



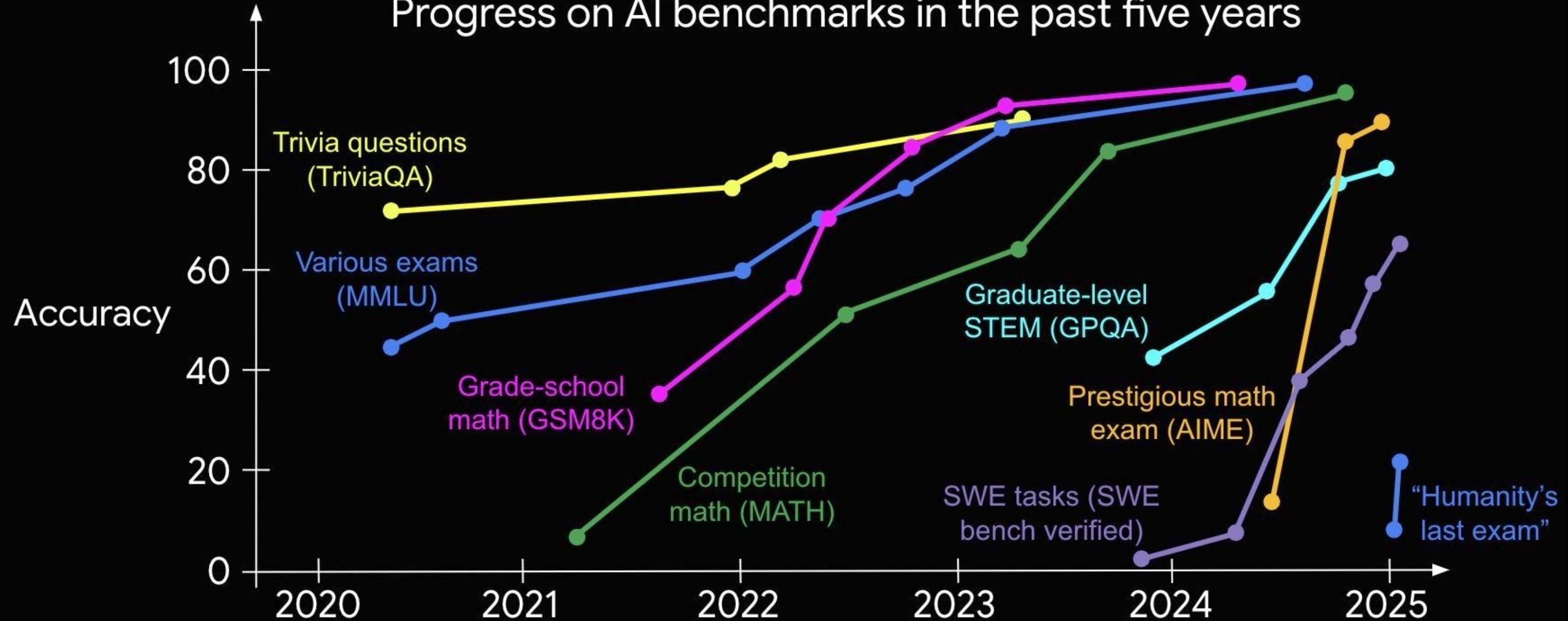
Learning to Reason in Agents

Zhiqing Sun

Apr. 24, 2025

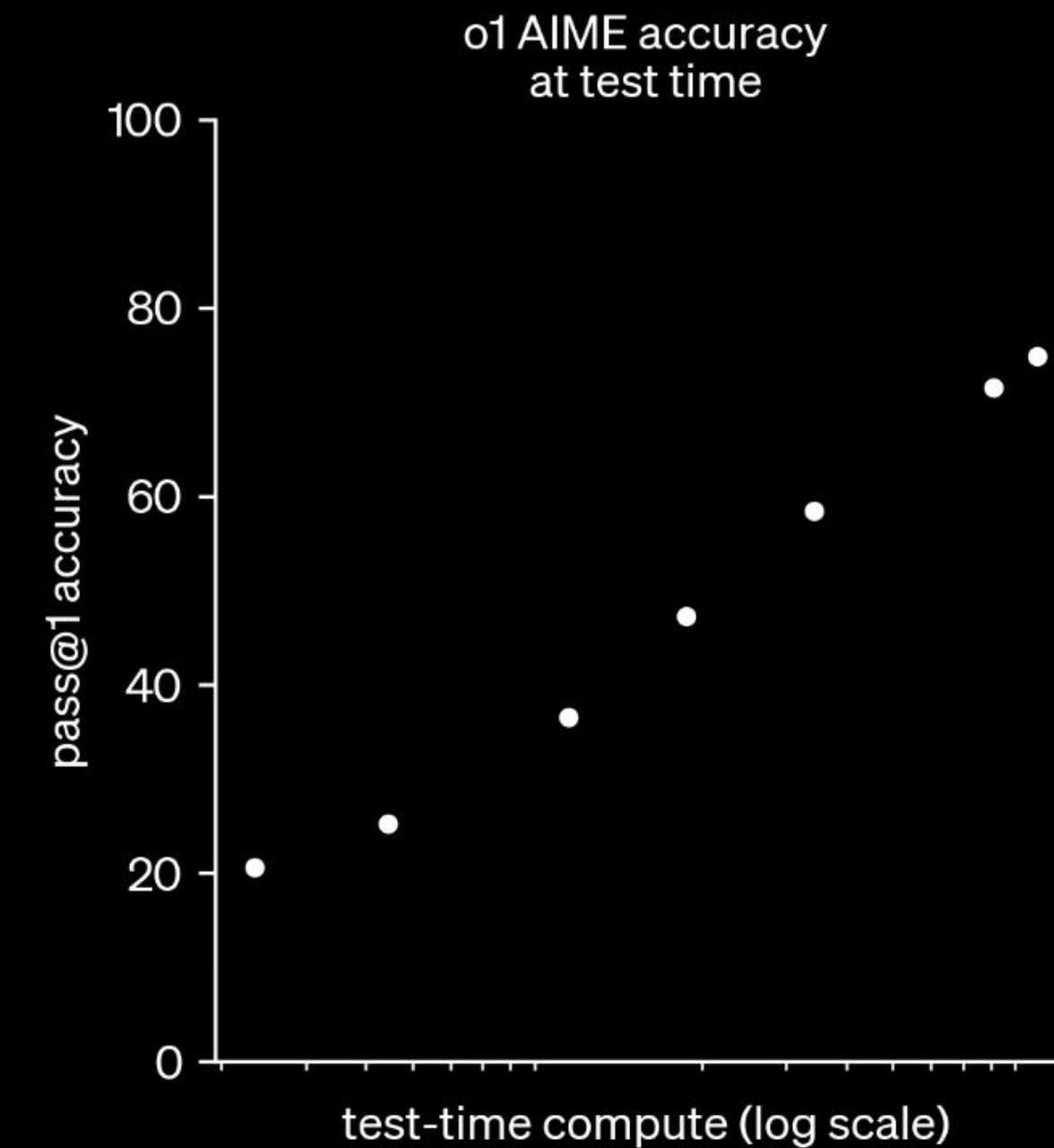
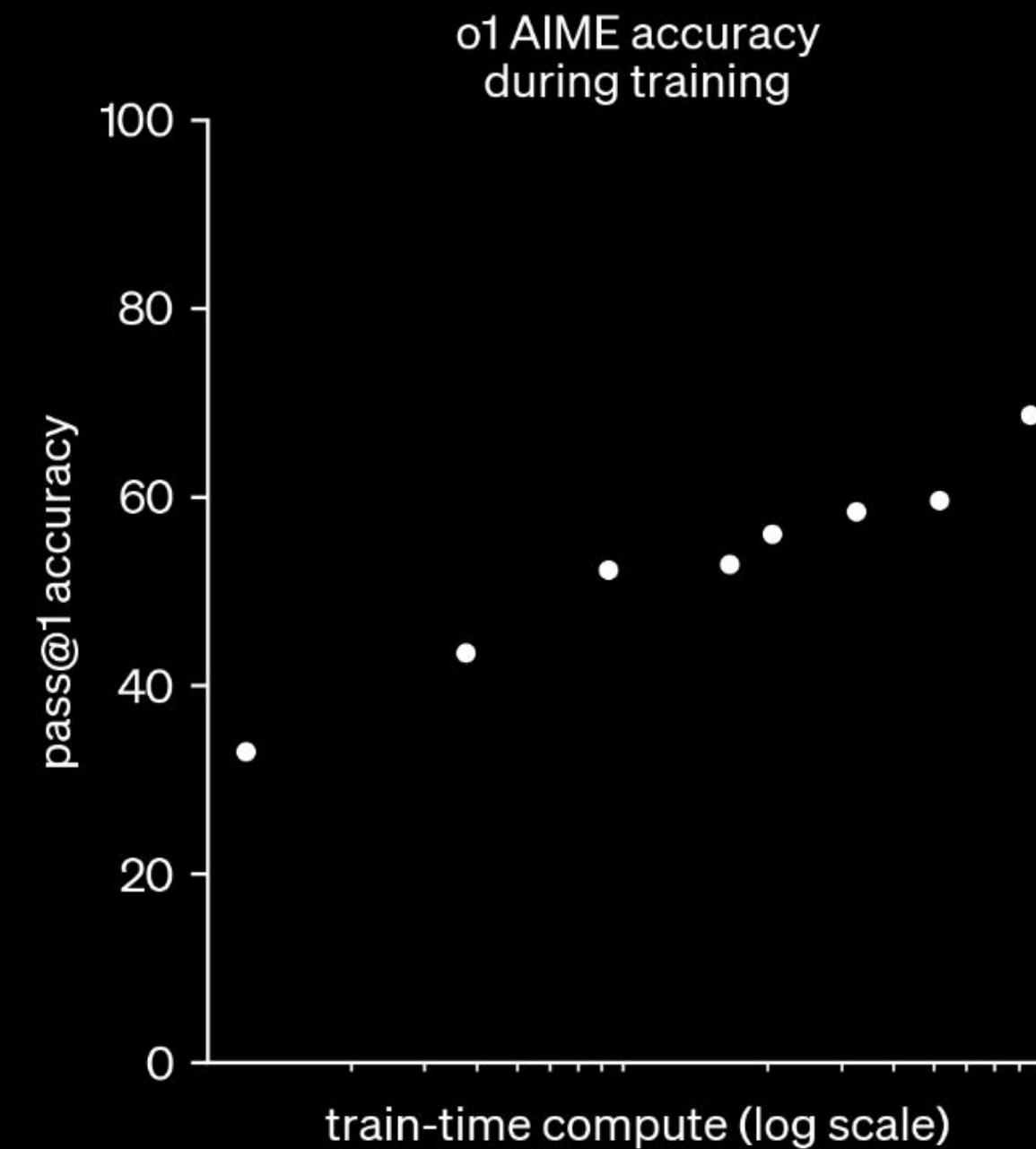


Progress on AI benchmarks in the past five years





Learning to Reason

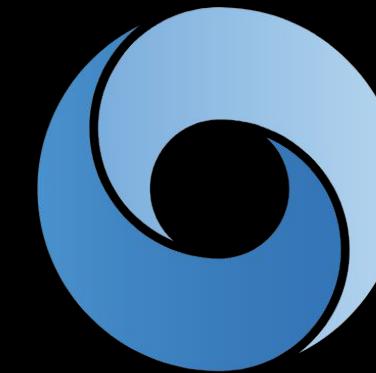




Test-Time Scaling in Learning to Reason



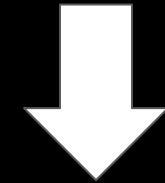
OpenAI o1



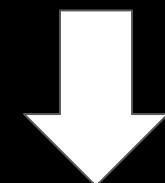
Gemini
Flash Thinking



DeepSeek r1



Chain-of-Thought^[1] works and scales on any problems

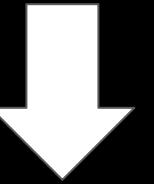


Fantastic CoTs and Where to Find Them?

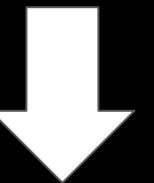


Test-Time Scaling in Learning to Reason

Chain-of-Thought^[1] works and scales on any problems



Fantastic CoTs and Where to Find Them?



