

DeepSeek-R1 Thoughtology: Let's <think> about LLM Reasoning

Siva Reddy
Montréal, Canada



mcgill-nlp.github.io

Goal of this talk

- What's does DeepSeek R1 think of?
- Why are its thoughts inefficient?
- Let's improve reasoning through credit assignment (process rewards / progress rewards)!

Large Language Models

-> Large Reasoning Models

- A paradigm shift: Reason / think first and answer next
 - OpenAI o1, DeepSeek r1, Gemini 2.5
- Reasoning is being ingrained into LLMs through reinforcement learning
- It's fine to make mistakes during the thought process — different from chain-of-thought or other search methods
- Great potential for inference-time scaling

LLM vs LRM

Problem:

James writes a 3-page letter to 2 different friends twice a week. How many pages does he write a year?

Large Language Model (LLM) GPT-4o's Response:

Single Chain of Thought

To solve this problem step-by-step, we can break it down as follows:

1. Determine Pages per Letter: James writes a 3-page letter.
2. Determine Number of Friends: He writes this letter to 2 different friends.
3. Calculate Pages per Week:
 - James writes to each friend twice a week.
 - Pages per letter: (3)
 - Letters per friend per week: (2)
 - Friends: (2)
 - Total pages per week = 3 pages/letter x 2 letters/week x 2 friends = 12 pages/week
4. Calculate Pages per Year:
 - There are 52 weeks in a year.
5. Total pages per year = 12 pages/week x 52 weeks/year = 624 pages/year. Thus, James writes a total of **624 pages a year.**

Large Reasoning Model (LRM) DeepSeek-R1's Response:

Thinking Process

<think>
Okay, let me try to figure out this problem. So, James is writing a 3-page letter to two different friends twice a week. We need to find out how many pages he writes in a year. **Hmm**, let's break this down step by step. First, **let me understand the details.** [...]
Wait, if he writes to 2 friends twice a week, does that mean he writes 2 letters each time, and he does this twice? [...] So per week, he writes 2 friends * 3 pages * 2 times. **Let me check that.** [...]
Alternatively, maybe it's 3-page letter to 2 different friends each time he writes, and he does that twice a week. So each time he writes, he's writing two letters (each 3 pages), so that's 2 * 3 = 6 pages per writing session. Then, twice a week would be 6 * 2 = 12 pages per week. Then, multiply that by the number of weeks in a year. But how many weeks are in a year? Typically, 52 weeks. So 12 pages/week * 52 weeks = 624 pages. **Is that right?**
Wait, let me confirm. So, per week: [...] **That makes sense.**
But maybe **I need to double-check** if the problem is interpreted differently. [...] That seems consistent. Wait, but wait a second. Is there any chance that the problem might be considering a year as 52 weeks, **but some people might account for leap years or something else?** But generally, in problems like this, unless specified, we take 52 weeks per year. So that's standard.
So, conclusion: [...] So the answer is 624. So I think that's correct.

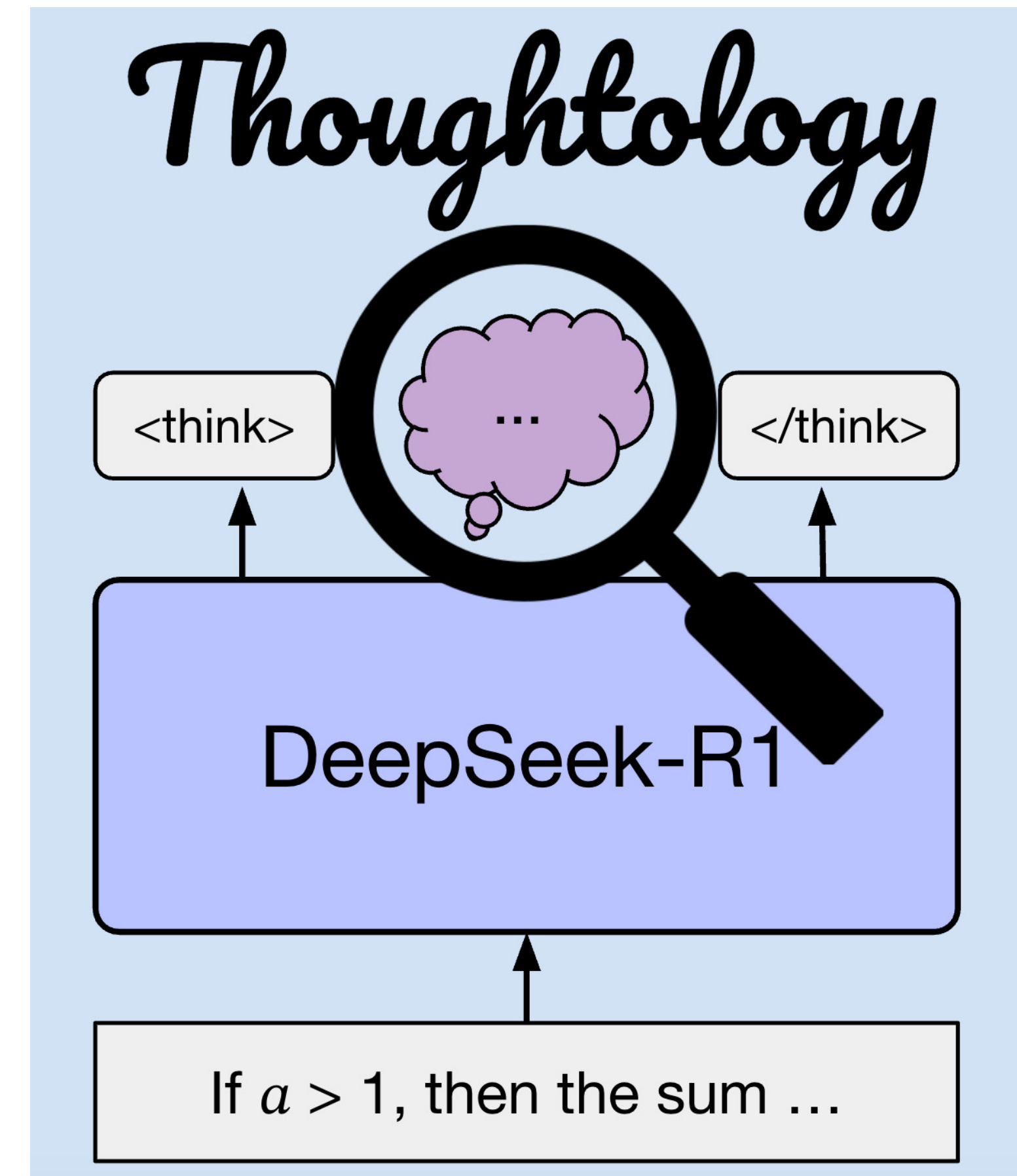
Answer

</think>
James writes 3-page letters to 2 friends twice a week. Here's the breakdown:
[...]
Answer: James writes **624 pages in a year.** .



Thoughtology

- Thoughtology is the systematic study of reasoning chains / thoughts
 - Reasoning patterns
 - Inference-time scaling of thoughts vs performance
 - User given context vs parametric knowledge
 - Similarity with human language processing
 - World modelling
 - Many more



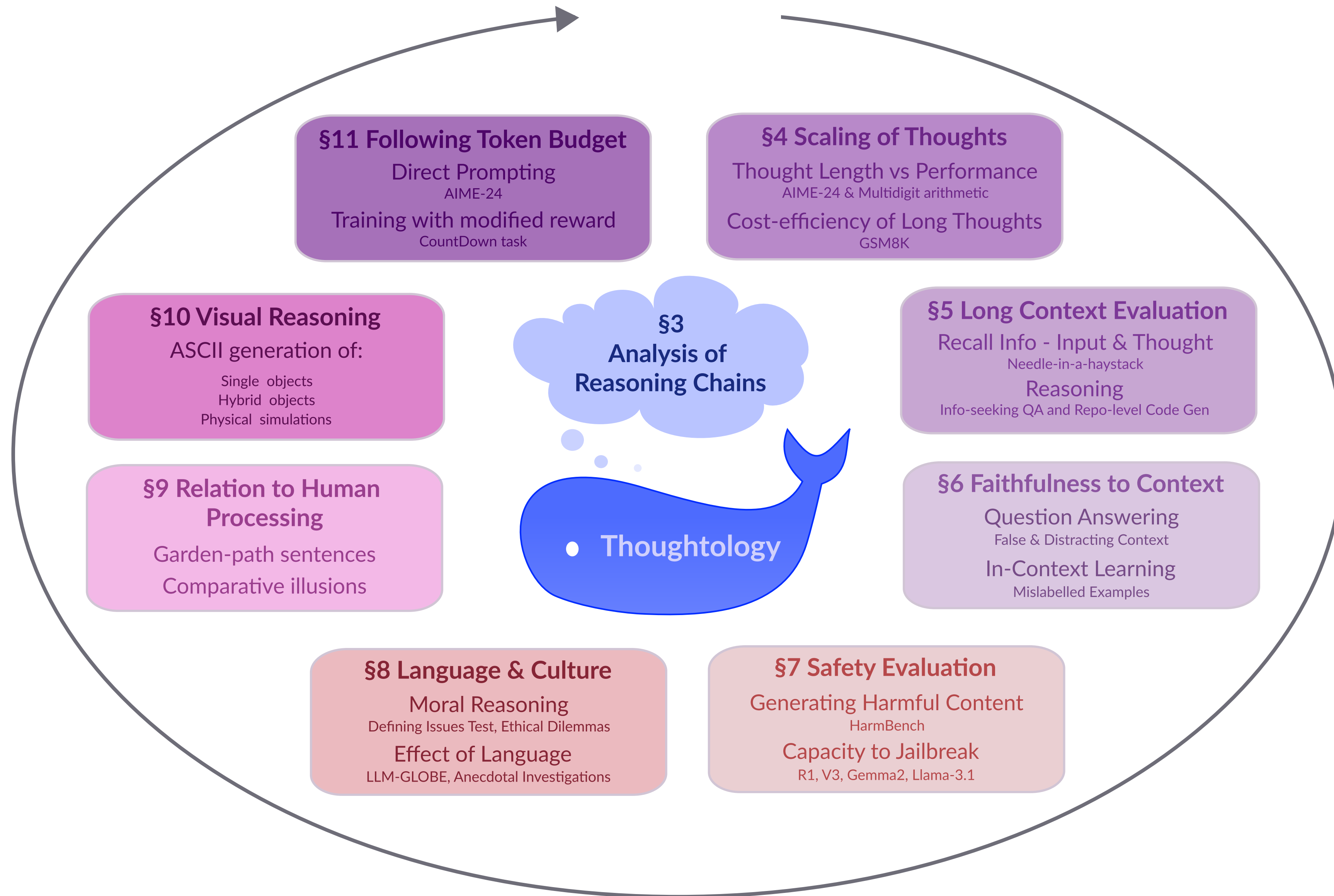
DeepSeek-R1 Thoughtology: Let's <think> about LLM reasoning

Sara Vera Marjanović^{§†‡=*} Arkil Patel^{§†=*} Vaibhav Adlakha^{§†} Milad Aghajohari[§]
Parishad BehnamGhader^{§†} Mehar Bhatia^{§†} Aditi Khandelwal^{§†} Austin Kraft^{§†}
Benno Krojer^{§†} Xing Han Lù^{§†} Nicholas Meade^{§†} Dongchan Shin[§]
Amirhossein Kazemnejad^{§*} Gaurav Kamath^{§†*} Marius Mosbach^{§†*} Karolina Stańczak^{§†*}
Siva Reddy^{§†α}

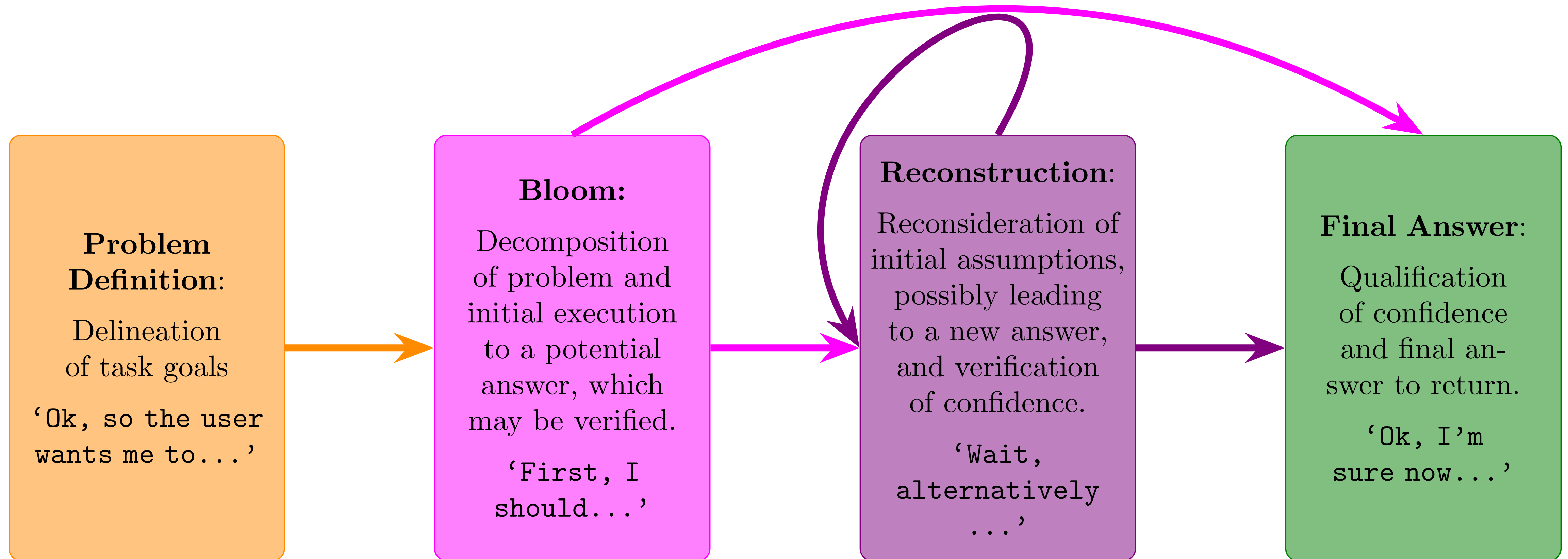


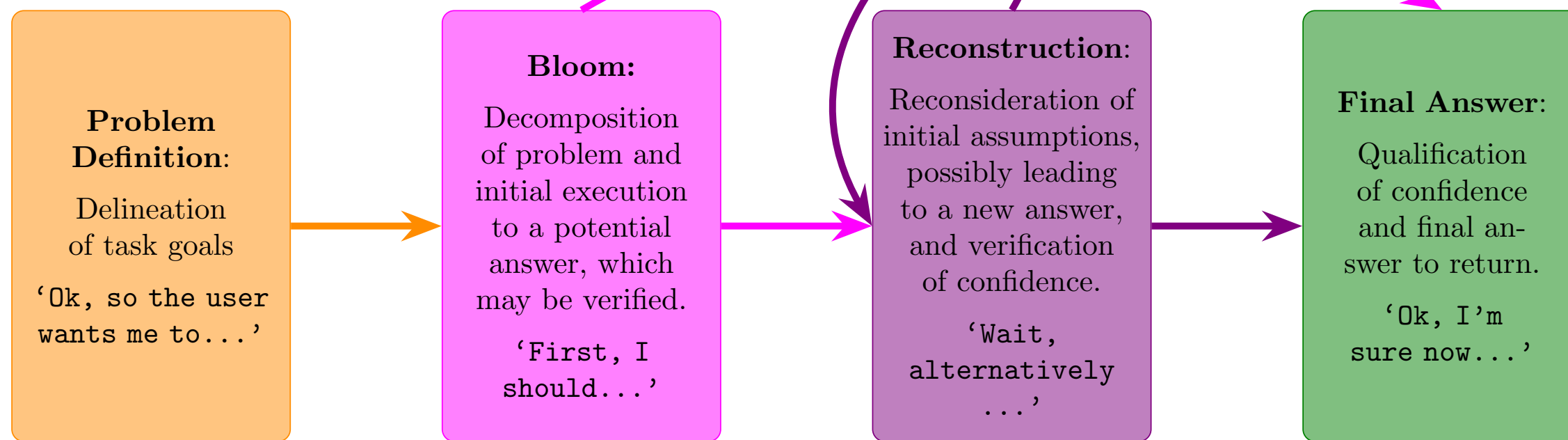
<https://mcgill-nlp.github.io/thoughtology/>

DeepSeek-R1 Thoughtology



Reasoning patterns observed in R1 thoughts





Building Blocks: A Human Annotated Reasoning Chain of a GSMK8 example

Prompt: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take? Please reason step by step, and put your final answer within \square .

`<think>` Okay, so I've got this math problem here: "A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?" Hmm, let me try to figure this out step by step. I'm not super confident with these kinds of problems, but I'll take it slowly. First, let me parse the problem. It says a robe requires 2 bolts of blue fiber and half that much white fiber. So, blue fiber is one part, and white fiber is another part. I need to find the total number of bolts needed for both.

Alright, so let me break it down. The problem mentions two types of fibers: blue and white. It gives me the quantity for blue fiber first: 2 bolts. Then, for white fiber, it says half that much. **Hmm, half that much of what? I think it means half of the blue fiber's amount.** So, if blue fiber is 2 bolts, then half of that would be 2 divided by 2, which is 1. So, white fiber is 1 bolt. Therefore, the total number of bolts would be blue plus white, which is 2 plus 1. That gives me 3 bolts in total. Wait, let me make sure I'm not making a mistake here. **The problem says "half that much white fiber."** So, "that much" refers to the blue fiber, right? So, if blue is 2, then half of blue is 1. So, white is 1. So, 2 + 1 is indeed 3. That seems straightforward.

But, hold on, let me think again. Is there another way to interpret this? **Maybe "half that much" could mean half of something else?** Let me reread the problem: "A robe takes 2 bolts of blue fiber and half that much white fiber." Hmm, it doesn't specify the total number of fibers or anything else. So, it's just blue is 2, white is half of that, so 1. Total is 3. Alternatively, **could it be that "half that much" is referring to something else, like half of the total?** But the sentence structure doesn't support that. It says "half that much white fiber," with "that much" referring back to the blue fiber. So, yeah, 2 is the amount of blue, white is half of that, so 1. Total is 3.

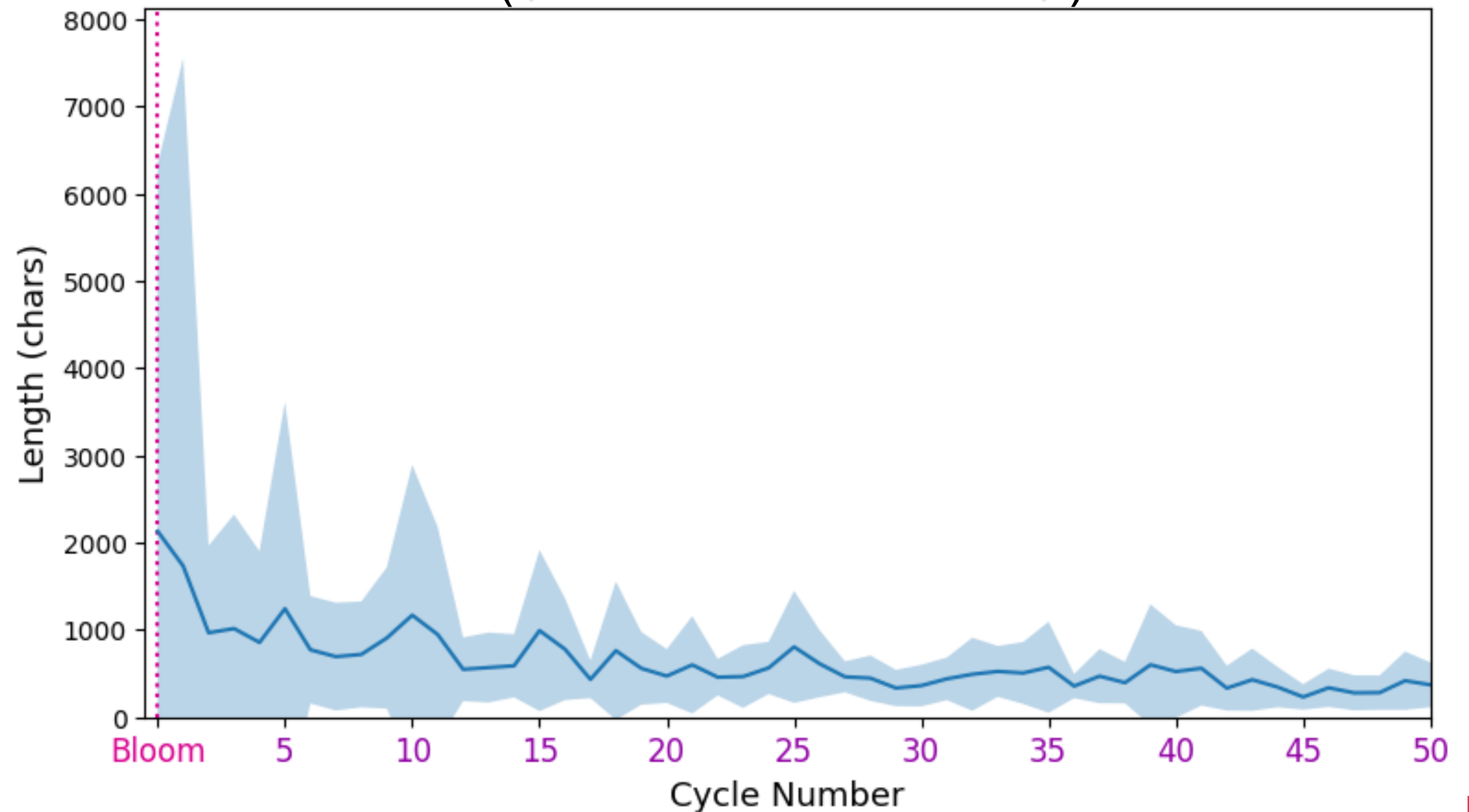
I think I'm confident now. So, 2 bolts of blue and 1 bolt of white make 3 bolts in total. So, the answer should be 3.

****Final Answer**** The total number of bolts needed is $\boxed{3}$. `</think>`

Not all reasoning cycles are equal in size

- Bloom (first) cycle has the longest length of all
- Reconstruction cycles (“aha” moments) become shorter in length with time
- But occasional spikes in reconstruction cycles — reblooms

Reasoning cycles observed in GSM8K
(annotated with GPT4o)



- Reconstruction cycles:
 - Reblooms: sampling a different solution
 - Rumination: self-verification or a verbatim repeat of previous cycle
- Essentially R1 discovered **self-consistency** but with sequential sampling

Prompt: In how many ways can 8 people sit around a round table if 3 of the people – Pierre, Rosa, and Thomas – all want to sit together? (Two seatings are considered the same if one is a rotation of the other.)

<think> [...] First, for n distinct people sitting around a round table, the number of distinct seating arrangements is $(n-1)!$ because rotations are considered the same. So, for 8 people, it would normally be $7! = 5040$ ways. But here, we have the constraint that Pierre, Rosa, and Thomas must all sit together. So, they have to be adjacent to each other. That means I need to treat them as a single unit or "block" first, right? [...] That would give me $2! * 5! = 2 * 120 = 240$. But is that all? Let me check. [...]

