

Advanced reasoning & Test-time scaling

CS 5624: Natural Language Processing
Spring 2025

<https://tuvllms.github.io/nlp-spring-2025>

Tu Vu



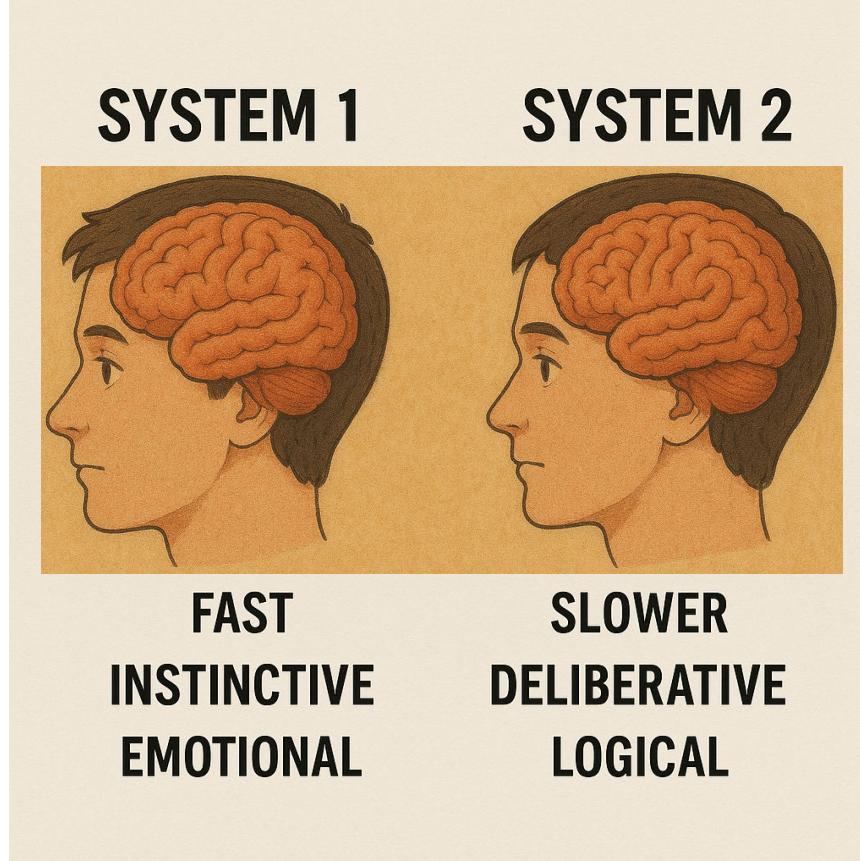
THINKING,
FAST AND SLOW



DANIEL
KAHNEMAN

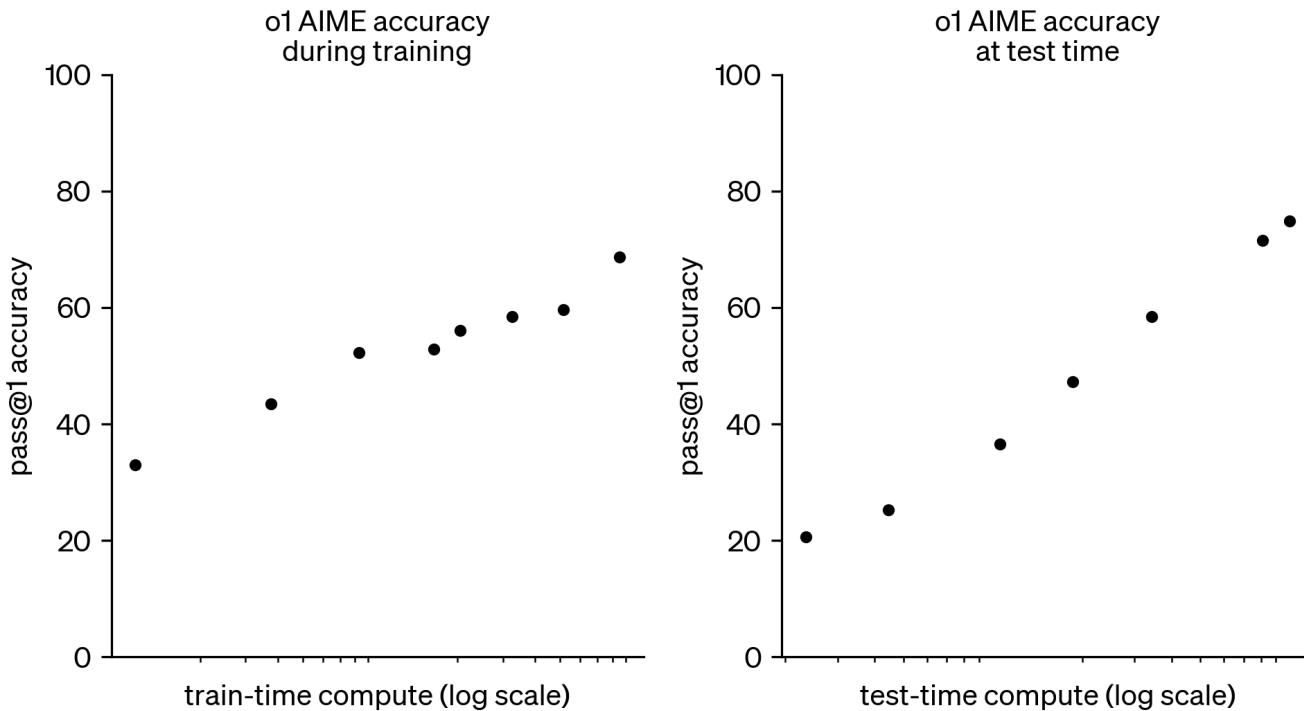
WINNER OF THE NOBEL PRIZE IN ECONOMICS

System 1 & System 2

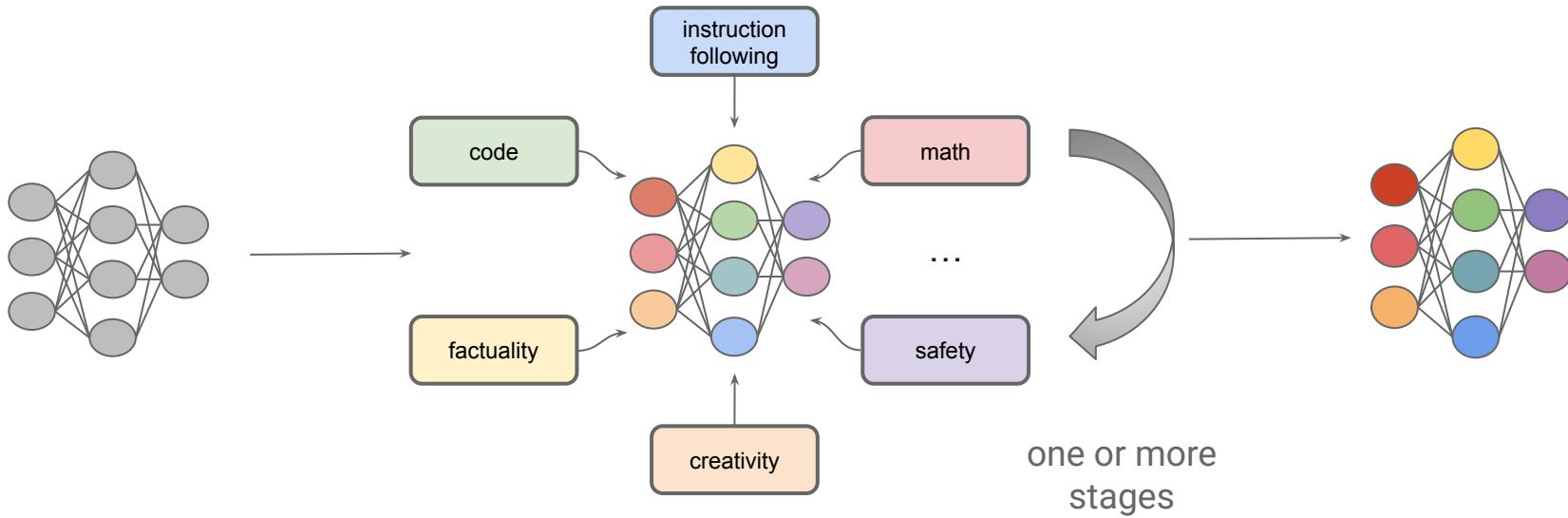


GPT-4o
image

o1 performance smoothly improves with both train-time and test-time compute



From System 1 to System 2