End-to-End RL



Chain of Thought

Similar to how a human may think for a long time before responding to a difficult question, o1 uses a chain of thought when attempting to solve a problem. Through reinforcement learning, o1 learns to hone its chain of thought and refine the strategies it uses. It learns to

How it works

Deep research was trained using end-to-end reinforcement learning on hard browsing and reasoning tasks across a range of domains. Through that training, it



Lean-STaR: Learning to Interleave Thinking and Proving

Haohan Lin^{2*}

Zhiqing Sun¹

Yiming Yang¹

Sean Welleck¹

¹Language Technologies Institute, Carnegie Mellon University

²Institute for Interdisciplinary Information Sciences, Tsinghua University

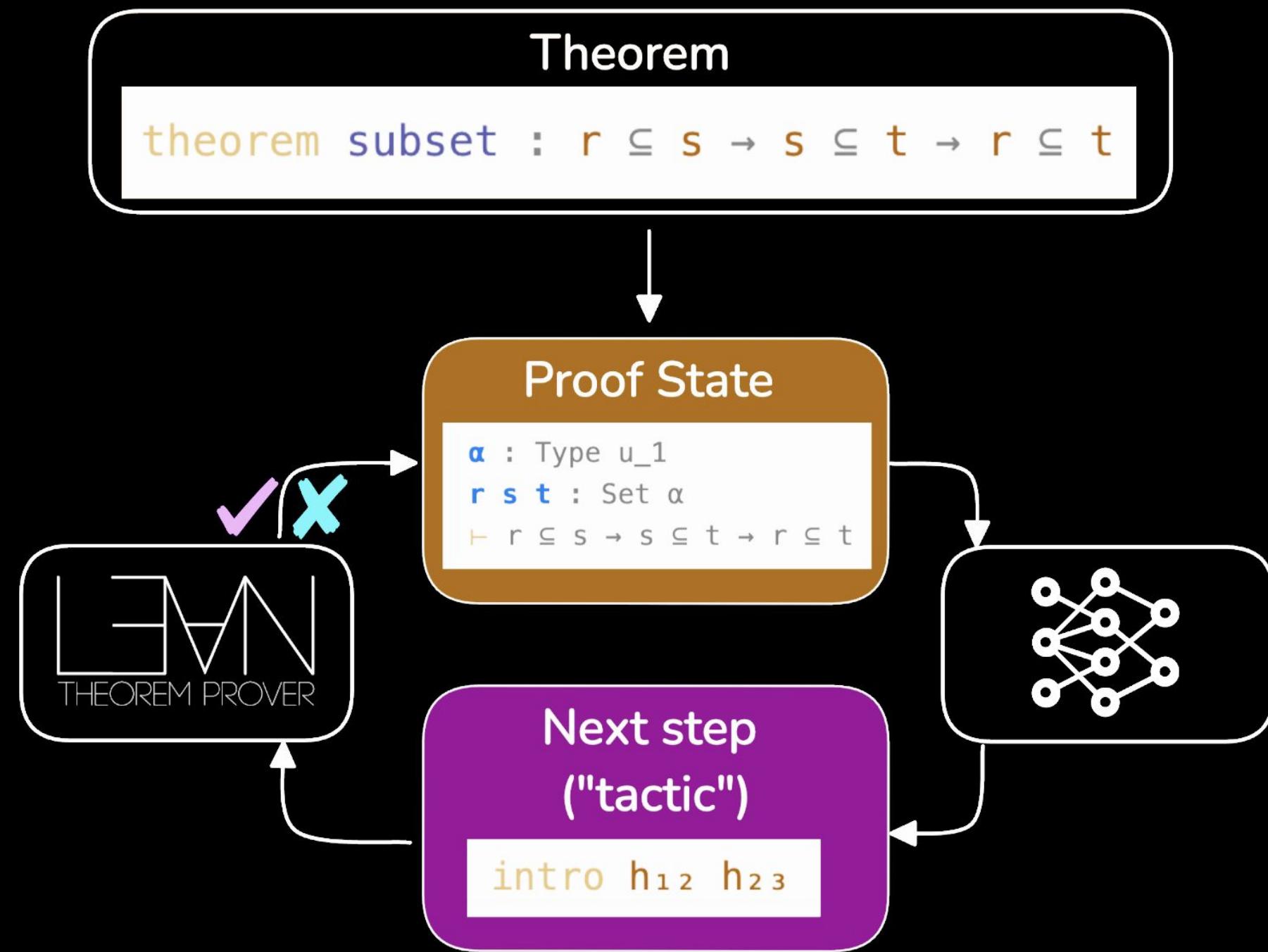
{haohanl,zhiqings,yiming,swelleck}@cs.cmu.edu

<https://leanstar.github.io/>



Teach models to think

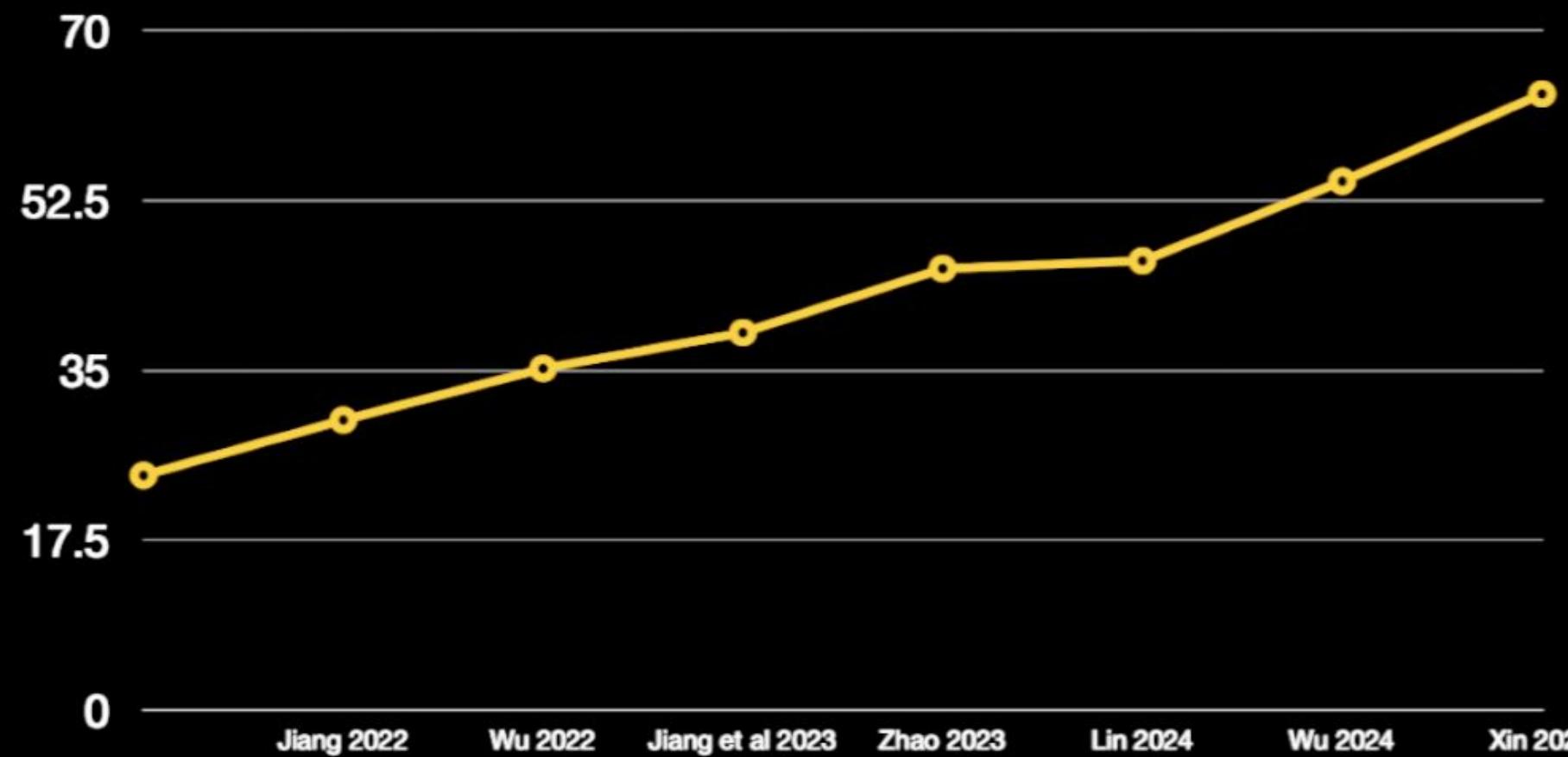
Neural theorem proving



- Math as checkable code
- Proof: sequence of (state, step)



Teach models to think



Generative Language Modeling for Automated Theorem Proving

Stanislas Polu
OpenAI
spolu@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

September 7, 2020 Publication

Fig: miniF2F benchmark performance, 2022-2024



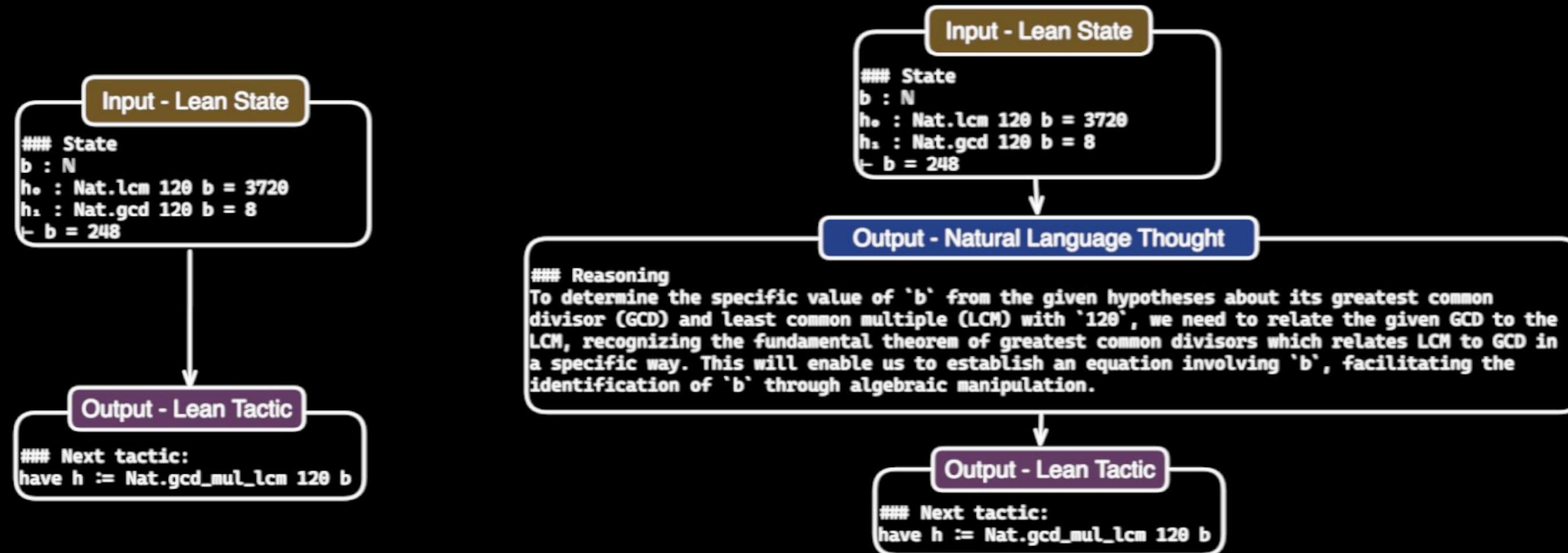
Teach models to think

```
theorem imo_1960_p2 (x : ℝ) (h0 : 0 ≤ 1 + 2 * x) (h1 : (1 - Real.sqrt (1 + 2 * x)) ^ 2 ≠ 0)
  (h2 : 4 * x ^ 2 / (1 - Real.sqrt (1 + 2 * x)) ^ 2 < 2 * x + 9) := -(1 / 2)
  ≤ x ∧ x < 45 / 8 := by
  norm_num at h0 h1 h2
  have h3 : 0 ≤ 1 + 2 * x := by linarith
  have h4 : 0 < 1 + Real.sqrt (1 + 2 * x) := by
    nlinarith [Real.sqrt_nonneg (1 + 2 * x)]
  have h5 : 4 * x ^ 2 / (1 - Real.sqrt (1 + 2 * x)) ^ 2 < 2 * x + 9 := by
    linarith
  have h6 : 1 - Real.sqrt (1 + 2 * x) ≠ 0 := by
    intro h
    apply h1
    nlinarith
  have h7 : 4 * x ^ 2 / (1 - Real.sqrt (1 + 2 * x)) ^ 2 = (1 + Real.sqrt (1 +
  2 * x)) ^ 2 := by
    field_simp [h6]
    nlinarith [sq_sqrt (show 0 ≤ 1 + 2 * x by linarith)]
  rw [h7] at h5
  constructor <;> nlinarith [sq_sqrt (show 0 ≤ 1 + 2 * x by linarith)]
```

Fig: Generated International Math Olympiad solution in Lean
(DeepSeek Prover-1.5B, Xin et al 2024)



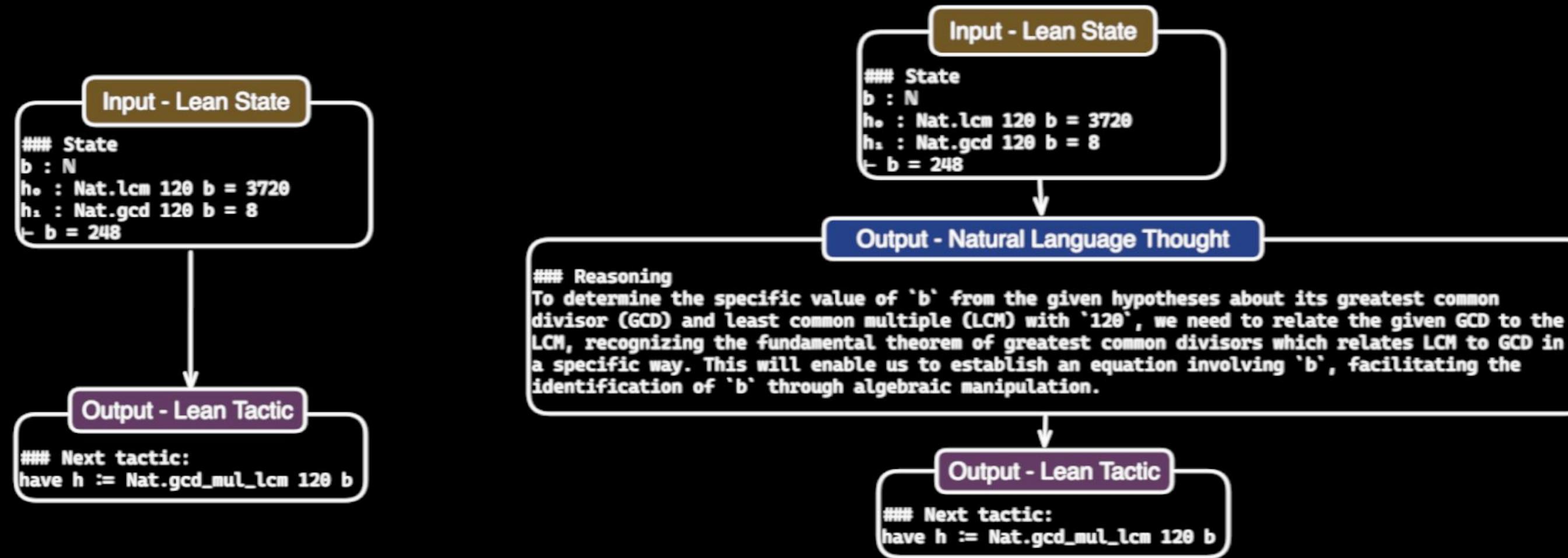
Teach models to think



Can we train a model to “think” in CoT before each step of formalized proving?



Teach models to think



Why?

