



# **CS 839: Foundation Models Reasoning, Verifiers, Rubrics**

Fred Sala

University of Wisconsin-Madison

**Oct. 23, 2025**

# Announcements

- **Logistics:** Grades for HW1 soon, project info as well
- **Presentations:** submit your proposals
- Class roadmap:

Thursday Oct. 23	More reasoning
Tuesday Oct. 28	Multimodal models
Thursday Oct. 30	Scaling Laws
Tuesday Nov. 4	Diffusion Models
Thursday Nov. 6	Security, Privacy, Toxicity + Future Areas

# Outline

- **Reasoning Review & Test-Time Scaling**
  - Back to CoT, reasoning types, test-time scaling, understanding & improving TTS, reasoning in pretraining
- **Verifiers and Rubrics**
  - Types of verifiers, weak verifiers, verifier design, rubrics, examples
- **Multimodal Reasoning**
  - From unimodal to multimodal reasoning, evals

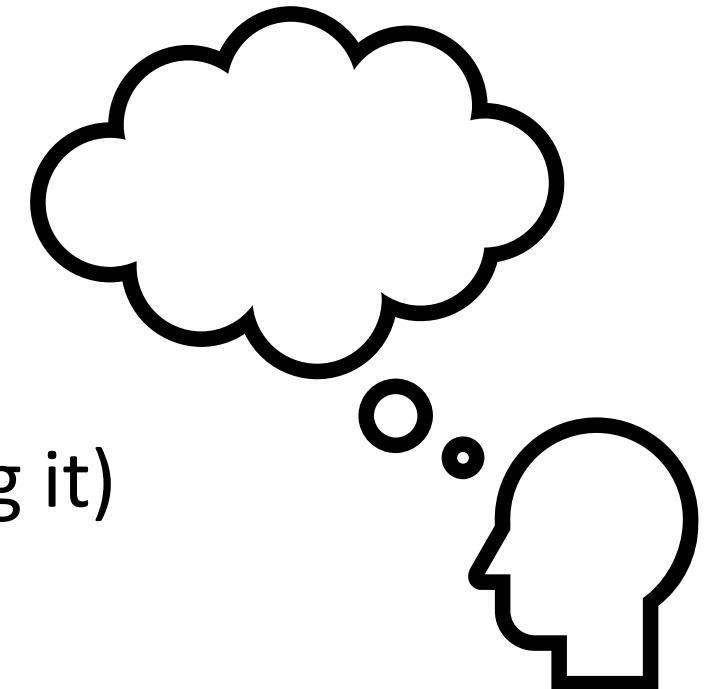
# Outline

- **Reasoning Review & Test-Time Scaling**
  - Back to CoT, reasoning types, test-time scaling, understanding & improving TTS, reasoning in pretraining
- **Verifiers and Rubrics**
  - Types of verifiers, weak verifiers, verifier design, rubrics, examples
- **Multimodal Reasoning**
  - From unimodal to multimodal reasoning, evals