System 1

Supervised Fine-tuning and/or
Reinforcement Learning on long
Chain-of-Thought data

System 2



DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

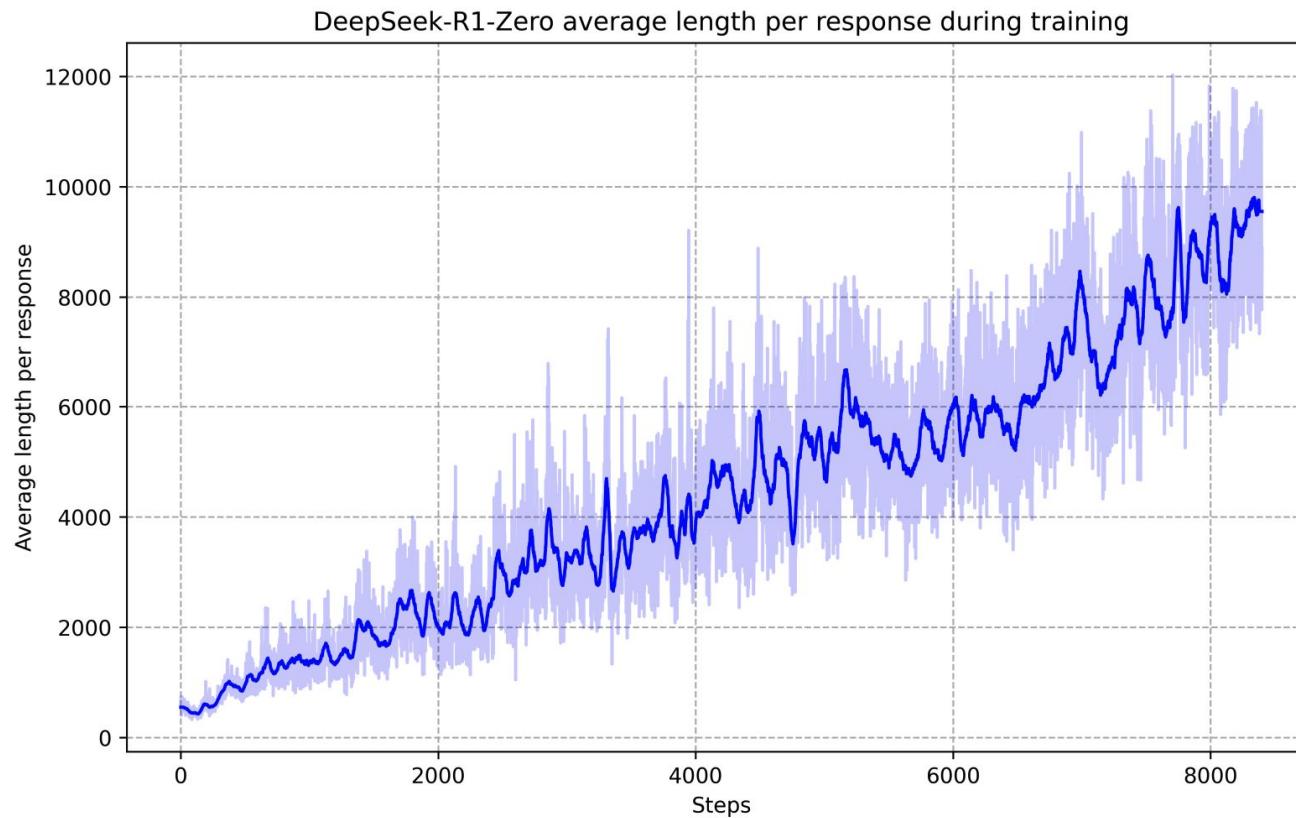
DeepSeek-AI

research@deepseek.com

Guided Chain-of-Thought (CoT) template

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think> </think>` and `<answer> </answer>` tags, respectively, i.e., `<think> reasoning process here </think>` `<answer> answer here </answer>`. User: **prompt**. Assistant:

DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time



An interesting “aha moment” of DeepSeek-R1-Zero

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a+x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a+x}} = x$, let's start by squaring both \dots

$$(\sqrt{a - \sqrt{a+x}})^2 = x^2 \implies a - \sqrt{a+x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

\dots

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be \dots

We started with the equation:

$$\sqrt{a - \sqrt{a+x}} = x$$

First, let's square both sides:

$$a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$$

Next, I could square both sides again, treating the equation: \dots

\dots

PPO (Proximal Policy Optimization) vs. GRPO (Group Relative Policy Optimization)

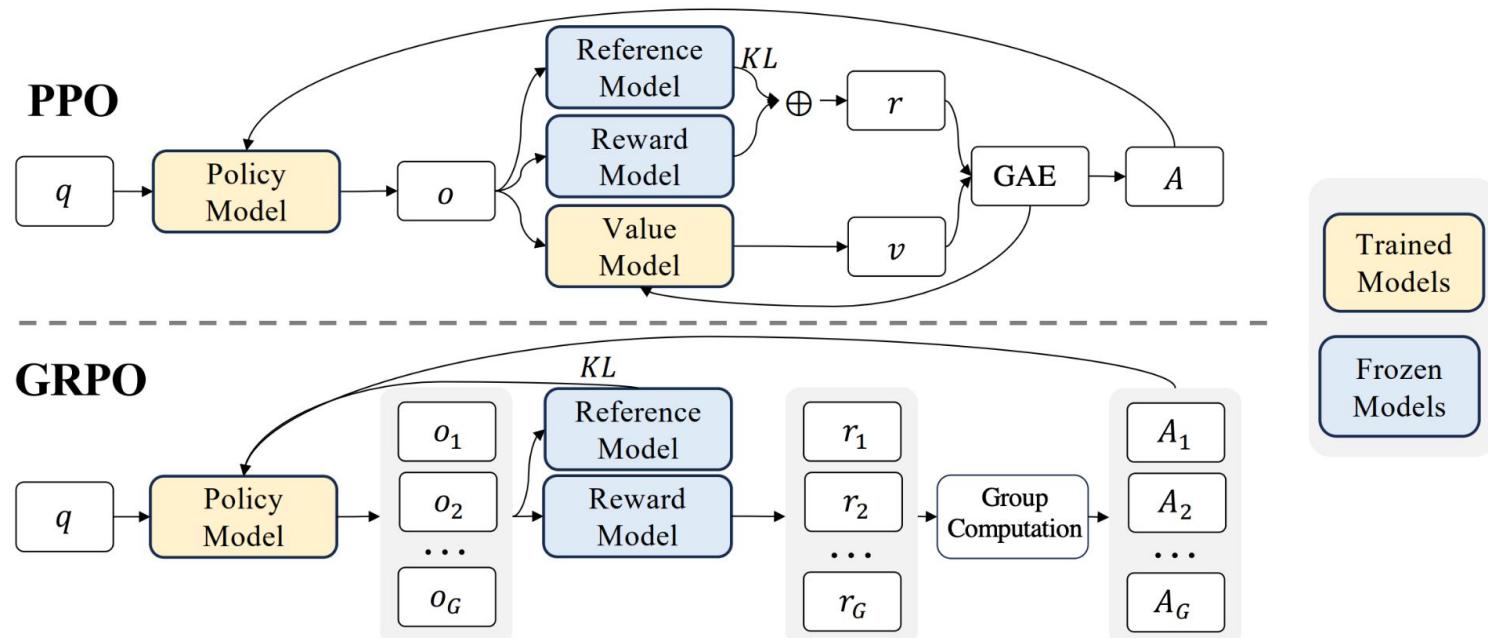


Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

Group Relative Policy Optimization In order to save the training costs of RL, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which foregoes the critic model that is typically the same size as the policy model, and estimates the baseline from group scores instead. Specifically, for each question q , GRPO samples a group of outputs $\{o_1, o_2, \dots, o_G\}$ from the old policy $\pi_{\theta_{old}}$ and then optimizes the policy model π_θ by maximizing the following objective:

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

$$\mathbb{D}_{KL}(\pi_\theta || \pi_{ref}) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \quad (2)$$

where ε and β are hyper-parameters, and A_i is the advantage, computed using a group of rewards $\{r_1, r_2, \dots, r_G\}$ corresponding to the outputs within each group:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})}. \quad (3)$$

<https://unsloth.ai/blog/r1-reasoning>



unsloth

About

Blog

Contact

Documentation

Easily finetune & train LLMs
Get *faster* with unsloth

 Join our Discord

Start for free



LIMO: Less is More for Reasoning

Yixin Ye* **Zhen Huang*** **Yang Xiao** **Ethan Chern** **Shijie Xia** **Pengfei Liu[†]**

SJTU, SII, GAIR

Superficial Alignment Hypothesis

- LIMA: Less is more for alignment ([Zhou et al., 2023](#))
 - 1,000 examples can be sufficient
- LIMO: even competition-level complex reasoning abilities can be effectively elicited through minimal but curated training samples
- LIMO: a promising technical pathway toward AGI - any sophisticated reasoning capability, no matter how complex, could potentially be activated with minimal samples given two key conditions:
 - (1) sufficient domain knowledge embedded during pre-training
 - (2) optimal cognitive reasoning chains for activation

