

A Selection of α in Stage I (Model Merge)

To select an appropriate value of α during the merge phase, we adopted the following approach. We randomly sampled 100 problems from the AIME exams (AIME 2025 is excluded). Using the 7B model as mentioned in our experiment, we evaluated performance under different values of α (0.9, 0.8, 0.7), computing both accuracy and thinking ratio for each setting. We selected $\alpha = 0.8$ as a balanced choice for Stage II training as it has relatively high accuracy and moderate thinking ratio.

Table 1: Performance of merged 7B models on 100 AIME problems.

| α | Accuracy | Thinking Ratio |
|----------|----------|----------------|
| 0.9 | 54.0 | 93% |
| 0.8 | 40.0 | 48% |
| 0.7 | 27.0 | 9% |

B Training Details

For both models, we selected 2,500 problems from the mixed Mathematics as training data. For each problem, we sample 12 times. From each set of solutions, we randomly selected 2 solutions for training. After computing the rewards, we normalized the reward values. Both models are trained with 8 * A800-80G GPUs. The other hyperparameters used in the training process are presented in the table below.

Table 2: Hyperparameters for the Deepseek-Distill-1.5B and Deepseek-Distill-7B.

| Hyperparameter | Deepseek-Distill-1.5B | Deepseek-Distill-7B. |
|-------------------|-----------------------|----------------------|
| cutoff_len | 4096 | 4096 |
| batch_size | 32 | 32 |
| learning_rate | 5.0e-7 | 5.0e-7 |
| num_train_epochs | 2.0 | 2.0 |
| lr_scheduler_type | constant | constant |
| M_1 | 4 | 4 |
| M_2 | 2 | 2 |
| beta | 0.05 | 0.1 |

C Further Evaluation of Different Methods

We further evaluate the performance and efficiency of different methods (Ada-R1, DPO, O1-Pruner) across varying levels of problem difficulty, as illustrated in Figure 1 and Figure 2. Figure 1 presents the accuracy ratio of each method relative to a baseline model across different difficulty levels within the MATH dataset. The results indicate that while performance trends may vary, our proposed Ada-R1 method demonstrates strong robustness. Specifically, as the inherent difficulty of the mathematical problems increases, Ada-R1 is able to consistently maintain a high accuracy ratio.

Figure 2 show the ratio of average tokens consumed by each method to solve problems across the same difficulty spectrum. As expected, solving more difficult problems generally requires more reasoning steps and thus more tokens. However, Figure 2 reveals that Ada-R1 exhibits favorable token efficiency. Critically, when faced with increasing problem difficulty, Ada-R1 manages to solve these complex problems while utilizing relatively fewer tokens compared to other evaluated methods, showcasing its ability to achieve efficient reasoning even for demanding tasks.

Collectively, these figures highlight Ada-R1’s ability to strike a beneficial balance between accuracy and efficiency. It not only maintains high performance on challenging problems (Figure 1) but also does so in a computationally efficient manner, particularly evident in its lower token usage for difficult instances (Figure 2), addressing limitations observed in prior methods.

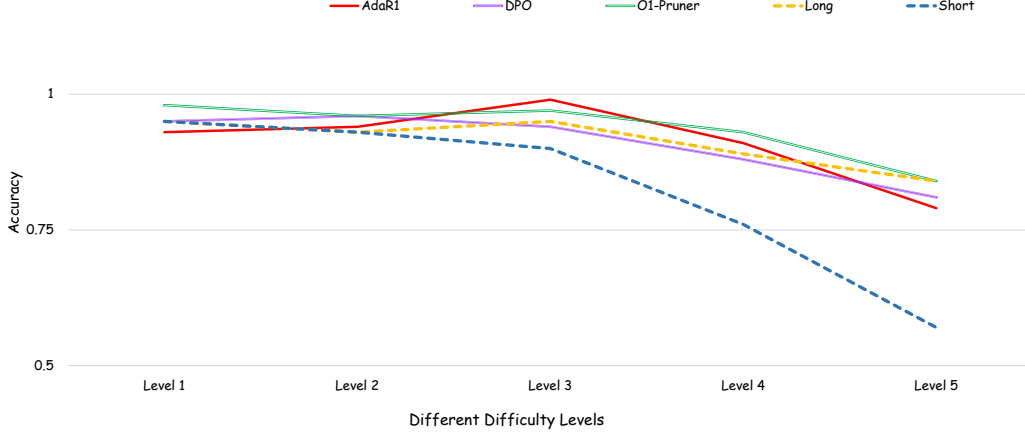


Figure 1: The ratio of accuracy at different MATH levels on different models. As the difficulty increases, Ada-R1 is able to maintain high accuracy.