# Back to Reasoning

We've seen some approaches for inducing reasoning in LLMs/FMs

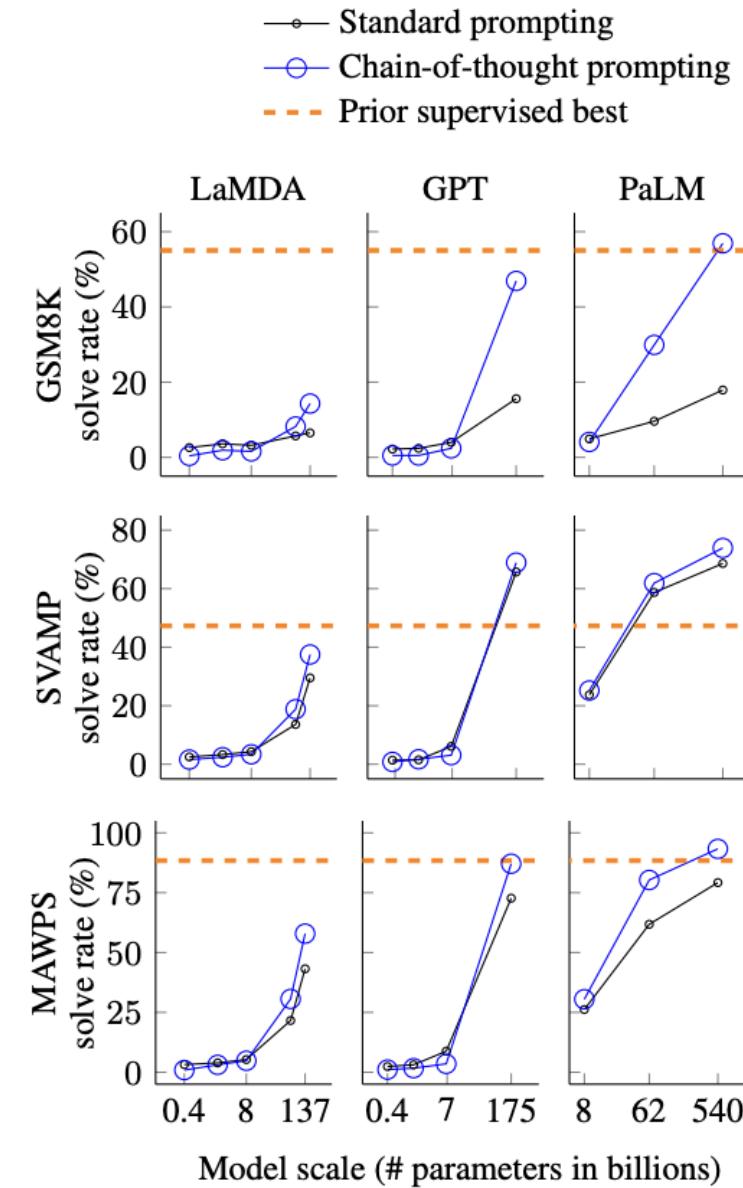
- In **multiple** ways,
  - At inference time: **chain-of-thought**
  - During post-training: via **SFT** or **RL**
  - Even during **pre-training**
- In all cases, we'll need signal for correctness (and lots of choices for getting it)



# Review: Chain-of-Thought

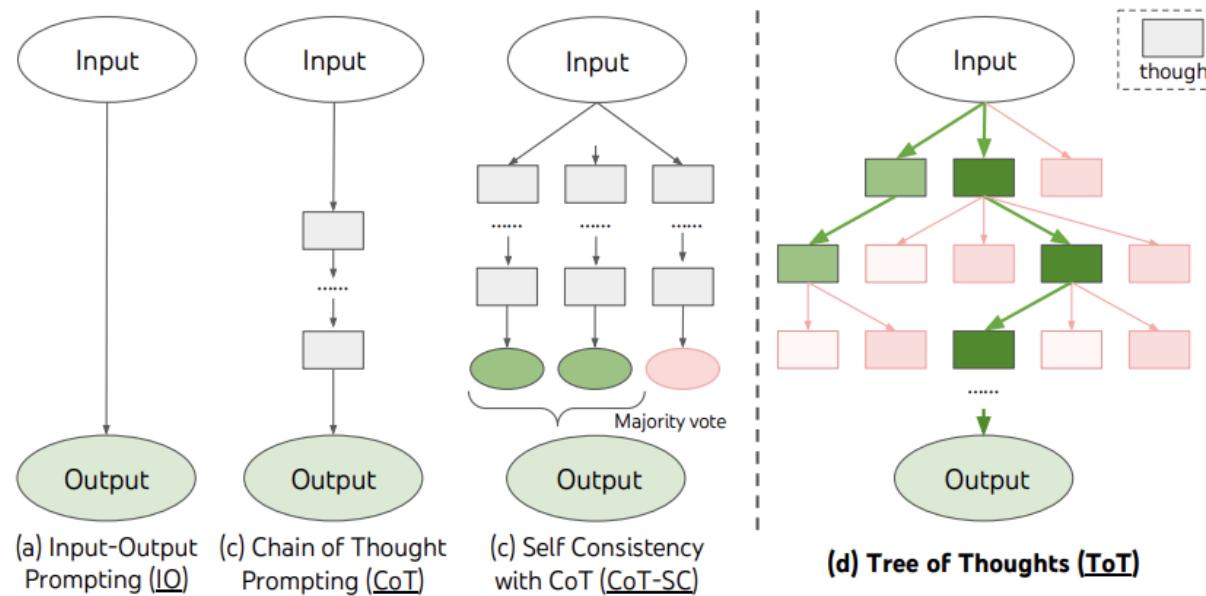
Simple but can obtain a big boost in performance