- Planning / Backtracking / Reflection
- More tokens can give more computational capacity
- ...

Ultimately, the RL Gods will decide how to use these tokens.



Teach models to think

Lean-STaR (Self-taught reasoner^[1])

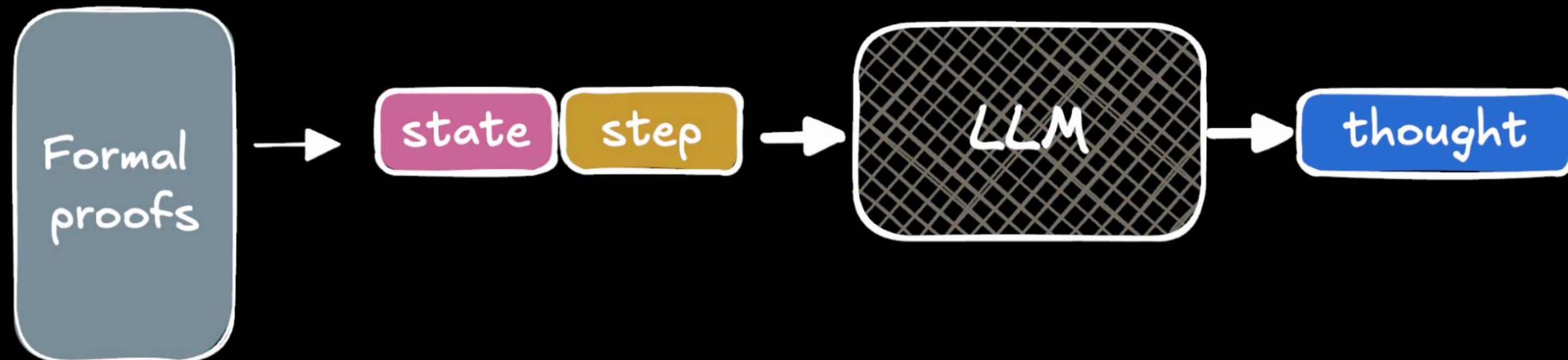
- Learn to generate Chain-of-Thoughts via reinforcement learning
 - Initialization
 - Reinforcement learning



Teach models to think

1. Initialization

Annotate thoughts "retrospectively"

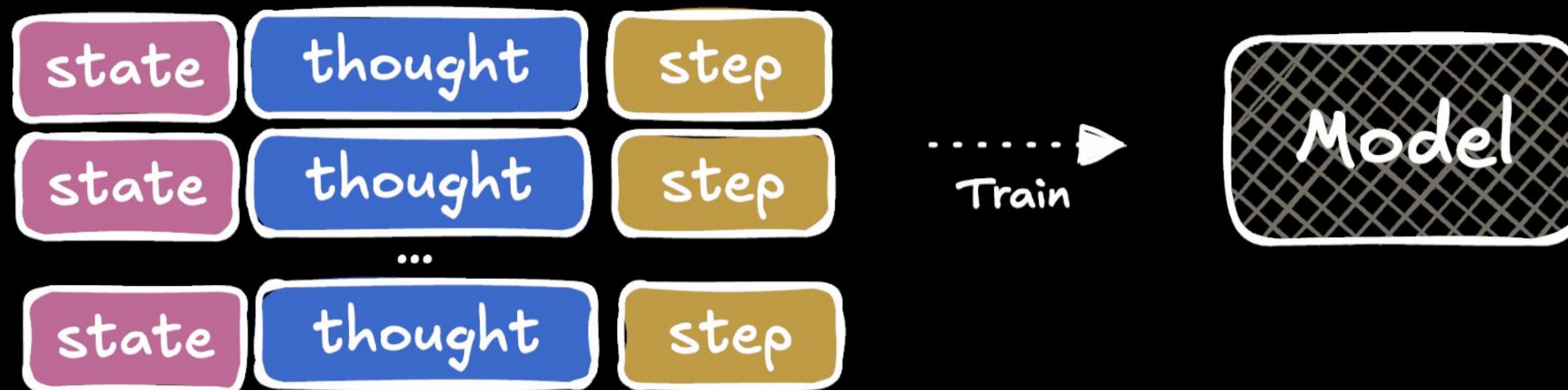




Teach models to think

1. Initialization

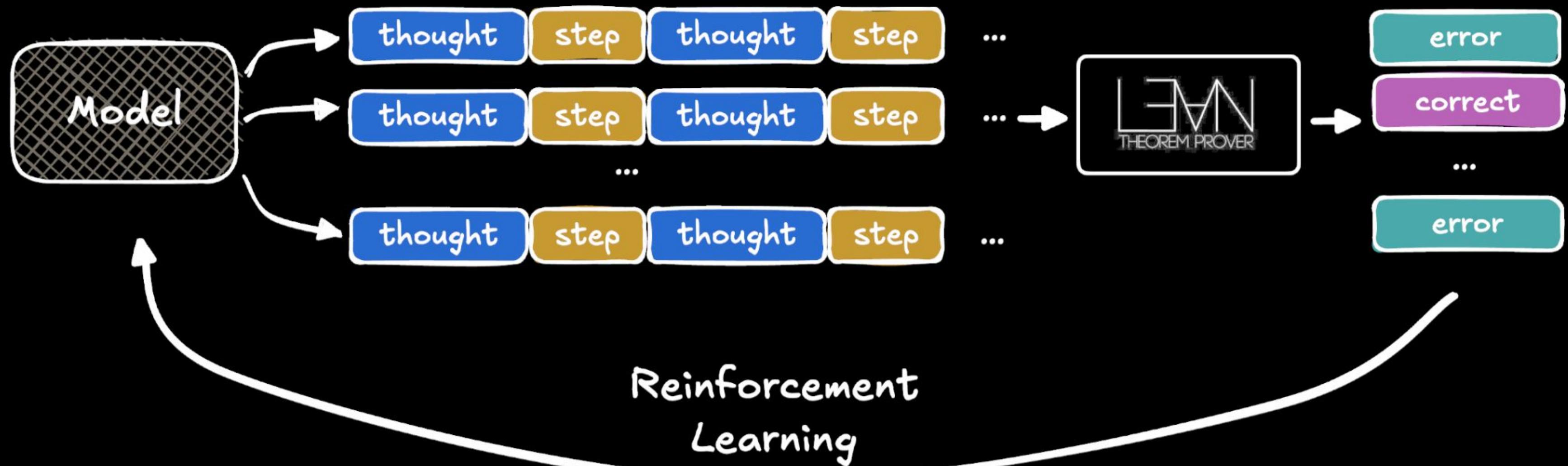
Train initial model on
(state, thought) -> step examples





Teach models to think

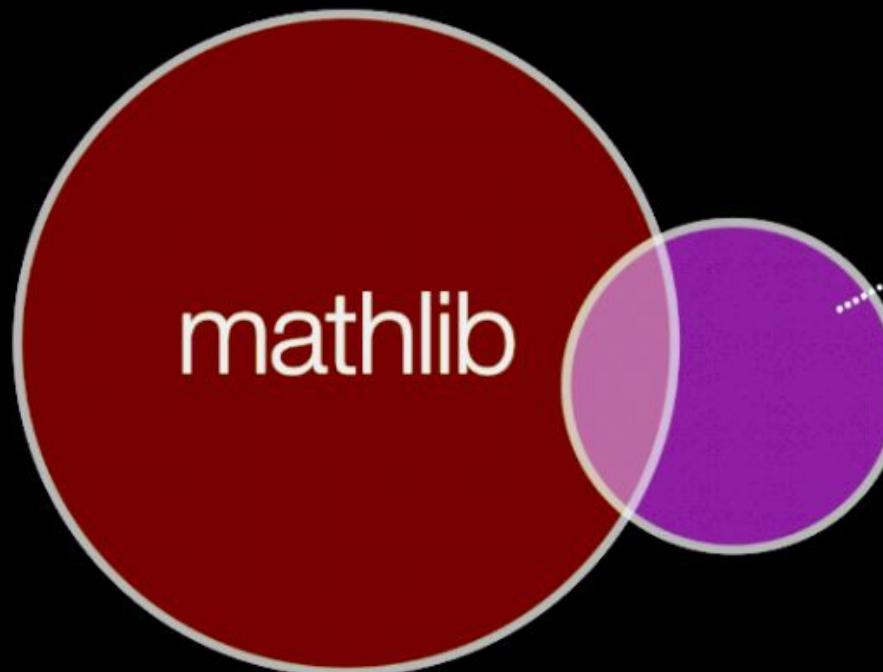
2. Reinforcement Learning (Expert Iteration)





Teach models to think

- train
- test



Problem 1959 IMO Problems/Problem 1

Prove that the fraction $\frac{21n + 4}{14n + 3}$ is irreducible for every natural number n .

theorem imo_1959_p1

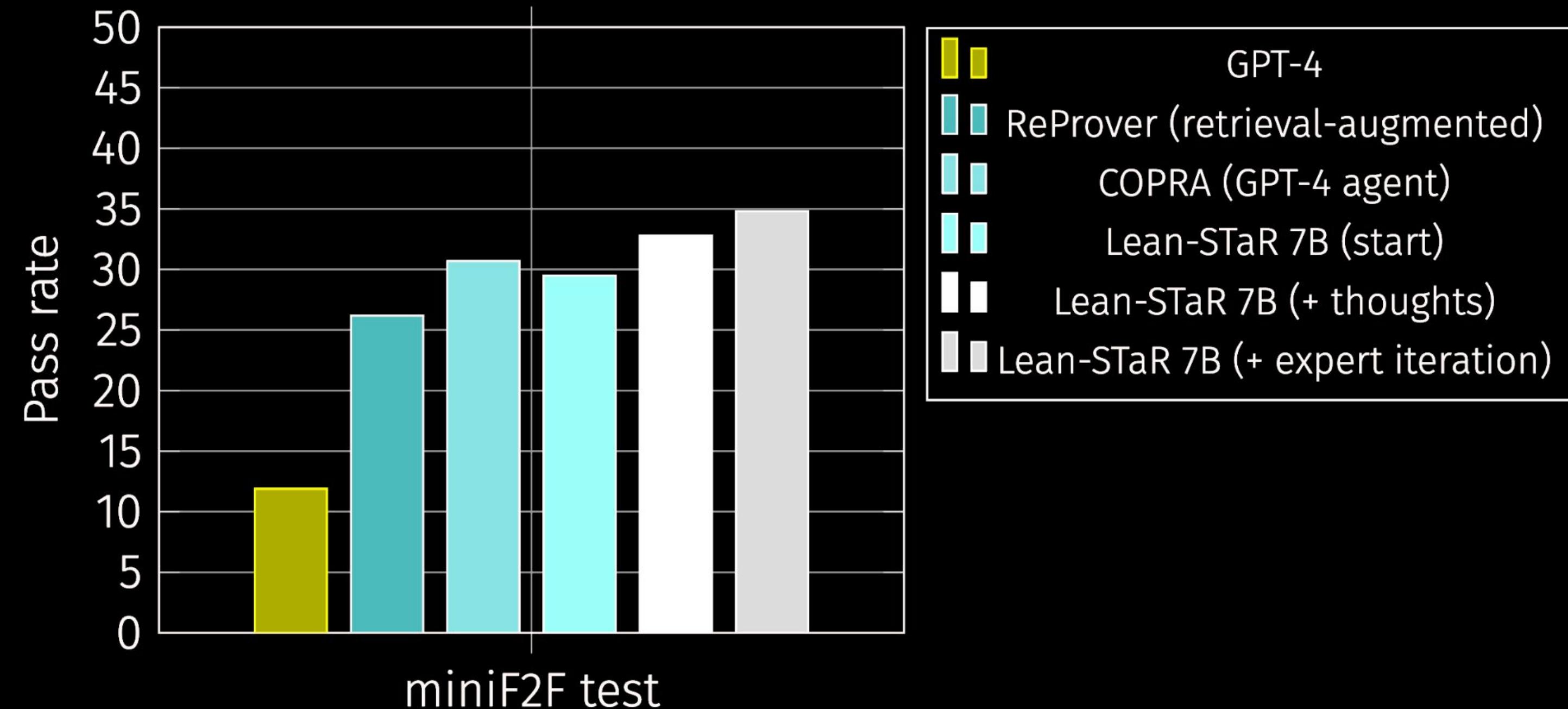
(n : ℕ)

(h₀ : 0 < n) :

