

# **Best-of- $\infty$ — Asymptotic Performance of Test- Time Compute**

**RIKEN AIP, LLM×ML Workshop, Oct 3 2025**

**Junpei Komiyama (MBZUAI / NYU / RIKEN AIP),**

**joint work with Daisuke Oba (Sci Tokyo), Masafumi Oyamada (NEC)**

(This slide was heavily assisted by LLMs, though the best-of-N strategy was not used.)

# Self-introduction

- Junpei Komiyama

- Visiting scholar in RIKEN AIP, Sequential Decision-making Team

- My timeline:

- Assistant professor at Business School, NYU (till Aug 2025)
- Visiting NEC lab (Now! Sep 2025-Nov 2025)
- Assistant professor in machine learning dept, MBZUAI (Jan 2026 -)

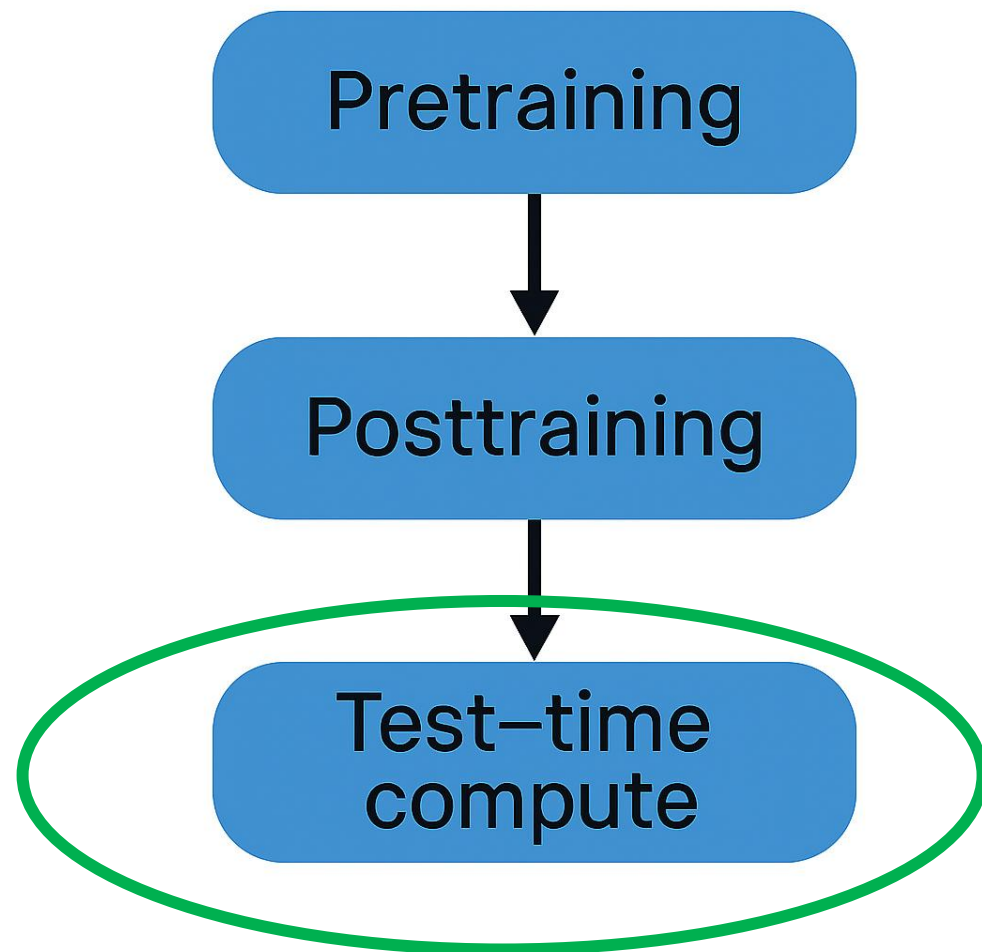
# Agenda

- Introduction
  - Best-of- $\infty$ , the limit of Best-of-N with large N
- Adaptive Sampling:
  - Efficient test-time compute, approaching Best-of- $\infty$  with limited computational budget
- LLM Ensemble
  - Complementary strength of multiple LLMs
- Summary

# Test-time compute

- This talk focuses on test-time compute, where we use LLMs are used to answer questions.


THIS TALK IS ABOUT →



# LLM on Math Reasoning

□ AIME2024 (high-school math competition, pre-Olympiad level):

Q. Eight circles of radius 34 are sequentially tangent, and two of the circles are tangent to sides  $AB$  and  $BC$  of triangle  $ABC$ , respectively. 2024 circles of radius 1 can be arranged in the same manner. The inradius of triangle  $ABC$  can be expressed as  $\frac{m}{n}$ , where  $m$  and  $n$  are relatively prime positive integers. Find  $m + n$ .



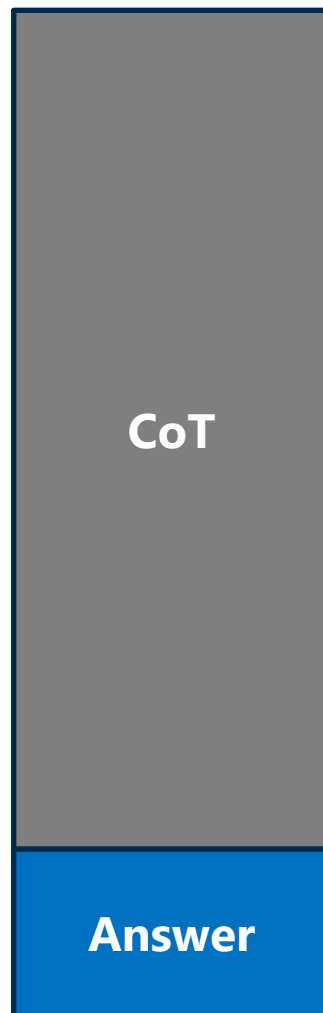
Challenging math question:  
And I ask a local LLM  
(say, gpt-oss-20b, phi-4-reasoning)  
for answering this question.

# Goal

- Math/sci reasoning tasks (e.g., AIME2024/2025, MATH500, GPQA-DIAMOND)
- If we have access to infinite test-time compute resource (= local LLMs), how accurate we can be, in answering such questions?
  - We define Best-of- $\infty$  performance.
- How close we can be, given a finite test-time compute budget?
  - Use computational resources efficiently.

# LLM uses Chain-of-thoughts (CoT) before answering

LLM's answer



30K tokens

>1K tokens

**Lengthy and repetitive "reasoning process"**

Step 1 is XX Step 2 is YY

Wait, Step 1 is wrong, the correct step is...

Alternatively Step 2 can be...

Wait, this is ...