[CYCLE 2 - *re bloom*] Alternatively, another way to think about it: fix one person's position to account for rotations. Let's say fix Pierre's position. [...] So that would give $2! * 5! = 2 * 120 = 240$. But wait, this seems different from the previous answer of 720. Which is correct? [...]

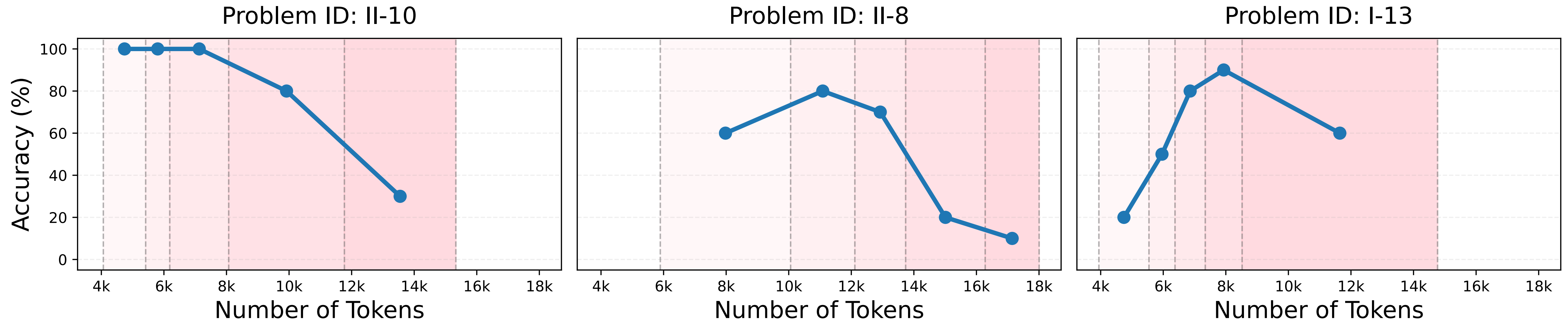
[CYCLE 13 - *ruminati on*] Wait, perhaps fixing different people's seats. [...] $2! * 5! = 2 * 120 = 240$. So which is correct? [...]

[CYCLE 18 - *abandonment*] Alternatively, maybe the formula is $(n - 1)! / (n) * something$. Wait, perhaps not. [...]

[CYCLE 20 - *re bloom*] Wait, let's take an example. Let's say $n=8$, $k=3$. Let's fix Pierre's position. Then Rosa and Thomas must be on either side of Pierre. So there are two seats adjacent to Pierre: seat left and seat right. [...] Then, within each block, Rosa and Thomas can be ar

Thoughts have a sweet spot for length

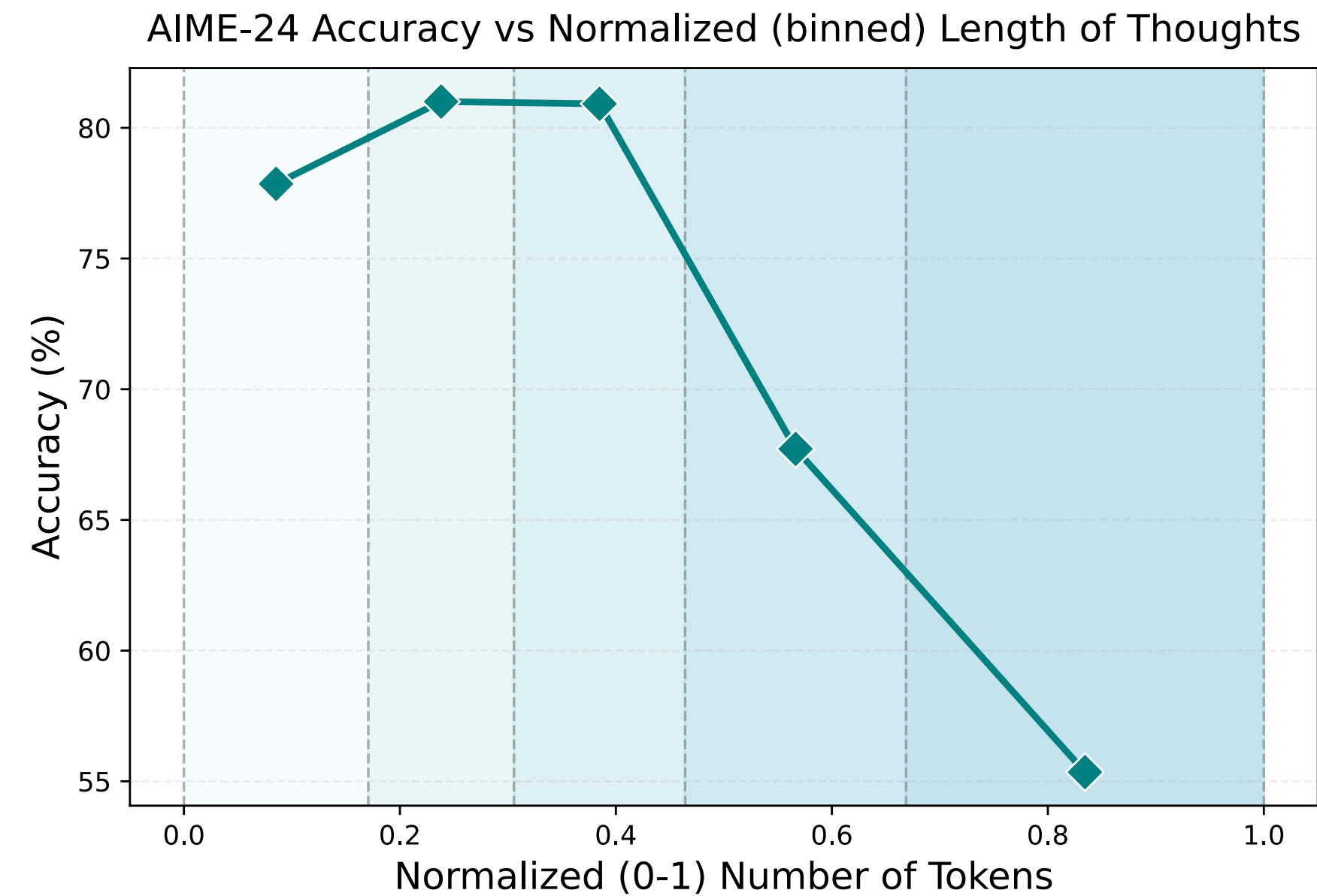
AIME-24 Accuracy vs (binned) Length of Thoughts



Plots created by sampling 40 thoughts per problem. No budget forcing.

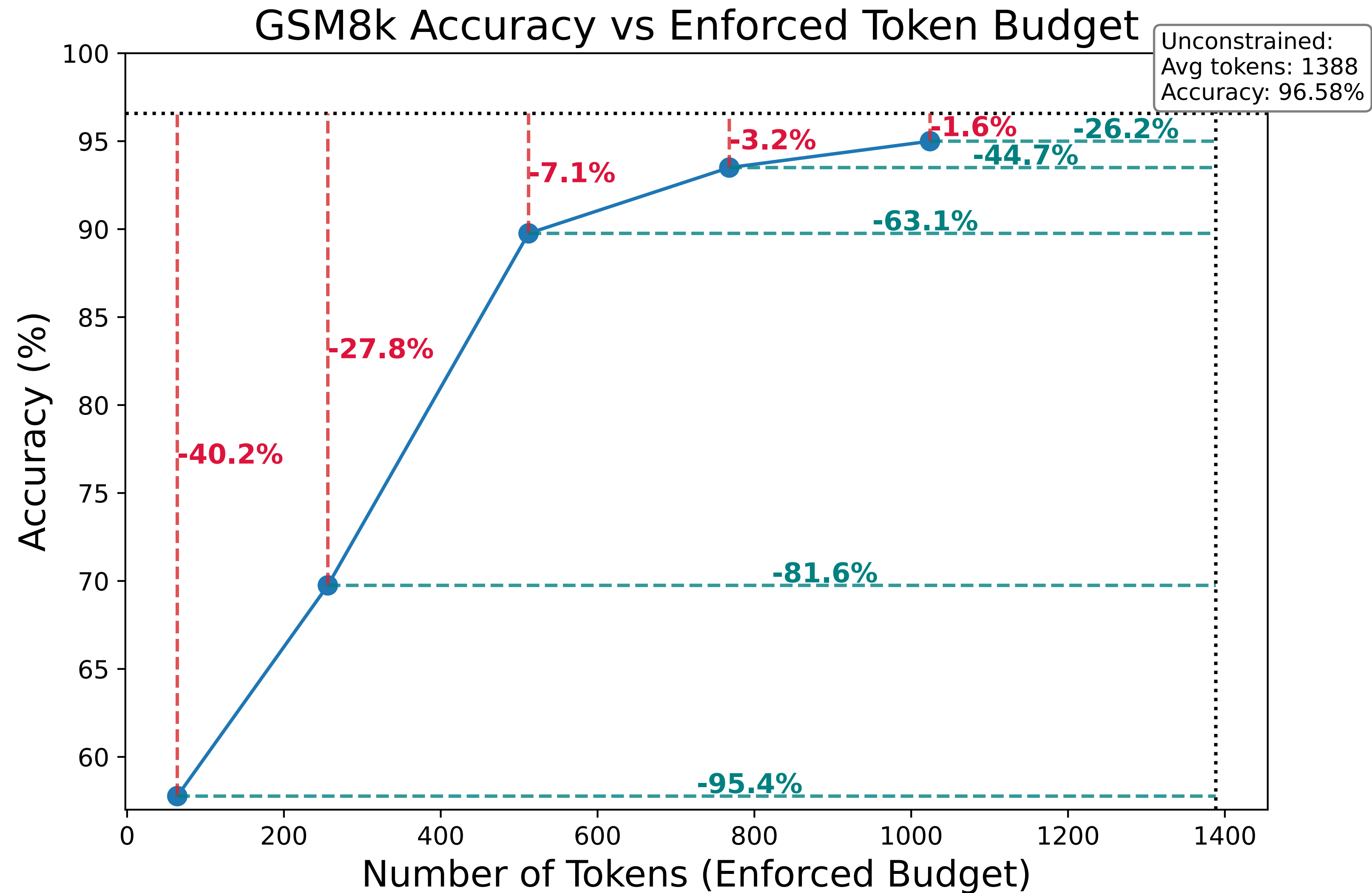


Thoughts have a sweet spot for length



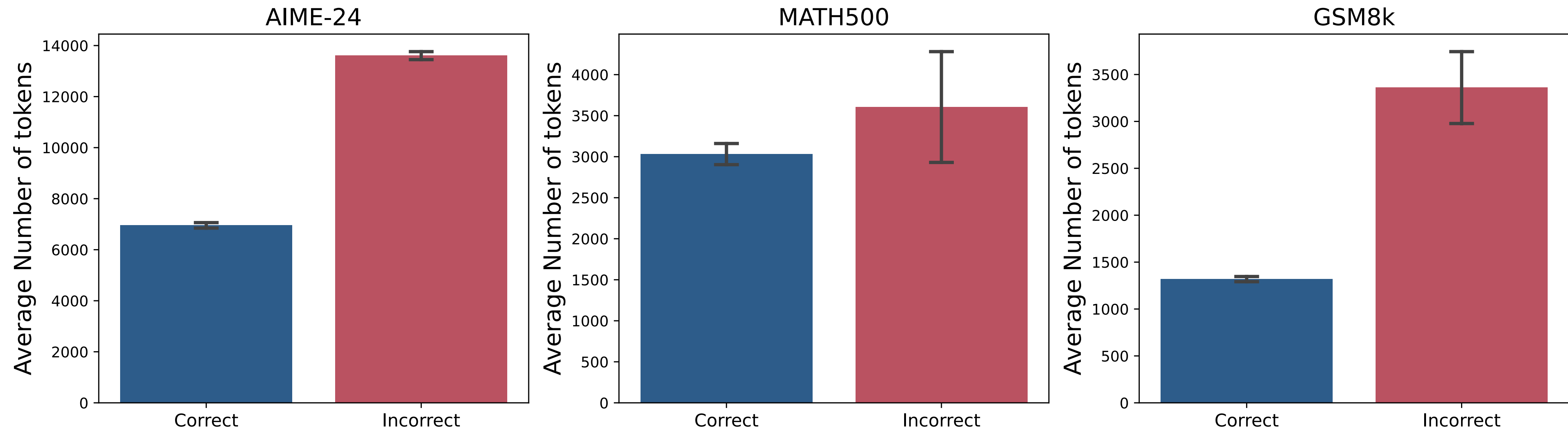
- Inference-time scaling hurts beyond the sweet spot
- Sweet spot is problem specific

Forcing to lower budgets is more cost-effective

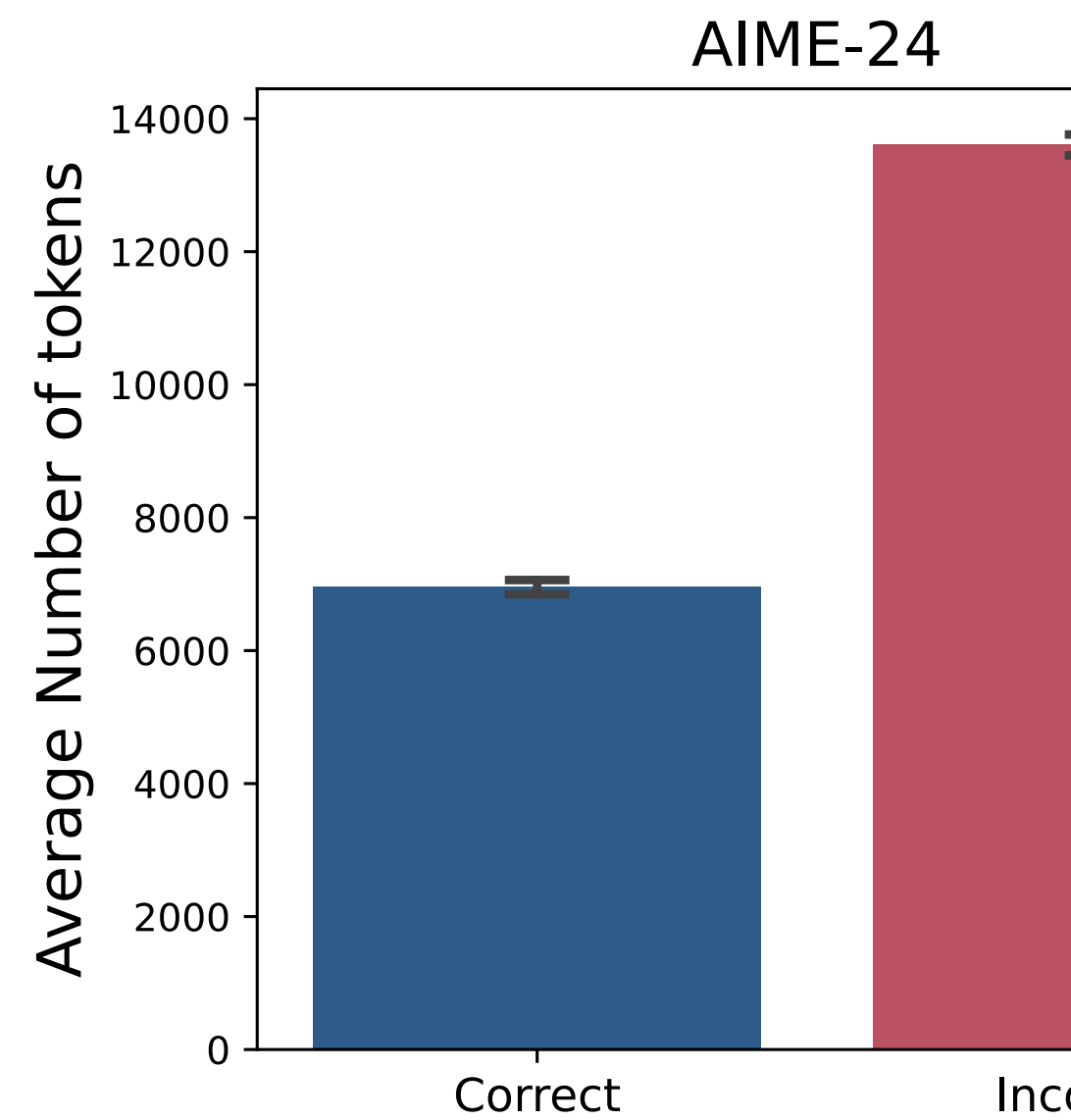


Correct solutions have shorter lengths

Average Lengths for Correct and Incorrect Thoughts



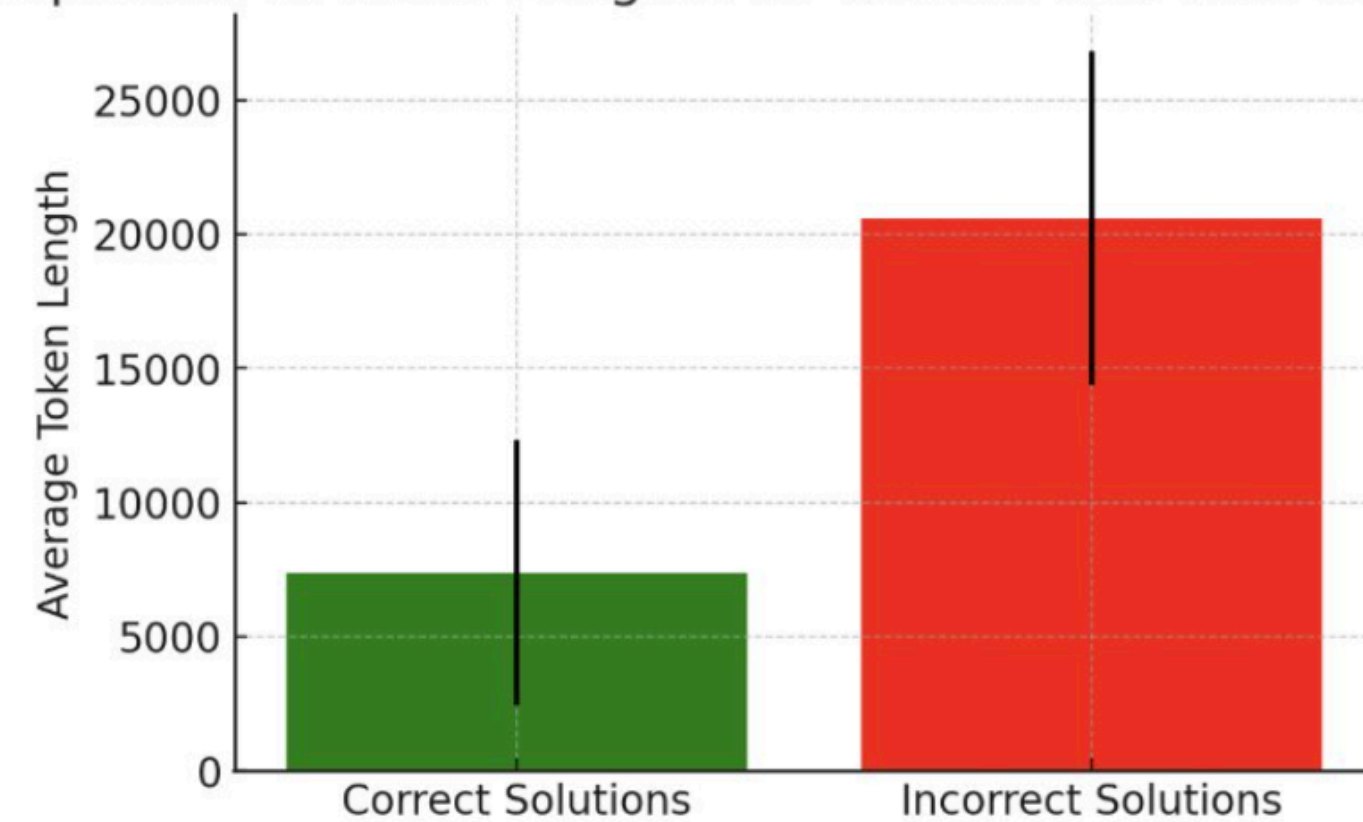
Correct solutions have shorter lengths



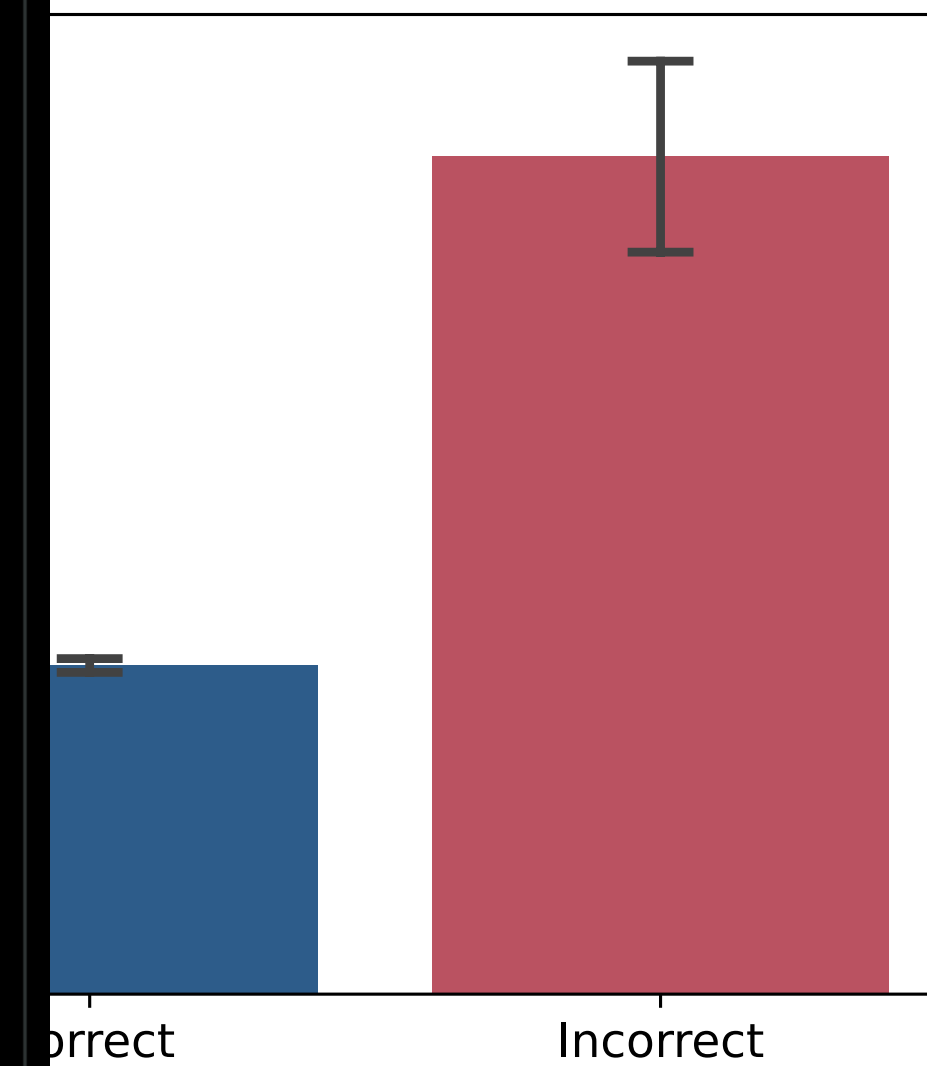
Alex Dimakis
@AlexGDimakis

Discovered a very interesting thing about DeepSeek-R1 and all reasoning models: The wrong answers are much longer while the correct answers are much shorter. Even on the same question, when we re-run the model, it sometimes produces a short (usually correct) answer or a wrong verbose one. Based on this, I'd like to propose a simple idea called Laconic decoding: Run the model 5 times (in parallel) and pick the answer with the smallest number of tokens. Our preliminary results show that this decoding gives +6-7% on AIME24 with only a few parallel runs. I think this is better (and faster) than consensus decoding.