Categorizing the reasoning chains into five

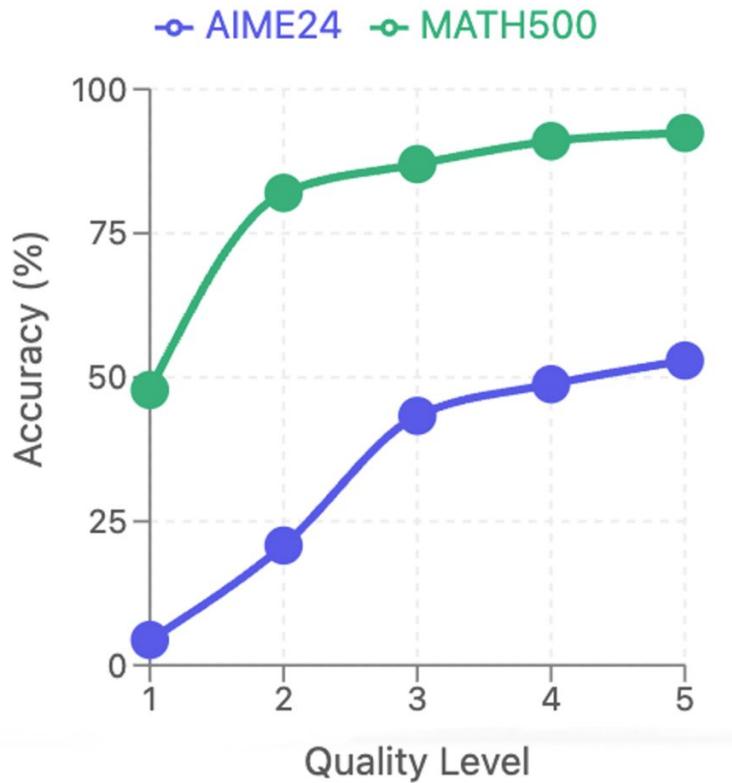
How well the reasoning steps were organized, whether important logical transitions were properly explained, and if the solution included self-verification steps

- **L5:** excellent organization with clear, well-explained steps and thorough self-verification
- **L4:** well-structured but perhaps with slightly less rigorous checking
- **L3:** decent organization but sometimes skipped over explaining crucial logical leaps
- **L2:** often provided abbreviated reasoning without much explanation
- **L1:** just listed basic steps with minimal elaboration and rarely included any verification

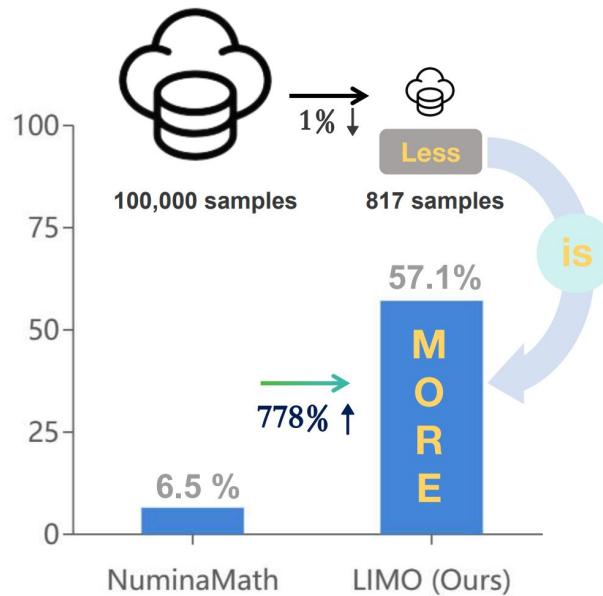
Statistical analysis of different quality levels

Data Quality Level	Avg. Tokens per response	Avg. Lines per response	Top 10 Frequently Occurring Keywords (in order)
Level 1	230	9.21	since, however, number, let, thus, which, get, two, triangle, theta
Level 2	444.88	50.68	number, need, times, which, find, list, thus, since, triangle, sum
Level 3	4956.11	375.60	perhaps, alternatively, consider, number, wait, which, sides, need, equal, seems
Level 4	4726.97	354.87	wait, which, number, perhaps, therefore, let, since, maybe, sides, two
Level 5	5290.26	239.29	wait, therefore, which, number, since, lets, two, sides, let, maybe

Comparison of models trained on reasoning chains of different quality levels

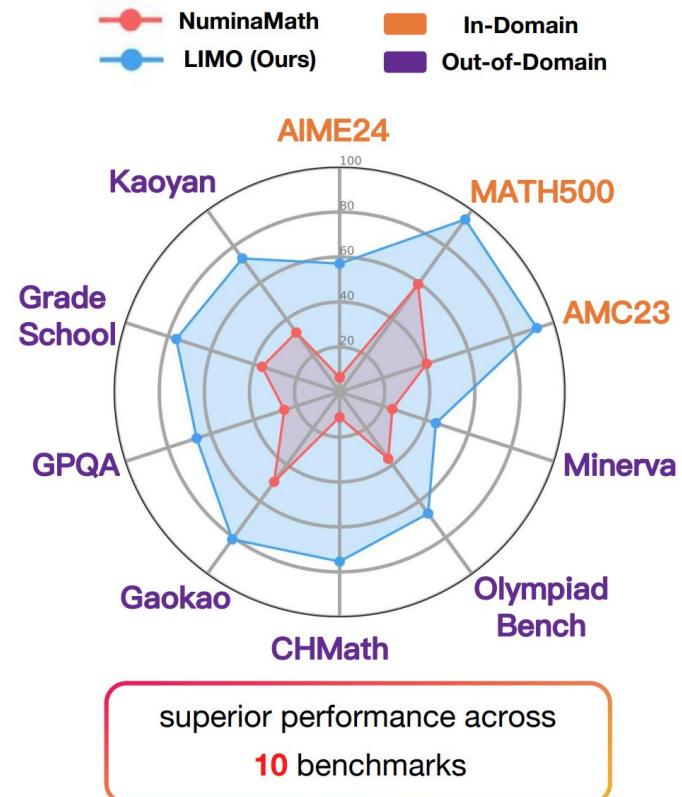


LIMO achieves substantial improvement over NuminaMath with fewer samples



completely **same** backbone
1% data → **778%** gain on AIME24 (pass@1)