- Downsides: linear path, limited exploration
  - Recall only helps in some areas
- Early form of **test-time scaling**
  - (more coming in a few slides)



# Review: Chain-of-Thought Variants

- **Tree-of-Thoughts et al:** permits branching and backtracking (important for more complex reasoning)
- Enables exploration of multiple reasoning paths
- External harness that requires intervention (plus lots of compute)



# Breaking Down Reasoning Types

Not all forms of reasoning will have equal performance,

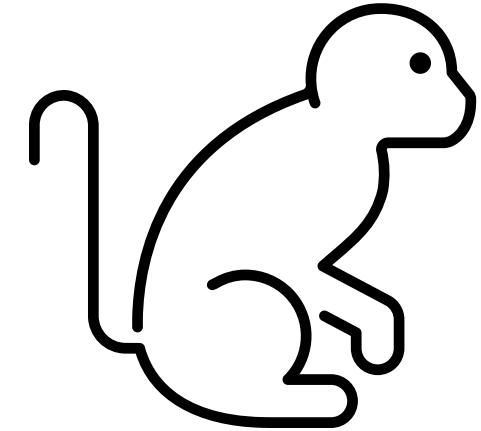
- Example: **deductive** vs. **abductive** reasoning gaps (Dougrez-Lewis et al., 2025)
- Need to evaluate specific forms of reasoning & improve them

Model	VitaminC			CLIMATE-FEVER			PHEMEPlus		
	F1 ↑	Deductive ↓	Abductive ↓	F1 ↑	Deductive ↓	Abductive ↓	F1 ↑	Deductive ↓	Abductive ↓
Claude ZS <i>No-Exp</i>	0.85	13.62	33.33	0.80	12.81	40.20	0.73	19.40	<b>38.89</b>
Claude M-CoT <i>No-Exp</i>	0.87	12.77	<b>23.33</b>	0.80	12.81	41.84	<b>0.76</b>	<u>18.53</u>	<b>38.89</b>
GPT-4 ZS <i>No-Exp</i>	0.86	12.13	30.00	<u>0.87</u>	<b>8.79</b>	<b>20.59</b>	0.69	20.69	<b>38.89</b>
GPT-4 M-CoT <i>No-Exp</i>	<b>0.90</b>	<b>8.30</b>	<u>26.67</u>	0.85	10.05	27.45	0.70	22.41	52.78
GPT-4o ZS <i>No-Exp</i>	0.88	10.43	40.00	0.84	9.55	33.33	0.72	20.04	<u>41.67</u>
GPT-4o M-CoT <i>No-Exp</i>	0.88	10.43	30.00	<b>0.92</b>	<u>9.05</u>	<u>25.49</u>	0.74	19.40	47.22

# Test-Time Scaling

We often talk about **scaling** up models

- **Traditional scaling:**
  - Model size, data amount, compute amount
- **Test-time scaling:** increase inference compute
  - More samples,
  - Longer chains (in specific ways)
- **Intuition:** “monkeys at typewriters” — more attempts at inference increase chance of hitting the correct answer.



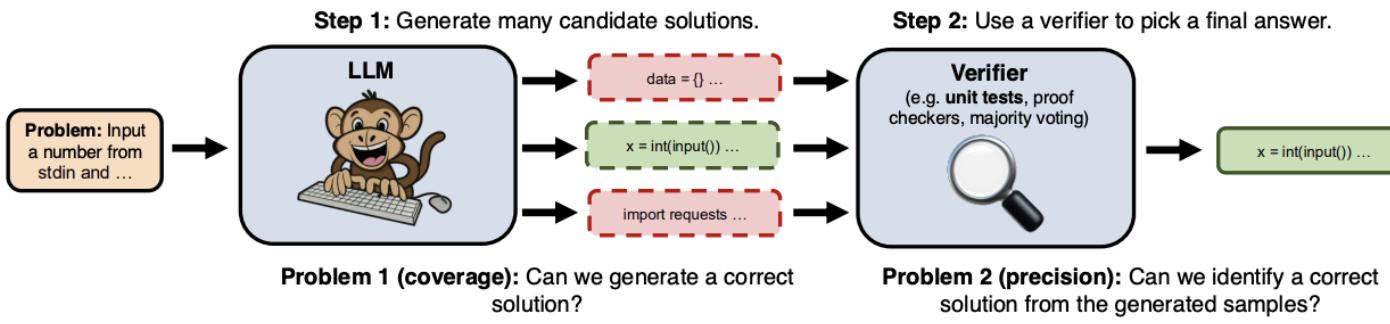
# Example: Large Language Monkeys

Generate candidate solutions,

- Rank & select with a verifier
  - AKA Best-of-n

Two key goals:

- Coverage, precision



## Large Language Monkeys: Scaling Inference Compute with Repeated Sampling

Bradley Brown\*†‡, Jordan Juravsky\*†, Ryan Ehrlich\*†, Ronald Clark‡, Quoc V. Le§, Christopher Ré†, and Azalia Mirhoseini†§

\*Department of Computer Science, Stanford University

†University of Oxford

§Google DeepMind

`bradley.brown@cs.ox.ac.uk`, `bjb@stanford.edu`, `ryanehrlich@cs.stanford.edu`,  
`ronald.clark@cs.ox.ac.uk`, `qvl@google.com`, `chrismre@stanford.edu`,  
`azalia@stanford.edu`

### Abstract

Scaling the amount of compute used to train language models has dramatically improved their capabilities. However, when it comes to inference, we often limit models to making only one attempt at a problem. Here, we explore inference compute as another axis for scaling, using the simple technique of repeatedly sampling candidate solutions from a model. Across multiple tasks and models, we observe that coverage – the fraction of problems that are solved by any generated sample – scales with the number of samples over four orders of magnitude. Interestingly, the relationship between coverage and the number of samples is often log-linear and can be modelled with an exponentiated power law, suggesting the existence of inference-time scaling laws. In domains like coding and formal proofs, where answers can be automatically verified, these increases in coverage directly translate into improved performance. When we apply repeated sampling to SWE-bench Lite, the fraction of issues solved with DeepSeek-Coder-V2-Instruct increases from 15.9% with one sample to 56% with 250 samples, outperforming the single-sample state-of-the-art of 43%. In domains without automatic verifiers, we find that common methods for picking from a sample collection (majority voting and reward models) plateau beyond several hundred samples and fail to fully scale with the sample budget.