Nat.gcd (21*n + 4) (14*n + 3) = 1 := by sorry



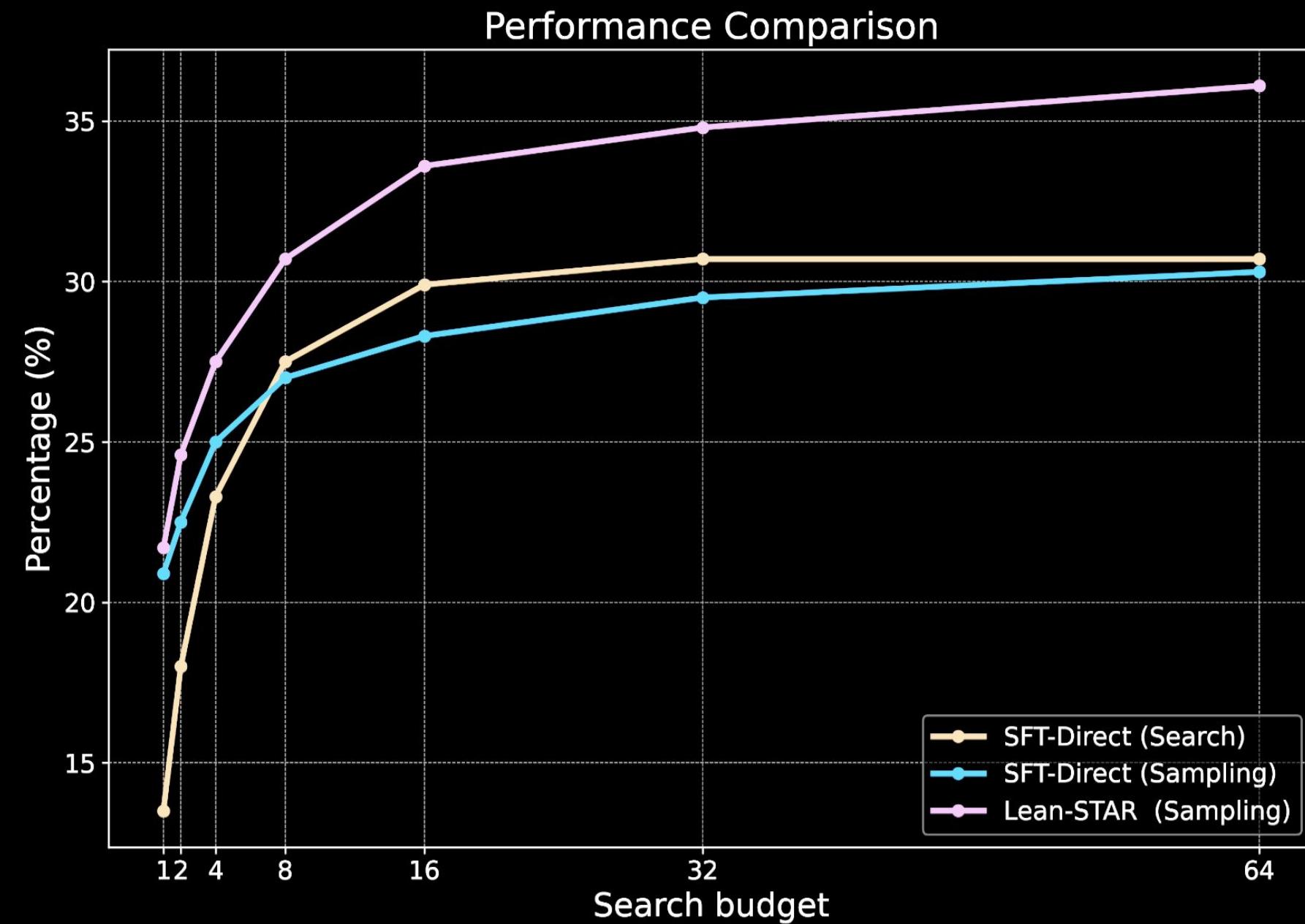
Teach models to think



SoTA on miniF2F (when we submitted it to NeurIPS)



Teach models to think



Takeaway: Chain of Thoughts is more effective when scaling up inference-time compute.



Teach models to think

`theorem amc12a_2002_p6 (n : ℕ) (h₀ : 0 < n)`

`: ∃ m, m > n ∧ ∃ p, m * p ≤ m + p :=`

`use n + 1, by linarith only [h₀]` -- Reasoning: To demonstrate the existence of such an `m` and `p` that satisfy the given inequalities, we leverage the property of natural numbers that ensures an increase in `n` from `0` leads to the possibility of finding such `m` and `p`. This approach is supported by the fundamental property of natural numbers and their arithmetic operations, suggesting that for a given `n` greater than `0`, there exists a successor `m` and a number `p` that satisfy the conditions outlined.

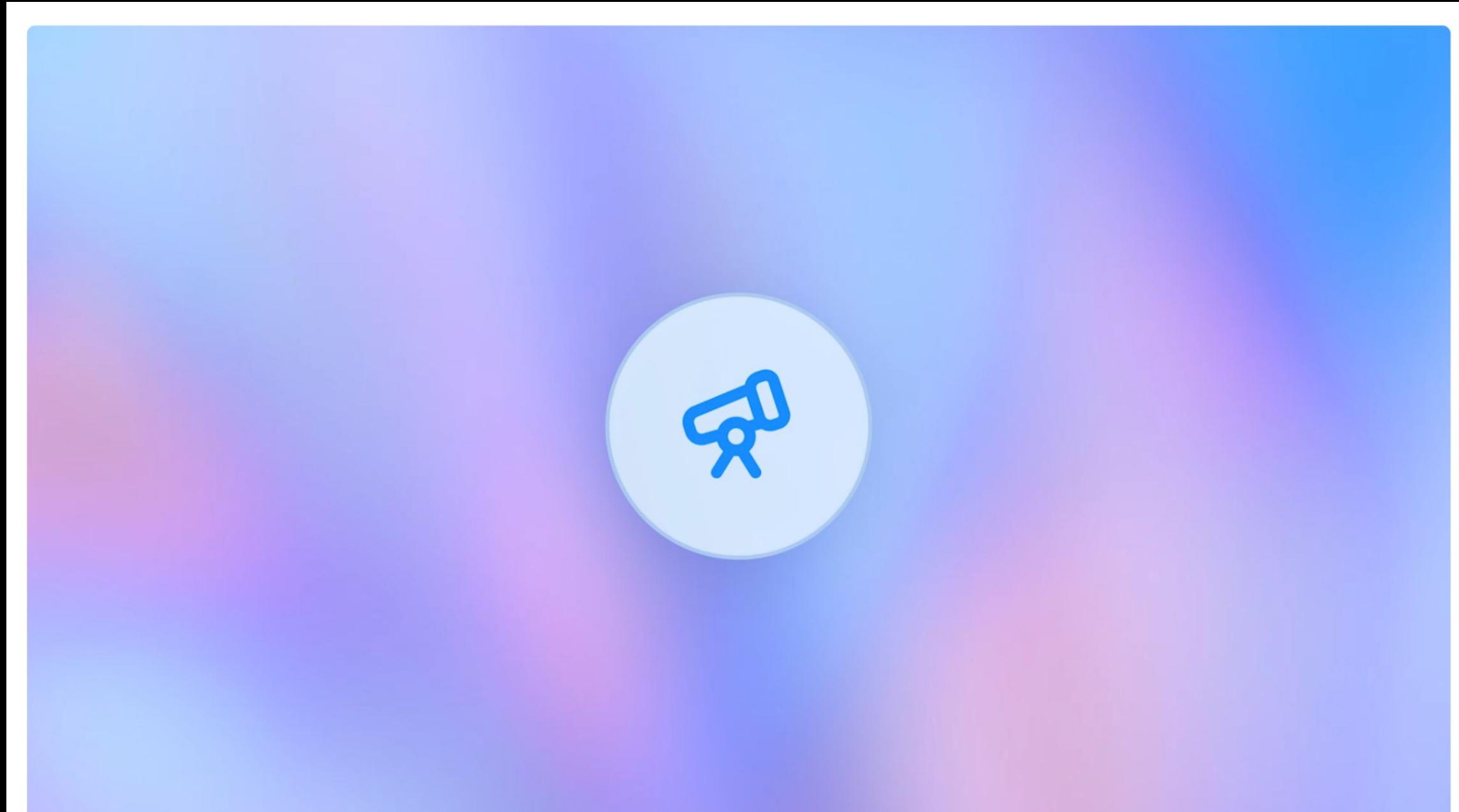
`exact ⟨0, by simp ⟩` -- Reasoning: To demonstrate that there exists an $|(n \in \mathbb{N}|)$ such that $|(n + 1) * p| \leq n + 1 + p|$ holds for some $|(p \in \mathbb{N}|)$, we recognize that setting $|(p = 0|)$ satisfies the inequality by the properties of zero in arithmetic. This is because $|(n + 1 > 0|)$ by assumption, ensuring the left-hand side is non-negative and the right-hand side simplifies to $|(n + 1|)$, making the inequality trivially true.

```
theorem aime_1983_p2 (x p : ℝ) (f : ℝ → ℝ) (h₀ : 0 < p ∧ p < 15) (h₁ : p ≤ x ∧ x ≤ 15)
  (h₂ : f x = abs (x - p) + abs (x - 15) + abs (x - p - 15))
  : 15 ≤ f x :=
cases1 h₁ with h₂ -- Reasoning: Given a function  $|f(x)|$  defined over the real numbers, where  $|f(x)|$  equals the sum of the absolute values of the differences  $|x - p|$ ,  $|x - 15|$ , and  $|x - p - 15|$ , along with conditions on  $|p|$  and  $|x|$  that ensure  $|x|$  lies between  $|p|$  and  $|15|$  and  $|p|$  is positive, we aim to show that  $|f(x)| \geq 15$ . To demonstrate this, we consider all possible cases for the position of  $|x|$  relative to  $|p|$  and  $|15|$ , specifically at  $|p|$  itself, between  $|p|$  and  $|15|$ , and at  $|15|$ . By evaluating  $|f(x)|$  for each case, we can establish a lower bound of  $|f(x)|$  that is greater than or equal to  $15$ .
simp [h₀, abs_of_nonneg (sub_nonneg.2 h₁), abs_of_nonneg (sub_nonneg.2 h₂), *]
-- Reasoning: The objective is to show that  $|f(x)|$  is at least 15 for  $|x|$  within the bounds  $|p| \leq x \leq 15|$ , where  $|p|$  is positive but less than 15. The provided expression for  $|f(x)|$  suggests using properties of absolute values and inequalities to simplify and analyze  $|f(x)|$ . By applying simplification rules for absolute values and inequalities, along with the given bounds, we aim to demonstrate the desired inequality  $|f(x)| \geq 15$ .
simp only [abs_of_nonneg] at * -- Reasoning: The objective is to establish an inequality involving an expression with absolute values and a sum, where all terms are functions of  $|x|$ . By recognizing that the absolute value functions are always non-negative, we can simplify the given expression by removing the absolute value symbols, thus focusing on the arithmetic manipulation of the terms to demonstrate the inequality.
simp only [abs_of_nonneg, sub_add_eq_add_sub] at * -- Reasoning: The objective is to demonstrate that the expression  $15 \leq x - p + |x - 15| + |x - p - 15|$  holds given the conditions on  $p$ ,  $x$ , and the function  $f$ . Since the expression in question involves absolute values which inherently involve non-negativity, manipulating these expressions to highlight their non-negativity or equality is a logical initial step. This involves recognizing that the absolute value of a difference can be rewritten as the difference of absolute values, which simplifies the manipulation of the expression to prove the inequality.
congr 1 -- Reasoning: To establish the inequality  $15 \leq x - p + |x - 15| + |x - p - 15| - p$ , we recognize that the left-hand side of the inequality involves a mix of additions and subtractions of real numbers. By introducing a congruence transformation that simplifies operations involving absolute values, we aim to align both sides of the inequality in a form that directly reflects the properties of real numbers under these operations.
simp only [abs_of_nonpos (sub_nonpos_of_le h₂), abs_of_nonneg (sub_nonneg_of_le h₁)] -- Reasoning: The objective is to prove an inequality involving absolute values and subtractions that aligns the given conditions and constraints. Simplification using the properties of absolute values and the provided bounds can directly lead to the desired conclusion by expressing the absolute values in terms of their underlying expressions and applying the given constraints.
simp [abs_of_pos, *] -- Reasoning: To prove the given inequality involving absolute values, simplification using algebraic properties and given hypotheses will directly lead us to the desired inequality by recognizing the algebraic manipulation and application of the known bounds  $'h₁'$  and  $'h₂'$ ; as well as the non-negativity of  $'p'$ .
rw [abs_of_nonpos] -- Reasoning: The objective is to demonstrate that  $15 \leq 15 + |x - p - 15| - p$  under the given constraints on  $'p'$  and  $'x'$ . Given  $'p < 15'$  and  $'p \leq x \leq 15'$ , it's apparent that  $'p - 15'$  is non-positive. This observation allows us to apply the property that the absolute value of a non-positive number is equal to its negative value, thereby simplifying the expression on the right-hand side of the inequality.
all_goals linarith -- Reasoning: The goal involves demonstrating a relationship between expressions involving inequalities and subtraction. The simplification of these expressions into a form that directly compares their numerical values can lead to a straightforward application of known inequalities and algebraic properties, demonstrating the inequality's validity under the given conditions.
```

Examples generated thoughts and proof from Lean-STaR



Teach models to think



Introducing deep research

Feb 2, 2025 · 8 min read

An agent that uses reasoning to synthesize large amounts of online information and complete multi-step research tasks

 Sam Altman  
@sama

congrats to the team, especially @isafulf and @EdwardSun0909, for building an incredible product.

my very approximate vibe is that it can do a single-digit percentage of all economically valuable tasks in the world, which is a wild milestone.

 What does the product do?

5:11 PM · Feb 2, 2025 · 1.1M Views



Teach models to think

GAIA task examples

Level 1 Level 2 Level 3

In July 2, 1959 United States standards for grades of processed fruits, vegetables, and certain other products listed as dehydrated, consider the items in the "dried and dehydrated section" specifically marked as dehydrated along with any items in the Frozen/Chilled section that contain the whole name of the item, but not if they're marked Chilled. As of August 2023, what is the percentage (to the nearest percent) of those standards that have been superseded by a new version since the date given in the 1959 standards?

[The remaining Chain-of-Thought messages and the final answer are hidden to protect the benchmark's ground truth answer.]

⌚ Noting the changes
I'm chronicling the transition from the "Dehydrated Orange Juice" standard to the comprehensive 1983 standard for orange juice, highlighting the updates in the late 1960s and 1983.

⌚ [browse action]

⌚ Figuring it out
I'm working through identifying which of seven standards from a 1959 list have received updates since then.

⌚ [search action]

GAIA: A Benchmark for General AI Assistants				
	Level 1	Level 2	Level 3	Avg.
Previous SOTA ↗	67.92	67.44	42.31	63.64
Deep Research (pass@1)	74.29	69.06	47.6	67.36
Deep Research (cons@64)	78.66	73.21	58.03	72.57

Scaling up with better base model / better RL / more data / better RM / browser + python tools



BrowseComp: A Simple Yet Challenging Benchmark for Browsing Agents

Jason Wei* **Zhiqing Sun*** **Spencer Papay**

Scott McKinney **Jeffrey Han** **Isa Fulford** **Hyung Won Chung**

Alex Tachard Passos **William Fedus** **Amelia Glaese**

OpenAI



Example of questions

What's the title of the scientific paper published in the EMNLP conference between 20182023 where the first author did their undergrad at Dartmouth College and the fourth author did their undergrad at University of Pennsylvania?