`<think>We are given: "Eight circles of radius 34 are sequentially tangent, and two of the circles are tangent to AB and BC of triangle ABC, respectively. 2024 circles of radius 1 can be arranged in the same manner. The inradius of triangle ABC can be expressed as  $m/n$ , where  $m$  and  $n$  are relatively prime positive integers. Find  $m+n$ ."`

`We need to re-read the problem: "Eight circles of radius 34 are sequentially tangent, and two of the circles are tangent to AB and BC of triangle ABC, respectively. 2024 circles of radius 1 can be arranged in the same manner. The inradius of triangle ABC can be expressed as  $m/n$ , where  $m$  and  $n$  are relatively prime positive integers. Find  $m+n$ ."`

`Understanding: Possibly the problem is about circles arranged tangent along the incircle of a triangle? Possibly it's the incircle chain. But reading: "Eight circles of radius 34 are sequentially tangent, and two of the circles are tangent to AB and BC of triangle ABC, respectively." Also "2024 circles of radius 1 can be arranged in the same manner." Possibly the arrangement follows the incircle chain. On side AB, the incircle touches AB at some point. There is a chain of circles arranged along the incircle chain. The word "sequentially" might indicate that there is a chain of circles arranged inside the triangle? They are tangent to each other sequentially and also tangent to sides of the triangle. And then "two of the circles are tangent to AB and BC of triangle ABC, respectively." So which ones? Possibly the circle tangent to AB is the first one in the chain and the circle tangent to BC is the last one in the chain. The chain lies along the incircle. The circles are arranged along a line, with one circle tangent to AB and one circle tangent to BC. Then "2024 circles of radius 1 can be arranged in the same manner." Possibly a similar chain of circles of radius 1 can be arranged in the incircle of the triangle. We are asked to find the inradius of triangle ABC.`

`Wait, need to re-read carefully. The circles of radius 34 are sequentially tangent to each other. They are arranged in a line, each tangent to the next, and they all have radius 34. And among these eight circles, two of them are tangent to sides AB and BC of triangle ABC respectively. So the arrangement is as follows: There is a chain of eight circles arranged such that they are tangent to each other sequentially. Two of these circles are externally tangent to two sides of triangle ABC and internally tangent to the incircle. So the chain lies in the corner at B? Possibly the chain of circles touches the incircle of the triangle along the arc from AB to BC? And then "2024 circles of radius 1 can be arranged in the same manner." That means that if we replace the circles of radius 34 with circles of radius 1, we can arrange 2024 of them in the same manner along the incircle? Possibly the chain of circles is placed inside the incircle of the triangle, with each circle tangent to its neighbors and tangent to the sides of the triangle (or incircle)? Let's re-read: "Eight circles of radius 34" suggests a large chain. And "2024 circles of radius 1" suggests a chain of many small circles arranged in the same pattern. The problem likely uses similarity of chain lengths to determine the incircle's radius.`

**Summary of answer**

Step 1: XX

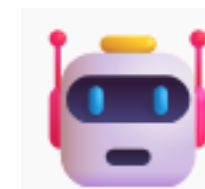
Step 2: YY

...

Thus, the final answer is **42**

`\boxed{2929}`.

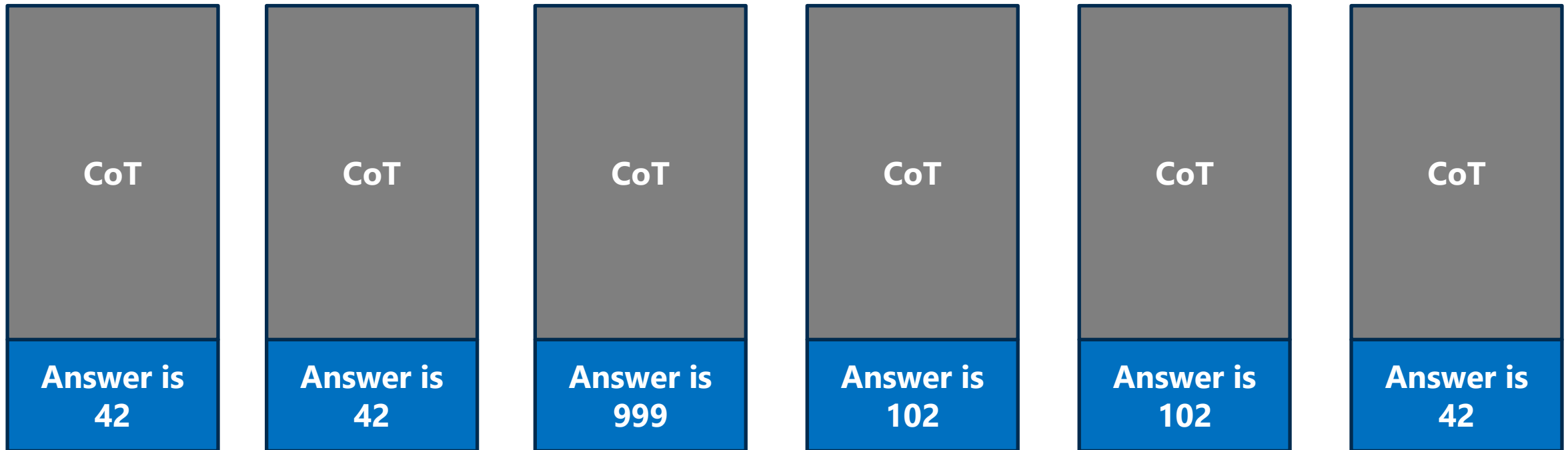
I need to think before answering a difficult question



LLM

# LLM reasoning: Best-of-N

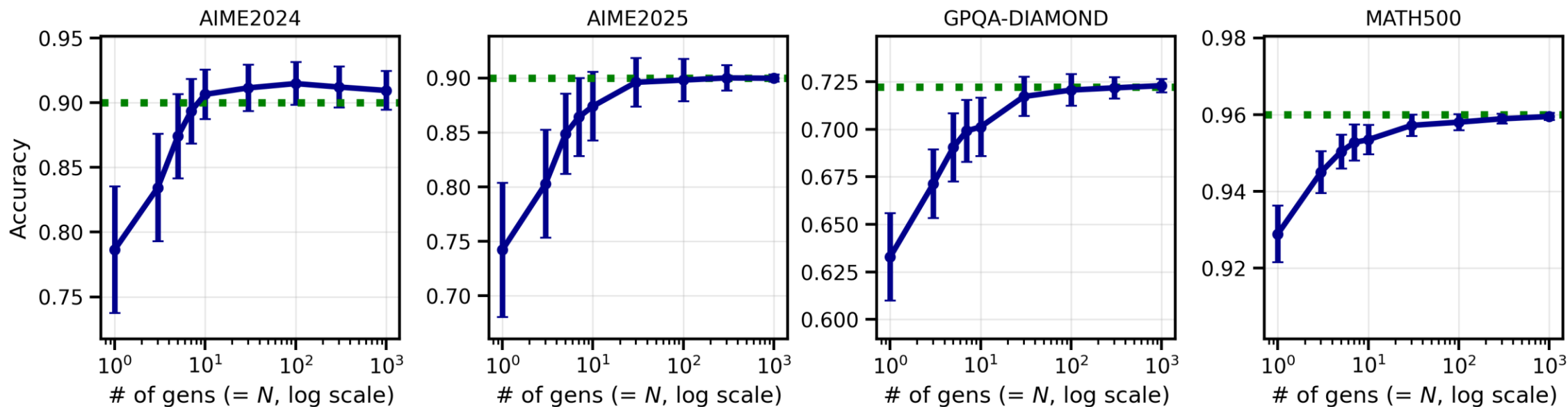
- ❑ LLM's decoding process is random. Sometimes it answers correctly/incorrectly.
- ❑ Use best-of-N: Generate N answers.
- ❑ Selection rule: Majority voting.





