Adaptive-Answer	50.7	4,612 (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.

We emphasize that not all methods are directly comparable using accuracy and response length alone. 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 confirms the effectiveness of our methods: on average, *Format-Adaptive-Answer* outperforms all other methods, followed by *Adaptive-Answer*. *Format-Adaptive-Answer* achieves the highest score on all math and question answering datasets except MATH-500, where it ranks second. Interestingly, on LiveCodeBench and LongBenchv2, simple baselines such as *SFT* and *Hard-Length 16k*—equivalent to RL without a length penalty—outperform all other efficient reasoning alternatives.

Component Ablations. To evaluate the contribution of each component of our approach, we perform ablations for each component individually. 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*).

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

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

Finally, SFT is a crucial stage for RL performance: although *RL (no SFT)* produces the second-shortest reasoning traces, it incurs a performance penalty and achieves a lower AUC_{OAA} compared to the full approaches (*Adaptive-Answer* and *Format-Adaptive-Answer*).

Efficiency and Model Size. We investigate how our approaches scale with model size and training regimes. For this set of experiments, we evaluate Qwen3-{1.7B, 8B, 32B}, and DeepSeek-R1-Qwen-7B-Distilled.

Across all models, the performance drop after applying our methods remains under 2.5 points—and is negligible for DeepSeek-R1-7B (see Table 3). Notably, the reduction in generated tokens increases with model size: *Adaptive-Answer* shortens responses by 22% on Qwen3-1.7B, 23% on Qwen3-8B, and 40% on Qwen3-32B, demonstrating that

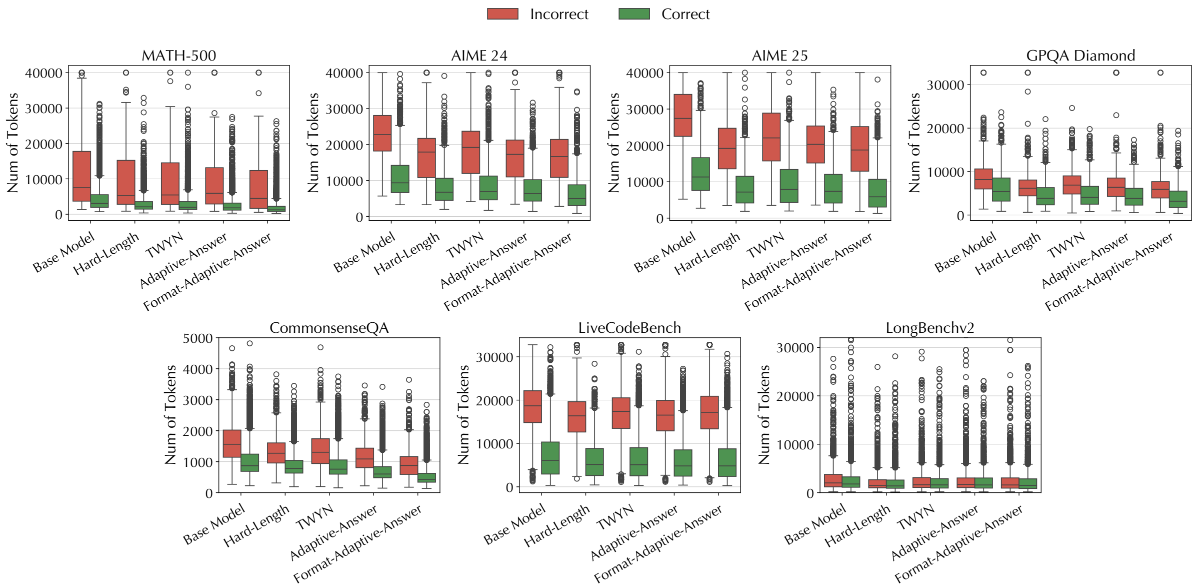

Figure 3: Response length distributions of some representative efficient reasoning methods applied to Qwen3-8B. We separate the correct and incorrect responses.

larger models benefit more from efficient reasoning. However, $\mathrm{AUC_{OAA}}$ does not always align perfectly with the accuracy–response length trade-offs, particularly for smaller models, highlighting that efficiency gains can sometimes come at a subtle cost to overall reasoning effectiveness. For instance, the fine-tuned DeepSeek-R1 dominates the base model in absolute accuracy but achieves a slightly lower $\mathrm{AUC_{OAA}}$ score. Overall, these results indicate that our methods are model-agnostic, consistently effective across different model sizes, and operating without significant performance loss.