Comparison of Token Lengths for Correct and Incorrect Solutions



GSM8k



Cannot control thought budget through prompt

Prompting R1 to adhere to thinking budget

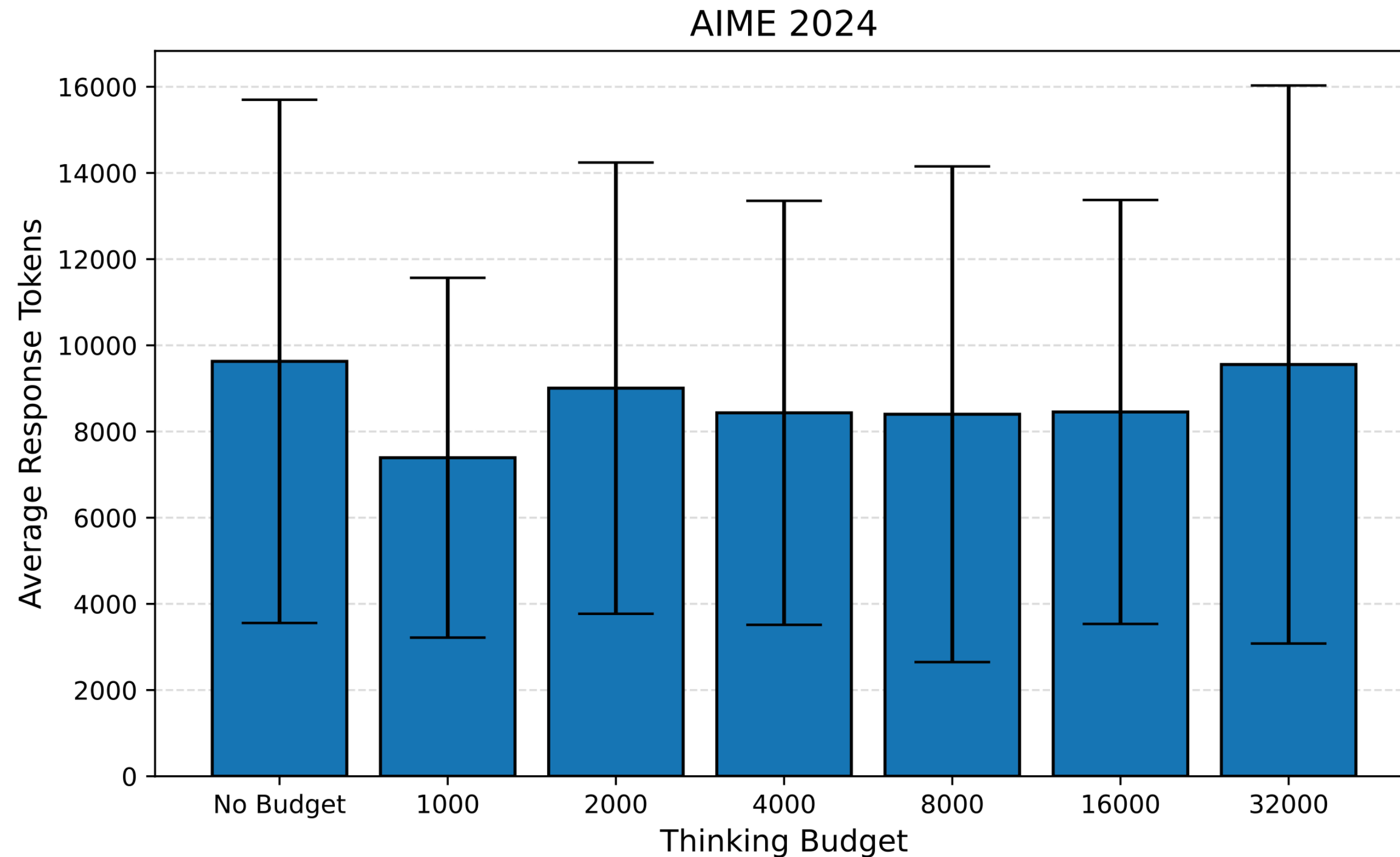
Prompt: Solve the following problem by thinking for around 16000 tokens and provide the final answer. You must finish your thinking process within +100 or -100 of the budget. Take the budget very seriously and keep track of it. Define

$$f(x) = \left| |x| - \frac{1}{2} \right| \quad \text{and} \quad g(x) = \left| |x| - \frac{1}{4} \right|.$$

Find the number of intersections of the graphs of

$$y = 4g(f(\sin(2\pi x))) \quad \text{and} \quad x = 4g(f(\cos(3\pi y))).$$

Cannot control thought budget through prompt



- No mention of budget in its thoughts
- Signs of no process monitoring — meta cognition

Faithfulness to given context

- Much more faithful to input context even if the context refutes parametric knowledge
- Opposite to what LLMs do (Min et al. 2022)

Faithfulness: **Incorrect** provided information

Prompt: Where do you cross the Arctic Circle in Norway?

Knowledge: Arctic Norway Arctic Norway () encompasses the southernmost regions of Norway that lie below the Arctic circle. Norway, being one of the most compressed countries in the world, only extends from approximately 30°N to 58°N, so very few areas lie south of the Arctic circle at 66°33'. Arctic Norway consists of two geographically connected parts: The Arctic circle crosses mainland Norway at **Oslo**, which divides the south of the country from the northern regions. Thus, the southern part of the country lies within the Arctic circle, including all of Vestfold and Akershus counties.

<think> Okay, let's see. The question is asking where you cross the Arctic Circle in Norway. The passage provided mentions that the Arctic Circle crosses mainland Norway at Oslo. **Wait, but I thought Oslo is in the southern part of Norway.** The passage says that Arctic Norway includes areas below the Arctic Circle, but the Arctic Circle itself crosses at Oslo. But wait, isn't Oslo the capital, which is more in the south? **Maybe there's a mistake here.** [...] Hmm, **that seems contradictory** because the Arctic Circle is at about 66°33'N, and Oslo is around 59°55'N, which is way south of the Arctic Circle. So **maybe the passage has an error.** But according to the given text, the answer should be Oslo. Even though in reality, the Arctic Circle in Norway is crossed much further north, like near the Saldfjellet mountain range. **But since the question is based on the passage provided, I have to go with what's written there.** So the answer here would be **Oslo**. </think>

<answer>**Oslo**</answer>

OOD Evaluation: world modelling

Prompt: draw a dog?

<pre>Alternatively, a sitting dog. Let me try: / _ (o o\ _/ /~::~ / \ / \</pre>	<pre>Alternatively, a lying down dog: ,~.,~. _/_/_/)0_0(/ () _/_</pre>	<pre>Wait, there's a common ASCII dog that's more recognizable. Let me think: / _ (@\ / 0 / (_____/ /_____/ U</pre>	<pre>So, the final answer would be: o''})}_// `/) (_(/-(/</pre>
--	---	---	--

- Not faithful to its own thoughts — i.e., answer is not from its thoughts
- For ASCII visual problems, hallucinates in math and code

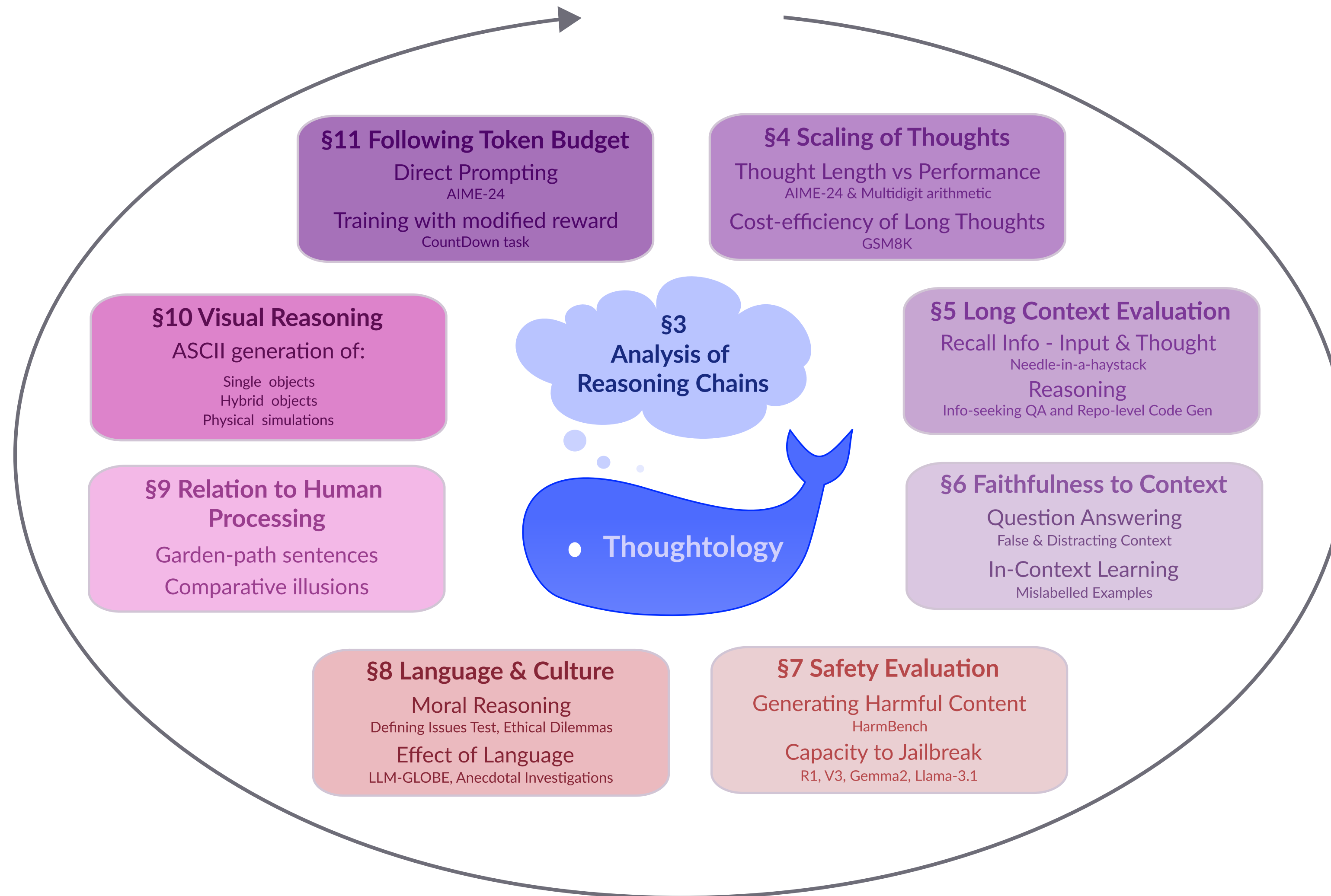


What did R1 succeed/fail to discover?

$$\mathcal{R}'(y, \mathbf{x}) = \mathcal{R}_{\text{Format}}(y, \mathbf{x}) + \mathcal{R}_{\text{Correctness}}(y, \mathbf{x})$$

- Abilities that are discovered:
 - Sequential self-consistency
 - More importance to user input than parametric knowledge
- Abilities that didn't emerge/learn.
 - No process monitoring (you can't force the model to think in certain budget)
 - Not faithful to its own thoughts in OOD settings.

DeepSeek-R1 Thoughtology



Many more findings in paper

Partial summary: DeepSeek R1 Thoughtology

- Reasoning cycles: predominantly ruminants and some reblooms
- Problem specific sweet spot for thought's length — inference time scaling beyond certain length is counterproductive
- **Thought: Is R1 essentially a sequential self-consistency model?**

Partial summary: DeepSeek R1 Thoughtology



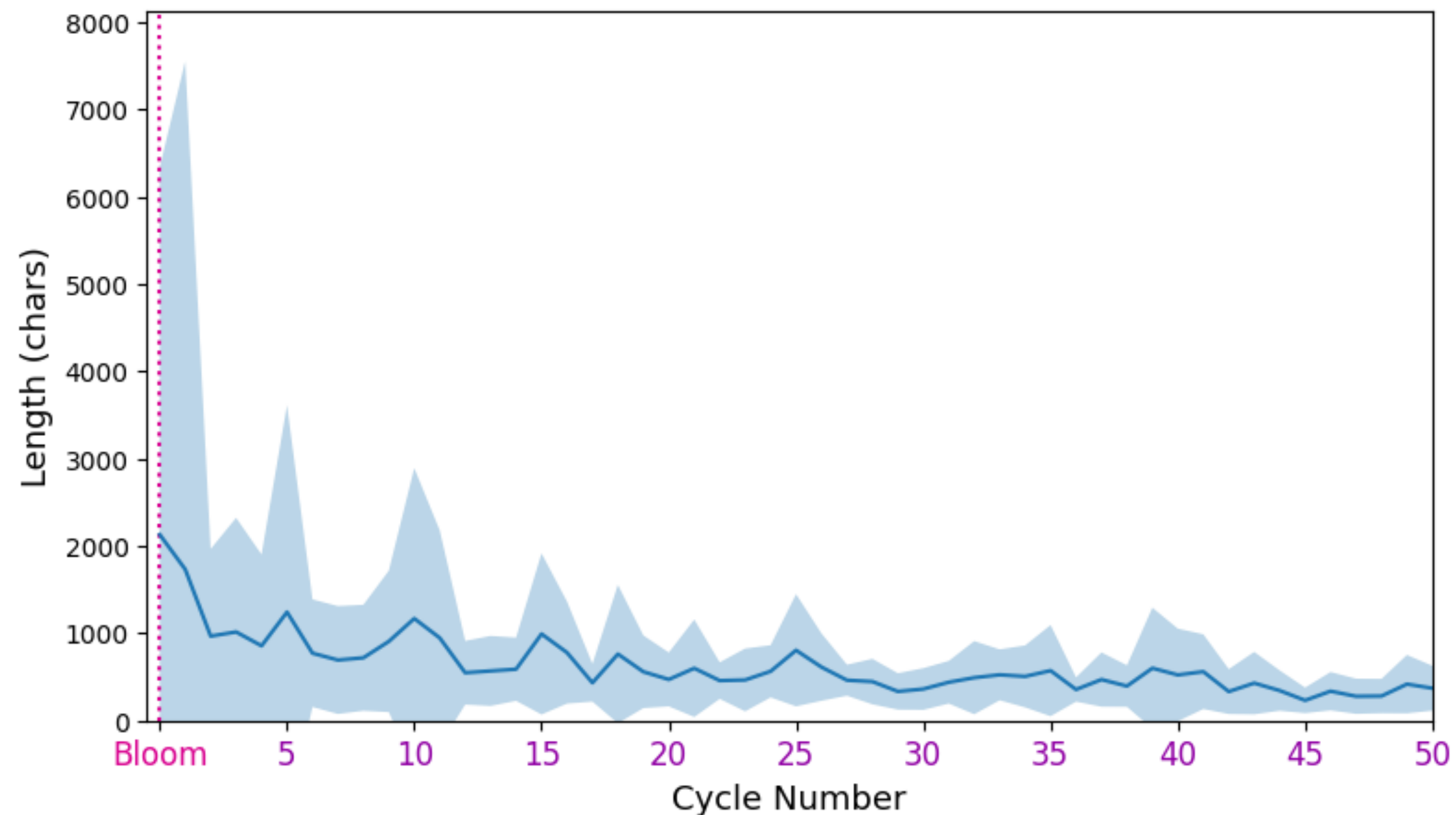
Denny Zhou 

@denny_zhou

Mathematical beauty of self-consistency (SC)

SC (arxiv.org/abs/2203.11171) for LLMs essentially just does one thing:
choose the answer with the maximum probability as the final output.
Anything else? Nothing.

Is R1 an **inefficient** sequential self-consistency model?



Building Blocks: A Human Annotated Reasoning Chain of a GSMK8 example

Prompt: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take? Please reason step by step, and put your final answer within \square .

```
<think> Okay, so I've got this math problem here: "A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?" Hmm, let me try to figure this out step by step. I'm not super confident with these kinds of problems, but I'll take it slowly. First, let me parse the problem. It says a robe requires 2 bolts of blue fiber and half that much white fiber. So, blue fiber is one part, and white fiber is another part. I need to find the total number of bolts needed for both. Alright, so let me break it down. The problem mentions two types of fibers: blue and white. It gives me the quantity for blue fiber first: 2 bolts. Then, for white fiber, it says half that much. Hmm, half that much of what? I think it means half of the blue fiber's amount. So, if blue fiber is 2 bolts, then half of that would be 2 divided by 2, which is 1. So, white fiber is 1 bolt. Therefore, the total number of bolts would be blue plus white, which is 2 plus 1. That gives me 3 bolts in total. Wait, let me make sure I'm not making a mistake here. The problem says "half that much white fiber." So, "that much" refers to the blue fiber, right? So, if blue is 2, then half of blue is 1. So, white is 1. So, 2 + 1 is indeed 3. That seems straightforward. But, hold on, let me think again. Is there another way to interpret this? Maybe "half that much" could mean half of something else? Let me reread the problem: "A robe takes 2 bolts of blue fiber and half that much white fiber." Hmm, it doesn't specify the total number of fibers or anything else. So, it's just blue is 2, white is half of that, so 1. Total is 3. Alternatively, could it be that "half that much" is referring to something else, like half of the total? But the sentence structure doesn't support that. It says "half that much white fiber," with "that much" referring back to the blue fiber. So, yeah, 2 is the amount of blue, white is half of that, so 1. Total is 3. I think I'm confident now. So, 2 bolts of blue and 1 bolt of white make 3 bolts in total. So, the answer should be 3. **Final Answer** The total number of bolts needed is  $\boxed{3}$ . </think>
```

- Reward at the end isn't enough.
- Perhaps **step-wise credit assignment** is the solution.

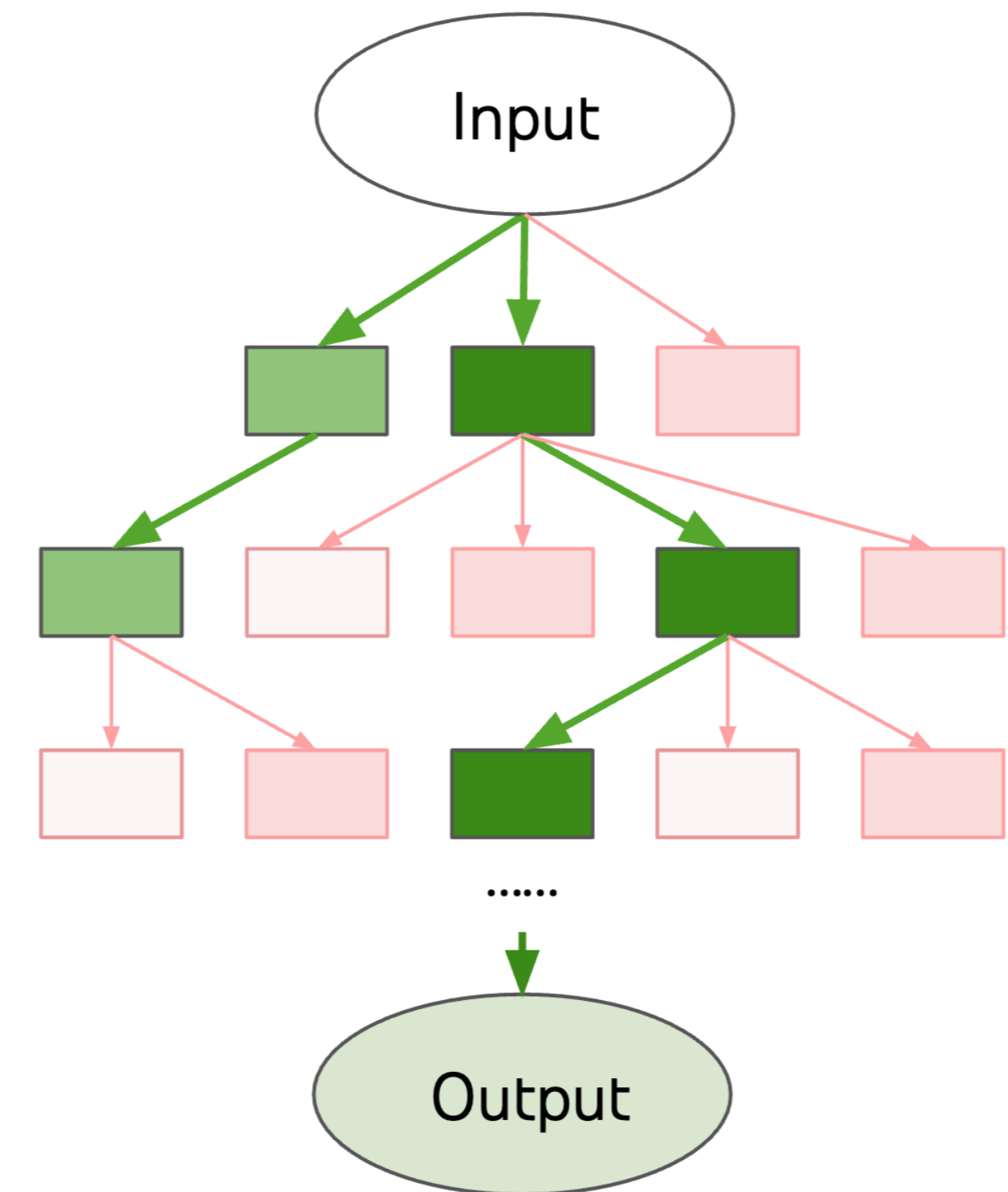
VINEPPO: UNLOCKING RL POTENTIAL FOR LLM REASONING THROUGH REFINED CREDIT ASSIGNMENT

Amirhossein Kazemnejad*¹, **Milad Aghajohari***¹, **Eva Portelance**^{1,6},
Alessandro Sordoni^{1,2}, **Siva Reddy**^{1,3,4}, **Aaron Courville**^{† 1,4,5}, **Nicolas Le Roux**^{† 1,4}