# Best-of-N improves accuracy

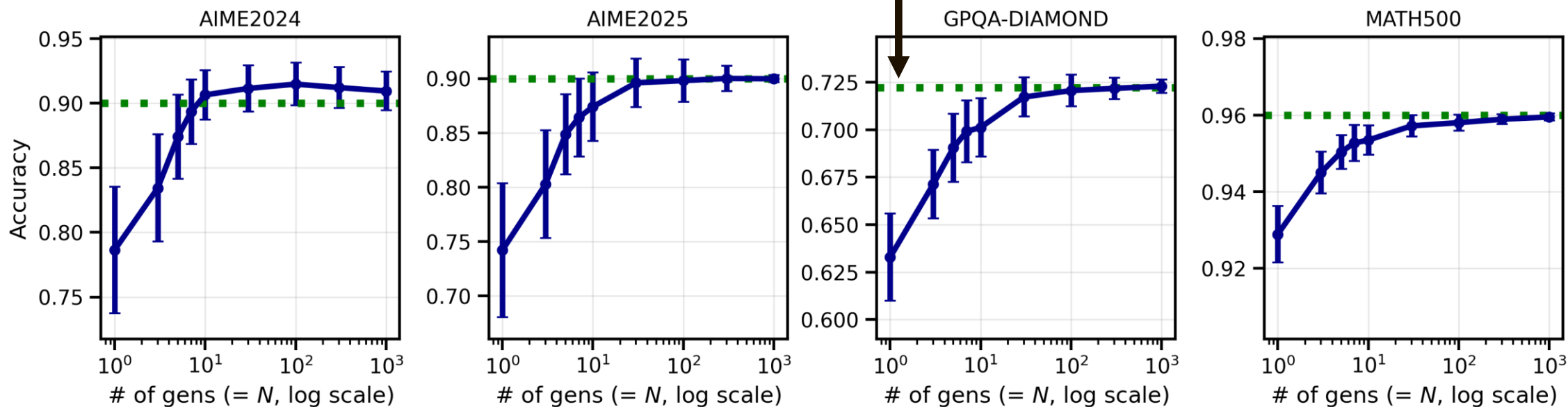
- Larger N, better accuracy.
  - Effective till  $N=100\sim 1000$ .



# Best-of-N improves accuracy

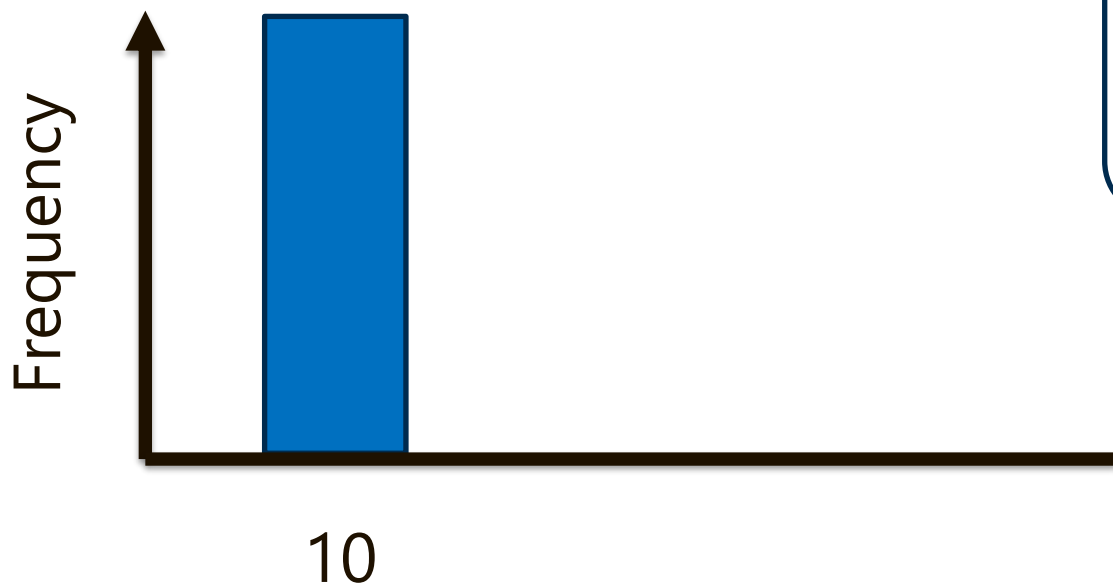
- Larger N, better accuracy.
  - Effective till N=100~1000.

Green dotted line  
= Best-of- $\infty$  performance

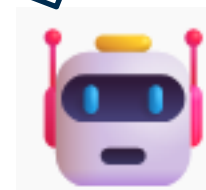


# Best-of- $\infty$ Performance

- At the limit of  $N \rightarrow \infty$ , there is a distribution of answers.
- For an **easy** question, LLM's answer is uniform (and correct)



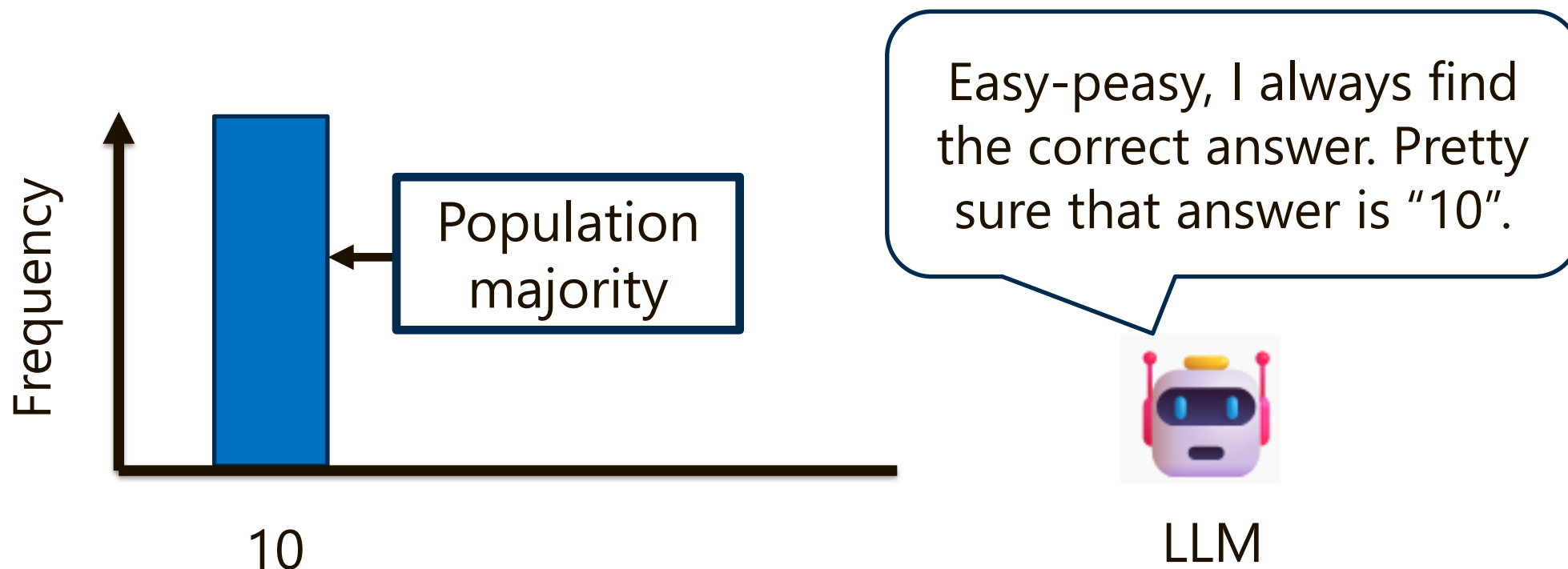
Easy-peasy, I always find the correct answer. Pretty sure that answer is "10".