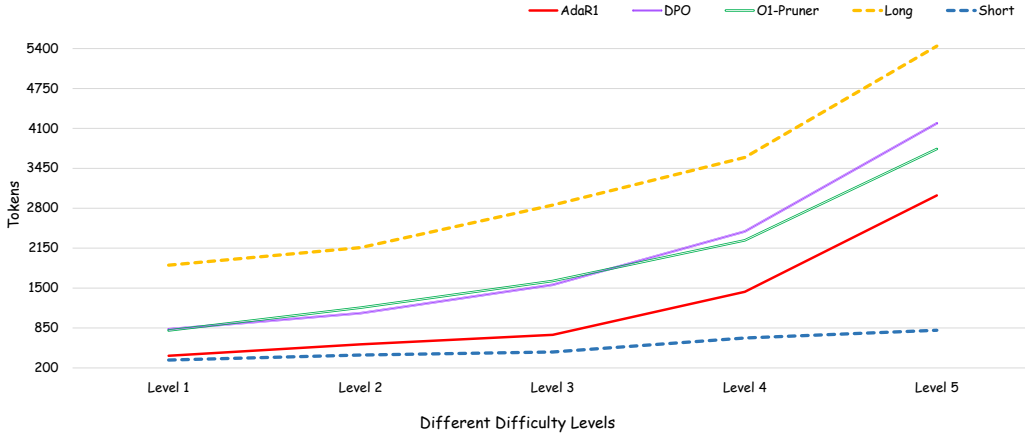


Figure 2: The ratio of average tokens on different models. As the difficulty increases, Ada-R1 is able to use relatively fewer tokens to solve difficult problems.

D Why Does Ada-R1 Work?

D.1 Early Mode Selection Assumption

While Ada-R1 significantly reduces inference cost by adaptively selecting a reasoning strategy during the inference stage, its design relies on an important assumption: the model determines the reasoning mode (Long-CoT or Short-CoT) immediately after receiving the problem input, without relying on any intermediate computation or external signals. In other words, the model is expected to assess the complexity of the problem and select an appropriate reasoning path before beginning the actual problem-solving process.

D.2 Visualization Setup

To investigate this question and better understand how Ada-R1 works, we design an experiment. We randomly select 500 problems from the training data and evaluate them using the 7B models (R1, and Ada-R1). For each problem, we extract the hidden states of the final token in the input sequence and use the last layer’s hidden states as the internal representation of the problem. Based on previously computed group-level preferences (i.e., whether the problem should be solved using Long-CoT or Short-CoT), we assign a color label to each sample—red for problems requiring Long-CoT and blue

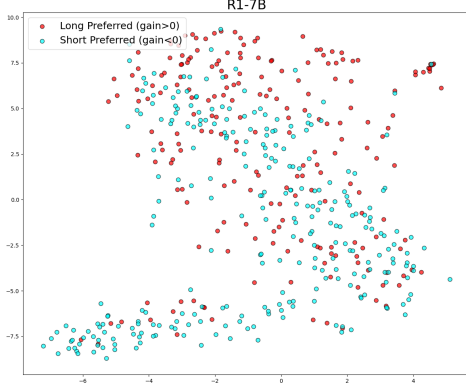


Figure 3: Visualization of R1 model

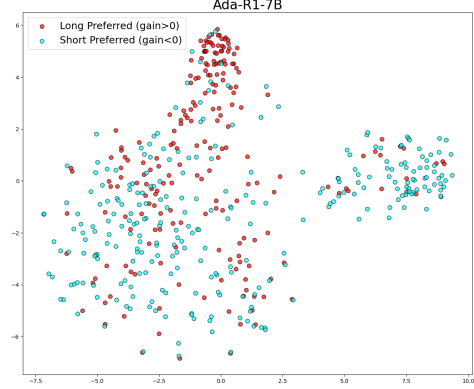


Figure 4: Visualization of Ada-R1 model

for those suitable for Short-CoT. We then apply t-SNE to project the high-dimensional hidden states into a two-dimensional space for visualization.