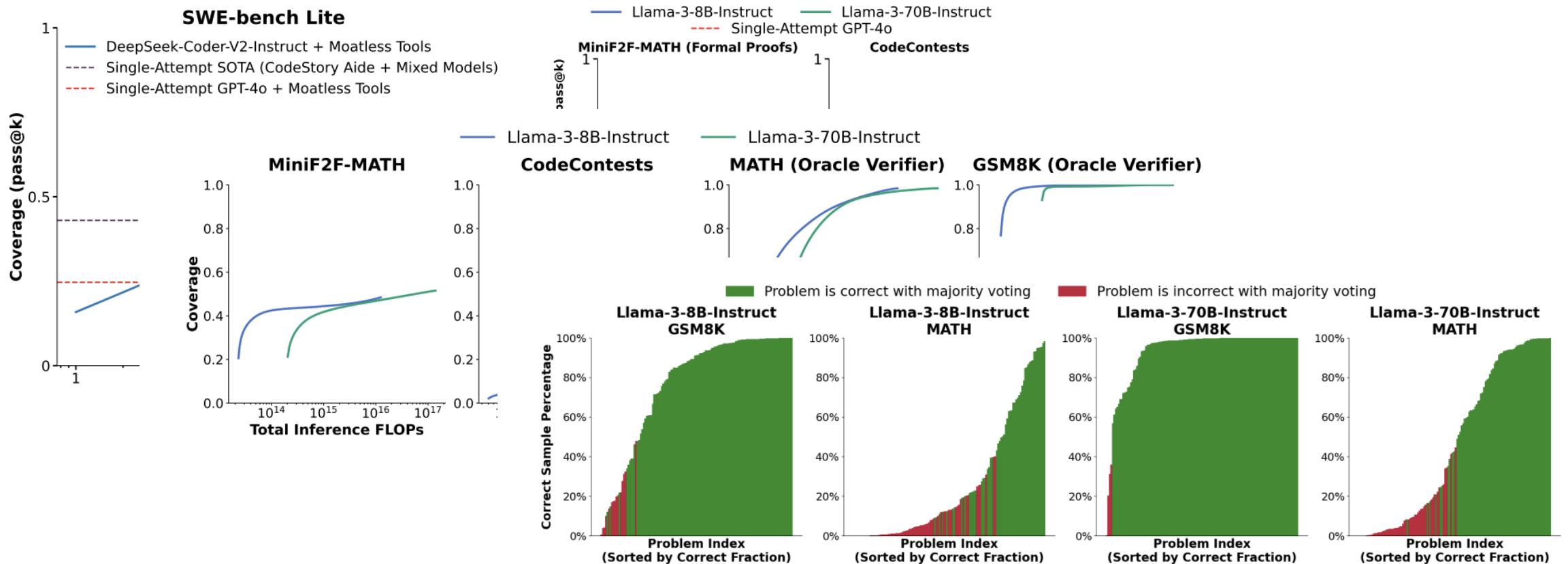
### Introduction

Ability of large language models (LLMs) to solve coding, mathematics, and other reasoning has improved dramatically over the past several years [47, 11, 2, 4]. Scaling the amount of compute through bigger models, longer pre-training runs, and larger datasets has been a consistent driver of these gains [27, 37, 28].

# Example: Large Language Monkeys

**Coverage:** depends on dataset, model

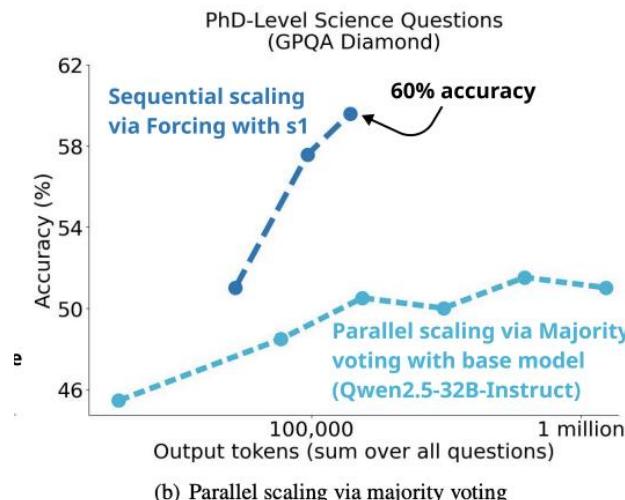
- **Precision:** sample dependent!



# How to Induce Test-Time Scaling?

## S1 paper: simple approach

- Curate a dataset and train on it
- Control amount of thinking
  - “Wait” (make longer)
  - “End” (keep short)



## s1: Simple test-time scaling

Niklas Muennighoff<sup>\*1 3 4</sup> Zitong Yang<sup>\*1</sup> Weijia Shi<sup>\*2 3</sup> Xiang Lisa Li<sup>\*1</sup> Li Fei-Fei<sup>1</sup> Hannaneh Hajishirzi<sup>2 3</sup>  
Luke Zettlemoyer<sup>2</sup> Percy Liang<sup>1</sup> Emmanuel Candès<sup>1</sup> Tatsunori Hashimoto<sup>1</sup>

### Abstract

Test-time scaling is a promising new approach to language modeling that uses extra test-time compute to improve performance. Recently, OpenAI’s o1 model showed this capability but did not publicly share its methodology, leading to many replication efforts. We seek the simplest approach to achieve test-time scaling and strong reasoning performance. First, we curate a small dataset s1K of 1,000 questions paired with reasoning traces relying on three criteria we validate through ablations: difficulty, diversity, and quality. Second, we develop budget forcing to control test-time compute by forcefully terminating the model’s thinking process or lengthening it by appending “Wait” multiple times to the model’s generation when it tries to end. This can lead the model to double-check its answer, often fixing incorrect reasoning steps. After supervised finetuning the Qwen2.5-32B-Instruct language model on s1K and equipping it with budget forcing, our model s1-32B exceeds o1-preview on competition math questions by up to 27% (MATH and AIME24). Further, scaling s1-32B with budget forcing allows extrapolating beyond its performance without test-time intervention: from 50% to 57% on AIME24. Our model, data, and code are open-source at <https://github.com/simplescaling/s1>.