LLM

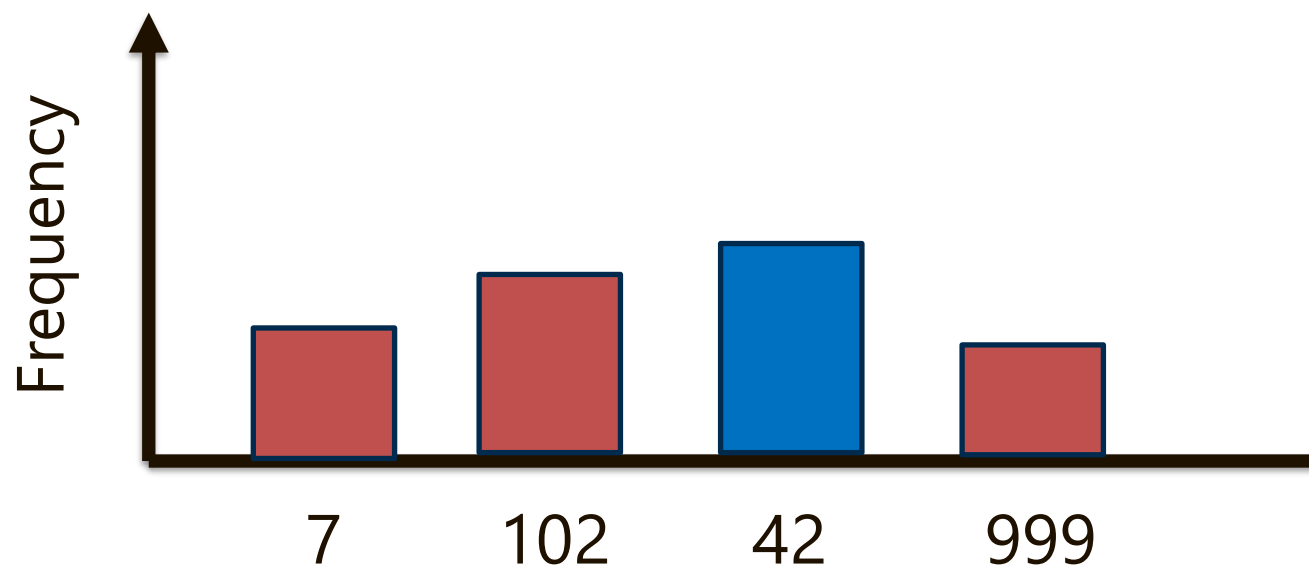
# Best-of- $\infty$ Performance

- At the limit of  $N \rightarrow \infty$ , there is a distribution of answers.
- For an **easy** question, LLM's answer is uniform (and correct)

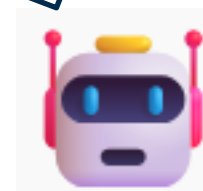


# Best-of- $\infty$ Performance

- At the limit of  $N \rightarrow \infty$ , there is a distribution of answers.
- For a **hard** question, LLM's answer varies



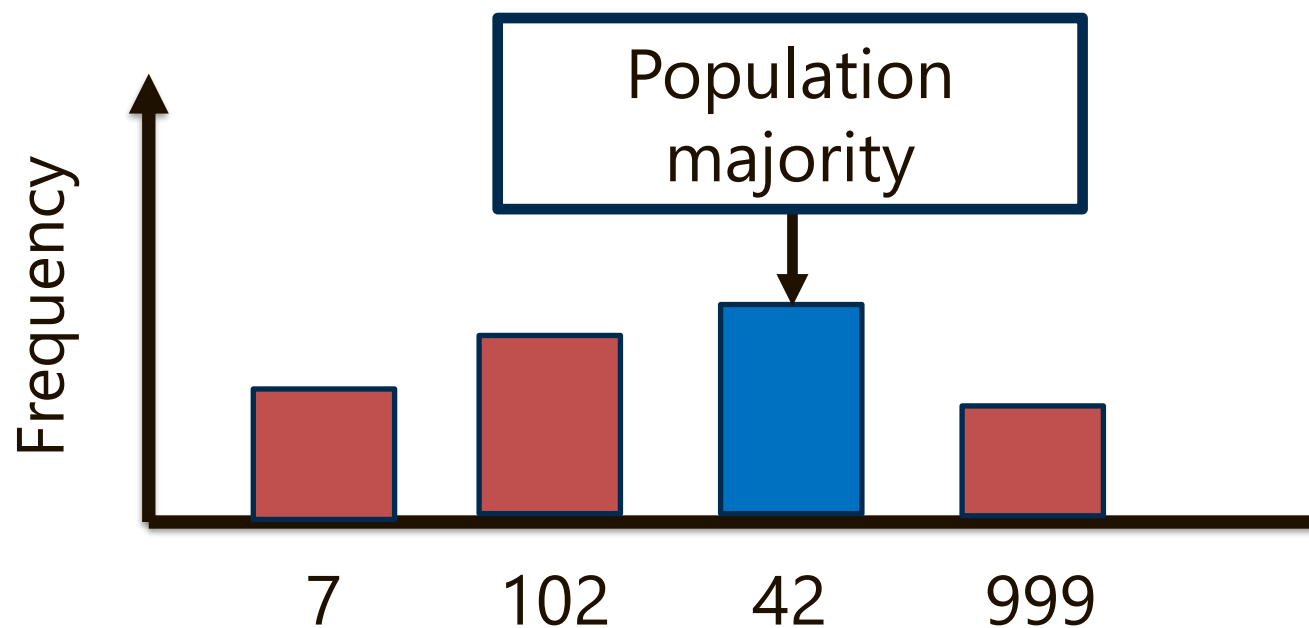
This question is challenging... My chain-of-thought sometimes finds the correct solution path, occasionally leads to an incorrect answer.



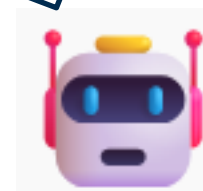
LLM

# Best-of- $\infty$ Performance

- At the limit of  $N \rightarrow \infty$ , there is a distribution of answers.
- For a **hard** question, LLM's answer varies



This question is challenging... My chain-of-thought sometimes finds the correct solution path, occasionally leads to an incorrect answer.



LLM

# Best-of- $\infty$ Performance

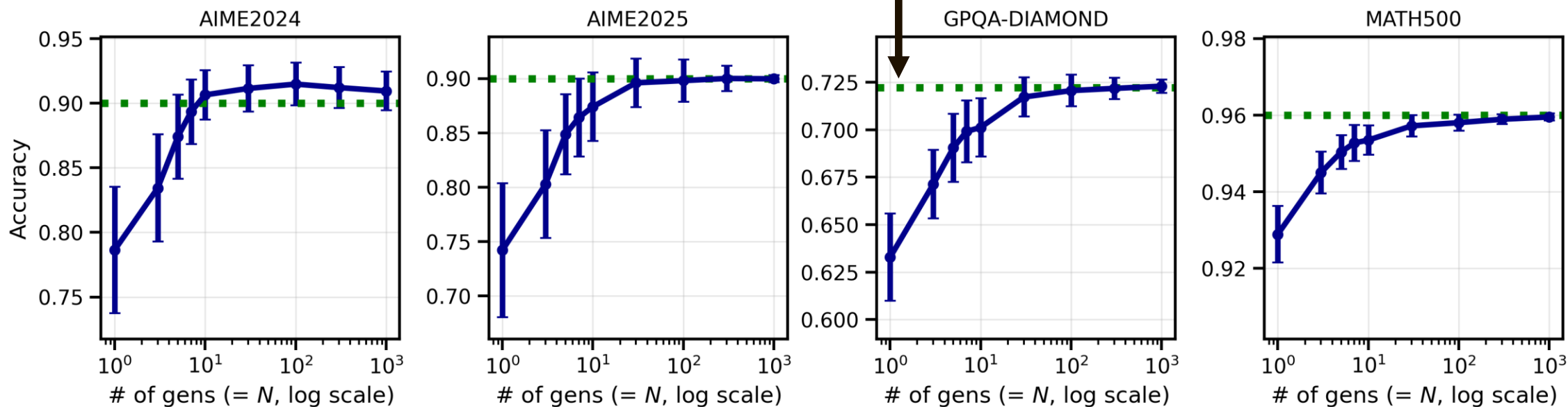
- Best-of- $\infty$  performance is defined as

$$\frac{1}{|\text{Questions}|} \sum_{q \in \text{Questions}} 1[\text{Gold answer} = \text{population majority answer}]$$

- For example, AIME2025 consists of 30 questions. Phi-4-reasoning (Microsoft's open weight LLM)'s population majority matches the gold answer in 25 out of 30 questions.
  - The Best-of- $\infty$  performance is 0.833 (= 25/30)

# Best-of- $\infty$ accuracy

To achieve Best-of- $\infty$  accuracy, we need infinite, or sufficiently large  $N$ , samples





# Adaptive Sampling

- Adaptive sampling: Ask LLM for the next generation or terminate.
- Better use of test-time compute budget:
  - For easy problems -> consistent answers -> early termination
  - For hard problems -> answer varies -> ask LLM for more answers
- Demo: <https://jkomiyama.github.io/bestofinfity/llm-consensus-demo.html>

# Adaptive Sampling: When to terminate?

- Consider it as a hypothesis testing

$H_0$  : The most frequent answer  $A_1$  is not the true majority.