5 Analysis

Response Length Distribution. Figure 3 shows the response length distributions for a representative set of efficient reasoning methods applied to Qwen3-8B. Across all three datasets, incorrect answers tend to have longer traces than correct ones, highlighting a correlation between excessive reasoning and errors. Importantly, our methods effectively shift the response length distribution for both correct and incorrect answers, showing that the models adapt their traces consistently regardless of the final answer. This indicates that our approach encourages concise reasoning across all outputs, not just the correct ones.

Intermediate Answers. The RL stage encourages the model to minimize unnecessary self-verification. To analyze this, we report the number of correct answers appearing in each reasoning trace for MATH-500, AIME 24, and AIME 25 (Figure 4). While this metric is a coarse proxy—since answers may be repeated or paraphrased—it provides qualitative insight into verification behavior. Both *Adaptive-Answer* and *Format-Adaptive-Answer* shift the distribution toward fewer intermediate correct answers, indicating reduced redundancy in reasoning.

Difficulty Analysis. We examine how problem difficulty affects accuracy, response length, and intermediate correct answers. Each MATH-500 problem is assigned a difficulty level from 1 to 5. As shown in Figure 5, our approaches maintain the base model's accuracy across all levels. Response length adapts to difficulty, increasing for harder problems and reflecting the need for more reasoning. Although both response length and intermediate correct answers rise with difficulty, they remain lower than those of the base model, demonstrating our methods' efficiency even on challenging problems.

Qualitative Analysis. Figure 6 compares reasoning traces for a math problem from AIME 24 produced by (*i*) the base model, (*ii*) *Adaptive-Answer*, and (*iii*) *Format-Adaptive-Answer* (long responses are trimmed). The base model performs seven unnecessary self-verifications after producing the first correct answer (*116*). In contrast, *Adaptive-Answer* reduces this to three, while

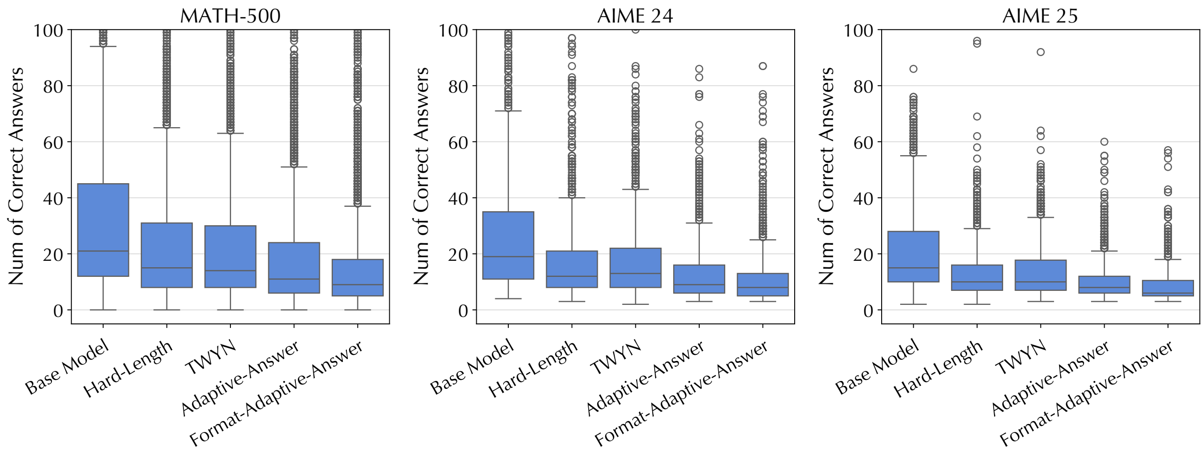

Figure 4: Distributions of the number of correct answers in the traces of some representative efficient reasoning methods applied to Qwen3-8B for MATH-500, AIME 24 and AIME 25.