Answer: Frequency Effects on Syntactic Rule Learning in Transformers

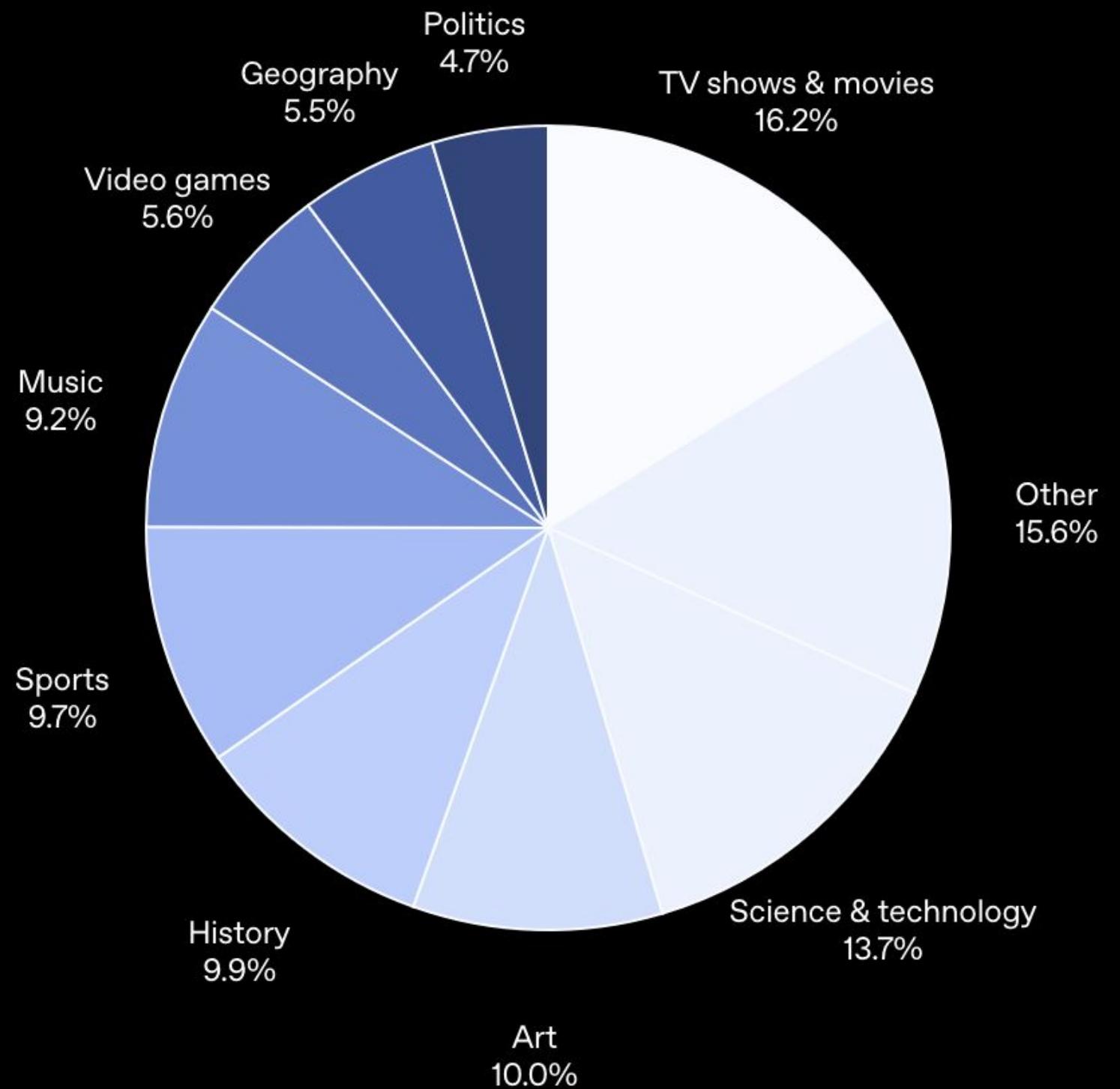
Please identify the fictional character who occasionally breaks the fourth wall with the audience, has a backstory involving help from selfless ascetics, is known for his humor, and had a TV show that aired between the 1960s and 1980s with fewer than 50 episodes.



Answer: Plastic Man



Dataset diversity & difficulty



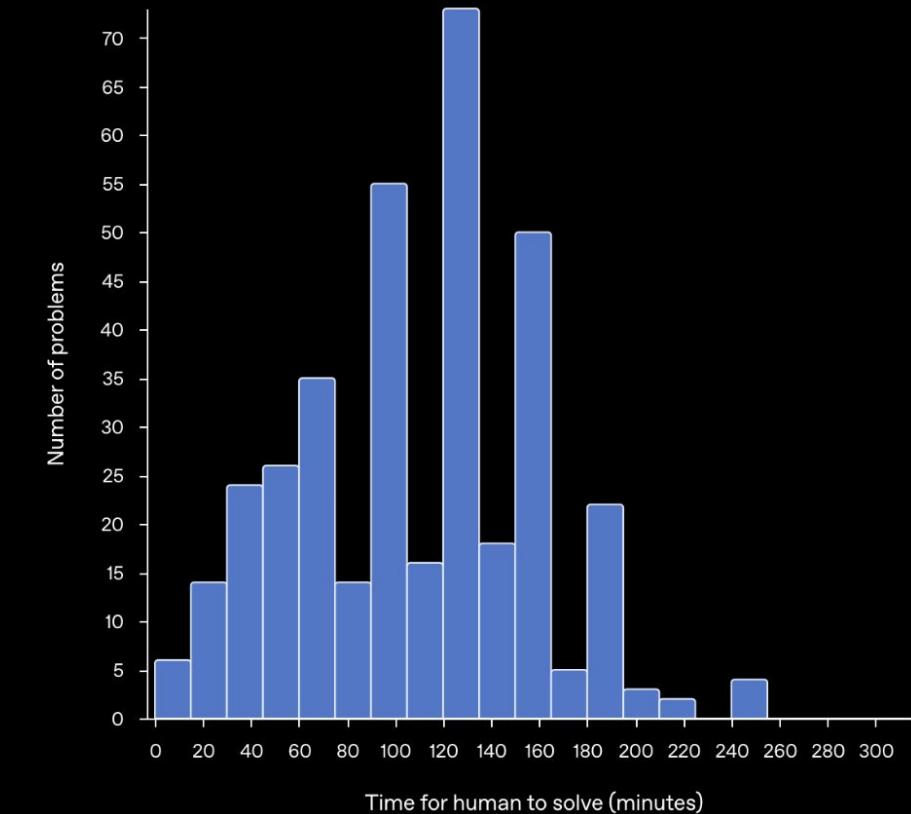


Q

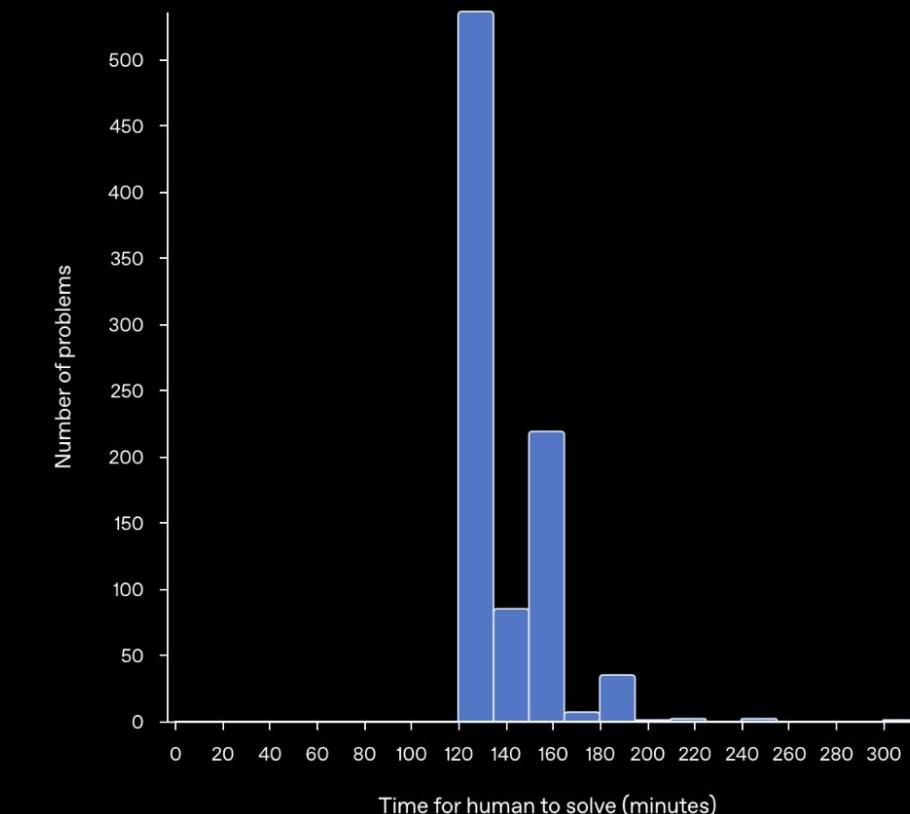
Human Performance

Total problems in verification campaign	1,255
Unsolvable	888 / 1,255 (70.8%)
Solvable	367 / 1,255 (29.2%)
Of solvable problems, trainer answer and reference answer agree	317 / 367 (86.4%)