$H_1$  : The most frequent answer  $A_1$  is the true majority.

- Confidence to  $H_1$ : Termination based on thresholding Bayes factor:

$$\begin{aligned} \text{BF}(n) &:= \frac{\mathbb{P}(\mathcal{D}(n)|H_1)}{\mathbb{P}(\mathcal{D}(n)|H_0)} = \frac{\mathbb{P}(H_1|\mathcal{D}(n))}{\mathbb{P}(H_0|\mathcal{D}(n))} \cdot \frac{\mathbb{P}(H_0)}{\mathbb{P}(H_1)} \quad (\text{Bayes' theorem}) \\ &\approx s(n) \frac{\mathbb{P}(H_1|\mathcal{D}(n))}{\mathbb{P}(H_0|\mathcal{D}(n))} \end{aligned}$$

# of unique answers so far

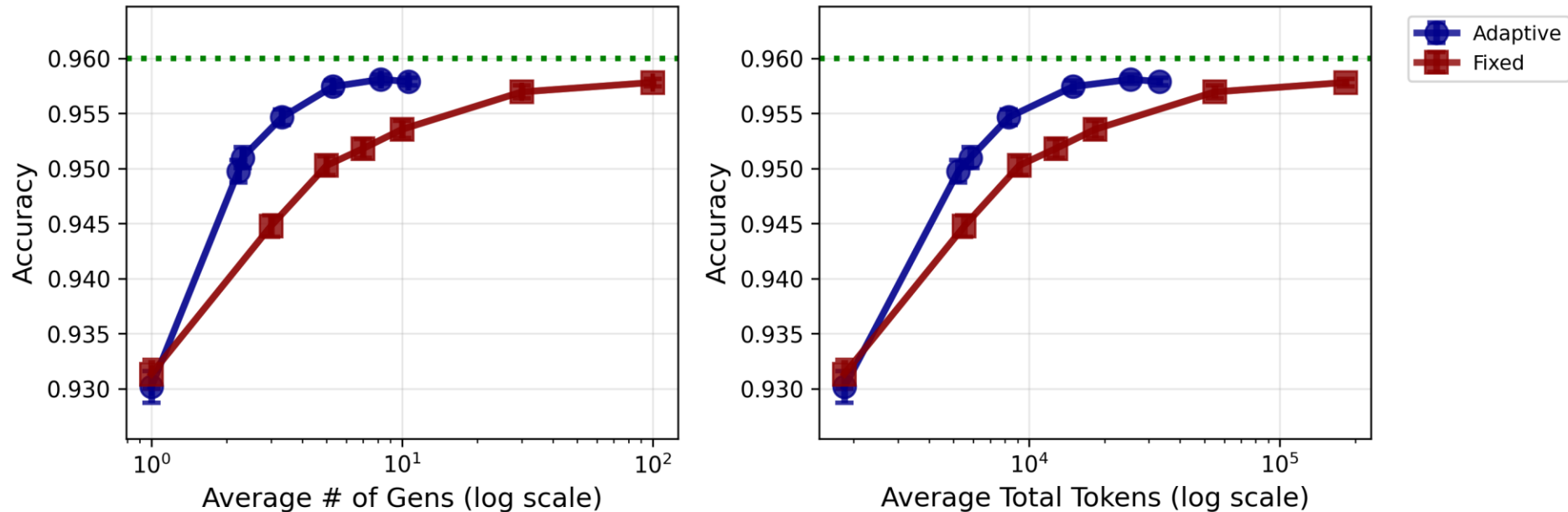
Posterior ratio, modelled as a Dirichlet process

# Algorithm for Adaptive Sampling

- Algorithm - Repeat the follows:
  - Sample (= ask LLM to generate) answers from the LLM.
  - Update counts and the Bayes factor after every generation.
  - If  $BF \geq B$  (threshold) or had  $N_{\max}$  generations, terminate.
- Output the majority answer among collected generations.
  
- Theorem (consistency): As  $B, N_{\max} \rightarrow \infty$ , the procedure matches Best-of- $\infty$  accuracy almost surely.

# Adaptive Sampling: Empirical Performance on MATH500

- Adaptive sampling based on Bayes factor is very efficient:
  - Average  $N = 3$  -> comparable to fixed sampling with  $N = 10$
  - Average  $N = 10$  -> comparable to fixed sampling with  $N = 100$



LLM: GPT-OSS-20B, Dataset: MATH500

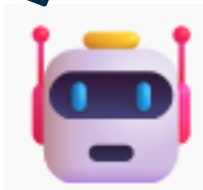
# Agenda

- Introduction
  - Best-of- $\infty$
- Adaptive Sampling:
  - Efficient test-time compute, approaching Best-of- $\infty$  with limited computational budget
- **LLM Ensemble**
  - **Complementary strength of multiple LLMs**
- Summary

# Why Ensembles?

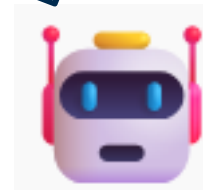
- ❑ Different LLMs offer complementary strengths on reasoning tasks.
- ❑ Weighted sampling per generation lets weaker-but-diverse models contribute.
- ❑ Objective: maximize majority-vote accuracy in the limit by **weighting** LLMs.