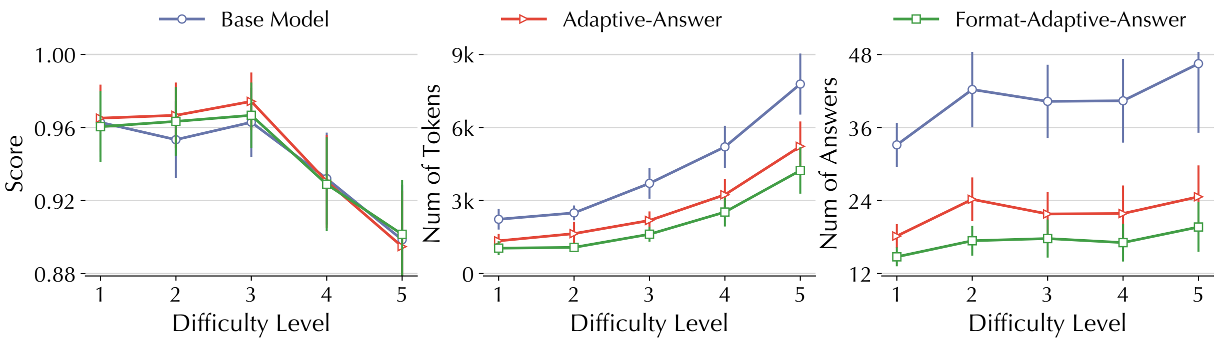

Figure 5: Accuracy, response length, and count of intermediate correct steps across difficulty levels on MATH-500.

Format-Adaptive-Answer produces an optimal trace with no self-verifications, directly generating the final answer without summarizing the reasoning.

There are multiple forms of self-verification. One involves verifying the solution using exactly the same approach, while another involves applying a different method in the hope of arriving at the same final answer. The latter is not merely repetitive behavior but rather a form of sanity checking. In Figure 6, the base model performs this second type of self-verification in its sixth attempt. We have observed that the trained model performs fewer such sanity checks, which may affect overall performance.

6 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., 2023; Snell et al., 2025). Models generate intermediate tokens in parallel—by sampling multiple traces (Wang et al., 2023)—or sequentially, by verifying and correcting their own outputs (Madaan et al., 2023; Kumar et al., 2025). Recent work uses reinforcement learning with verifiable rewards to further enhance reasoning capabilities, leading to longer CoT as well as self-verification and self-correction behaviors (OpenAI, 2024; DeepSeek-AI, 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., 2025; Yang et al., 2025c; Wu et al., 2026). Several methods have been proposed to address this issue (Zhu and Li, 2025). The most direct approach, budget forcing (Muennighoff et al., 2025; Yang et al., 2025a), interrupts generation once a predefined threshold is exceeded. Other methods construct synthetic datasets by shortening model-generated reasoning traces (via rejection sampling or pruning) and then perform SFT (Yang et al., 2025c; Xia et al., 2025; Cui et al., 2025).

Base Model

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

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 = $\boxed{\textbf{116}}$. Wait, but before I conclude, let me verify once more because sometimes with probability . . . Therefore, the answer is 1 + 115 = $\boxed{\textbf{116}}$. But just to make sure, let me think of another way . . . I think the answer is 116.**Final Answer**$\boxed{\textbf{116}}$</think> To solve the problem, we are asked to find the probability . . . These are relatively prime.$m + n = 1 + 115 = \boxed{\textbf{116}}$

Format-Adaptive-Answer

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

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.

A different line of work to which our method belongs uses reinforcement learning with a length penalty in addition to the correctness reward (Liu et al., 2025; Lou et al., 2025; Zhang et al., 2025; Tu et al., 2025). 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 a threshold (Yu et al., 2025; Aggarwal and Welleck, 2025). In both cases, a maximum budget needs to be manually set a priori. 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 approaches incentivize the model to infer an input-dependent budget, since not all problems require the same number of self-verifications.

7 Conclusion

Large language models (LLMs) often perform better on reasoning-intensive tasks by producing longer chains of thought. However, these chains are often unnecessarily long, increasing inference costs without improving accuracy. To address this, we propose a multi-stage efficient reasoning framework that consists of supervised fine-tuning—via rejection sampling or trace reformatting—followed by reinforcement learning with a reward that penalizes tokens generated after the first correct answer. Our approach effectively shortens response length (28% for Qwen3-8B and 40% for Qwen3-32B) with only minor performance drops (up to 2.5 accuracy points) and outperforms existing state-of-the-art efficient reasoning methods by 2.5 points when evaluated using the unified metric AUC_{OAA}.