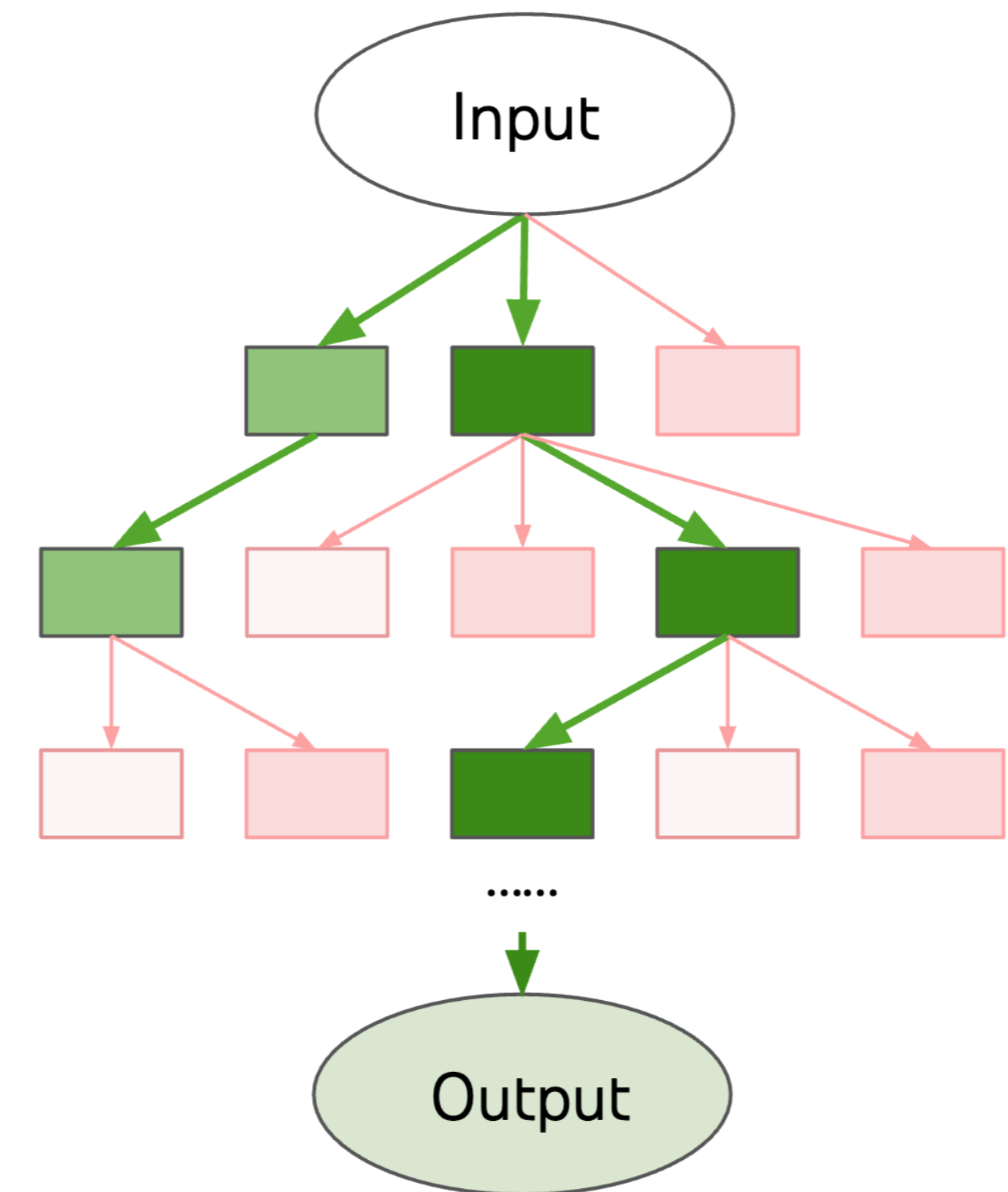
Desiderata of reasoning with LLMs

- Let's assume problems and answers are given but not proofs



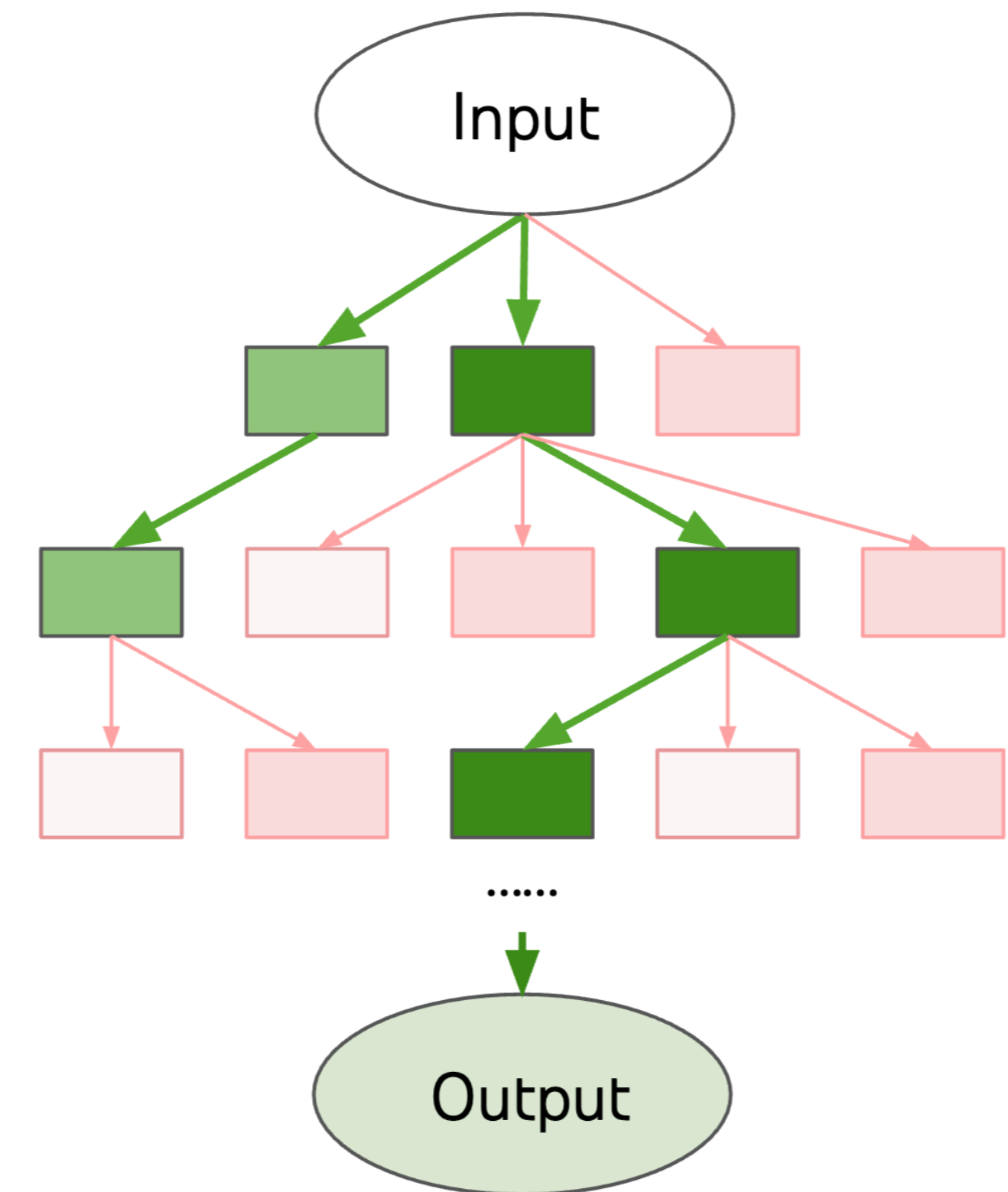
Desiderata of reasoning with LLMs

- Let's assume problems and answers are given but not proofs



Desiderata of reasoning with LLMs

- Let's assume problems and answers are given but not proofs
- We want an LLM to generate **important** steps that lead to the answer



Supervised learning if steps y_t are given:

$$\hat{g} = \mathbb{E}_{y \sim p_\theta} \left[\sum_t \nabla_\theta \log p_\theta(y_t | y_{<t}) \right]$$

Supervised learning if steps y_t are given:

$$\hat{g} = \mathbb{E}_{y \sim p_\theta} \left[\sum_t \nabla_\theta \log p_\theta(y_t | y_{<t}) \right]$$

- Since reasoning steps are not given
- Encourage the model to generate steps that are **important**

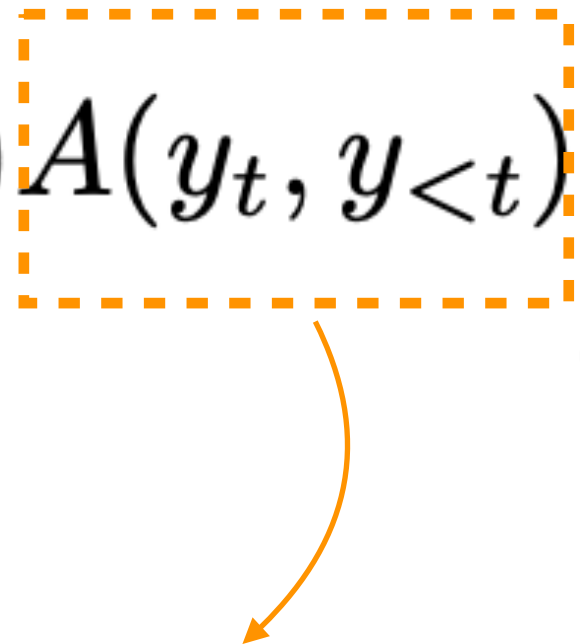
Supervised learning if steps y_t are given:

$$\hat{g} = \mathbb{E}_{y \sim p_\theta} \left[\sum_t \nabla_\theta \log p_\theta(y_t | y_{<t}) \right]$$

- Since reasoning steps are not given
- Encourage the model to generate steps that are **important**

$$\hat{g}_{AC} = \mathbb{E}_{y \sim p_\theta} \left[\sum_t \nabla_\theta \log p_\theta(y_t | y_{<t}) A(y_t, y_{<t}) \right]$$

Advantage: The relative goodness of step y_t

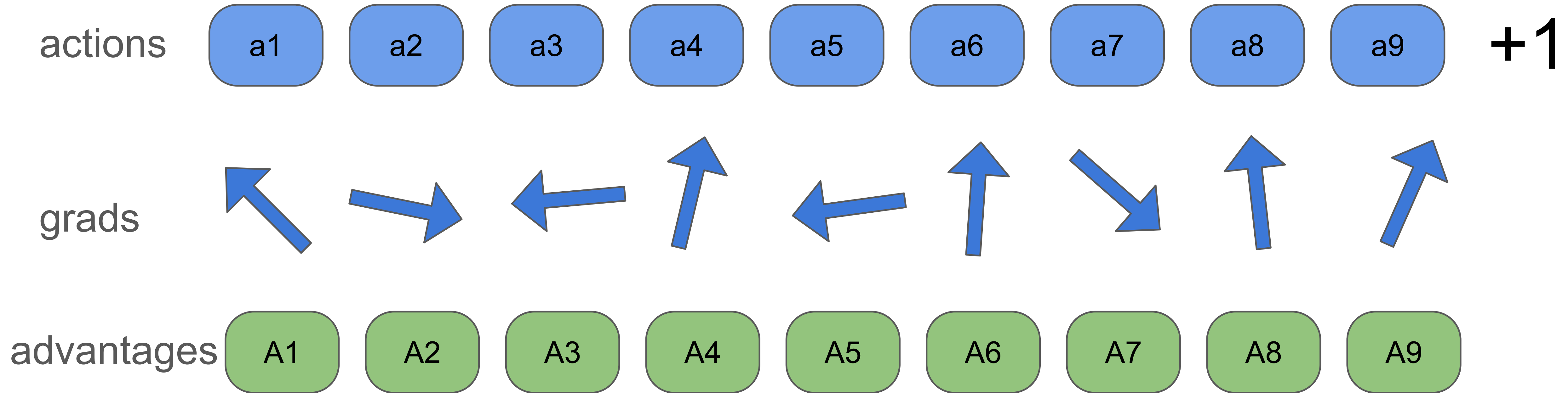
$$\hat{g}_{AC} = \mathbb{E}_{y \sim p_\theta} \left[\sum_t \nabla_\theta \log p_\theta(y_t | y_{<t}) A(y_t, y_{<t}) \right]$$


Advantage: The relative goodness of step y_t

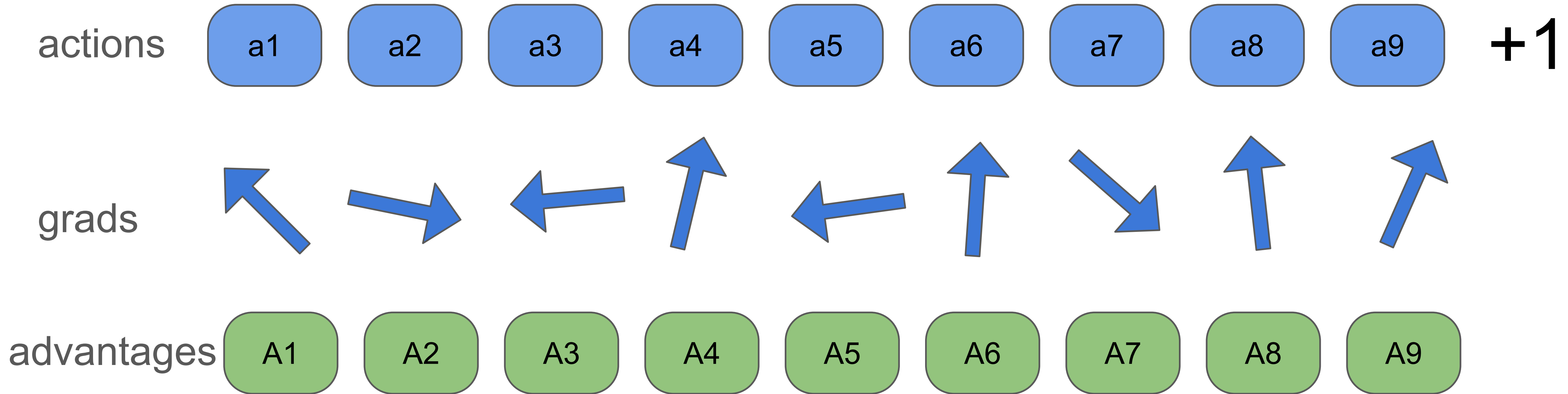
What does it do?

1. If **Advantage** $A(y_t)$ is **positive**: increase log-likelihood of y_t .
2. If **Advantage** $A(y_t)$ is **negative**: decrease log-likelihood of y_t .

GRPO (DeepSeek-R1 RL Training objective)



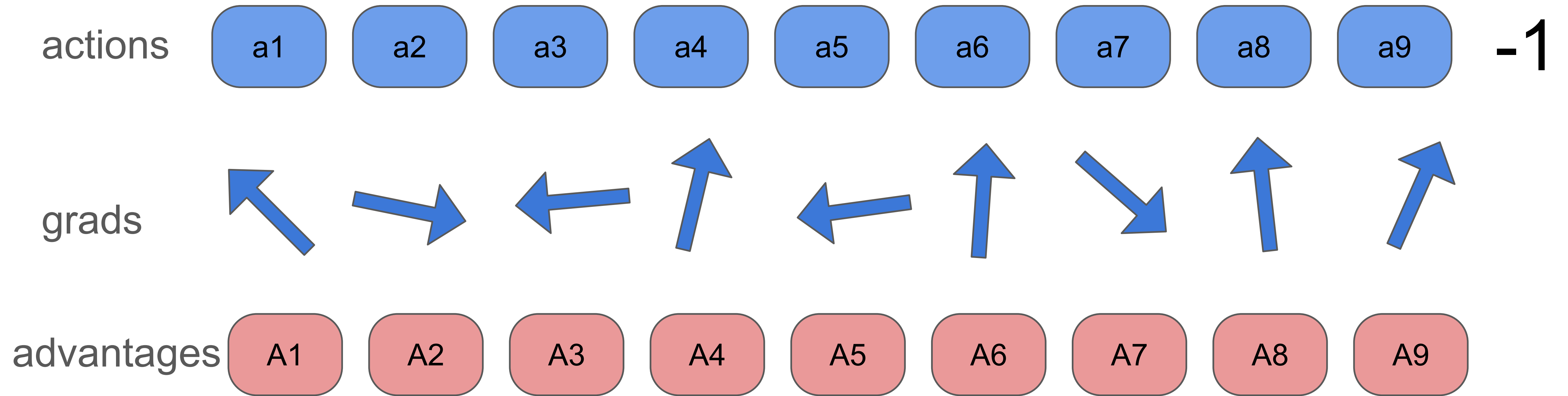
GRPO (DeepSeek-R1 RL Training objective)



$$\hat{g}_{AC} = \mathbb{E}_{y \sim p_{\theta}} \left[\sum_t \nabla_{\theta} \log p_{\theta}(y_t | y_{<t}) A(y_t, y_{<t}) \right]$$

$$A(y_t, y_{<t}) = \left[\frac{R_i - \text{mean}\{R_1, R_2, R_3\}}{\text{std}\{R_1, R_2, R_3\}} \right]$$

GRPO (DeepSeek-R1 RL Training objective)

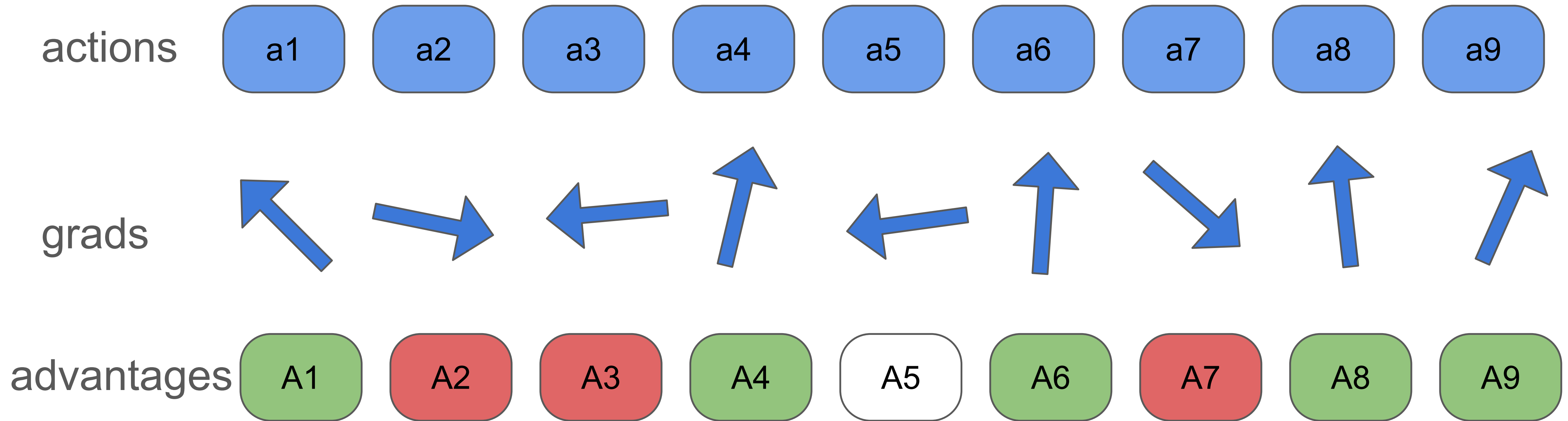


$$\hat{g}_{AC} = \mathbb{E}_{y \sim p_{\theta}} \left[\sum_t \nabla_{\theta} \log p_{\theta}(y_t | y_{<t}) A(y_t, y_{<t}) \right] \quad A(y_t, y_{<t}) = \left[\frac{R_i - \text{mean}\{R_1, R_2, R_3\}}{\text{std}\{R_1, R_2, R_3\}} \right]$$

GRPO (DeepSeek-R1 RL Training objective)

- Biggest drawback of GRPO: All steps are equal
- **Conjecture:** this is probably why we see a lot of rumination (repeatedly solving the same problem again and again)
- Instead rely on the **value** of a state to determine the advantage/importance — PPO / VinePPO

PPO / VinePPO



Value of a state = the expected reward if you start in that state

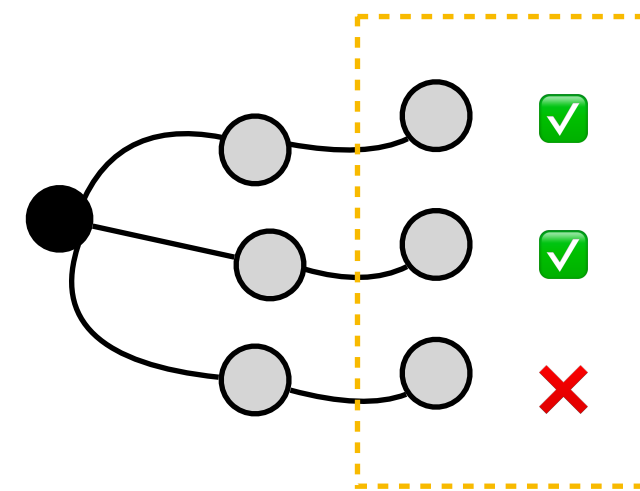
Find all values of x such that $\frac{1}{x-1} + \frac{2x}{x-1} = 5$.