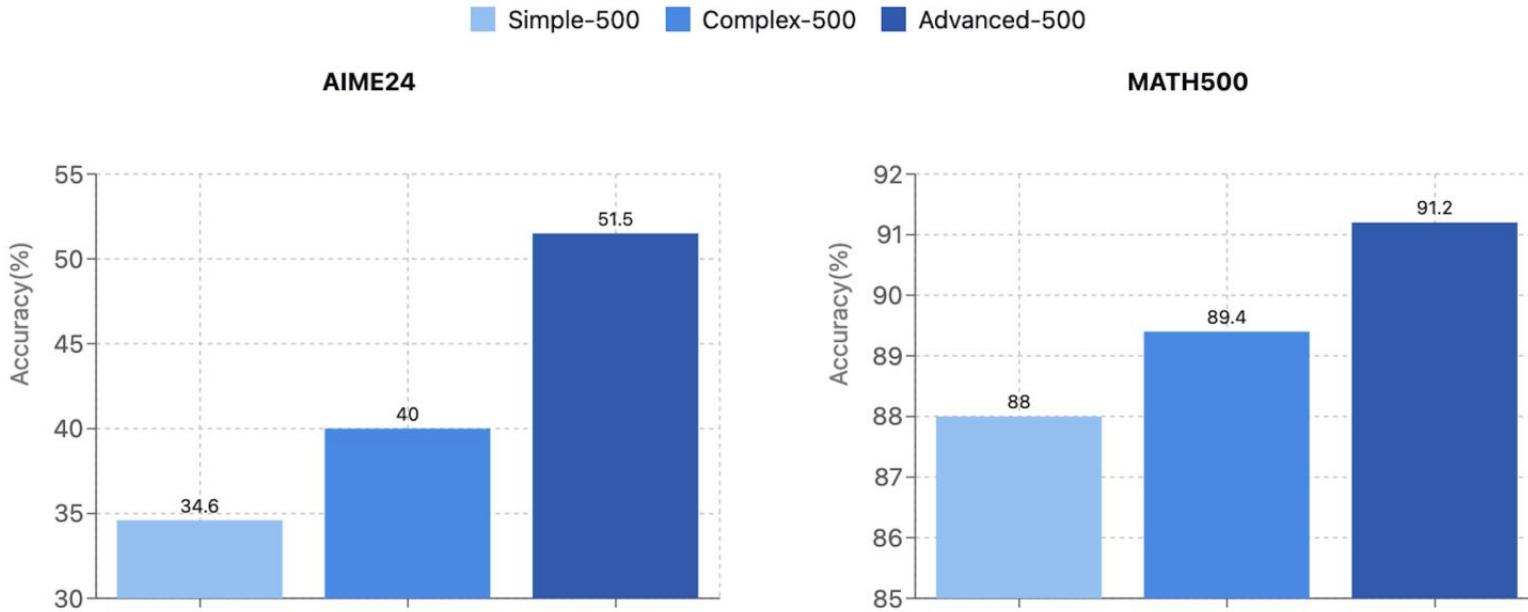
... while excelling across diverse mathematical and multi-discipline benchmarks



LIMO achieves superior performance despite using significantly fewer training examples

Datasets	OpenAI-o1 -preview	Qwen2.5-32B -Instruct	QwQ-32B- preview	OpenThoughts (114k)	NuminaMath (100k)	LIMO ours(817)
In Domain						
AIME24	44.6	16.5	50.0	50.2	6.5	57.1
MATH500	85.5	79.4	89.8	80.6	59.2	94.8
AMC23	81.8	64.0	83.6	80.5	40.6	92.0
Out of Domain						
OlympiadBench	52.1	45.3	58.5	56.3	36.7	66.8
CHMath	50.0	27.3	68.5	74.1	11.2	75.4
Gaokao	62.1	72.1	80.1	63.2	49.4	81.0
Kaoyan	51.5	48.2	70.3	54.7	32.7	73.4
GradeSchool	62.8	56.7	63.8	39.0	36.2	76.2
Minerva	47.1	41.2	39.0	41.1	24.6	44.9
GPQA	73.3	48.0	65.1	42.9	25.8	66.7
AVG.	61.1	49.9	66.9	58.3	32.3	72.8

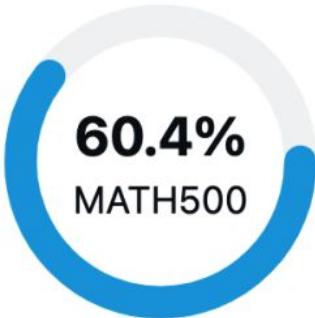
Models trained on different question quality



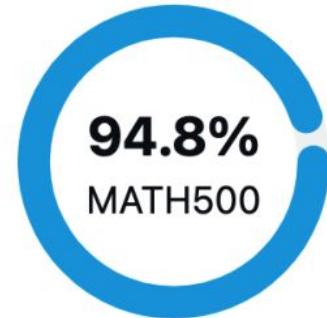
- **Simple-500:** 500 simple problems randomly selected problems from MATH levels 1 and 2.
- **Complex-500:** 500 complex problems randomly selected problems from MATH levels 3, 4, and 5.
- **Advanced-500:** 500 advanced problems randomly selected problems from past AIME tests.

Impact of pretrained model choice

Qwen1.5-32B-Chat



Qwen2.5-32B-Instruct



Example model outputs

Problem: Every morning Aya goes for a 9 km long walk and stops at a coffee shop afterwards. When she walks at a constant speed of s kilometers per hour, the walk takes her 4 hours, including t minutes spent in the coffee shop. When she walks $\$s+2\$$ kilometers per hour, the walk takes her 2 hours and 24 minutes, including $\$t\$$ minutes spent in the coffee shop. Suppose Aya walks at $\$s+\frac{1}{2}\$$ kilometers per hour. Find the number of minutes the walk takes her, including the $\$t\$$ minutes spent in the coffee shop.

Qwen2.5-32B-Instruct

To solve the problem, we need to determine the time Aya spends walking [omitted for brevity...]

The time spent walking is the distance divided by the speed, so the time spent walking is $(9/s)$ hours. Therefore, the equation for the total time is: $9/s + t = 240$, where t is the time spent in the coffee shop in minutes.

Next, we know that [omitted for brevity...]. The time spent walking is $9/(s+2)$ hours. Therefore, the equation for the total time is: $9/(s+2) + t = 144$.

We now have two equations:

$$1. 9/s + t = 240$$

$$2. 9/(s+2) + t = 144$$

[omitted for brevity...]

So, $s = 3 / 16$ km/h. Now we substitute $s = 3 / 16$ km/h back into equation (1) to find t :

$$9 / (3 / 16) + t = 240, 48 + t = 240, t = 192$$

[omitted for brevity...]