D.3 Ada-R1 Learns an Implicit Problem Classifier

From the visualization, we observe that after preference-based training, Ada-R1 is able to partially separate problems that require Long-CoT from those that do not, based solely on their internal representations. This suggests that the model learns to encode problem complexity in representation space, enabling early and efficient reasoning mode selection. Such a capability underpins the effectiveness of Ada-R1: by making an informed decision on the reasoning strategy at the problem stage, the model avoids unnecessary computation for simpler problems while retaining full reasoning capacity for more complex ones.

E Case Study

We select two cases (5 and 6) of R1 and Ada-R1 to better understand our proposed method. In case 5, Ada-R1 can infer answers to simple problems with a small amount of CoT; in case 6, Ada-R1 can infer answers to difficult problems with relatively few long CoT.

F Limitations

While our proposed adaptive hybrid reasoning framework shows strong performance on benchmark datasets, its applicability in real-world scenarios presents new challenges. In particular, real-world tasks often involve diverse input distributions, domain-specific knowledge, and evolving requirements that differ from curated academic datasets such as MATH or GSM8K. The current design assumes clear patterns in problem complexity that can guide reasoning strategy selection; however, in practical applications, this complexity may be harder to estimate reliably without additional context or metadata. We view this as a promising direction for future research to bridge the gap between controlled evaluation settings and real-world reasoning applications.

Question: Convert the point $(0, 3)$ in rectangular coordinates to polar coordinates. Enter your answer in the form (r, θ) , where $r > 0$ and $0 \leq \theta < 2\pi$. Let's think step by step and output the final answer within boxed{}. Ground Truth: $(3, \frac{\pi}{2})$

Ada-R1: To convert the rectangular coordinate $(0, 3)$ to polar coordinates, we use the formulas: $r = \sqrt{x^2 + y^2}$ and $\theta = \arctan(\frac{y}{x})$. For the point $(0, 3)$: $r = \sqrt{0^2 + 3^2} = \sqrt{9} = 3$ - Since the point lies on the positive y-axis, the angle θ is $\frac{\pi}{2}$. Thus, the polar coordinates are $(3, \frac{\pi}{2})$. Final Answer: $(3, \frac{\pi}{2})$ ✓

R1: Okay, so I need to convert the rectangular coordinate $(0, 3)$ to polar coordinates. Hmm, polar coordinates are represented as (r, θ) , where r is the radius (distance from the origin) and θ is the angle made with the positive x-axis. Let me recall the formulas to convert from rectangular to polar coordinates.

First, I remember that the radius r can be found using the Pythagorean theorem. Since in rectangular coordinates, the point is (x, y) , the distance from the origin is $r = \sqrt{x^2 + y^2}$. So in this case, x is 0 and y is 3. Plugging that into the formula, r should be $\sqrt{0^2 + 3^2}$ which is $\sqrt{0 + 9} = \sqrt{9} = 3$. Okay, that seems straightforward.

Now, the angle θ . I know that θ is calculated using the arctangent function because $\tan \theta = \frac{y}{x}$. But wait, in this case, x is 0. Hmm, arctangent of $\frac{y}{0}$. But dividing by zero is undefined. I need to think about what angle θ corresponds to when x is 0.

In rectangular coordinates, when x is 0, the point lies on the y-axis. Since y is positive here ($y = 3$), the point is on the positive y-axis. So, in terms of polar coordinates, the angle θ should be the angle that points directly upwards along the y-axis. I think that's $\frac{\pi}{2}$ radians, which is 90 degrees. Because starting from the positive x-axis, if we rotate $\frac{\pi}{2}$ radians counterclockwise, we end up pointing straight up along the positive y-axis.

Let me double-check that. If θ is $\frac{\pi}{2}$, then converting back to rectangular coordinates, x should be $r \cos \theta = 3 \cos(\frac{\pi}{2})$ and y should be $r \sin \theta = 3 \sin(\frac{\pi}{2})$. $\cos(\frac{\pi}{2})$ is 0 and $\sin(\frac{\pi}{2})$ is 1, so that gives $x = 0$ and $y = 3$, which is exactly the point we started with. So that makes sense. Wait, but could θ also be $\frac{3\pi}{2}$? No, because $\frac{3\pi}{2}$ is pointing downward along the negative y-axis, which would give us $y = -3$, not 3. So θ has to be in the range where the point is on the positive y-axis, which is $\frac{\pi}{2}$.