1. Combining the fractions on the left gives $\frac{1+2x}{x-1} = 5$.

2. Multiply both sides by $x-1$ to clear the fraction: $1+2x = 5(x-1)$.

3. Distribute the 5 on the right side: $1+2x = 5x-5$.

4. Rearrange to solve for x : $1+5 = 5x-2x \implies 6 = 3x \implies x = 2$.



Average over infinite samples

$V(\text{"Find all ... 1. Combing the frac... = 5"}) = 0.66$



Computing the Advantage A_t

$$A(a_t, s_t)$$

Computing the Advantage A_t

$$A(a_t, s_t) = r(a_t)$$

**Given that $\gamma = 1$ and the environment is fully deterministic.*

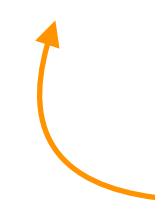
Computing the Advantage A_t

$$A(a_t, s_t) = r(a_t)$$

**Given that $\gamma = 1$ and the environment is fully deterministic.*

Computing the Advantage A_t

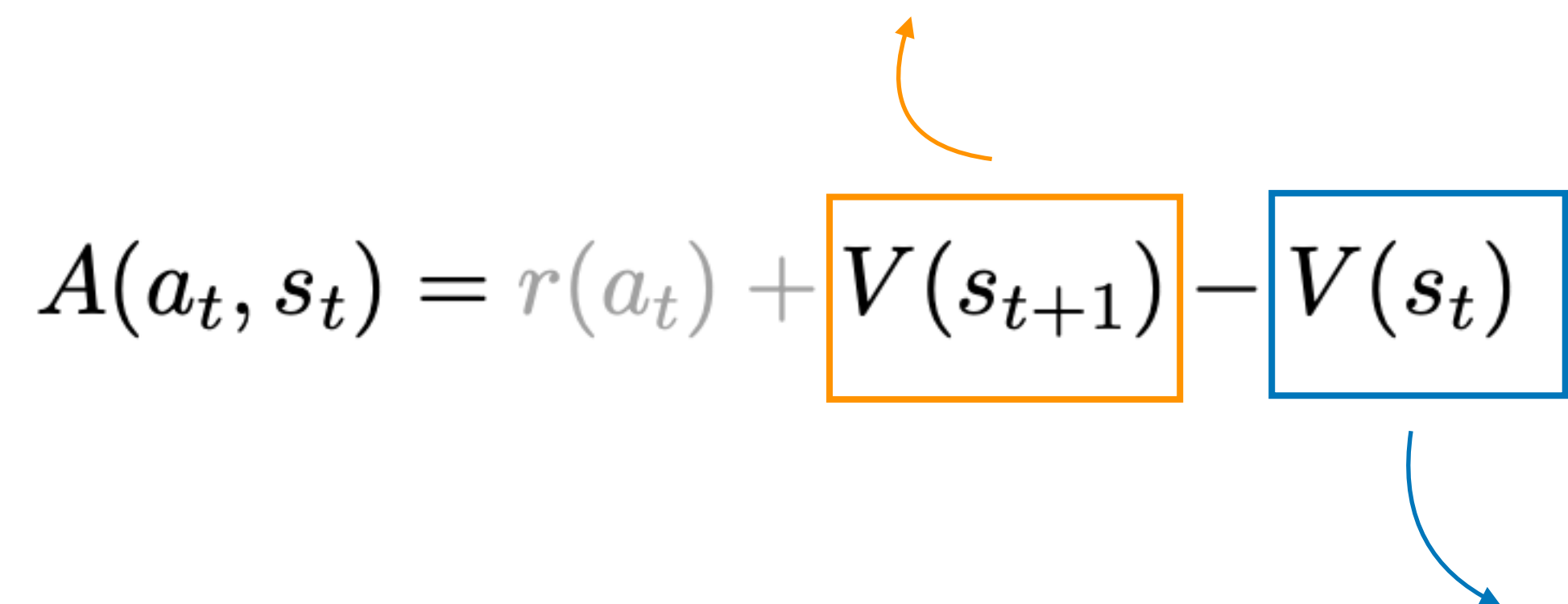
On average, how well we perform **after** a_t
i.e. sampling from $s_{t+1} = [s_t; a_t]$

$$A(a_t, s_t) = r(a_t) + \boxed{V(s_{t+1})}$$


**Given that $\gamma = 1$ and the environment is fully deterministic.*

Computing the Advantage A_t

On average, how well we perform **after** a_t
i.e. sampling from $s_{t+1} = [s_t; a_t]$

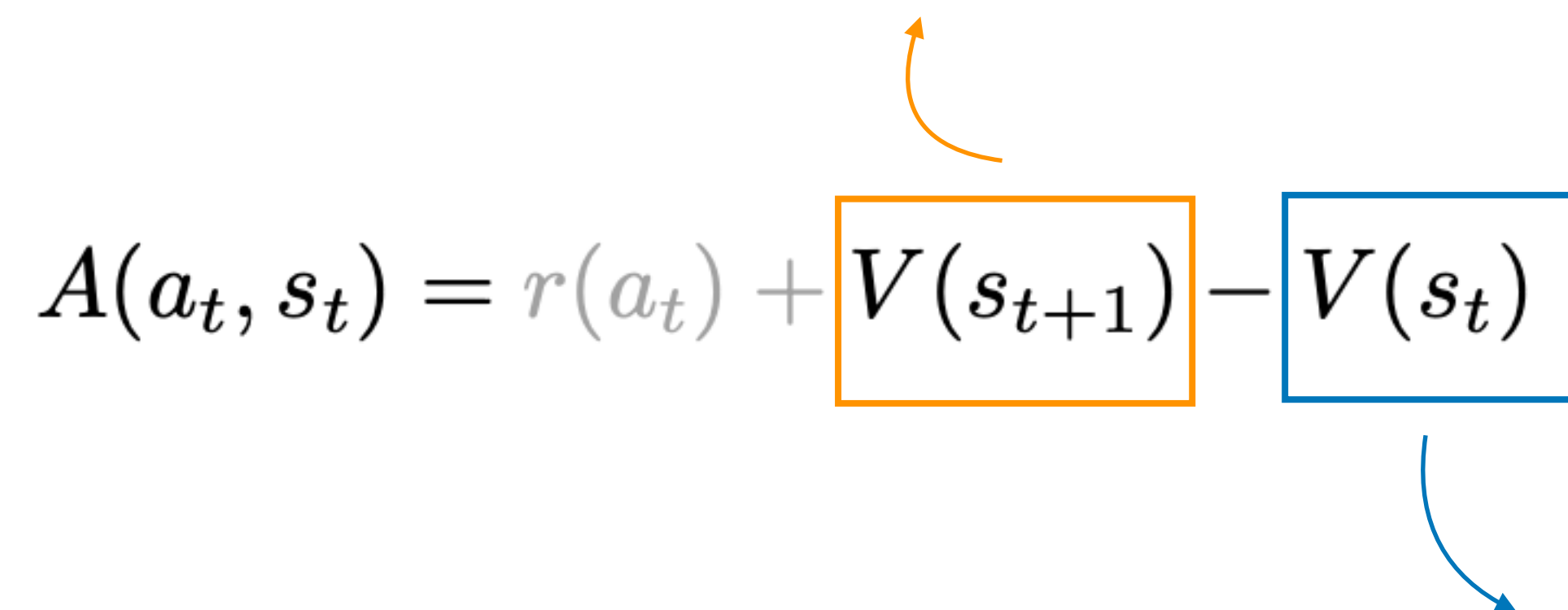
$$A(a_t, s_t) = r(a_t) + \boxed{V(s_{t+1})} - \boxed{V(s_t)}$$
An orange arrow points from the text 'i.e. sampling from s_{t+1} = [s_t; a_t]' to the boxed term V(s_{t+1}). A blue arrow points from the boxed term V(s_t) to the text 'On average, how well we already perform in s_t'.

On average, how well we **already** perform **in** s_t

**Given that $\gamma = 1$ and the environment is fully deterministic.*

Computing the Advantage A_t

On average, how well we perform **after** a_t
i.e. sampling from $s_{t+1} = [s_t; a_t]$

$$A(a_t, s_t) = r(a_t) + \boxed{V(s_{t+1})} - \boxed{V(s_t)}$$
The diagram shows the equation $A(a_t, s_t) = r(a_t) + V(s_{t+1}) - V(s_t)$. The term $V(s_{t+1})$ is enclosed in an orange box, and an orange arrow points from the text "i.e. sampling from $s_{t+1} = [s_t; a_t]$ " above to this box. The term $V(s_t)$ is enclosed in a blue box, and a blue arrow points from this box to the text "On average, how well we already perform in s_t " below.

On average, how well we **already** perform in s_t

- A formulation used in many RL optimization algorithms like **PPO**

**Given that $\gamma = 1$ and the environment is fully deterministic.*

Value of a state = the **expected reward** if you start in that state

Find all values of x such that $\frac{1}{x-1} + \frac{2x}{x-1} = 5$.

1. Combining the fractions on the left gives
$$\frac{1+2x}{x-1} = 5.$$

2. Multiply both sides by $x-1$ to clear the fraction: $1+2x = 5(x-1)$.

3. Distribute the 5 on the right side:
 $1+2x = 5x-5$.

4. Rearrange to solve for x :
 $1+5 = 5x-2x \implies 6 = 3x \implies x = 2.$



Value of a state = the expected reward if you start in that state

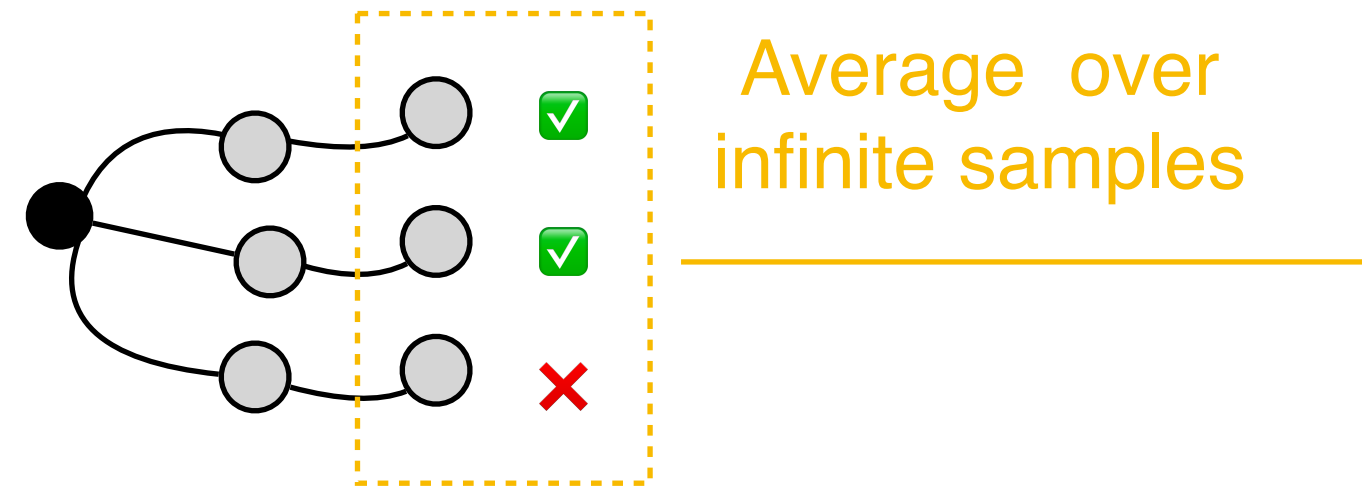
Find all values of x such that $\frac{1}{x-1} + \frac{2x}{x-1} = 5$.

1. Combining the fractions on the left gives
 $\frac{1+2x}{x-1} = 5$.

2. Multiply both sides by $x-1$ to clear the fraction: $1+2x = 5(x-1)$.

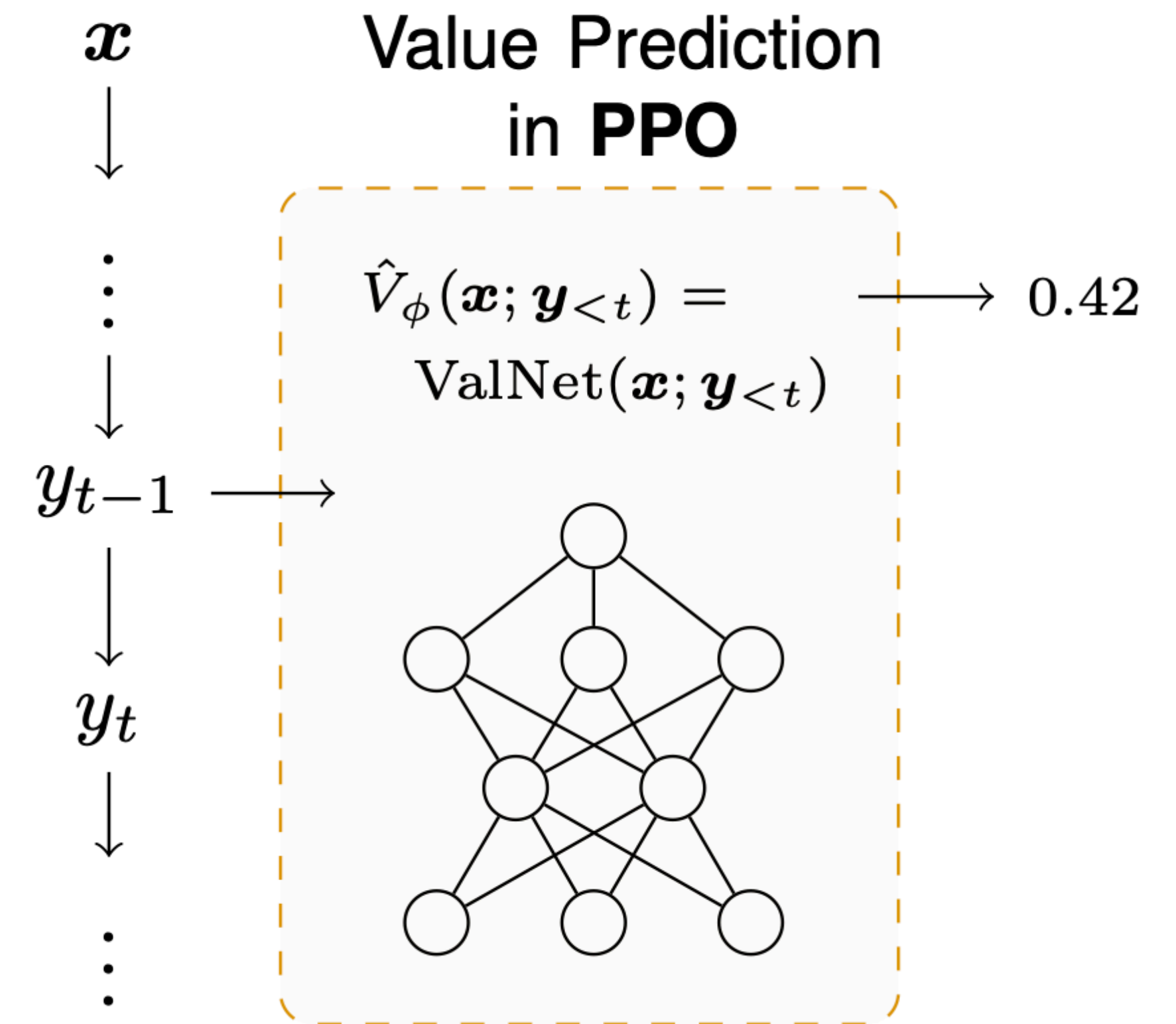
3. Distribute the 5 on the right side:
 $1+2x = 5x-5$.

4. Rearrange to solve for x :
 $1+5 = 5x-2x \implies 6 = 3x \implies x = 2$



PPO

- PPO as in RLHF pipeline (Ouyang et al. 2022) is the SOTA algorithm for RL-based finetuning of LLMs.
- PPO uses a value network to tackle credit assignment.
- ValNet is the same size as and initialized from LLM itself.



PPO uses value network to predict values

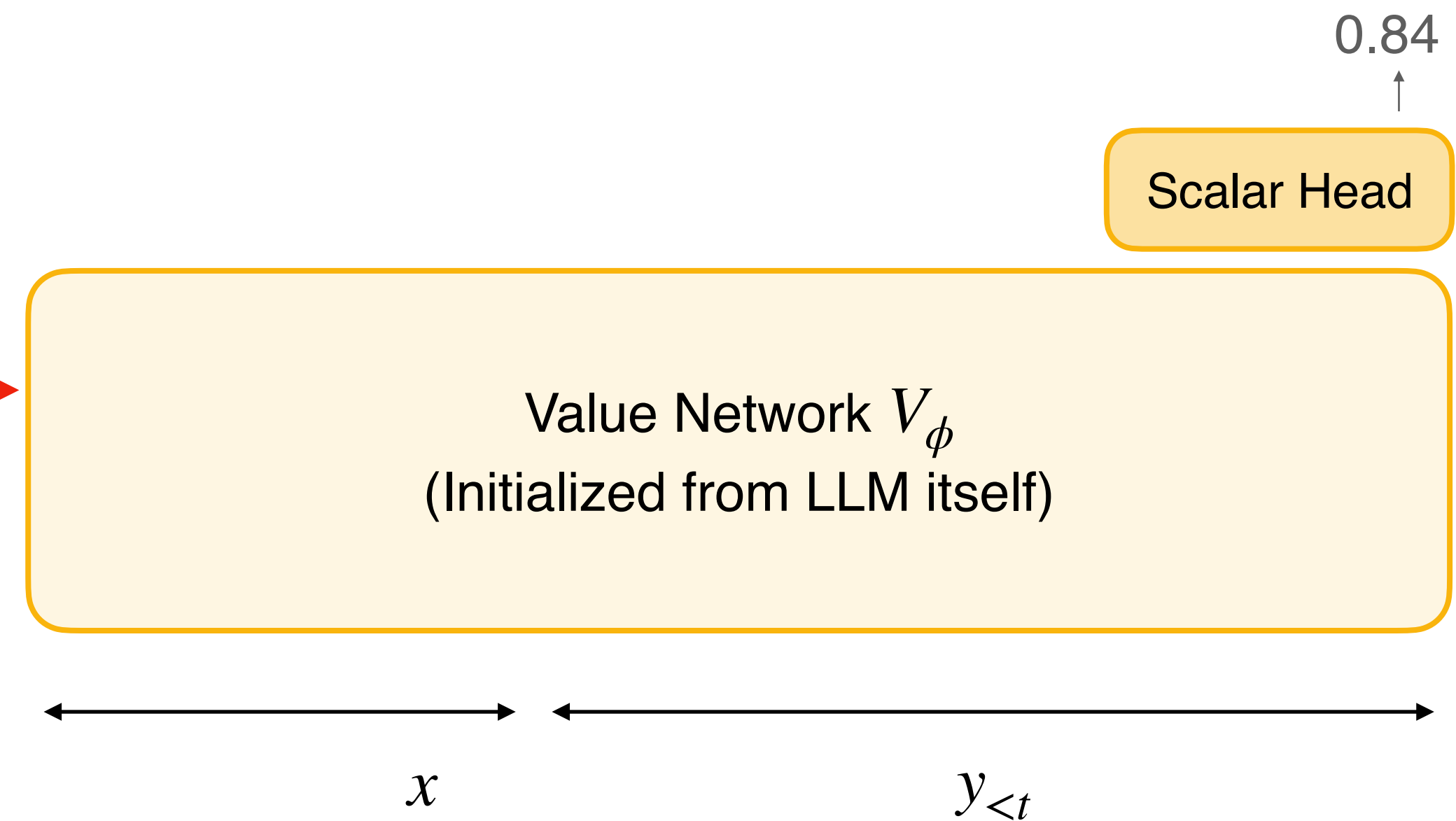
Find all values of x such that $\frac{1}{x-1} + \frac{2x}{x-1} = 5$.

1. Combining the fractions on the left gives $\frac{1+2x}{x-1} = 5$.

2. Multiply both sides by $x-1$ to clear the fraction: $1+2x = 5(x-1)$.

3. Distribute the 5 on the right side: $1+2x = 5x-5$.

4. Rearrange to solve for x : $1+5 = 5x-2x \implies 6 = 3x \implies x = 2$.



- **RQ1:**
How accurately do **value networks** actually perform for LLM post-training?

- **RQ1:**
How accurately do **value networks** actually perform for LLM post-training?

They provide **inaccurate estimates**.

- **RQ2:**
If we improve credit assignment, to what extent would it enhance LLM performance?

- **RQ1:**
How accurately do **value networks** actually perform for LLM post-training?

(a) They provide inaccurate estimates.

- **RQ2:**
If we **improve credit assignment**, to **what extent** would it **enhance LLM performance**?

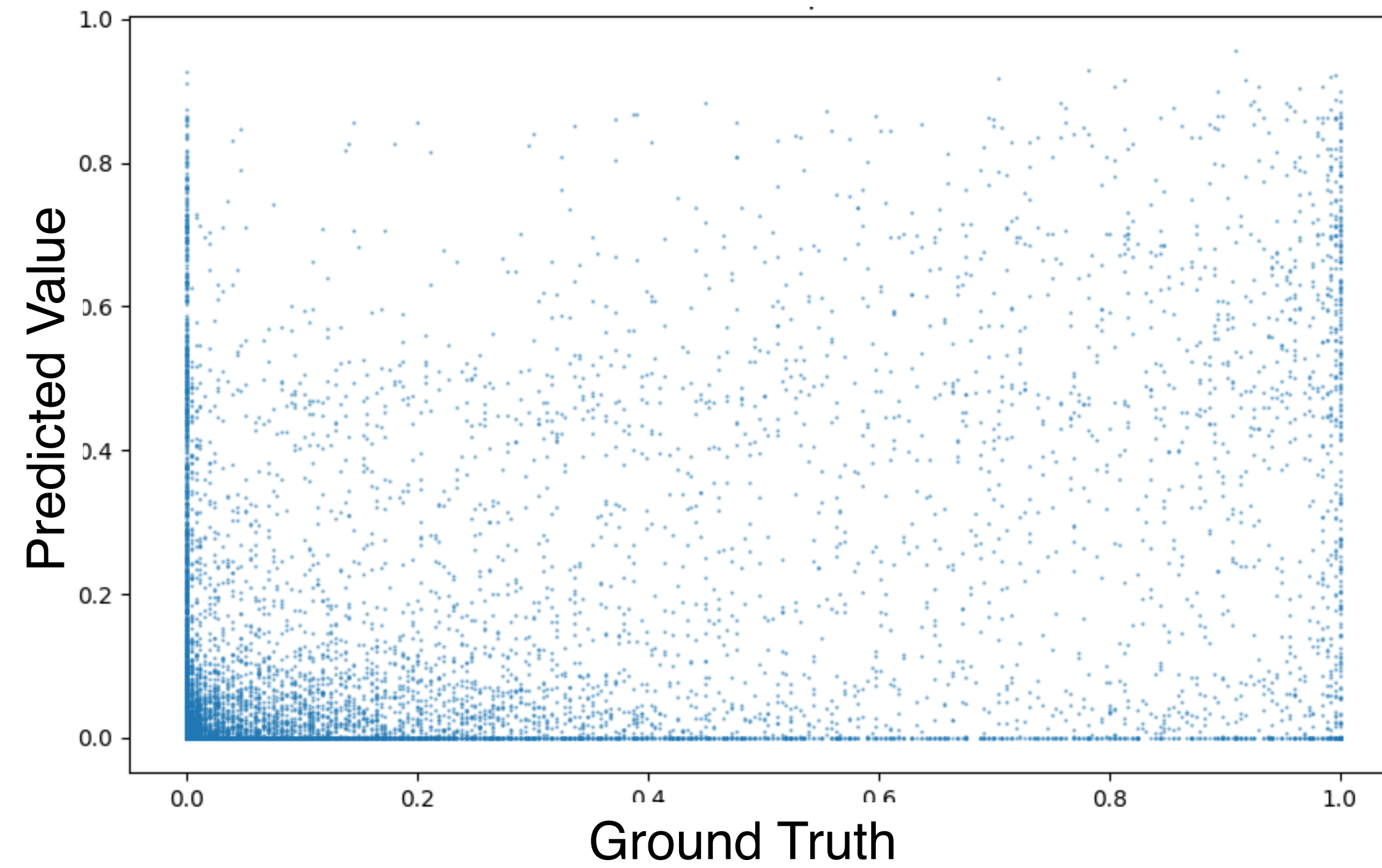
(a) We introduce **VinePPO** that uses MC samples to improve credit assignment.
(b) It outperforms standard PPO and RL-free methods (e.g. DPO)
(c) The effect of enhanced credit assignment is so important that it surpasses PPO in **fewer steps (9x)**, **less wall-clock time (3x)**, and **lower KL**.

Experimental Setup

- **Datasets:**
 - MATH and GSM8K
 - Final answer is given but not proofs
- **Models**
 - DeepSeekMath 7B and RhoMath 1.1B
- **Baselines**
 - GRPO, Standard PPO, and RL-free methods: DPO+, RestEM
- **Hyperparameters**
 - Search done on PPO.
 - VinePPO shares the exact same ones; $K=9$

Accuracy of Value Networks is poor

ValNet



Our VinePPO uses Monte Carlo Estimation

Find all values of x such that $\frac{1}{x-1} + \frac{2x}{x-1} = 5$.