I perform well on  
questions 1 and 3



LLM 1

I perform well on  
questions 2 and 4



LLM 2

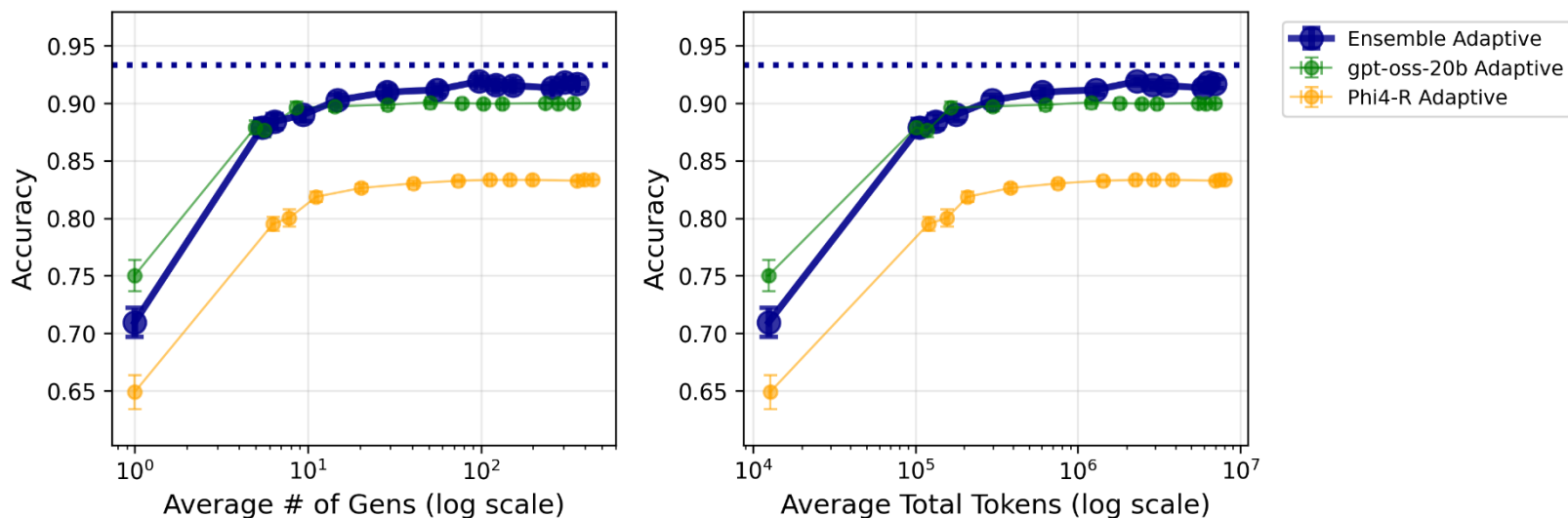
# Adaptive LLM Sampling: Algorithm

- Input: List of LLMs  $i = 1, 2, \dots, K$
- Repeat the followings:
  - **Select LLM with weight vector  $\mathbf{w} = (w_1, w_2, \dots, w_K)$ .**
  - Ask the LLM to generate answers from the LLM.
  - Update counts and the Bayes factor after every generation.
  - If  $BF \geq B$  or had  $N_{\max}$  generations, terminate.
- Output the majority answer among collected generations.

# Best-of- $\infty$ of Ensemble > max Best-of- $\infty$ of single LLM

□ Example: On AIME2025,

- Best-of- $\infty$  of gpt-oss-20b: 0.900 (27/30)
- Best-of- $\infty$  of Phi-4-reasoning: 0.833 (25/30)
- Best-of- $\infty$  of weighted mixture: 0.933 (28/30)





# Optimal weight

- The weight  $w_1, w_2, \dots, w_K$  optimization is reduced to the following **mixed integer linear programming** (MILP).

$$\max_{w \in \Delta^K, y \in \{0,1\}^N} \sum_q y_q$$

# of correctly answered questions

$$\text{s.t. } w_i \geq 0 \quad \forall_i$$

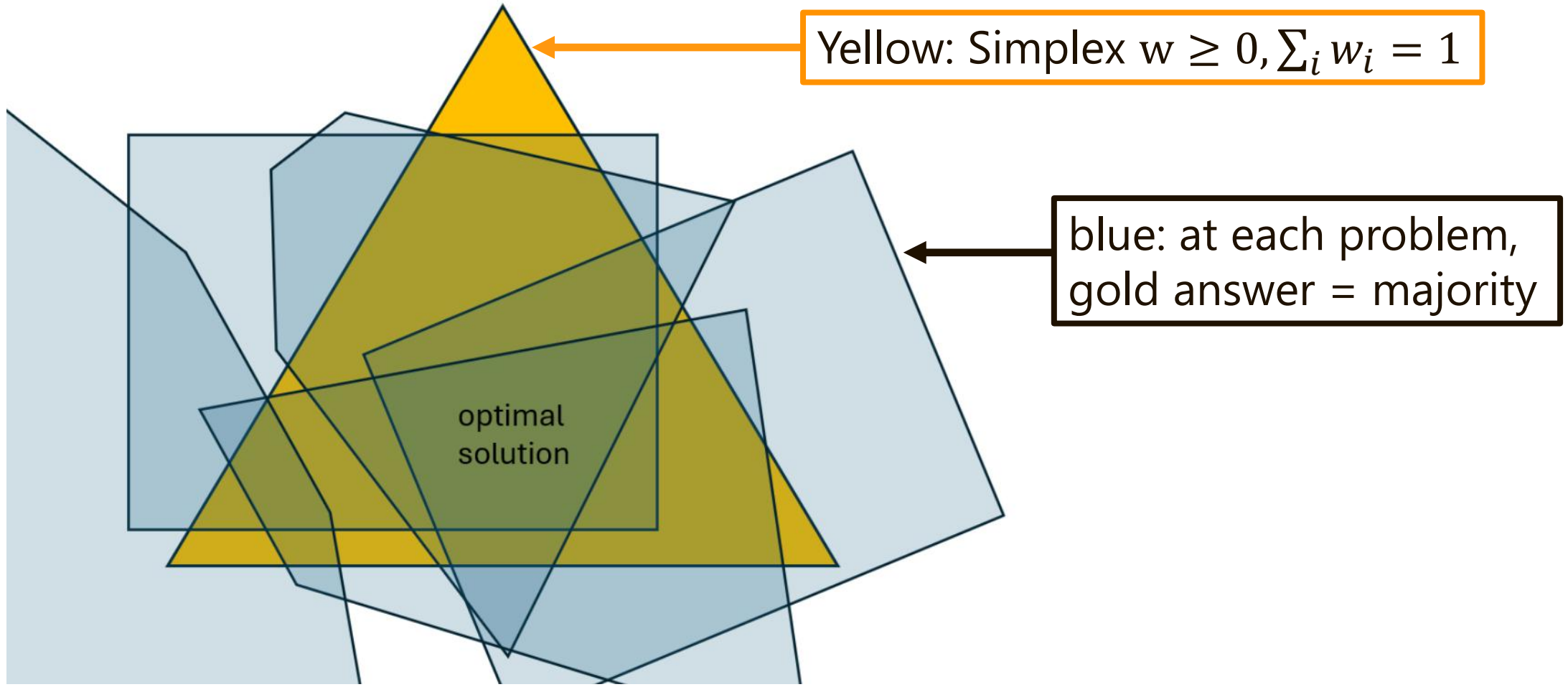
$$\sum_i w_i = 1$$

weight is on simplex (summation 1)