So, the number of minutes the walk takes her, including the t minutes spent in the coffee shop, is 205.

- unable to correct false statements above
- Did not cross verify equation 2

DeepSeek-R1

Okay, let's try to figure out this problem step by step. So, Aya goes for a 9-kilometer walk every morning and then stops at a coffee shop. [omitted for brevity...]

First, let's parse the problem. Let's denote the walking [omitted for brevity...]. Our goal is to find the total time when she walks at $s+1/2$ km/h. Hmm, so we need to figure out the values of s and t first. Let me write down the equations based on the given information. [omitted for brevity...]

Now we have two equations:

$$1) 9/s + t/60 = 4, 2) 9/(s+2) + t/60 = 2.4$$

We can subtract the second equation from the first to eliminate [omitted for brevity...]

So the left side becomes: [omitted for brevity...]

Multiply both sides by: [omitted for brevity...]

Now, solving this quadratic equation [omitted for brevity...]. So, Aya's original speed is 2.5 km/h.

Now, let's find t [omitted for brevity...]. **Wait, let me confirm.** But wait, the question says [omitted for brevity...]. Therefore, the answer should be 204 minutes.

But let me just check again to make sure I didn't make any mistakes in the quadratic equation solving. Let's verify that $s = 2.5$ is correct. [omitted for brevity...]. Thus, the answer should be 204.

think with anthropomorphic tone

detailed steps in calculation

self-reflection

LIMO

Okay, let's see. So Aya goes for a 9-km walk every morning and then stops at a coffee shop. The time she takes for the walk, including the time in the coffee shop, depends on her walking speed. [omitted for brevity...]. When she walks a bit faster, at $s + 2$ kilometers per hour, the total time is 2 hours and 24 minutes, which is 2 hours and 24 minutes, so that's 2.4 hours?

Wait, 24 minutes is 0.4 hours? Wait, no. Wait, 60 minutes is 1 hour, so 24 minutes is 24/60, which is 0.4 hours. So 2 hours and 24 minutes is 2.4 hours. [omitted for brevity...]

So, let's set up equations. Let's denote the walking time as T_1 when she walks at s km/h, and T_2 when she walks at $s+2$ km/h. Then, $T_1 + t = 4$ hours, and $T_2 + t = 2.4$ hours. **But t is in minutes, so we need to convert that to hours or convert the total time to minutes. Maybe it's better to keep everything in hours.** So t minutes is $t/60$ hours. [omitted for brevity...]

For the first case: walking time $T_1 = 9/s$ hours, then total time $T_1 + t/60 = 4$ hours.

For the second case: walking time $T_2 = 9/(s+2)$ hours, then total time $T_2 + t/60 = 2.4$ hours. So now we have two equations: 1) $9/s + t/60 = 4$, 2) $9/(s+2) + t/60 = 2.4$.

Let me write that as:

$$9*(1/s - 1/(s+2)) = 1.6. \text{ So, } 1/s - 1/(s+2) = 1.6/9$$

Let me compute 1.6 divided by 9. 1.6 divided by 9 is 0.1777... or 16/90, which simplifies to 8/45. So $1.6/9 = 8/45 \approx 0.1777\ldots$ So, $1/s - 1/(s+2) = 8/45$

Now, let's compute the left side: [omitted for brevity...]. Multiply both sides by 2: [omitted for brevity...]. Let me write that as a quadratic equation: [omitted for brevity...]. Now, let's solve this quadratic equation: [omitted for brevity...]. So, s is 2.5 km/h.

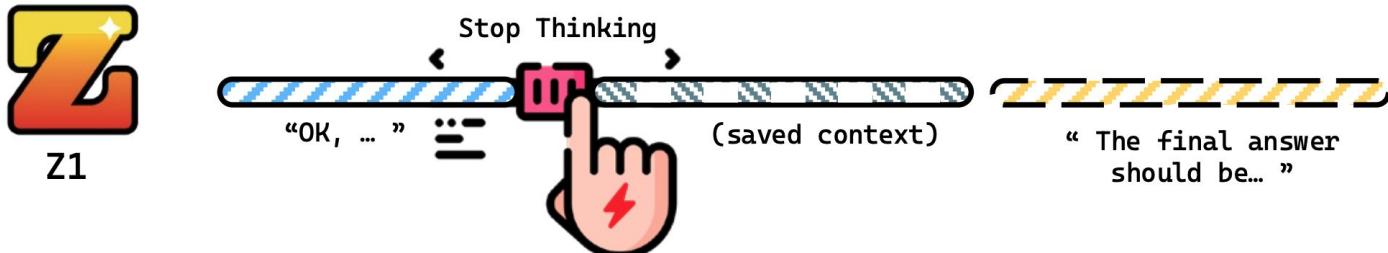
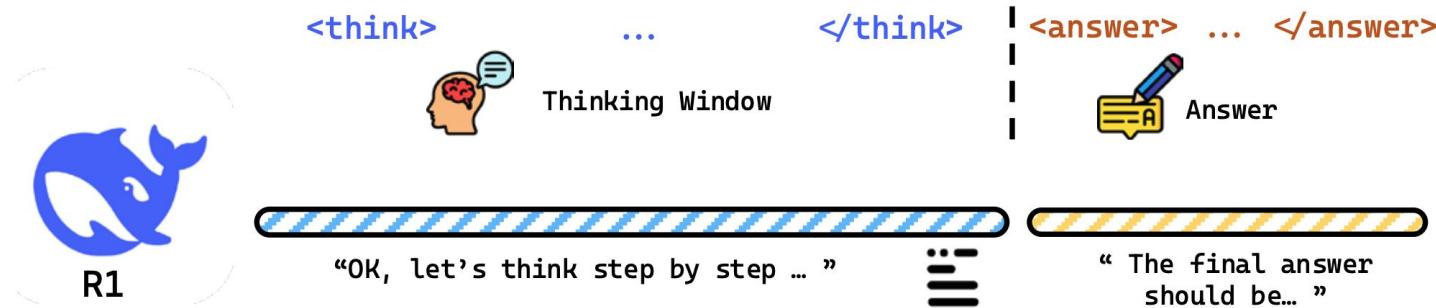
Let's verify that. If s is 2.5 km/h, then walking 9 km would take $9/2.5 = 3.6$ hour [omitted for brevity...]