Problems solved by humans

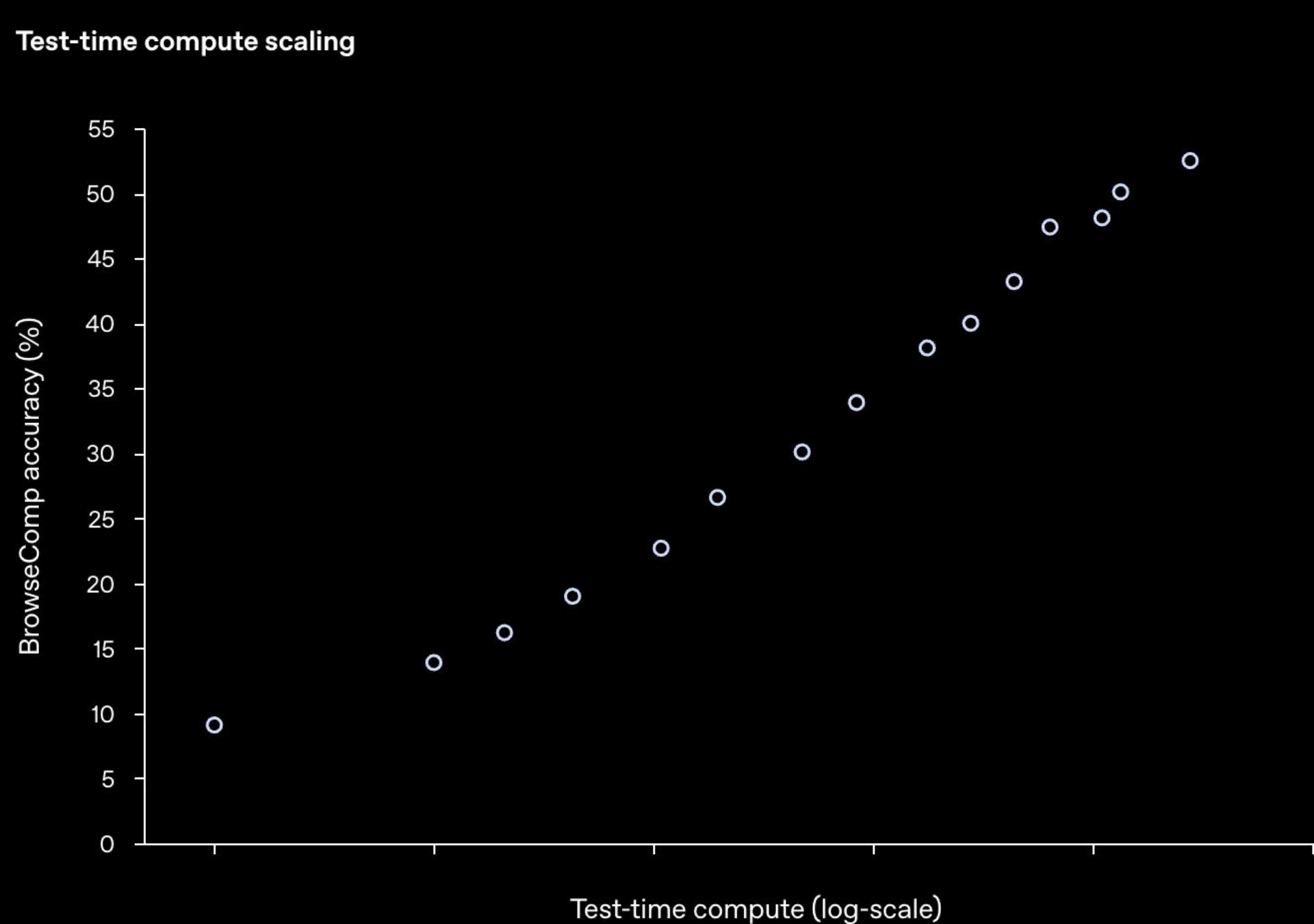


Problems where humans gave up

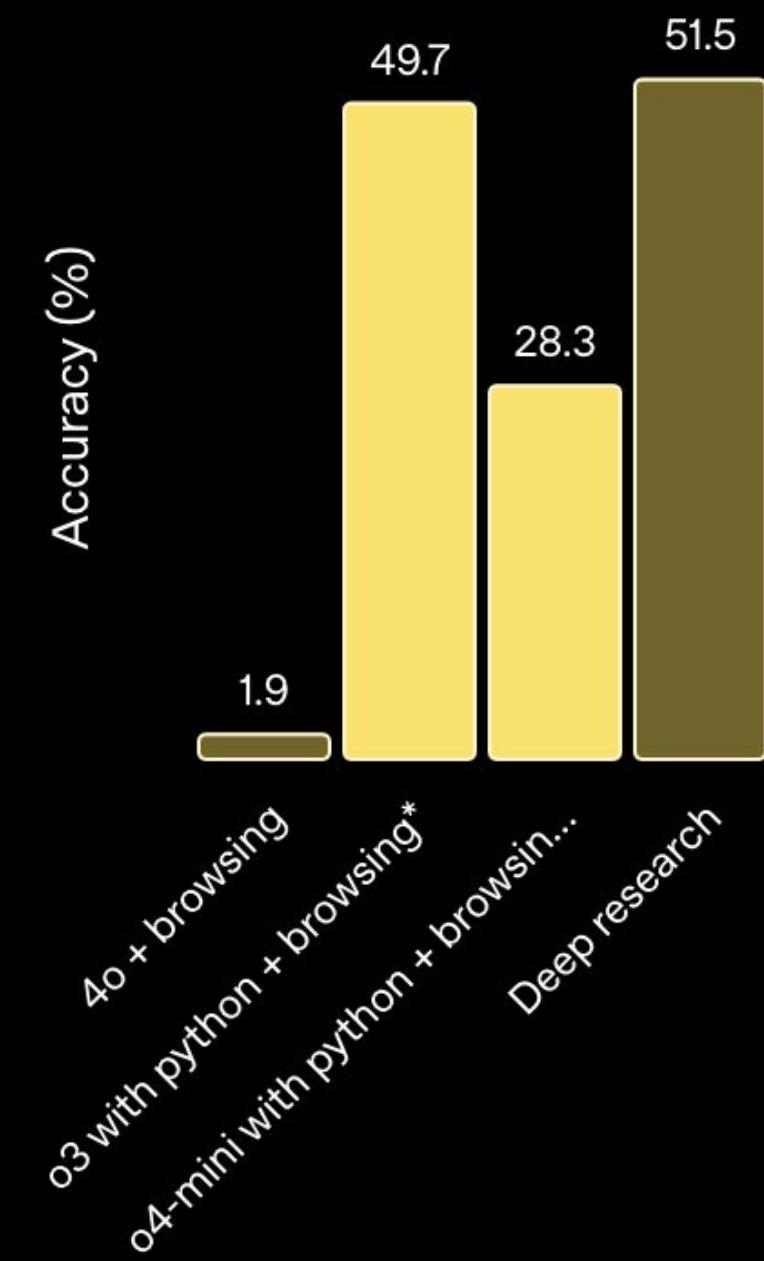




Test-Time Scaling & OpenAI Model Performance



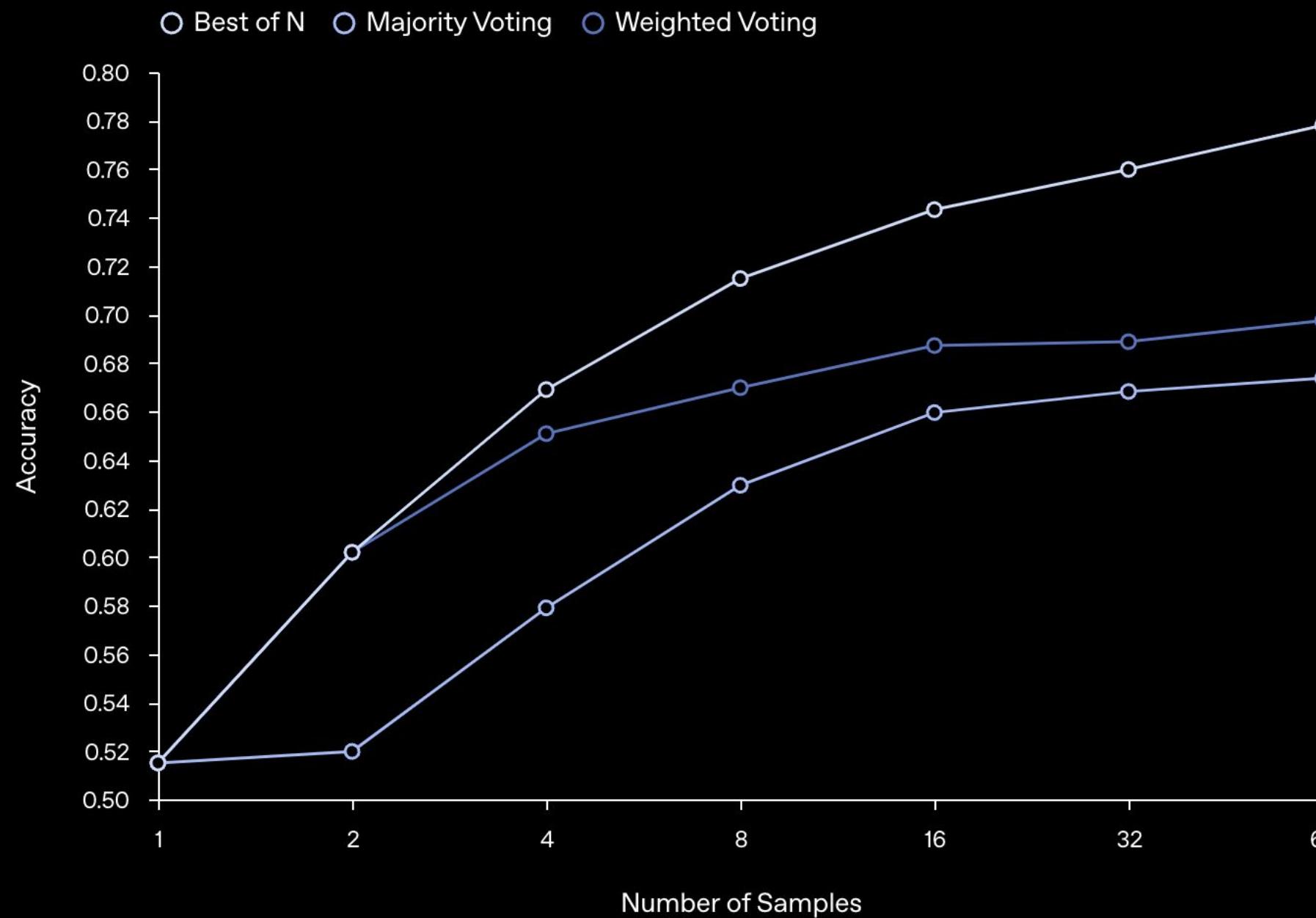
BrowseComp
Agentic Browsing





Parallel Sampling & Voting

Parallel Sampling and Confidence-Based Voting: Deep Research



HLE Prompt

Your response should be in the following format:

Explanation: {your explanation for your final answer}

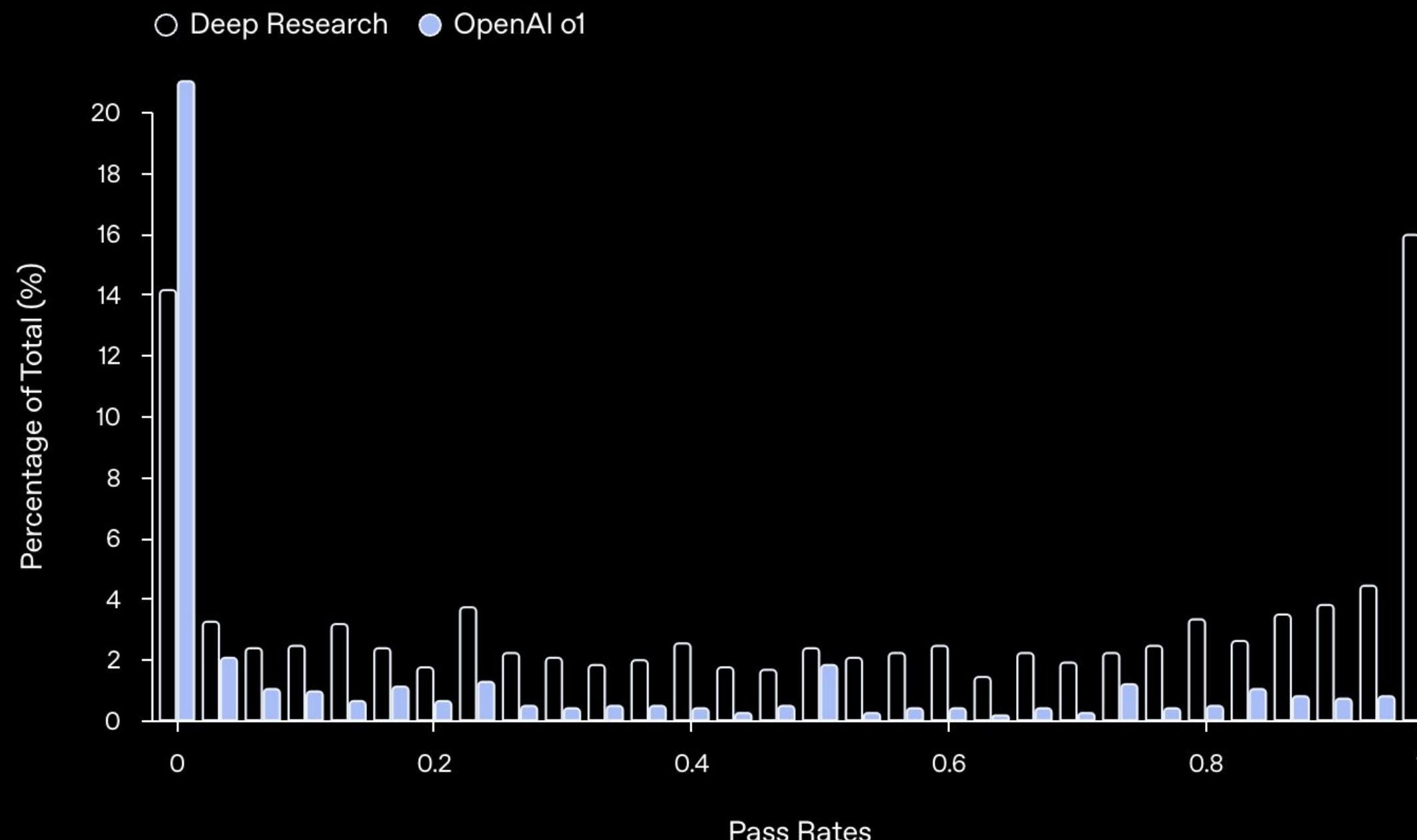
Exact Answer: {your succinct, final answer}

Confidence: {your confidence score between 0% and 100% for your answer}



Difficulty Analysis

BrowserComp Task Difficulty: Pass Rate Breakdown





Easy-to-Hard Generalization: Scalable Alignment Beyond Human Supervision

Zhiqing Sun^{1*}, Longhui Yu^{2*}, Yikang Shen³, Weiyang Liu^{4,5},
Yiming Yang^{1†}, Sean Welleck^{1†}, Chuang Gan^{3,6†}

¹Carnegie Mellon University, ²Peking University, ³MIT-IBM Watson AI Lab

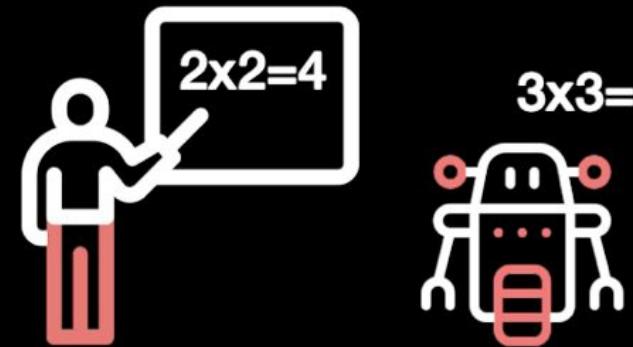
⁴University of Cambridge, ⁵Max Planck Institute for Intelligent Systems, ⁶UMass Amherst

Code: [Edward-Sun/easy-to-hard](#)



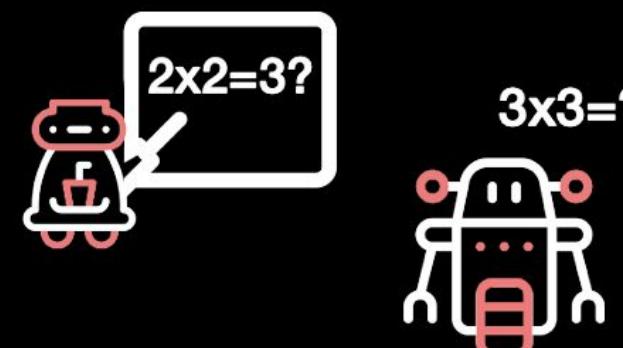
Scalable Oversight

Traditional Alignment



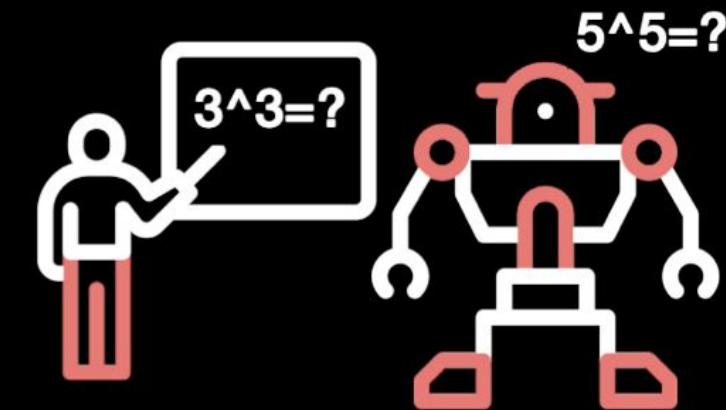
humans supervise strong models on hard tasks

Burns' Analogy on Weak-to-Strong Generalization



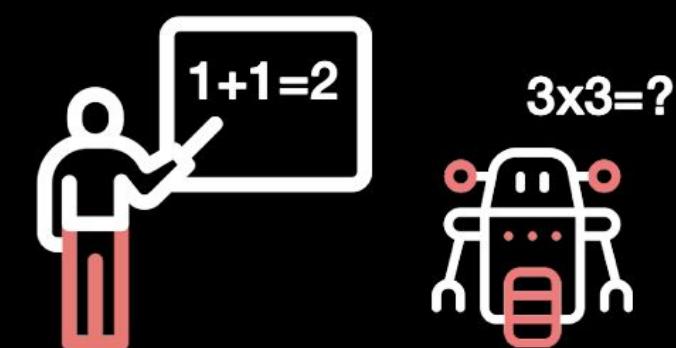
weak models unreliable supervise **strong** models
on hard tasks that humans can evaluate