$$A_q w \geq -m(1 - y_q) \quad \forall q$$

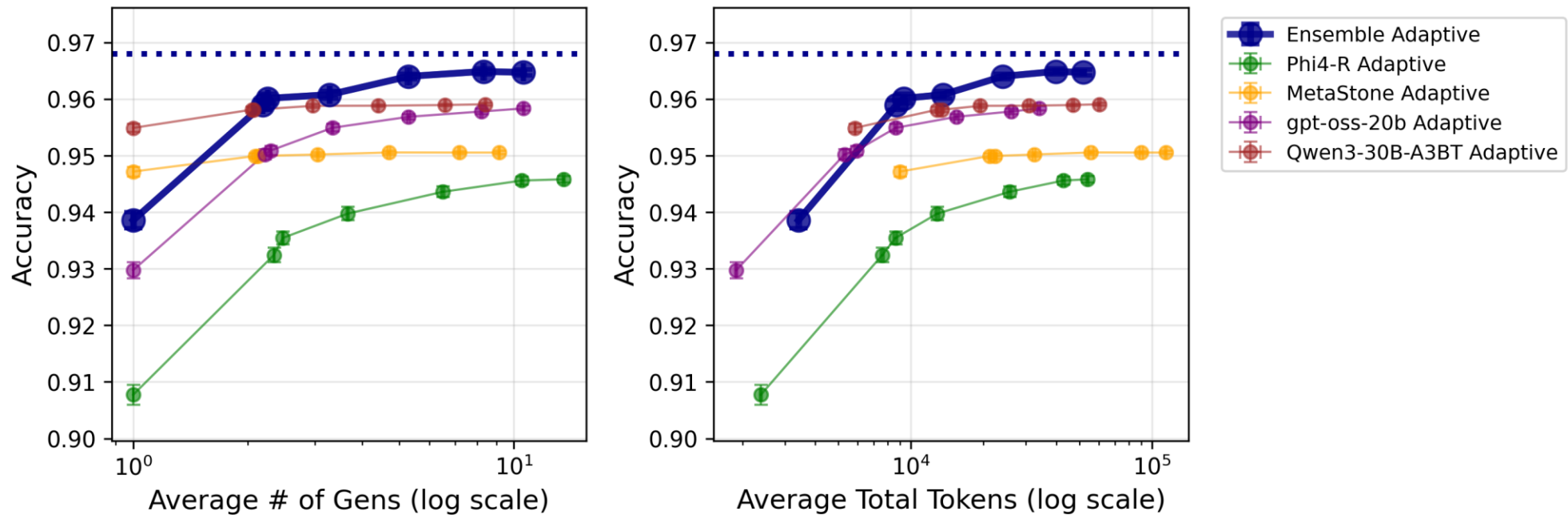
gold answer = population majority

# Illustration of the optimal weight



# Scalability

- ❑ MILP is NP-complete to solve, does not scale for extremely large instance.
- ❑ In practice, Optimization easily scales up to  $\sim 10$  LLMs and  $\sim 500$  questions.



LLM Ensemble performance in the MATH500 dataset

# We have released the generation dataset

- To estimate Best-of- $\infty$  accuracy, we generate  $\geq 80$  answers for each question.
- Over 500k generated solutions, up to 800M tokens per LLM.

11 LLMs



| LLM                         | # of files | total generated tokens | total file size (MB) |
|-----------------------------|------------|------------------------|----------------------|
| AM-Thinking-v1              | 4,800      | 79,438,111             | 185.95               |
| Datarus-R1-14B-preview      | 4,800      | 49,968,613             | 127.03               |
| EXAONE-Deep-32B             | 60,640     | 478,575,594            | 1,372.35             |
| GPT-OSS-20B                 | 68,605     | 244,985,253            | 98.59 <sup>6</sup>   |
| LIMO-v2                     | 6,095      | 77,460,567             | 219.45               |
| MetaStone-S1-32B            | 60,757     | 806,737,009            | 2,458.48             |
| NVIDIA-Nemotron-Nano-9B-v2  | 60,640     | 295,466,626            | 897.82               |
| Phi-4-reasoning             | 168,138    | 558,980,037            | 1,841.06             |
| Qwen3-4B                    | 20,640     | 547,170,887            | 1,704.28             |
| Qwen3-14B                   | 44,800     | 666,466,780            | 1,822.13             |
| Qwen3-30B-A3B-Thinking-2507 | 60,640     | 436,865,220            | 1,234.28             |

Table 1: Statistics of the large-scale generation dataset that we used in our experiments. Each file corresponds to a single answer. We release it with our code.

# Summary

- Best-of- $N$  with adaptive  $N$  approaches to Best-of- $\infty$  performance.
- Weighted LLM Ensemble.
  - Outperforms any single LLM.
  - Optimized weights via mixed-integer linear programming (MILP).
- Publicly available dataset: 11 open-weight reasoning LLMs  $\times$  4 hard benchmarks  $\times$  80+ generations each.

Paper URL (QR below):

<https://jkomiyama.github.io/bestofinfy/>



# We are hiring!

- I am joining MBZUAI, UAE in Jan 2026 and hiring/hosting at any levels (post-docs, visiting scholars etc.).
  - Shoot me an email if you are interested. I will be around Tokyo till the end of November.



Paper URL (QR below):

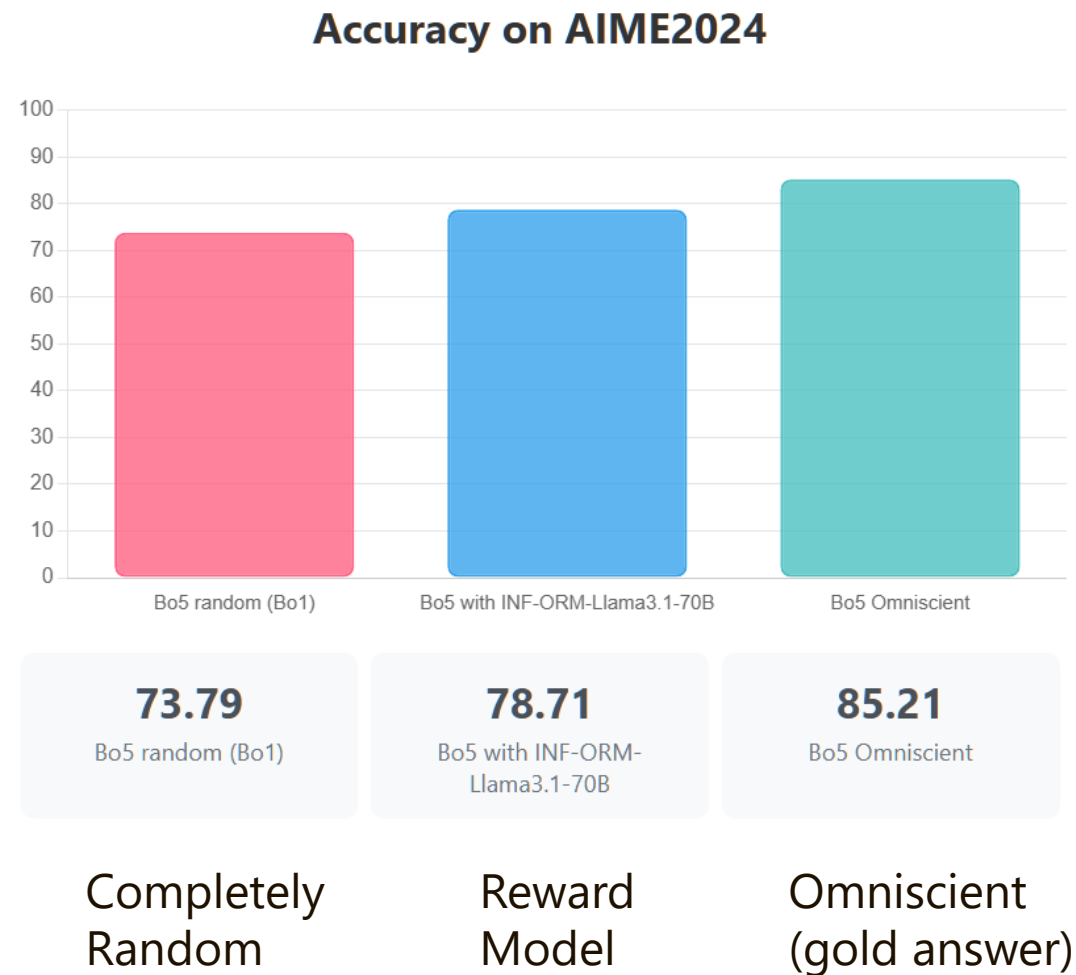
<https://jkomiyama.github.io/bestofinfy/>



**(the following are supplementary materials)**

# LLM reasoning: Best-of-N

- Reward-hacking: Best answer in terms of reward is NOT always the correct answer.
- On heavy-reasoning tasks, best reward models are still random-ish.
- How can we get close to “omniscient” BoN?