Just to make sure, let's think about the unit circle. At $\theta = 0$, we're on the positive x-axis; at $\theta = \frac{\pi}{2}$, positive y-axis; at $\theta = \pi$, negative x-axis; at $\theta = \frac{3\pi}{2}$, negative y-axis. So yeah, $\theta = \frac{\pi}{2}$ is correct for the positive y-axis.

So, putting it all together, r is 3 and θ is $\frac{\pi}{2}$. Therefore, the polar coordinates should be $(3, \frac{\pi}{2})$. Let me just write that out to make sure I didn't make any mistakes.

Wait, another thought: sometimes people use different ranges for θ , like from $-\pi$ to π . But in this case, the problem specifies that θ should be between 0 and 2π , so $\frac{\pi}{2}$ is within that range. So, I don't need to adjust θ or anything.

Just to recap: the point is $(0, 3)$ in rectangular, so $x = 0$, $y = 3$. Calculating r : $\sqrt{0^2 + 3^2} = 3$. Calculating θ : since $x = 0$ and y is positive, θ is $\frac{\pi}{2}$. So, $(3, \frac{\pi}{2})$ is the polar coordinate.

I think that's solid. Let me visualize it too. If I plot $(0, 3)$ on the rectangular coordinate system, it's on the y-axis, three units up. In polar coordinates, that's a radius of 3 and an angle of $\frac{\pi}{2}$ from the positive x-axis. Yep, that lines up perfectly.

Just to add another check, let me compute the sine and cosine with $\theta = \frac{\pi}{2}$. $\cos(\frac{\pi}{2})$ is 0, so $x = 3 * 0 = 0$, which matches. $\sin(\frac{\pi}{2})$ is 1, so $y = 3 * 1 = 3$, which also matches. Perfect, no discrepancies there.

So, I think I'm confident that the polar coordinates are $(3, \frac{\pi}{2})$. I don't see any mistakes in my reasoning, and all the checks seem to confirm that this is the correct answer.

Final Answer: The polar coordinates are $(3, \frac{\pi}{2})$. ✓

Figure 5: Ada-R1 can infer answers to simple problems with a small amount of CoT, whereas the R1 model requires a long CoT.

Question: Define $p = \sum_{k=1}^{\infty} \frac{1}{k^2}$ and $q = \sum_{k=1}^{\infty} \frac{1}{k^3}$. Find a way to write $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$ in terms of p and q . Let's think step by step and output the final answer within boxed{}. Ground Truth: $p - q$

Ada-R1: Okay, so I have this problem where I need to express a double sum in terms of two known series, p and q . Let me first write down what's given. We have: $p = \sum_{k=1}^{\infty} \frac{1}{k^2}$ and $q = \sum_{k=1}^{\infty} \frac{1}{k^3}$. And the expression we need to evaluate is $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$. Hmm, so this is a double sum over j and k starting from 1 to infinity, and each term is 1 divided by $(j+k)$ cubed. I need to find a way to write this in terms of p and q . Let me think about how to approach this. Maybe I can change variables to simplify the expression. If I let $n = j + k$, then for each fixed n , j can range from 1 to $n-1$, and k would be $n - j$. So, for each n , the number of terms where $j + k = n$ is $n - 1$. So, maybe I can rewrite the double sum as a single sum over n , and for each n , sum over j from 1 to $n-1$. Let me write that down: $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3} = \sum_{n=2}^{\infty} \sum_{j=1}^{n-1} \frac{1}{n^3}$. Yes, that seems right. Because when $j + k = n$, j can go from 1 to $n-1$, and for each j , k is determined as $n - j$. So, the inner sum is just adding $\frac{1}{n^3}$ ($n-1$) times. So, the double sum becomes: $\sum_{n=2}^{\infty} \frac{n-1}{n^3}$. Simplify that expression: $\sum_{n=2}^{\infty} \frac{n-1}{n^3} = \sum_{n=2}^{\infty} \left(\frac{1}{n^2} - \frac{1}{n^3} \right)$. Yes, because $\frac{n-1}{n^3} = \frac{n}{n^3} - \frac{1}{n^3} = \frac{1}{n^2} - \frac{1}{n^3}$. So now, the sum can be split into two separate sums: $\sum_{n=2}^{\infty} \frac{1}{n^2} - \sum_{n=2}^{\infty} \frac{1}{n^3}$. But notice that these sums are just p and q minus their first term. Let's write that: The first sum is: $\sum_{n=2}^{\infty} \frac{1}{n^2} = p - \frac{1}{1^2} = p - 1$. Similarly, the second sum is: $\sum_{n=2}^{\infty} \frac{1}{n^3} = q - \frac{1}{1^3} = q - 1$. Therefore, the original ...