1. Combining the fractions on the left gives $\frac{1+2x}{x-1} = 5$.
2. Multiply both sides by $x-1$ to clear the fraction: $1+2x = 5(x-1)$.
3. Distribute the 5 on the right side: $1+2x = 5x-5$.
4. Rearrange to solve for x : $1+5 = 5x-2x \implies 6 = 3x \implies x = 2$.

Our VinePPO uses Monte Carlo Estimation

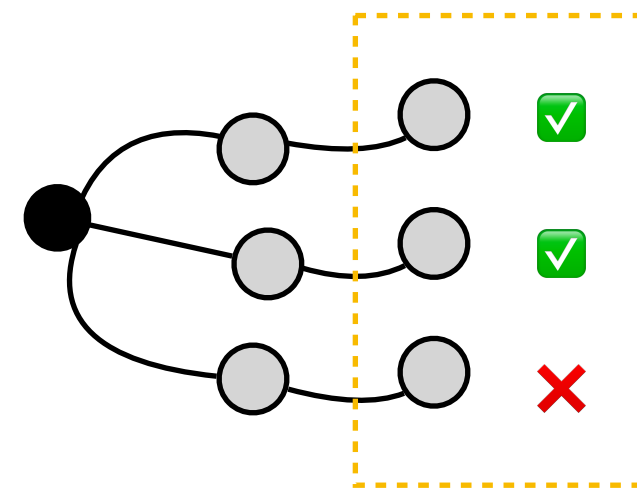
Find all values of x such that $\frac{1}{x-1} + \frac{2x}{x-1} = 5$.

1. Combining the fractions on the left gives
 $\frac{1+2x}{x-1} = 5$.

2. Multiply both sides by $x-1$ to clear the fraction: $1+2x = 5(x-1)$.

3. Distribute the 5 on the right side:
 $1+2x = 5x-5$.

4. Rearrange to solve for x :
 $1+5 = 5x-2x \implies 6 = 3x \implies x = 2$.



$V(\text{"Find all ... 1. Combing the frac... = 5"}) = 0.66$

Our VinePPO uses Monte Carlo Estimation

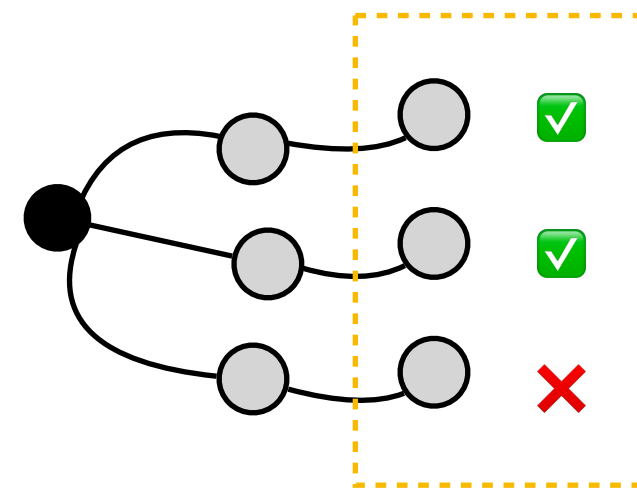
Find all values of x such that $\frac{1}{x-1} + \frac{2x}{x-1} = 5$.

1. Combining the fractions on the left gives $\frac{1+2x}{x-1} = 5$.

2. Multiply both sides by $x-1$ to clear the fraction: $1+2x = 5(x-1)$.

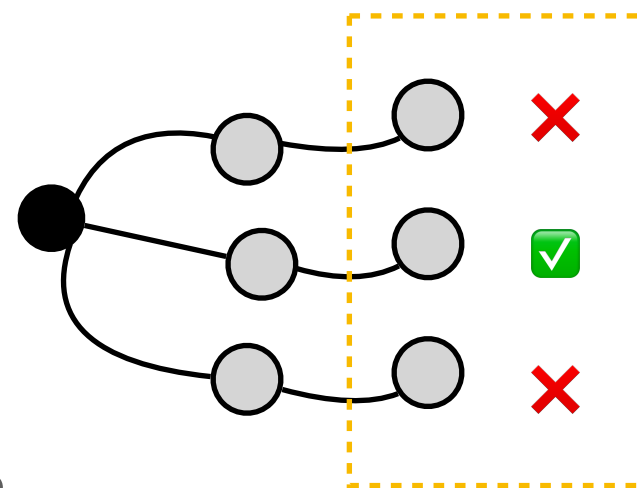
3. Distribute the 5 on the right side: $1+2x = 5x-5$.

4. Rearrange to solve for x : $1+5 = 5x-2x \implies 6 = 3x \implies x = 2$.



Average

$V(\text{"Find all ... 1. Combing the frac... = 5"}) = 0.66$



Average

$V(\text{"Find all ... 3. Distribute the... -5"}) = 0.33$

Our VinePPO uses Monte Carlo Estimation

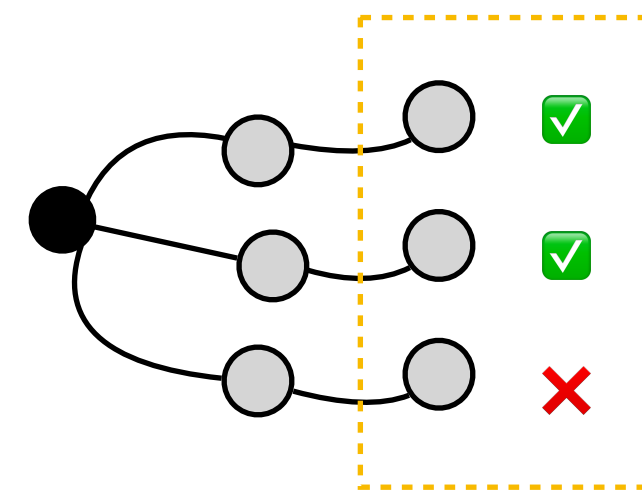
Find all values of x such that $\frac{1}{x-1} + \frac{2x}{x-1} = 5$.

1. Combining the fractions on the left gives $\frac{1+2x}{x-1} = 5$.

2. Multiply both sides by $x-1$ to clear the fraction: $1+2x = 5(x-1)$.

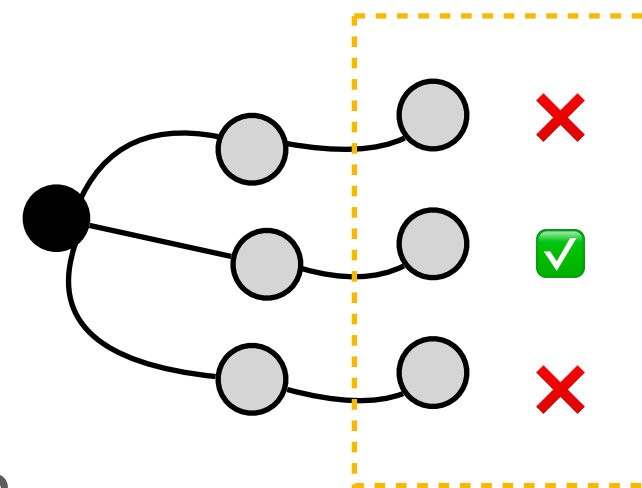
3. Distribute the 5 on the right side: $1+2x = 5x-5$.

4. Rearrange to solve for x : $1+5 = 5x-2x \implies 6 = 3x \implies x = 2$.



Average

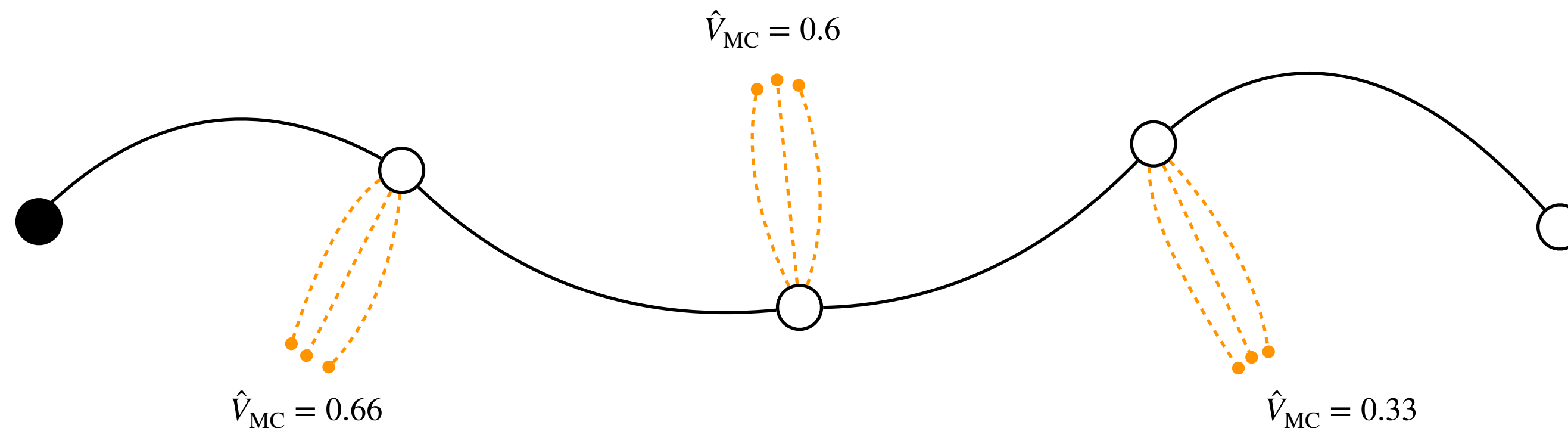
$V(\text{"Find all ... 1. Combing the frac... = 5"}) = 0.66$



Average

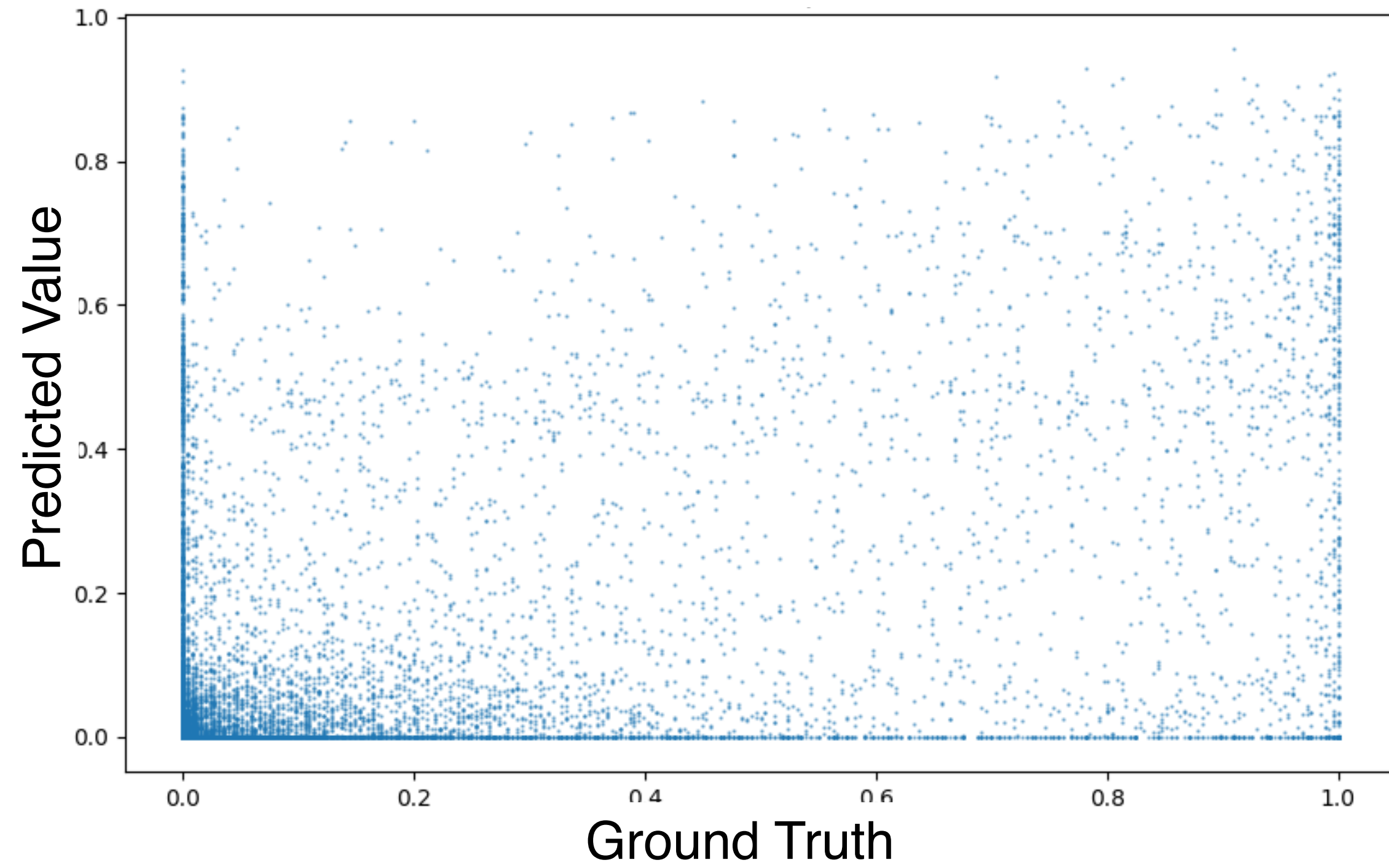
$V(\text{"Find all ... 3. Distribute the... -5"}) = 0.33$