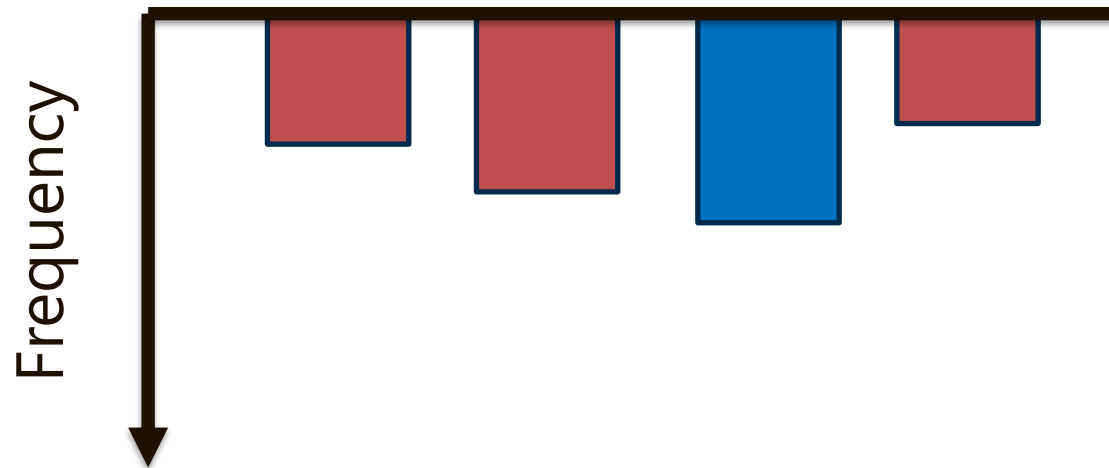




(A)

(B)

(C)



(A) (A) (A)

→ Terminate

(D)



(B) (C) (A) (C) (D) (C)

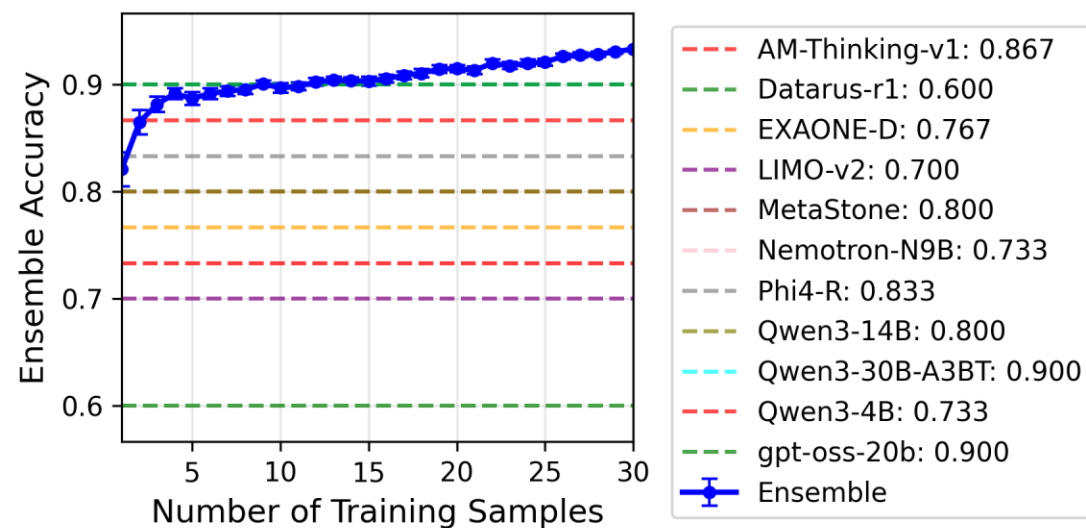
→ Terminate

# Additional experimental findings

- ❑ Transfer learning: Weights trained on AIME2024 beat or tie best single model on AIME2025 in 64% of 165 combinations.
  - Moderately effective?
- ❑ Selection comparison (Bo5 on AIME2025): Majority voting outperforms reward models and LLM-as-a-judge variants.
  - Majority voting is still one of the most useful aggregation.

# How many gold answers are needed to obtain good weights?

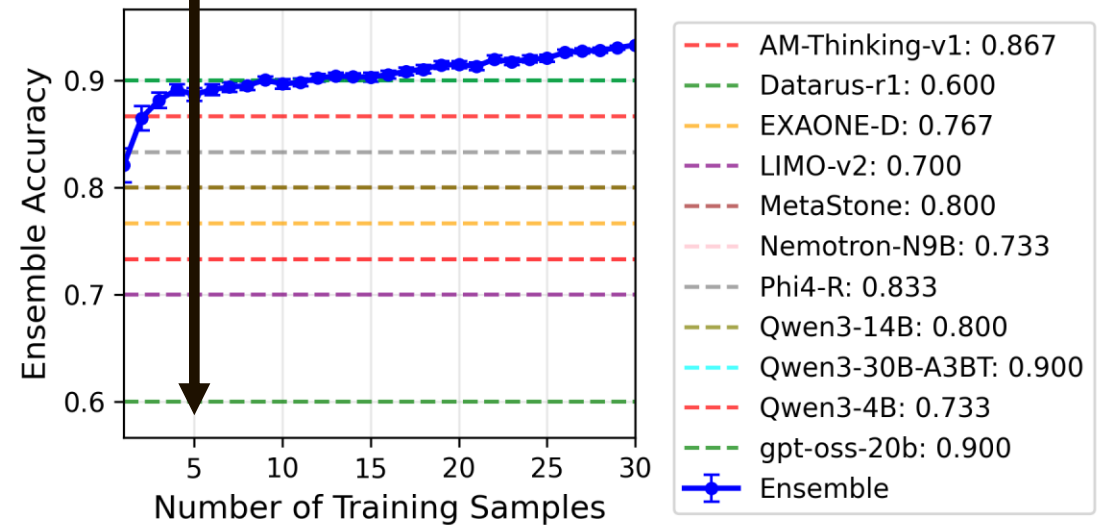
AIME2025,  
11 LLMs



# How many gold answers are needed to obtain good weights?

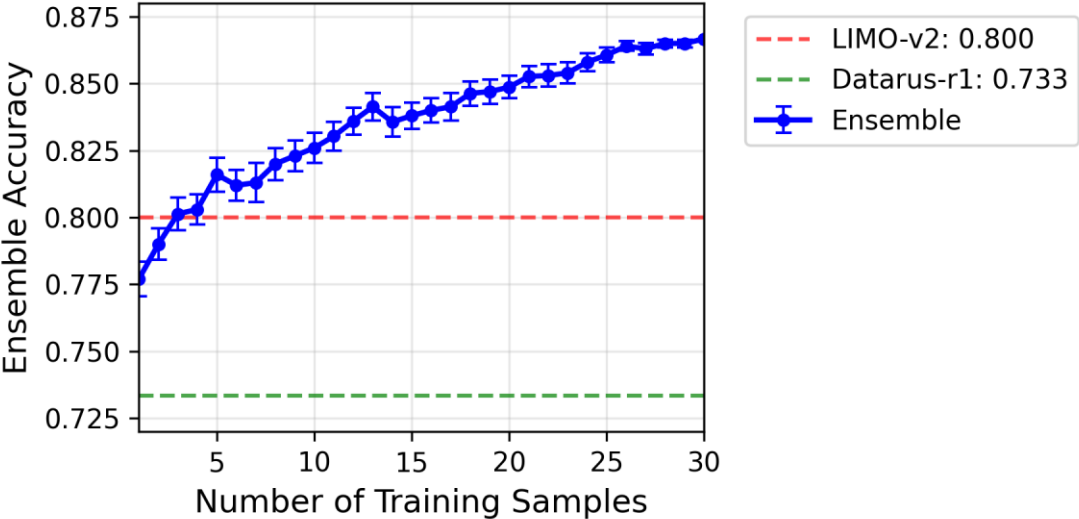
5 gold answers to find a reasonable weight vector

AIME2025,  
11 LLMs



# How many gold answers are needed to obtain good weights?

AIME2024,  
2 LLMs



AIME2025,  
11 LLMs