But let me check again. Maybe I made a mistake in calculations. First, $s = 2.5$ km/h. Then, walking time at s is $9/2.5 = 3.6$ hours...[omitted for brevity...]. **Alternatively, we could have converted everything to minutes from the start.** Let's see: First walk: $9/(s)$ hours + t minutes = 4 hours. 4 hours is 240 minutes. [omitted for brevity...]. I think that's solid. So the answer is 204.

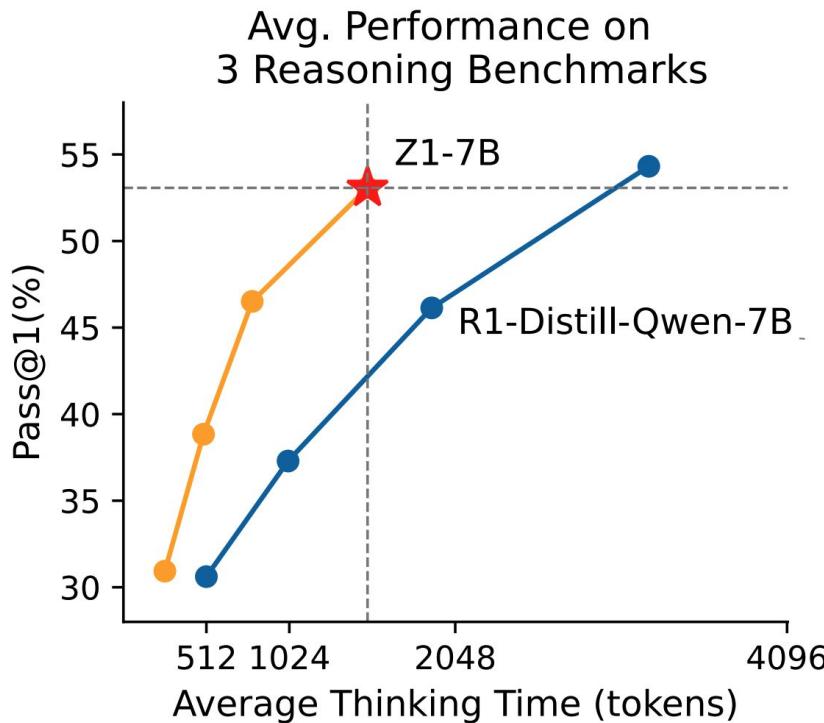
Z1: Efficient Test-time Scaling with Code

Zhaojian Yu¹, Yinghao Wu¹, Yilun Zhao², Arman Cohan², Xiao-Ping Zhang¹
¹Tsinghua University ²Yale University

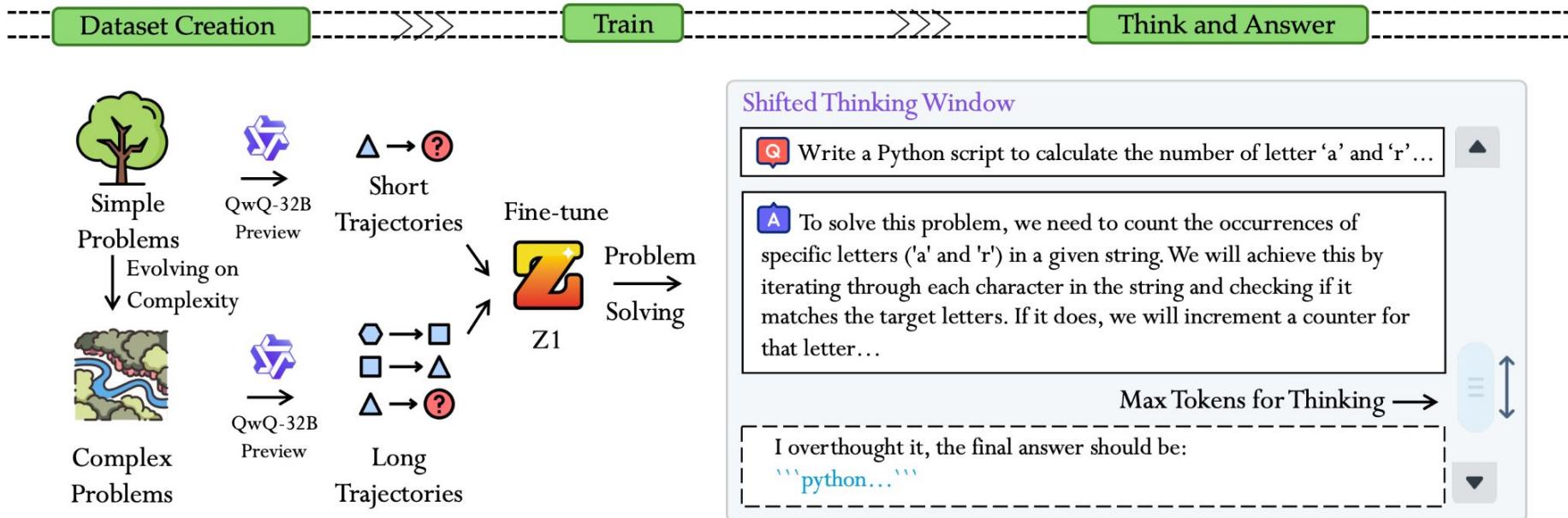
Z1 uses a shifted thinking window



Z1-7B matches R1-Distill-Qwen-7B performance with about 30% of its average thinking tokens



Fine-tuned with long and short trajectory data, Z1 could solve simple and complex problems in shifted thinking window efficiently