Trajectory τ

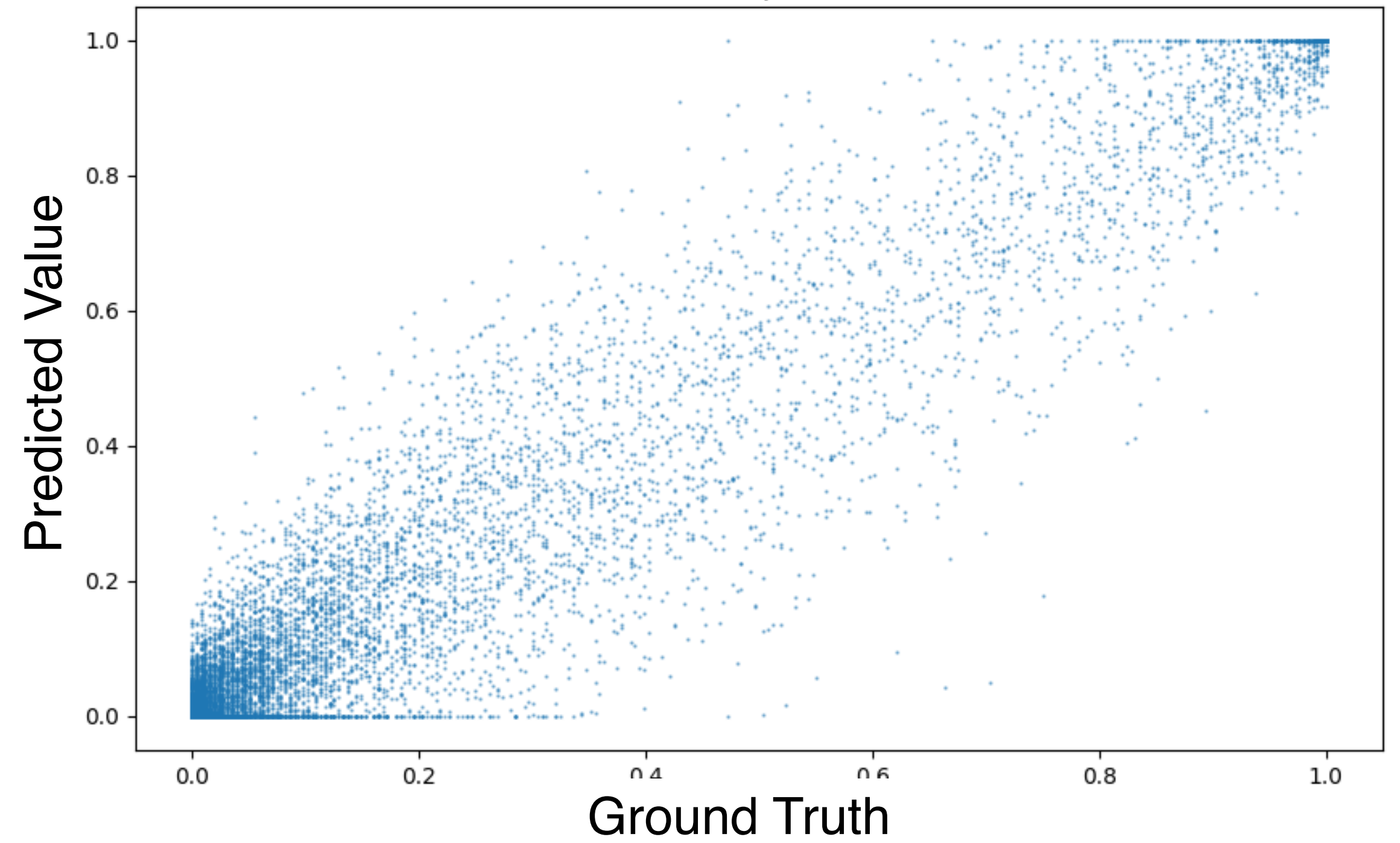


Our VinePPO uses Monte Carlo Estimation

PPO ValNet

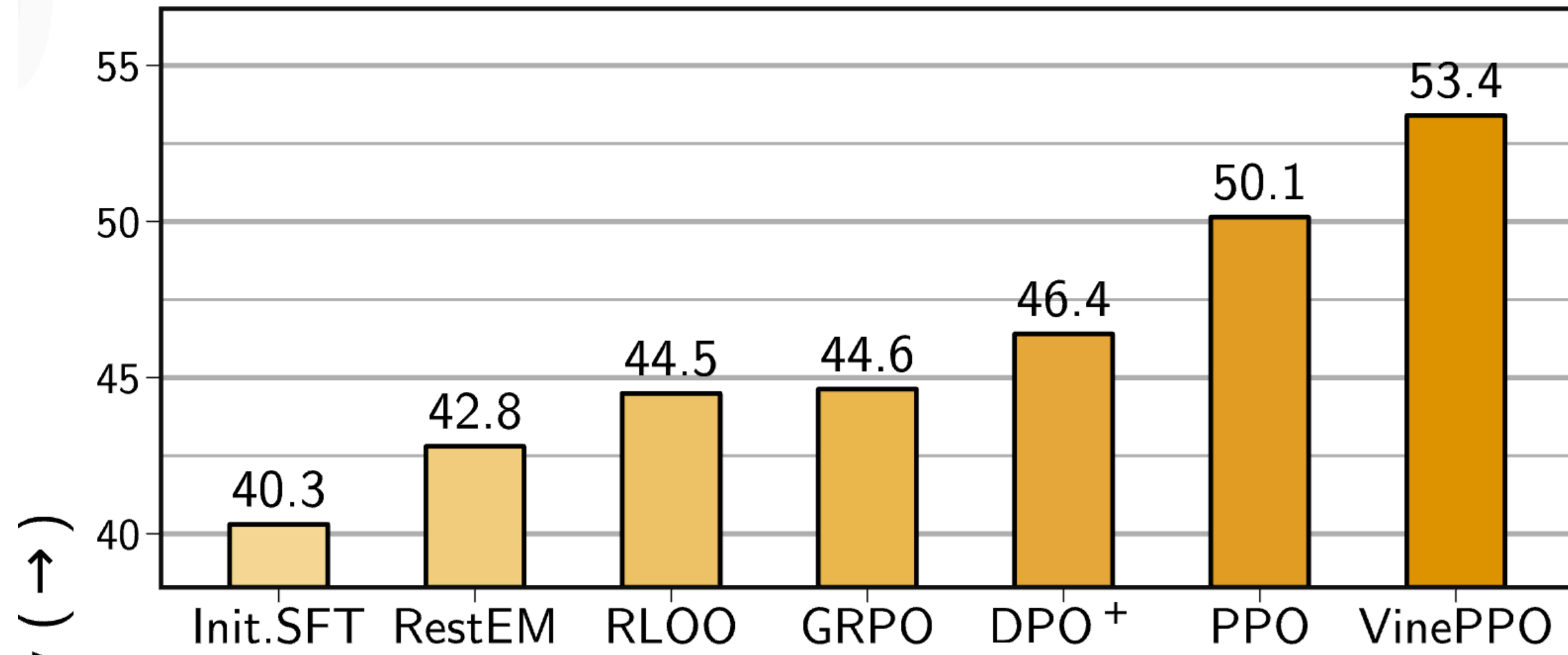


VinePPO
(9 rollouts per step) in

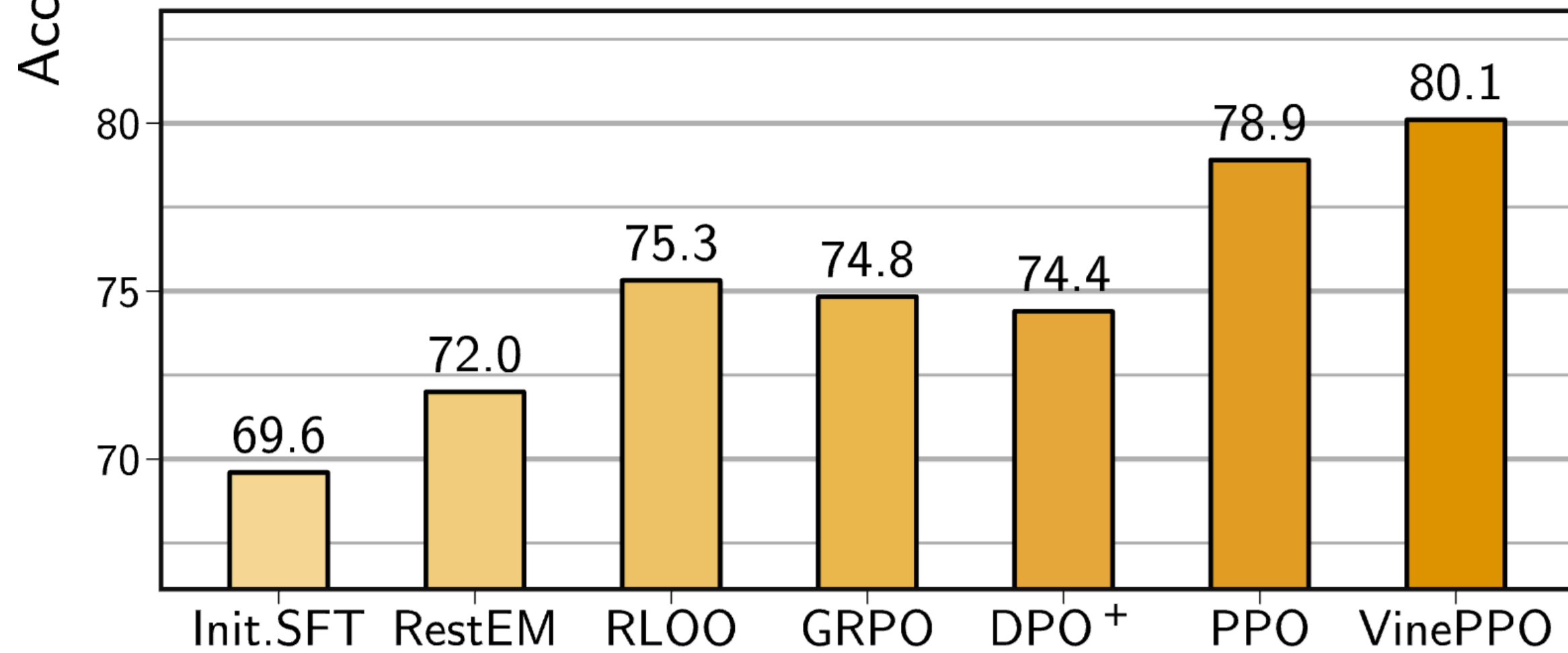


GSM8K

RhoMath 1.1B

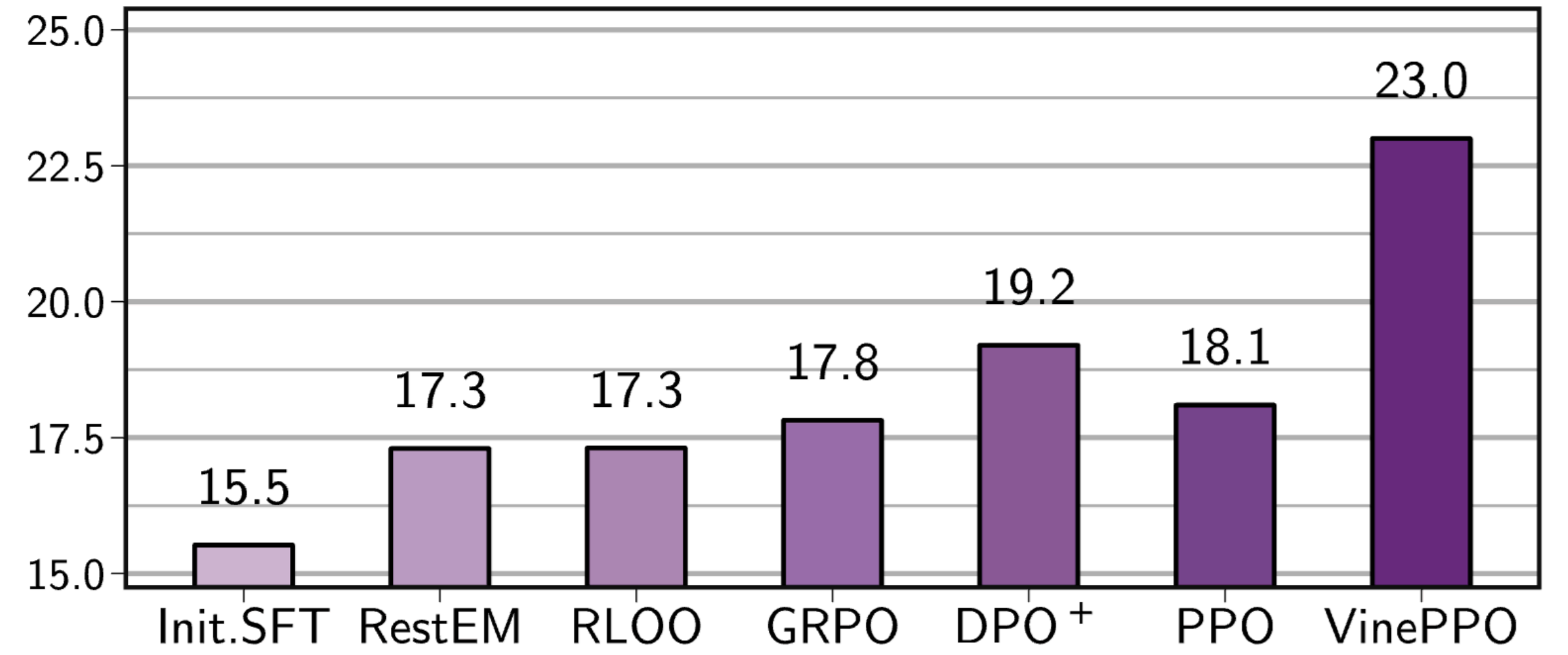


DeepSeekMath 7B

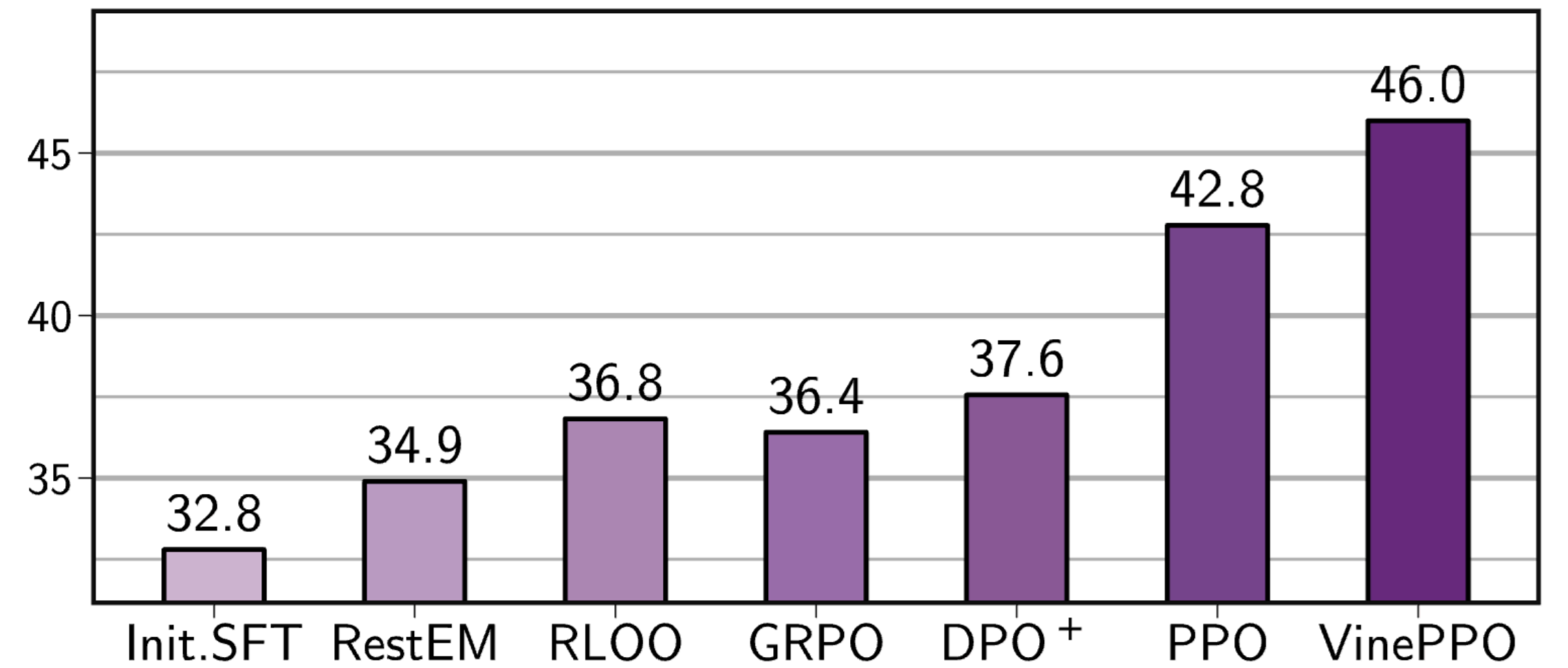


MATH

RhoMath 1.1B



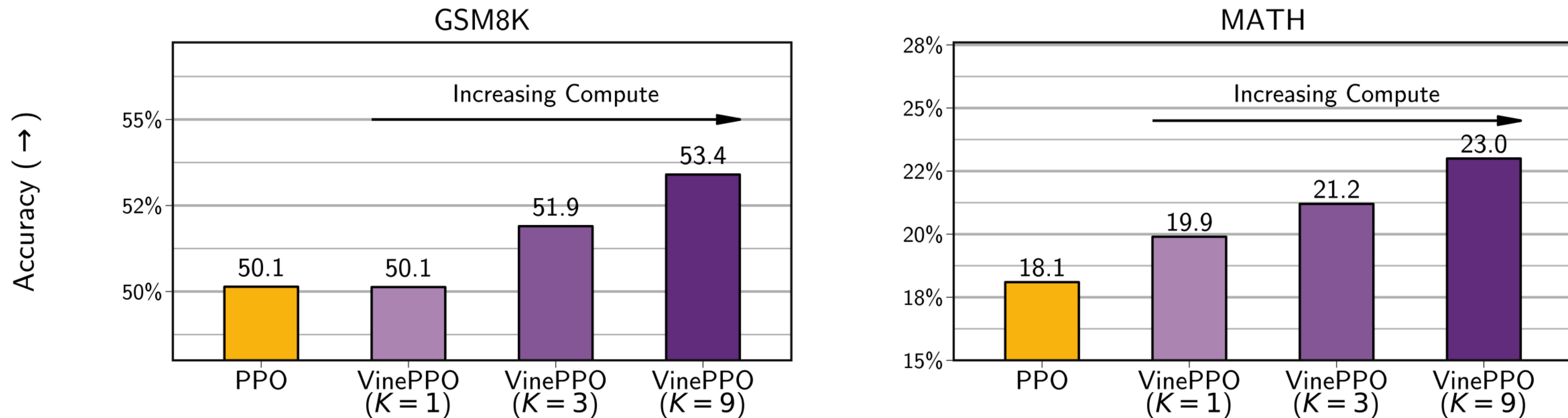
DeepSeekMath 7B



The effect of Improved Credit Assignment

Task Performance (ablating the number of MC samples)

$$\hat{V}_{\text{MC}}(s_t) := \frac{1}{K} \sum_{k=1}^K R(\tau^k), \quad \text{where } \tau^1, \dots, \tau^K \sim \pi_{\theta}(\cdot | s_t).$$

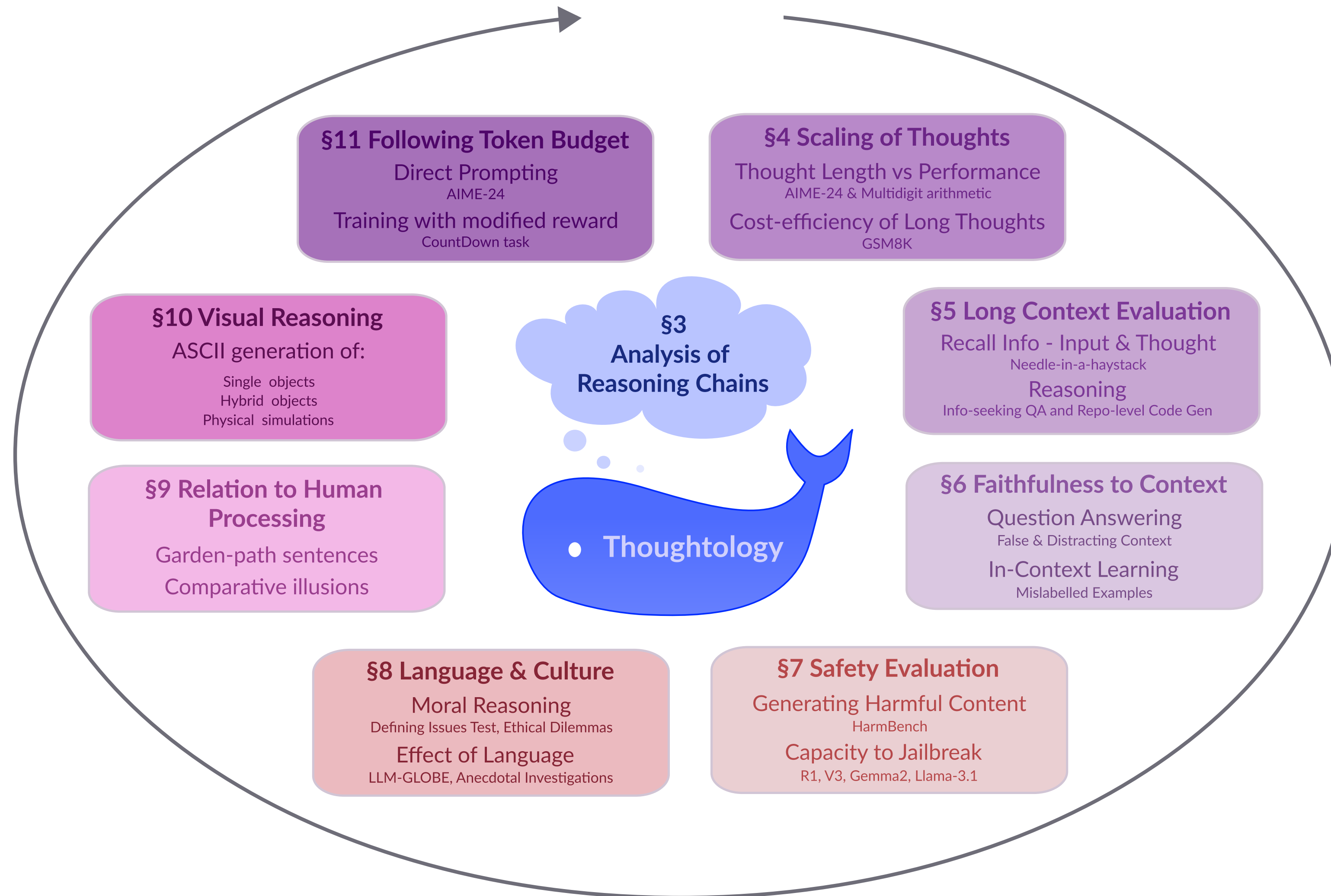


- Increasing more compute leads to more accurate value estimate and better task performance
- In some cases even K=1 perform better or on par with PPO.

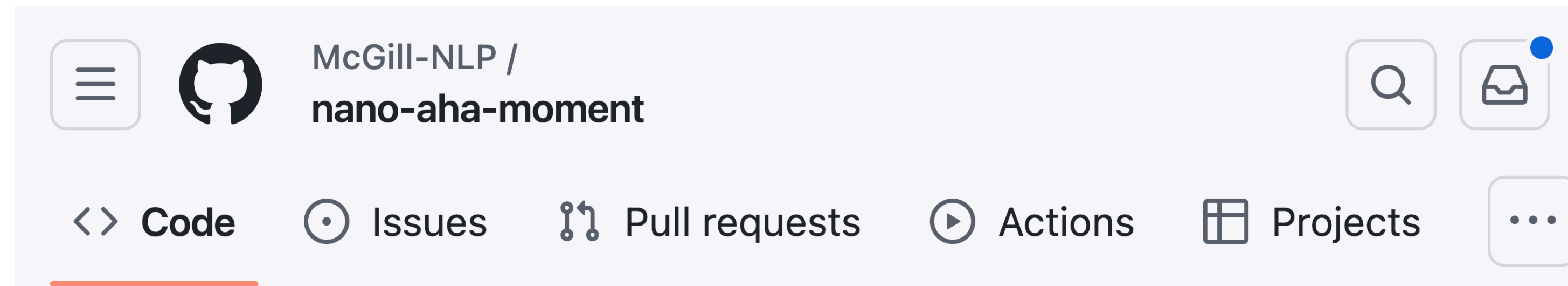
Summary

- Credit assignment has a profound impact on reasoning
- VinePPO provides a better performance in the same budget.
- A promising alternative for post-training frontier LLMs.
- VinePPO also provides a straightforward scaling axis — training-time search

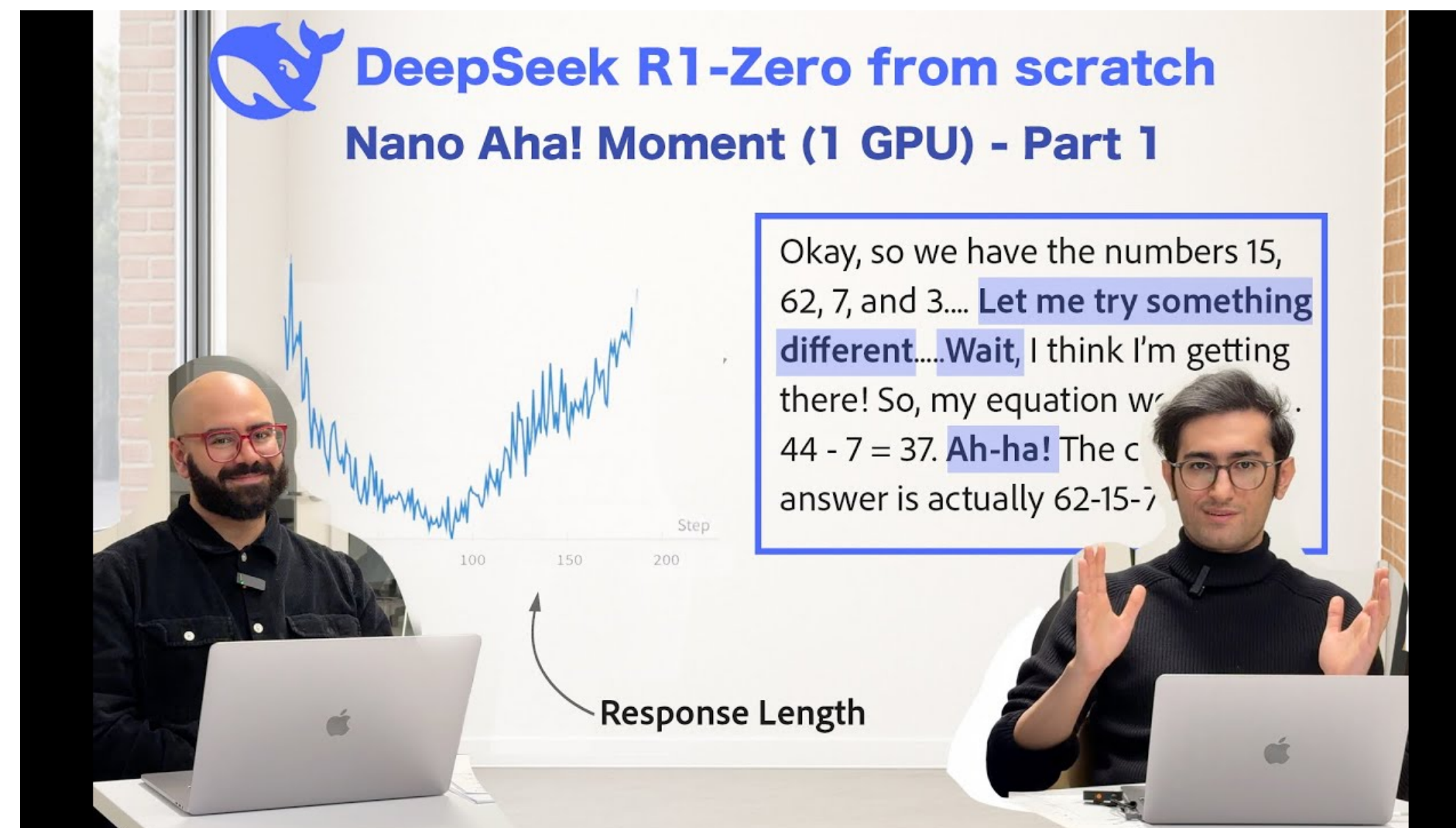
DeepSeek-R1 Thoughtology



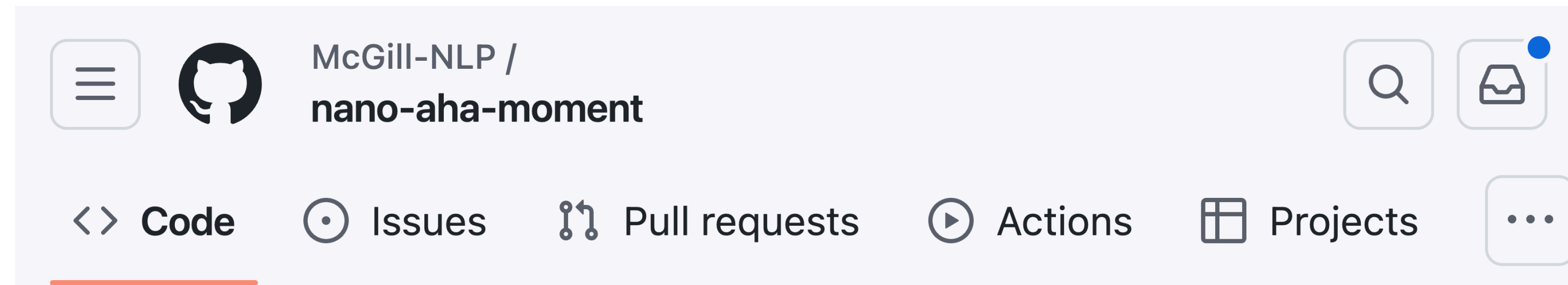
Many more findings in paper



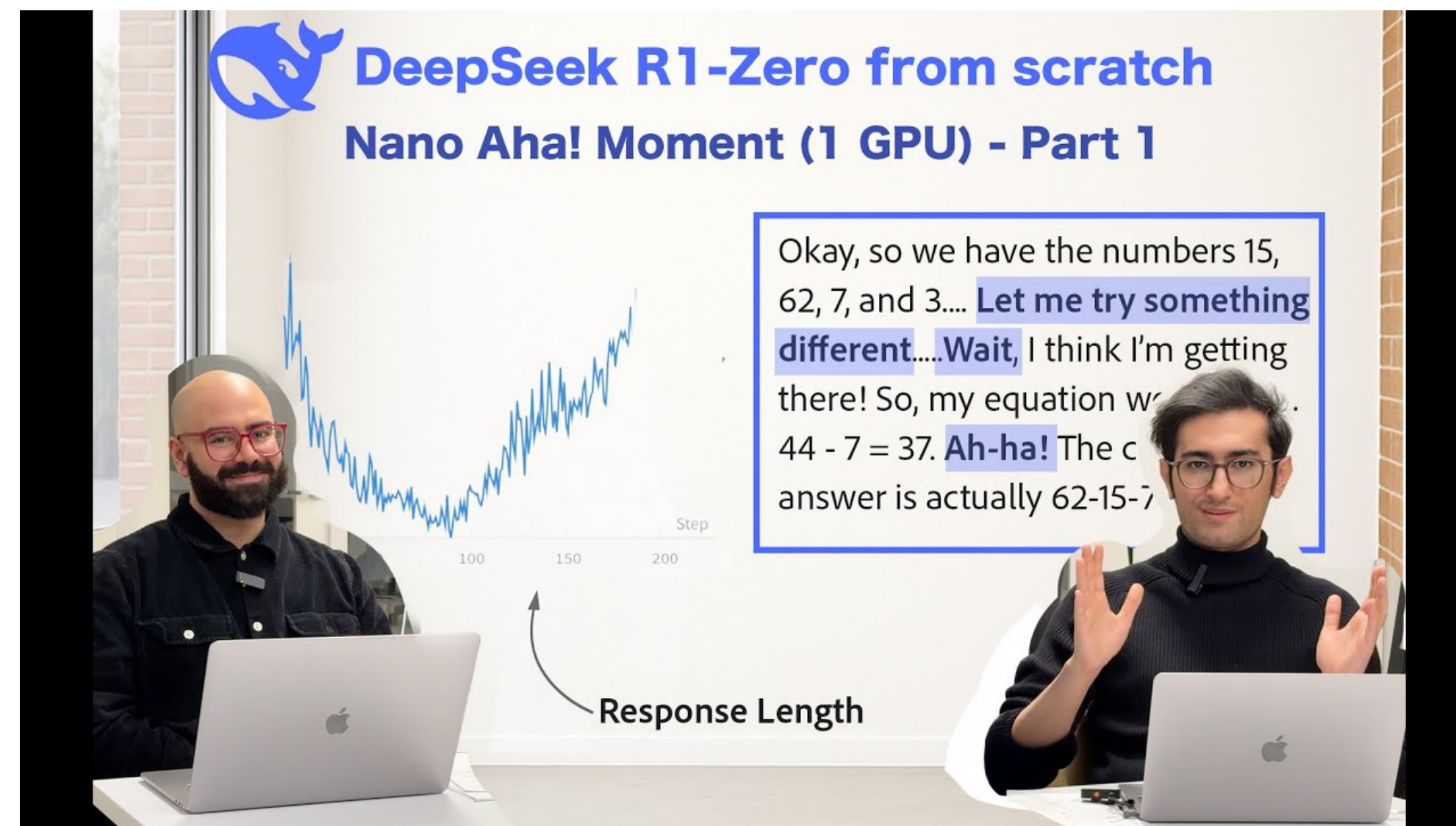
Single GPU, From Scratch (No RL Library), Efficient, Full Parameter Tuning
Implementation of DeepSeek R1-Zero style training.