Scalable Alignment (Superalignment)



humans cannot reliably supervise
superhuman models on the **hardest** tasks

Our Analogy on Easy-to-Hard Generalization

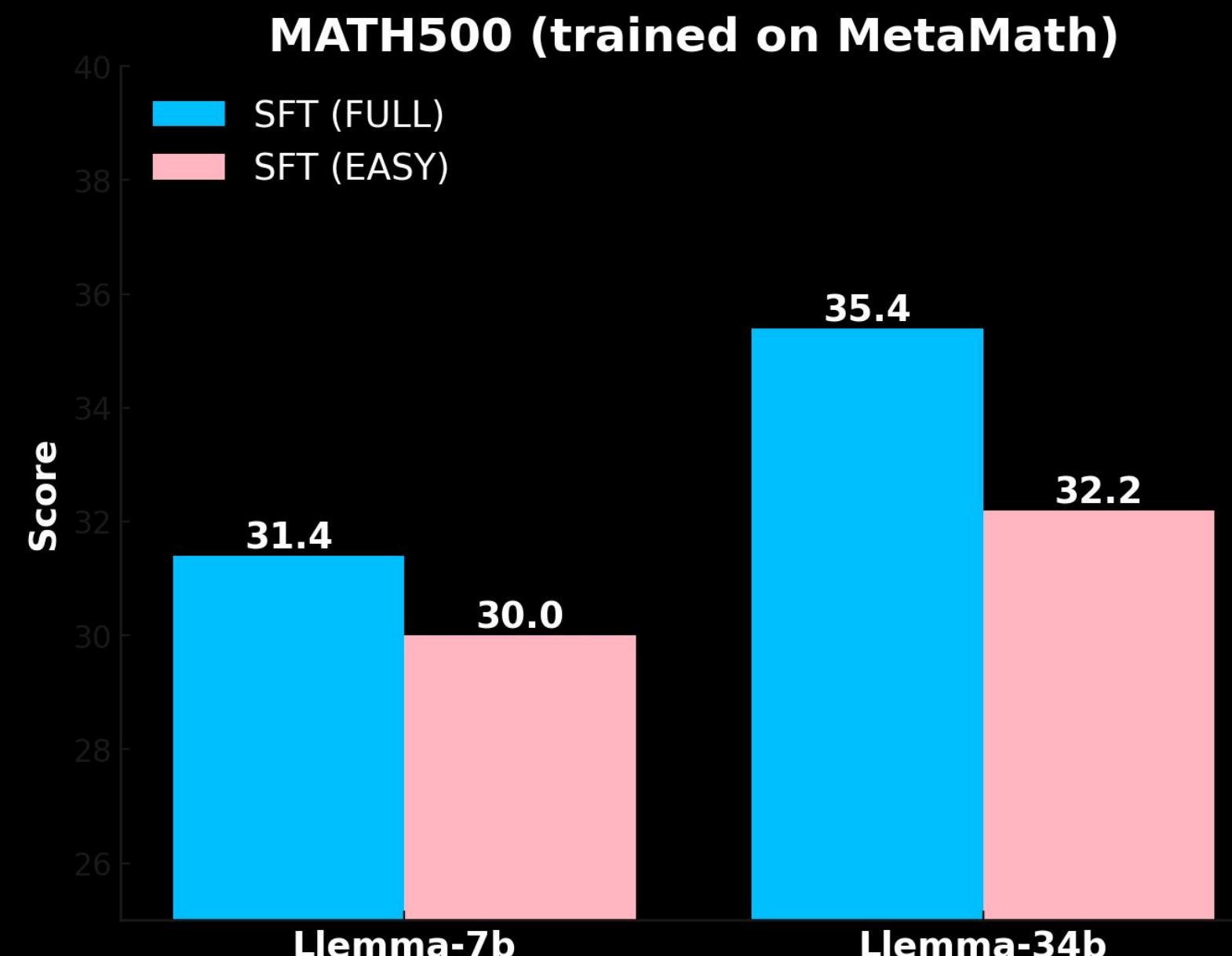


humans reliably supervise strong models
on **easy** tasks and evaluate them on **hard** tasks



Easy-to-Hard Generalization

- Observation: SFT fails to generalize





Easy-to-Hard Generalization

Insights:

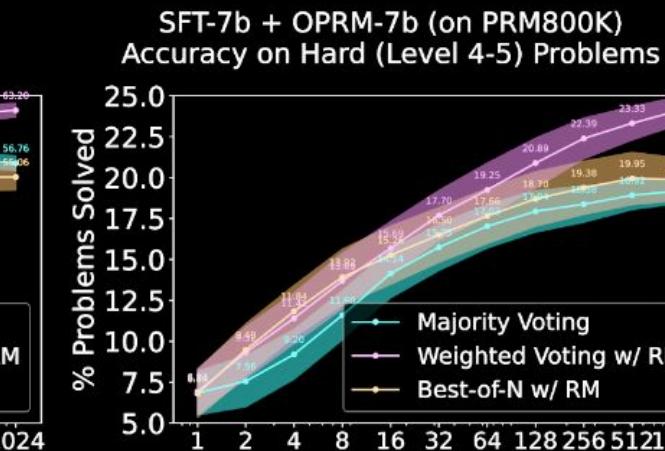
- Evaluation is easier than generation (NP problems)

Both SFTs and RMs are trained on the easy data

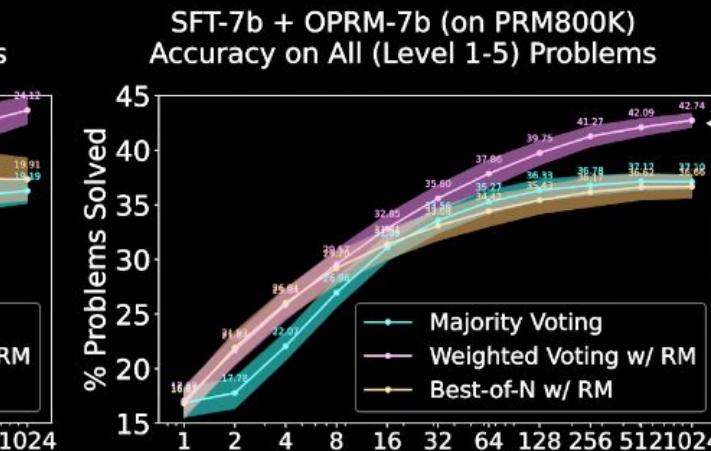
Lemma-7b



Hard



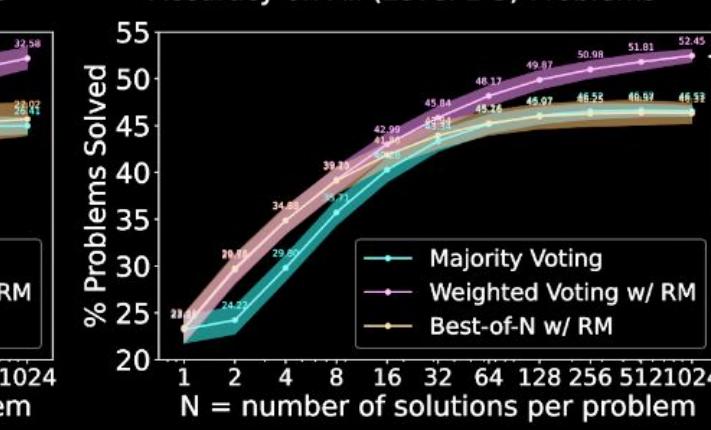
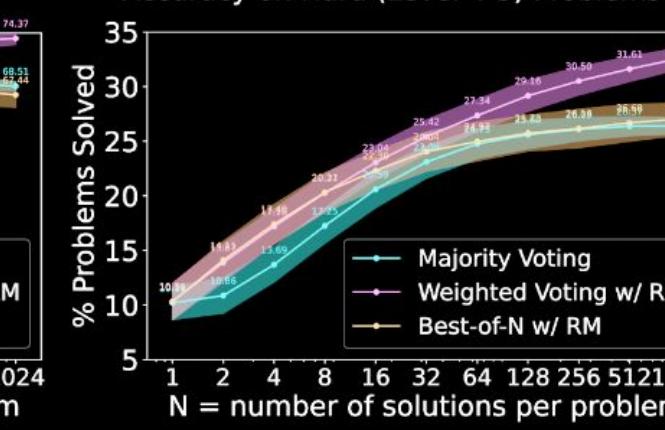
All



42.7

Ref:
Old GPT-4 (52.9)
Gemini Ultra (53.2)

Lemma-34b

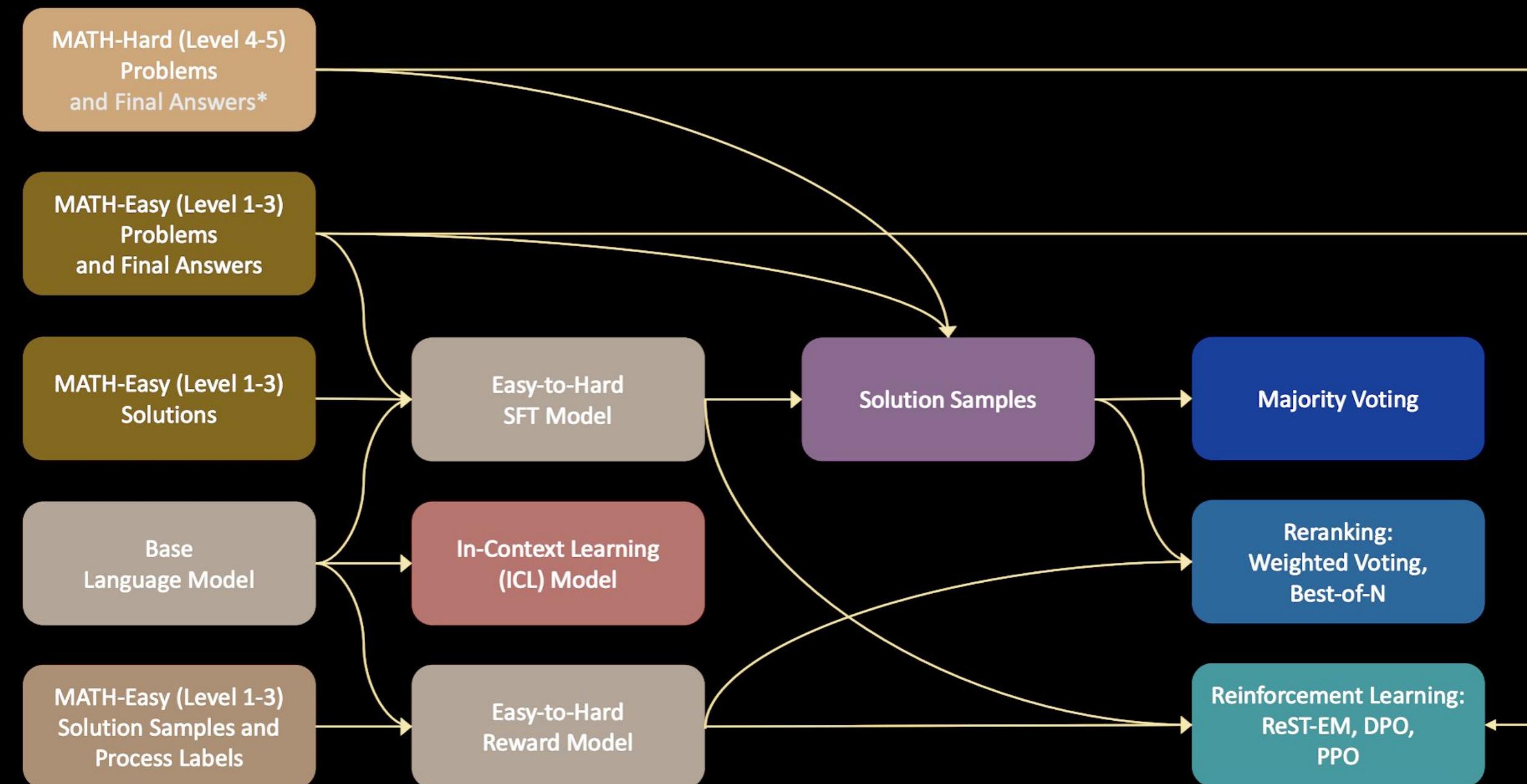


52.5



Easy-to-Hard Generalization

Formal Experimental Setup

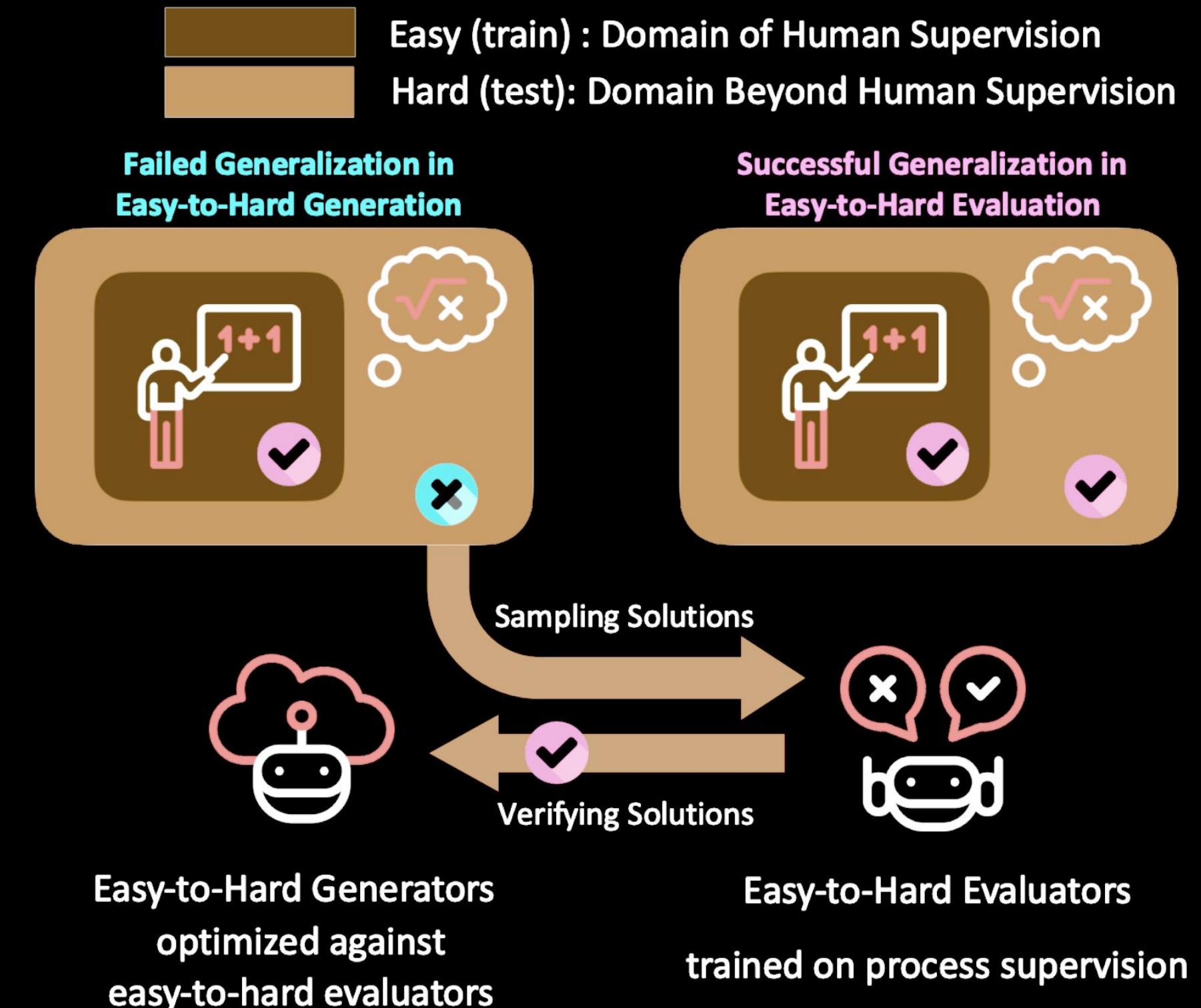




Easy-to-Hard Generalization

Insights:

- Evaluation is easier than generation (NP problems)





Easy-to-Hard Generalization

Insights:

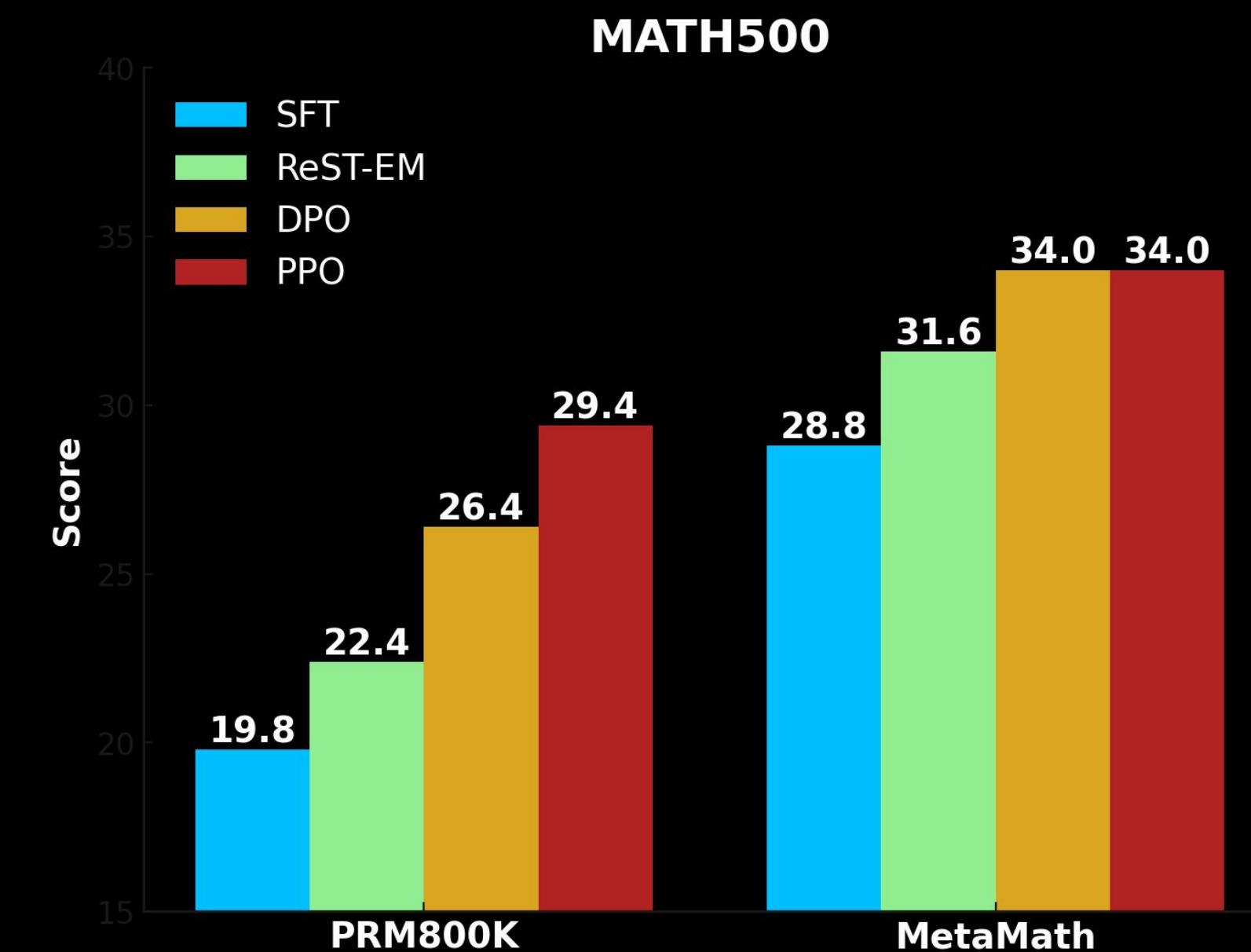
- Evaluation is easier than generation (NP problems)

PRM800K

Low-quality SFT
High-quality PRM

MetaMath

High-quality SFT
Low-quality PRM

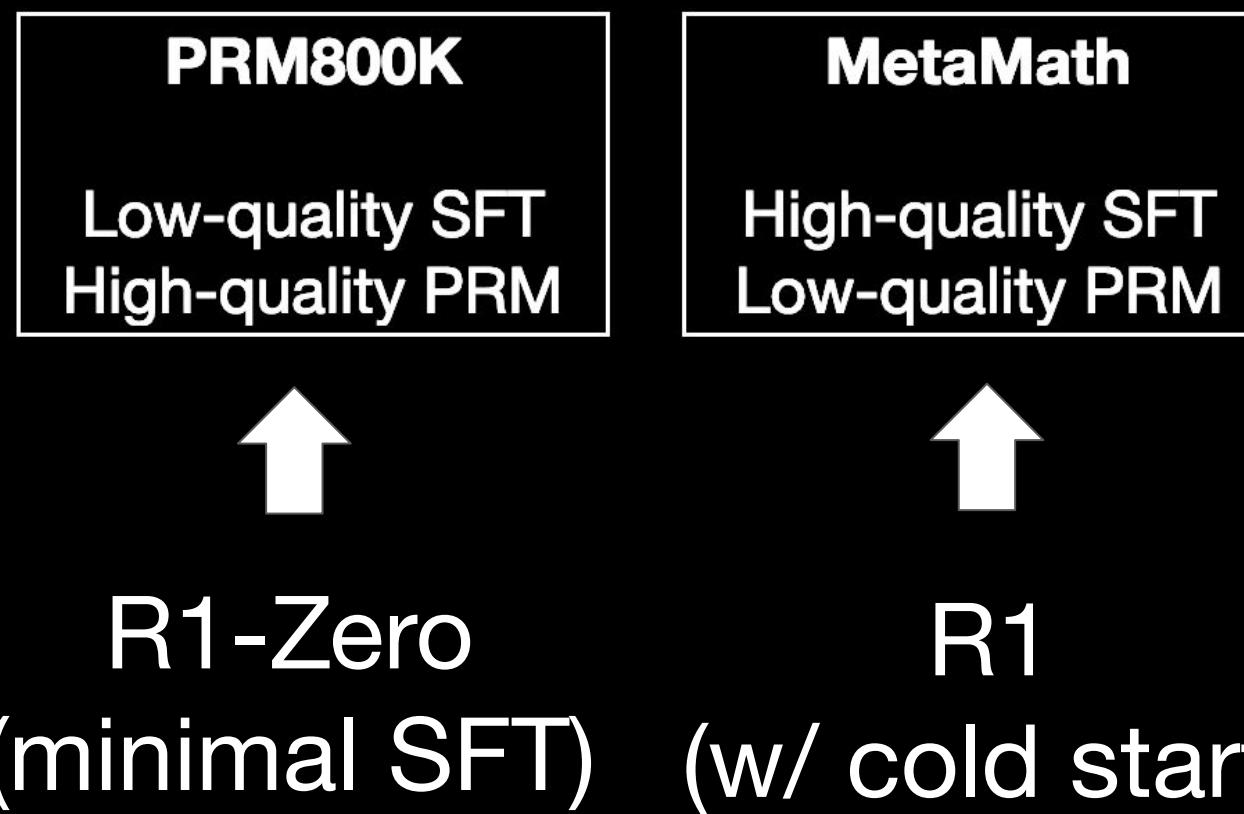




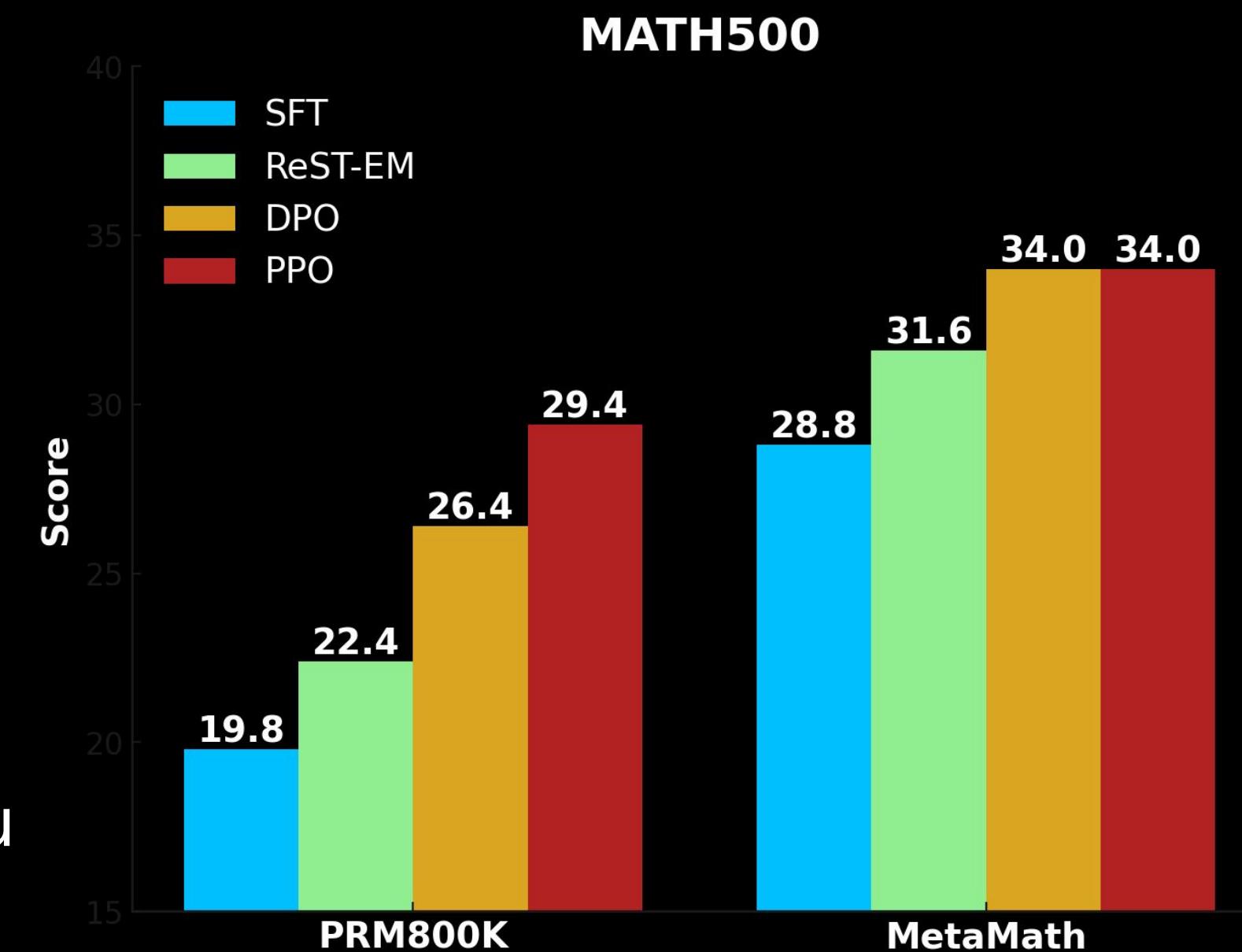
Easy-to-Hard Generalization

Insights:

- Evaluation is easier than generation (NP problems)



Takeaway: use PPO or its variants when you don't have cold start data.





Easy-to-Hard Generalization

What is the total surface area, in square inches, of a cube with a volume of \$1\$ cubic foot?

PRM800K - SFT

The volume of a cube is given by $V = s^3$, where s is the length of one side.

We are given that $V = 1$ cubic foot, so $s^3 = 1$.

Taking the cube root of both sides, we find that $s = 1$.

The surface area of a cube is given by $A = 6s^2$, so the total surface area of this cube is $A = 6(1^2) = \boxed{6}$ square inches.

Answer

6

PRM800K - PPO

So the volume of a cube is s^3 .

And we know the volume is 1 .

So $s^3 = 1$.

So $s = 1$.

The surface area of a cube is $6s^2$.

So the surface area is $6 \cdot 1^2 = 6$.

So the answer is $\boxed{6}$.

Answer

6

Takeaway: be cautious about crazier reward hacking when using PRMs



Questions?