Z1 exhibits efficient test-time compute scaling

Model	Data Source	MATH 500	GPQA Diamond	LiveCode Bench	BigCode Bench-Hard	Avg
API only						
o1-preview	N/A	85.5	73.3	43.2	23.0	56.3
o1-mini	N/A	90.0	60.0	53.7	27.7	57.9
Open Weights						
Deepseek-R1	N/A	97.3	71.5	77.9	29.7	67.6
R1-Distill-Qwen-32B	R1/800K	94.3	62.1	-	23.6	-
R1-Distill-Qwen-7B	R1/800K	83.3	49.1	40.5	3.4	44.1
QwQ-32B-Preview	N/A	90.6	60.0	59.9	25.0	58.9
Non-reasoning Model						
Deepseek-V3	N/A	90.2	59.1	56.3	27.7	58.3
GPT-4o-0513	N/A	75.8	46.5	43.4	25.0	47.7
Qwen2.5-Coder-7B-Ins	N/A	68.6	37.4	32.3	20.3	39.7
Open Weights and Data						
Sky-T1-32B-Preview	QwQ/17K	88.6	56.8	-	26.4	-
s1.1-7B	R1/1K	79.2	31.8	15.2	4.7	31.7
OpenThinker-7B	R1/114K	83.0	42.4	25.3	17.6	42.1
Z1-7B	QwQ/107K	76.4	47.5	35.3	22.3	45.4

Z1-Code-Reasoning-107K data ablations

Subset	Full	Random	Longest	Shortest
Training Dataset				
Dataset Size (Samples)	107K	16K	64K	33K
Dataset Size (Tokens)	124M	19M	74M	74M
Mean Trajectory Length	1,159	1,157	1,156	2,216
Evaluation				
GPQA Diamond	47.5	40.9	41.9	42.4
<i>Average Thinking Time</i>	2,470	1,797	2,241	2,695
LiveCode Bench	35.3	32.2	34.1	32.7
<i>Average Thinking Time</i>	866	864	811	927
MATH 500	76.4	72.4	74.4	77.2
<i>Average Thinking Time</i>	1,185	1,046	1,118	1,229
AVG	53.1	48.5	50.1	50.8
<i>Average Thinking Time</i>	1,507	1,236	1,390	1,617

Example model outputs

Write a Python script to calculate the number of letter 'a' and 'r' in a string.

Input:

s = 'strawberry'

Output:

'a' : 1, 'r' : 3

[Qwen2.5-Coder-7B-Ins]

Here is a Python script that can solve the problem:

```
def count_letters(s):
    a_count = s.count('a')
    r_count = s.count('r')
    return {'a': a_count,
            'r': r_count}

s = 'strawberry'
print(count_letters(s))
```

In this script, we define a function `count_letters`...
[120+ tokens omitted]

[Z1-7B]

To solve this problem, we need...
[50+ tokens omitted]

1. Approach

[94 tokens omitted]

2. Solution Code

```
def count_letters(s):
    count_a = 0
    count_r = 0
    for char in s:
        if char == 'a':
            count_a += 1
        elif char == 'r':
            count_r += 1
    print("'a' :", count_a)
    print("'r' :", count_r)
```

3.Explanation

[104 tokens omitted]

This approach ensures that we...

[28 tokens omitted]

[R1-Distill-Qwen-7B]

<think>
Okay, I need to write a Python script that counts the number of 'a's and 'r's...
[1787 tokens omitted]

</think>
To solve this problem...

[45 tokens omitted]

1. Approach

[92 tokens omitted]

2. Solution Code

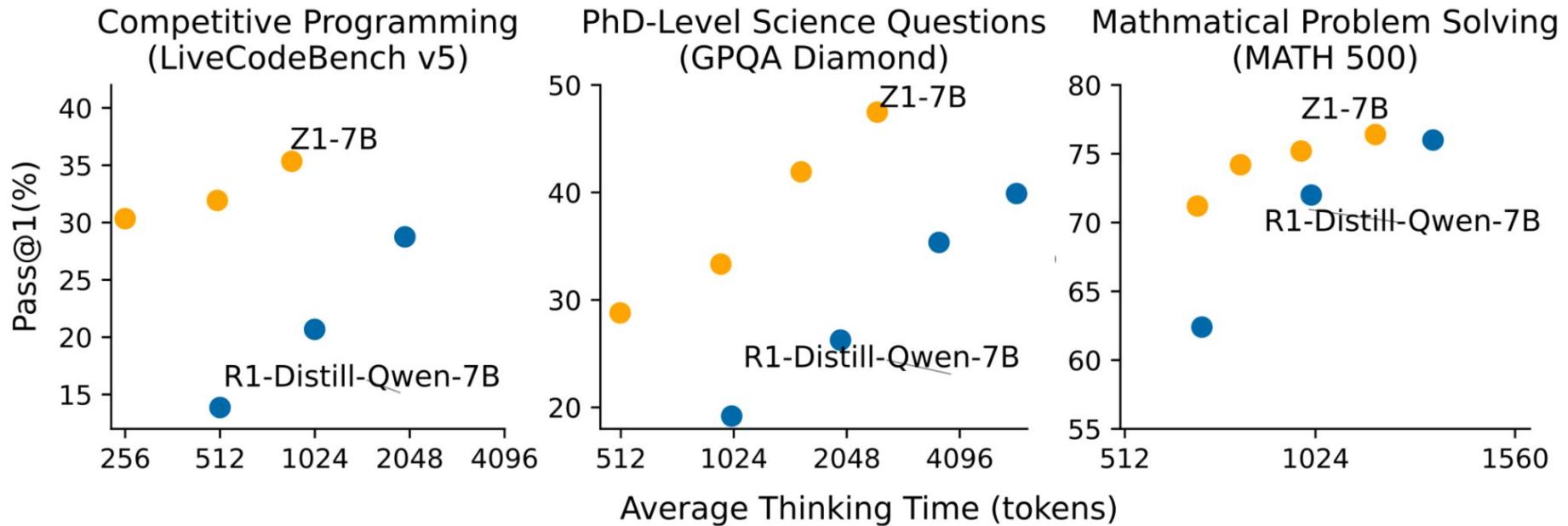
```
s = input().strip()
a_count = s.count('a')
r_count = s.count('r')
print(f'a: {a_count}')
print(f'r: {r_count}')
```

3.Explanation

[100+ tokens omitted]

This approach ensures that the solution is...
[22 tokens omitted]

Test-time scaling comparison between Z1-7B and R1-Distill-Qwen-7B



Thank you!