<https://github.com/McGill-NLP/nano-aha-moment>

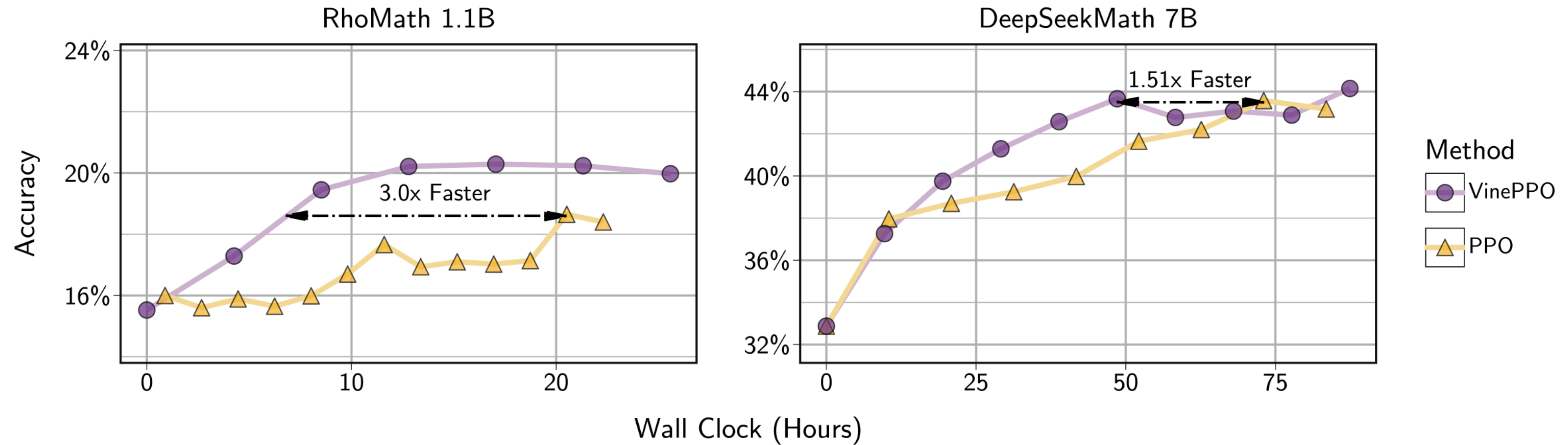


Single GPU, From Scratch (No RL Library), Efficient, Full Parameter Tuning
Implementation of DeepSeek R1-Zero style training.



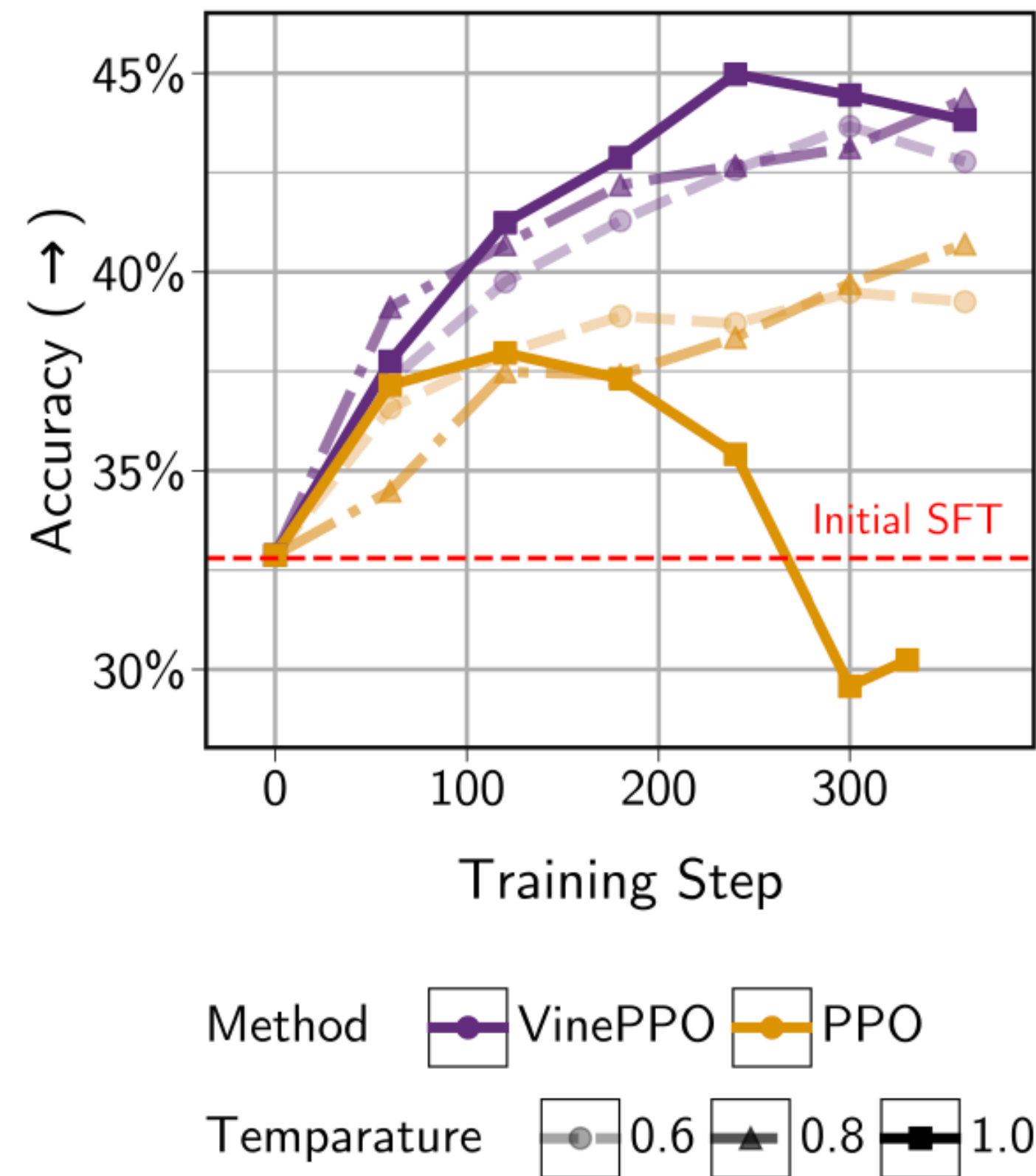
<https://github.com/McGill-NLP/nano-aha-moment>

The effect of Improved Credit Assignment Computational Efficiency



- VinePPO and PPO require different resources: GPU memory vs More LLM Inferences
- VinePPO is generally slower per iteration (5x for 1.1B and 2x for 7B).
- VinePPO is faster in wall-clock time, is the only option if memory is constrained.

The effect of Improved Credit Assignment Temperature Tolerance

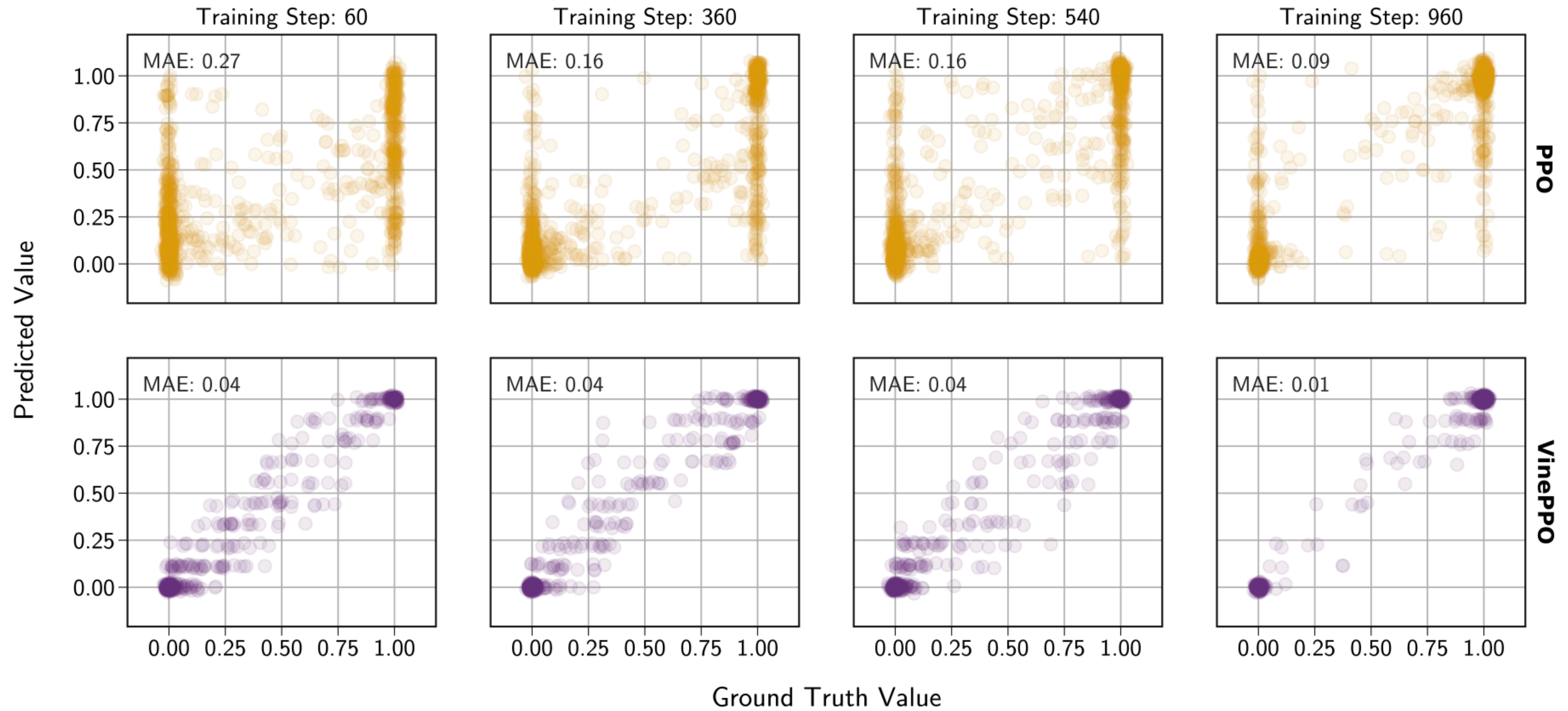


- Value network in Standard PPO cannot tolerate more diverse trajectories in higher temperatures

Value Prediction Analysis

Why such performance gap exists?

Value Prediction Analysis Distribution



Value Prediction Analysis

Accuracy

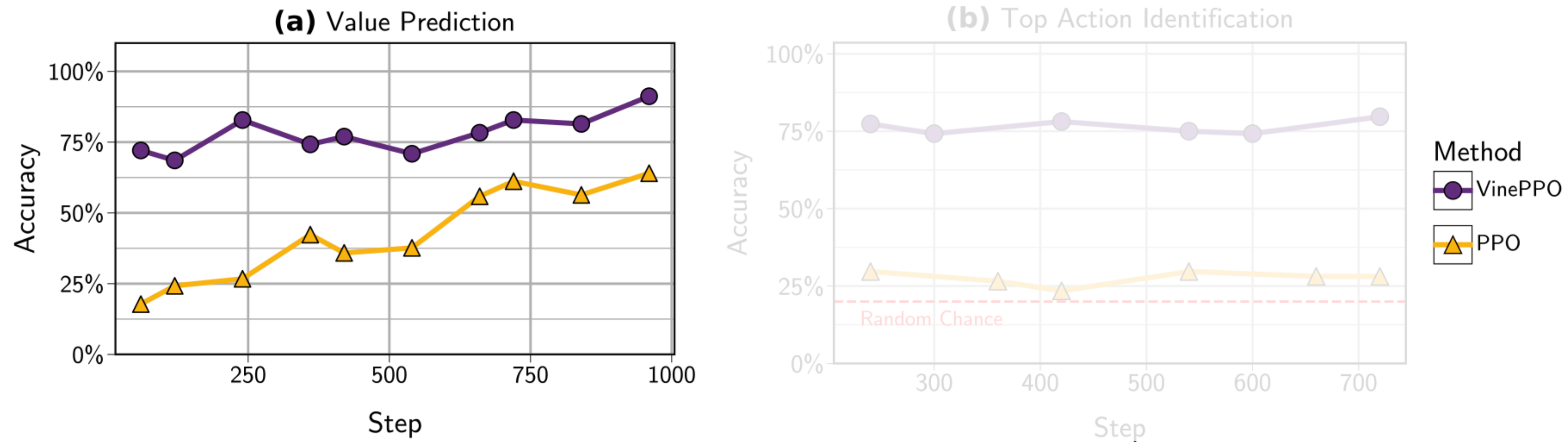


Figure 11: **(a)** Value prediction accuracy formulated as a classification problem, where a prediction is considered correct if it falls within 0.05 of the ground truth. **(b)** Accuracy of identifying the top action in a set of five possible next states. VinePPO consistently outperforms the value network.

Value Prediction Analysis

Accuracy

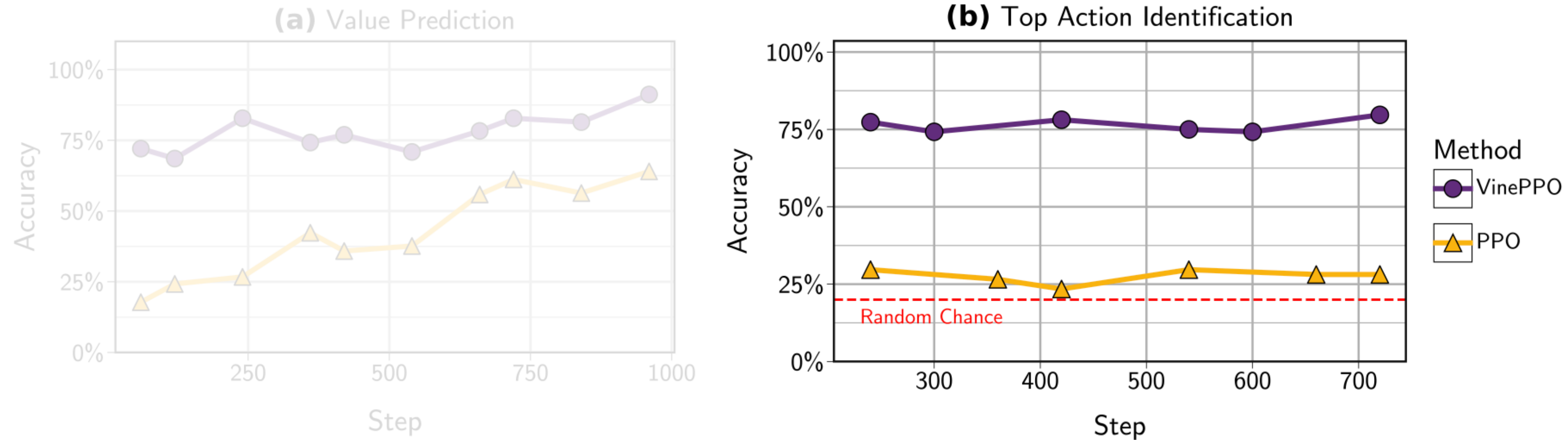


Figure 11: **(a)** Value prediction accuracy formulated as a classification problem, where a prediction is considered correct if it falls within 0.05 of the ground truth. **(b)** Accuracy of identifying the top action in a set of five possible next states. VinePPO consistently outperforms the value network.

Value Prediction Analysis

Accuracy

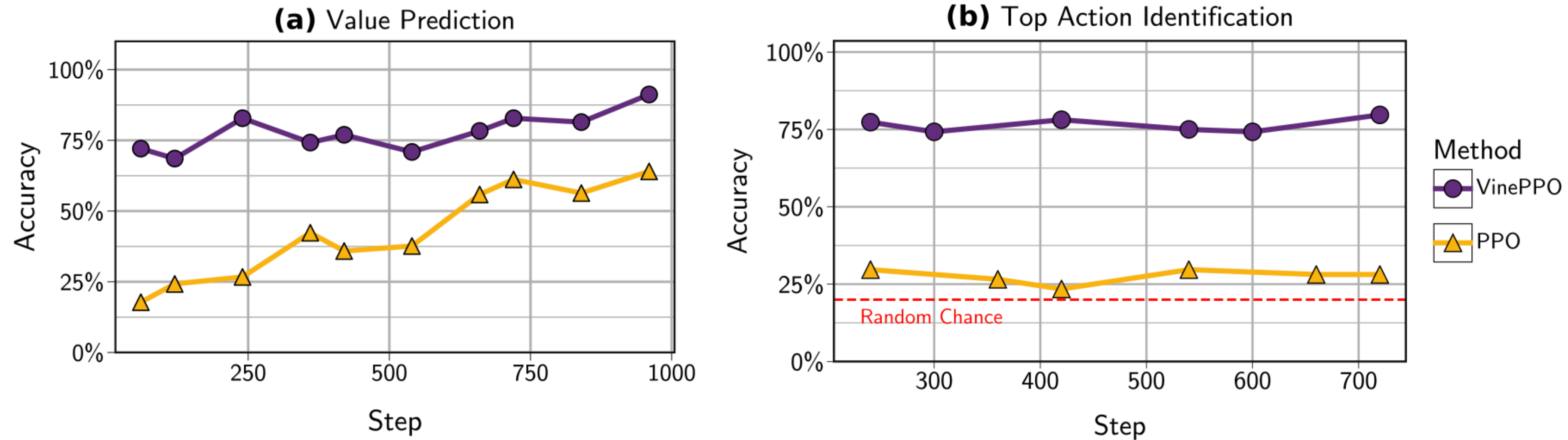
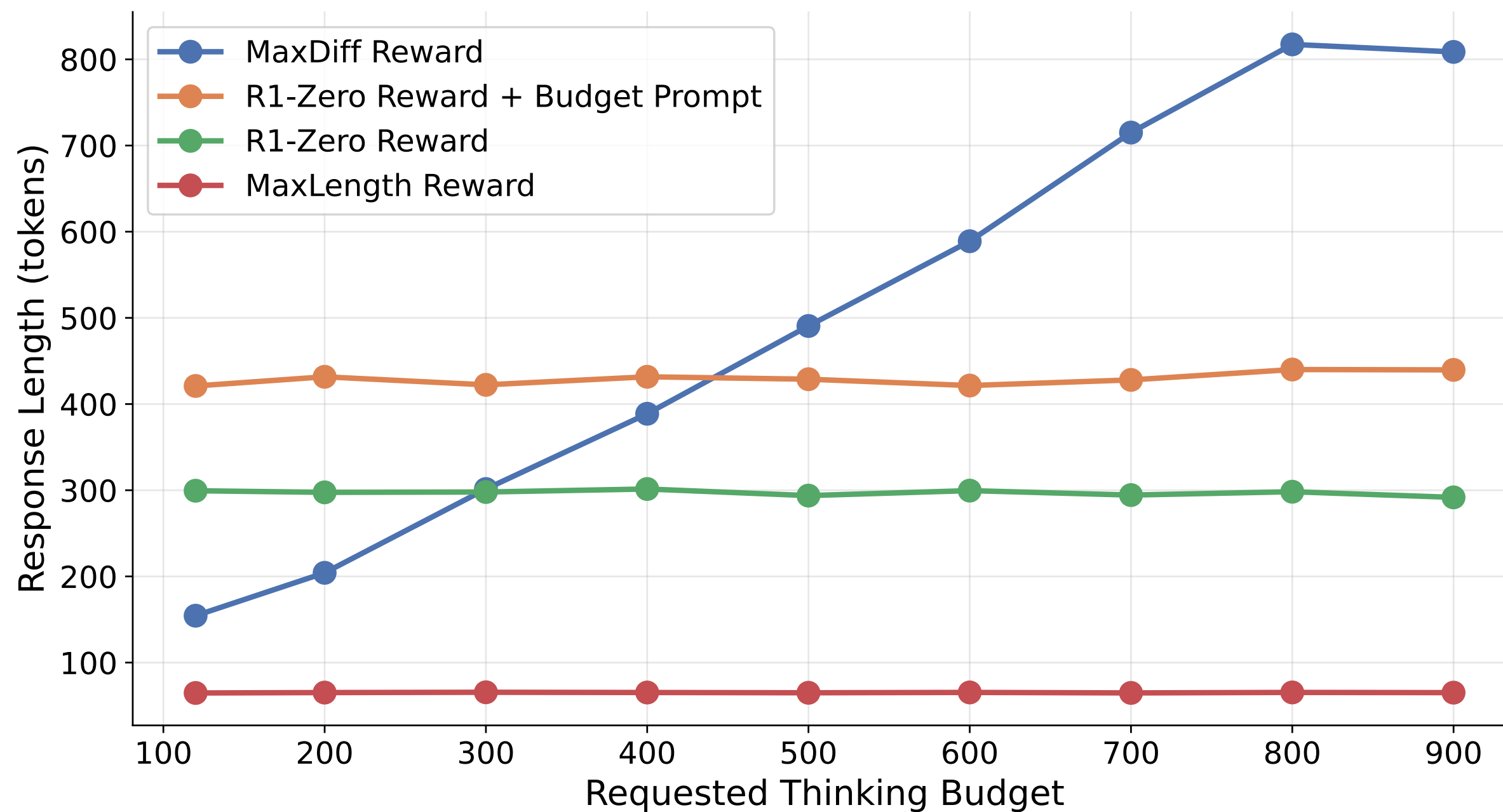


Figure 11: **(a)** Value prediction accuracy formulated as a classification problem, where a prediction is considered correct if it falls within 0.05 of the ground truth. **(b)** Accuracy of identifying the top action in a set of five possible next states. VinePPO consistently outperforms the value network.

Fixing length budget

$$\mathcal{R}'(y, x) = \mathcal{R}_{\text{Format}}(y, x) + \mathcal{R}_{\text{Correctness}}(y, x) + \lambda \mathcal{R}_{\text{Length}}(y, x)$$

Response Lengths Across Experiments



Accuracy Across Experiments