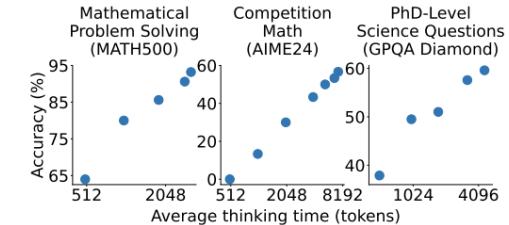


Figure 1. Test-time scaling with s1-32B. We benchmark s1-32B on reasoning-intensive tasks and vary test-time compute.

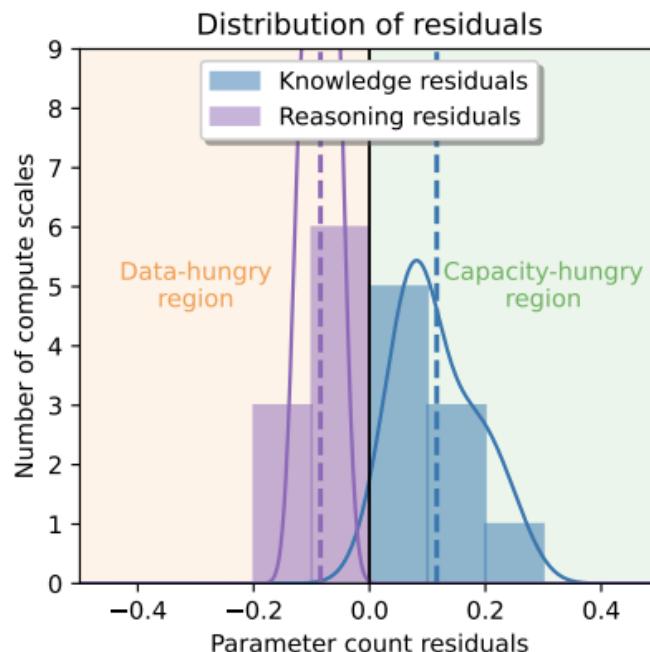
of this approach is to increase the compute at test time to get better results. There has been much work exploring this idea (Snell et al., 2024; Welleck et al., 2024), and the viability of this paradigm was recently validated by OpenAI o1 (OpenAI, 2024). o1 has demonstrated strong reasoning performance with consistent gains from scaling test-time compute. OpenAI describes their approach as using large-scale reinforcement learning (RL) implying the use of sizable amounts of data (OpenAI, 2024). This has led to various attempts to replicate their models relying on techniques like Monte Carlo Tree Search (Gao et al., 2024b; Zhang et al., 2024a), multi-agent approaches (Qin et al., 2024), and others (Wang et al., 2024a; Huang et al., 2024b; 2025). Among these approaches, DeepSeek R1 (DeepSeek-AI et al., 2025) has successfully replicated o1-level performance, also employing reinforcement learning via millions of samples and

Muennighoff et al ‘25

# And In Pretraining...

**Scaling laws** (coming up) predict performance

- But are coarse (just use compute, data, model size)
- Reasoning & knowledge have **dissimilar scaling laws**.



## Compute Optimal Scaling of Skills: Knowledge vs Reasoning

Nicholas Roberts<sup>u†</sup> Niladri Chatterji<sup>σ</sup> Sharan Narang<sup>σ</sup> Mike Lewis<sup>σ</sup> Dieuwke Hupkes<sup>σ</sup>

<sup>u</sup>University of Wisconsin <sup>σ</sup>GenAI at Meta

<sup>†</sup>Work done during an internship at Meta.

Correspondence: [nick11roberts@cs.wisc.edu](mailto:nick11roberts@cs.wisc.edu) [dieuwkehupkes@meta.com](mailto:dieuwkehupkes@meta.com)

### Abstract

Scaling laws are a critical component of the LLM development pipeline, most famously as a way to forecast training decisions such as ‘compute-optimally’ trading-off parameter count and dataset size, alongside a more recent growing list of other crucial decisions. In this work, we ask whether compute-optimal scaling behaviour can be skill-dependent. In particular, we examine knowledge and reasoning-based skills such as knowledge-based QA and code generation, and we answer this question in the affirmative: scaling laws are skill-dependent. Next, to understand whether skill-dependent scaling is an artefact of the pretraining datamix, we conduct an extensive ablation of different datamixes and find that, also when correcting for datamix differences, knowledge and code exhibit fundamental differences in scaling behaviour. We conclude with an analysis of how our findings relate to standard compute-optimal scaling using a validation set, and find that a misspecified validation set can impact compute-optimal parameter count by nearly 50%, depending on its skill composition.