Acknowledgments

We thank the anonymous reviewers for their helpful questions and comments, which helped us improve the quality of the paper. We also thank Ionut-Teodor Sorodoc, Gianni Barlacchi, Dennis Fucci, and Sebastian Steindl for their insightful discussions and valuable feedback throughout the project.

UPF was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101019291). This paper reflects the authors' view only, and the funding agency is not responsible for any use that may be made of the information it contains.

Limitations

Although we evaluate our methods on datasets from multiple domains, such as math, coding, science, and long-context reasoning, our training is performed exclusively on math datasets. This limits generalization to non-math datasets, and extending training to a more diverse set of tasks could yield a better accuracy–response length trade-off. Furthermore, the reward function relies on the presence of a correct answer in the reasoning trace, preventing its application to more open-ended or weakly verifiable tasks. Future work could modify the reward function to remove unnecessary self-verifications while broadening its applicability to a wider range of tasks. In addition, the efficient reasoning methods we propose are post hoc interventions; we do not explore incorporating the adaptive length penalty directly into the initial RL training. Due to resource constraints, we also limit our experimental scope to models of different sizes within the dense Qwen3 family, and future work could extend our approach to more model families and other architectures, such as mixture-of-experts models. Finally, we focus exclusively on performance on reasoning tasks and do not measure changes in performance on task groups that are less dependent on 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, it may also reduce the readability of the reasoning traces.

References

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason E. Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. OptimalThinkingBench: Evaluating Over and Underthinking in LLMs. In *NeurIPS 2025 Workshop on Efficient Reasoning*, NeurIPS '25 Workshop, New Orleans, Louisiana, USA.

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling How Long a Reasoning Model Thinks with Reinforcement Learning. In *Proceedings of the Second Conference on Language Modeling*, COLM '25, Montreal, Canada.

AIME. 2025. AIME Problems and Solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Amazon AGI. 2025. Amazon Nova 2: Multimodal reasoning and generation models. *Amazon Technical Reports*.

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. 2025. LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-Context Multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '25, pages 3639–3664, Vienna, Austria.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Do NOT Think That Much for 2+3=? On the Overthinking of Long Reasoning Models. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML '25, Vancouver, Canada.

Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, Suhang Wang, Yue Xing, Jiliang Tang, and Qi He. 2025. Stepwise Perplexity-Guided Refinement for Efficient Chain-of-Thought Reasoning in Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2025*, Findings '25, pages 18581–18597, Vienna, Austria.

DeepSeek-AI. 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, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2025. Omni-MATH: A Universal Olympiad Level Mathematic Benchmark for Large Language Models. In *The Thirteenth International Conference on Learning Representations*, ICLR '25, Singapore, Singapore.

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. *Transactions on Machine Learning Research*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. In *The Thirteenth International Conference on Learning Representations*, ICLR '25, Singapore, Singapore.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2025. Training Language Models to Self-Correct via Reinforcement Learning. In *The Thirteenth International Conference on Learning Representations*, ICLR '25, Singapore, Singapore.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations*, ICLR '24, Vienna, Austria.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025. Learn to Reason Efficiently with Adaptive Length-Based Reward Shaping. *arXiv preprint arXiv:2505.15612*.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. 2025. AdaCoT: Pareto-Optimal Adaptive Chain-of-Thought Triggering via Reinforcement Learning. *arXiv preprint arXiv:2505.11896*.

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, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems 36*, volume 36 of *NeurIPS '23*, New Orleans, Louisiana, USA.

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. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, EMNLP '25, pages 20275–20321, Suzhou, China.

OpenAI. 2024. OpenAI o1 System Card. *arXiv preprint arXiv:2412.16720*.

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`.

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 *Proceedings of the First Conference on Language Modeling*, COLM '24, Philadelphia, Pennsylvania, USA.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. 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. 2025. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pages 1279–1297, Rotterdam, Netherlands.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *The Thirteenth International Conference on Learning Representations*, ICLR '25, Singapore, Singapore.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, NAACL '19, pages 4149–4158, Minneapolis, Minnesota, USA.