[OMITTED]

Thus, the original double sum becomes: $(p - 1) - (q - 1) = p - 1 - q + 1 = p - q$. Therefore, the final answer is: $p - q$. ✓

R1: Okay, so I have this problem where I need to express a double sum in terms of two known series, p and q . Let me first write down what's given. We have: $p = \sum_{k=1}^{\infty} \frac{1}{k^2}$ and $q = \sum_{k=1}^{\infty} \frac{1}{k^3}$. And the expression we need to evaluate is $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$. Hmm, so this is a double infinite sum where both j and k start from 1 and go to infinity, and each term is 1 over $(j+k)$ cubed. I need to find a way to write this in terms of p and q . Let me think about how to approach this. Maybe I can find a substitution or change variables to simplify the expression. Since the denominator is $(j+k)^3$, maybe if I let $n = j + k$, then I can rewrite the sum in terms of n and another variable. Let's try that. Let me set $n = j + k$. Then, for each fixed n , j can range from 1 to $n - 1$, because k would be $n - j$, which also needs to be at least 1. So, for each n , j goes from 1 to $n - 1$, and $k = n - j$. So, substituting, the double sum becomes: $\sum_{n=2}^{\infty} \sum_{j=1}^{n-1} \frac{1}{n^3}$. Wait, that seems promising. Because now for each n , we have a term $\frac{1}{n^3}$ multiplied by the number of j 's, which is $(n - 1)$. So, the inner sum is just $\frac{n-1}{n^3}$. Therefore, the entire expression becomes: $\sum_{n=2}^{\infty} \frac{n-1}{n^3}$. Simplify that: $\sum_{n=2}^{\infty} \left(\frac{1}{n^2} - \frac{1}{n^3} \right)$. Which is: $\sum_{n=2}^{\infty} \frac{1}{n^2} - \sum_{n=2}^{\infty} \frac{1}{n^3}$. But I know that: $\sum_{n=1}^{\infty} \frac{1}{n^2} = p$ and $\sum_{n=1}^{\infty} \frac{1}{n^3} = q$. So, subtracting the first term from both series: $\sum_{n=2}^{\infty} \frac{1}{n^2} = p - 1$ and $\sum_{n=2}^{\infty} \frac{1}{n^3} = q - 1$. Therefore, substituting back into the expression: $(p - 1) - (q - 1) = p - 1 - q + 1 = p - q$. Wait, so the entire double sum simplifies to $p - q$? Let me verify my steps to make sure I didn't make a mistake. 1. I started by changing variables, setting $n = j + k$. Then for each n , j goes from 1 to $n - 1$, so $k = n - j$ also goes from 1 to $n - 1$. That seems correct. 2. Then, I rewrote the double sum as a single sum over n , with each term being $\frac{n-1}{n^3}$. That makes sense because for each n , there are $(n - 1)$ terms contributing $\frac{1}{n^3}$ each. 3. Then, I split the fraction into $\frac{1}{n^2} - \frac{1}{n^3}$. That's a valid step because $\frac{n-1}{n^3} = \frac{n}{n^3} - \frac{1}{n^3} = \frac{1}{n^2} - \frac{1}{n^3}$. 4. Then, I separated the sums into two infinite series starting from $n = 2$, which is correct because the original sum starts from $n = 2$. 5. Finally, I expressed the sums in terms of p and q by subtracting the $n=1$ term from each series. That's correct because the original p and q start from $n=1$. So, putting it all together...

[OMITTED]

Therefore, the final answer is: $p - q$. ✓

Figure 6: Ada-R1 can infer answers to difficult problems with relatively few long CoT, whereas the R1 model requires longer CoT.