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*.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. 2025. Learning When to Think: Shaping Adaptive Reasoning in R1-Style Models via Multi-Stage RL. In *The*

Thirty-Ninth Annual Conference on Neural Information Processing Systems, NeurIPS '25, San Diego, California, USA.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*, ICLR '23, Kigali, Rwanda.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems 35*, volume 35 of *NeurIPS '22*, New Orleans, Louisiana, USA.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2026. When More is Less: Understanding Chain-of-Thought Length in LLMs. In *The Fourteenth International Conference on Learning Representations*, ICLR '26, Kigali, Rwanda.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. TokenSkip: Controllable Chain-of-Thought Compression in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, EMNLP '25, pages 3351–3363, Suzhou, China.

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. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, NeurIPS '25, San Diego, California, USA.

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*.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025. AdaptThink: Reasoning Models Can Learn When to Think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, EMNLP '25, pages 3696–3715, Suzhou, China.

Jason Zhu and Hongyu Li. 2025. Towards concise and adaptive thinking in large reasoning models: A survey. *arXiv preprint arXiv:2507.09662*.

A Accuracy–Response Length Trade-off

Table 4 quantifies the accuracy–length trade-off. Tight hard-length constraints reduce average response length from 8.3k tokens (Base Model) to 5.4k, but incur a 3.0-point average accuracy drop and a severe degradation on AIME 25 (68.5 → 55.5). Normalized-Length achieves the shortest outputs (4.6k tokens on average) but suffers the largest performance loss (69.9 → 64.8). In contrast, adaptive methods preserve accuracy more effectively: Adaptive-Answer reduces average length by 23.5% (8.3k → 6.3k) with only a 0.9-point accuracy decrease, while Format-Adaptive-Answer achieves a 28.6% reduction (5.9k tokens) with a 1.2-point drop. Among all efficient reasoning strategies, our proposed methods consistently occupy the Pareto-optimal region, yielding the best overall accuracy–efficiency trade-offs across benchmarks. These results indicate that instance-level length adaptation yields a substantially better efficiency–accuracy trade-off than fixed or normalized constraints.

AI use disclosure: We used AI to assist with code writing and manuscript typesetting.

Model	**MATH-500**		**AIME 24**		**AIME 25**		**GPQA Diamond**		**Common-senseQA**		**LiveCode-Bench**		**Long-Benchv2**		**Average**	
	Acc.↑	#Tok.↓	Acc.↑	#Tok.↓	Acc.↑	#Tok.↓	Acc.↑	#Tok.↓	Acc.↑	#Tok.↓	Acc.↑	#Tok.↓	Acc.↑	#Tok.↓	Acc.↑	#Tok.↓
Base Model	93.6	4,837	75.5	14,191	**68.5**	17,402	55.5	7,284	83.5	1,130	**73.1**	10,261	39.7	2,980	**69.9**	8,298
SFT	94.0	4,292	75.9	12,381	66.1	16,058	55.8	6,459	84.0	939	72.5	9,708	39.5	2,915	69.7	7,536
Hard-Length 16k	94.0	4,388	**76.2**	12,803	67.0	15,707	55.9	6,712	83.5	1,087	72.5	9,909	39.3	2,822	69.8	7,633
Hard-Length 8k	93.8	3,369	72.9	10,633	61.9	12,758	54.9	5,535	83.8	959	**73.1**	8,772	**39.9**	2,332	68.6	6,337
Hard-Length 8k → 4k	93.4	2,703	71.2	**9,114**	55.5	10,770	54.5	4,642	83.7	844	71.3	7,972	38.6	1,994	66.9	5,434
Soft-Length	93.4	3,129	72.0	10,105	58.9	11,944	54.9	5,128	83.6	900	71.6	8,335	38.8	2,137	67.6	5,954
Normalized-Length	92.2	**1,734**	67.0	11,723	50.2	**7,475**	53.8	**3,011**	83.7	**537**	69.4	**6,273**	37.1	**1,326**	64.8	**4,583**
TWYN	**94.2**	3,377	74.1	11,491	63.6	14,243	54.4	5,978	83.8	964	72.4	9,259	38.7	2,579	68.8	6,841
Adaptive-Answer	94.0	3,098	75.1	10,261	63.9	13,017	55.5	5,560	**84.2**	786	71.7	9,089	39.1	2,634	69.0	6,349
Format-Adaptive-Answer	93.8	2,403	73.2	9,583	62.4	11,965	**58.5**	4,931	**84.2**	620	71.3	9,416	37.6	2,559	68.7	5,925

Table 4: Accuracy (Acc.↑) and response length (#Tok.↓) of all approaches applied to Qwen3-8B. Best values per column are in bold.