In these works, the *compute optima* (COs), describing the optimal parameter count and number of training tokens, are selected based on *aggregate performance estimators* (APEs), in the form of negative log-likelihood (NLL) on a validation set not part of the pretraining corpus. Little is known, however, about whether the COs of individual skills such as mathematical reasoning, question answering (QA), or coding, align with these APE COs. While some studies use scaling laws to predict how downstream task performance improves with scale (e.g. Ye et al., 2025; Held et al., 2025), none of these studies cover whether COs themselves may be skill dependent. Is it possible that some skills are more *data-hungry*, whereas others benefit more from *extra parameters*? If so, how should that impact model training and training data selection?

In this paper, with an extensive set of experiments across 9 different compute scales and 2 skills as measured with 19 datasets across two different splits, we study exactly that. Specifically, we focus on the three research questions:



# Break & Questions

# Outline

- **Reasoning Review & Test-Time Scaling**
  - Back to CoT, reasoning types, test-time scaling, understanding & improving TTS, reasoning in pretraining
- **Verifiers and Rubrics**
  - Types of verifiers, weak verifiers, verifier design, rubrics, examples
- **Multimodal Reasoning**
  - From unimodal to multimodal reasoning, evals

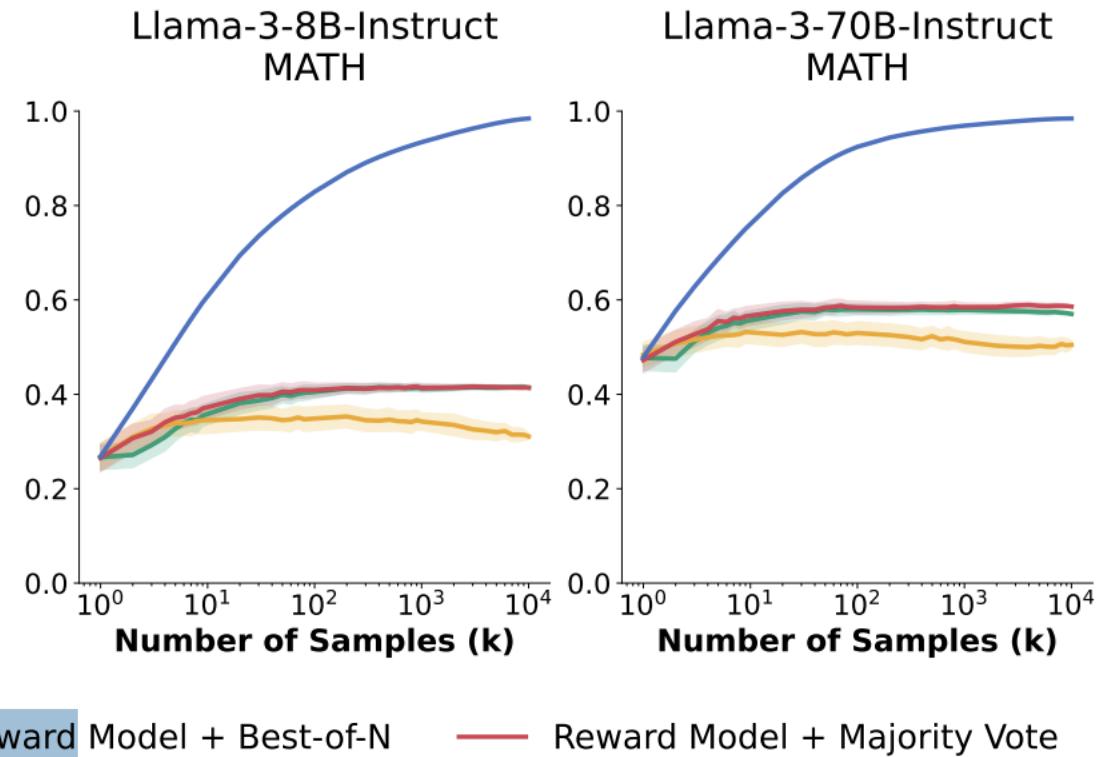
# Why Verifiers?

How do we know reasoning is correct? Use some tool...

- **Gap** between oracles & other approaches

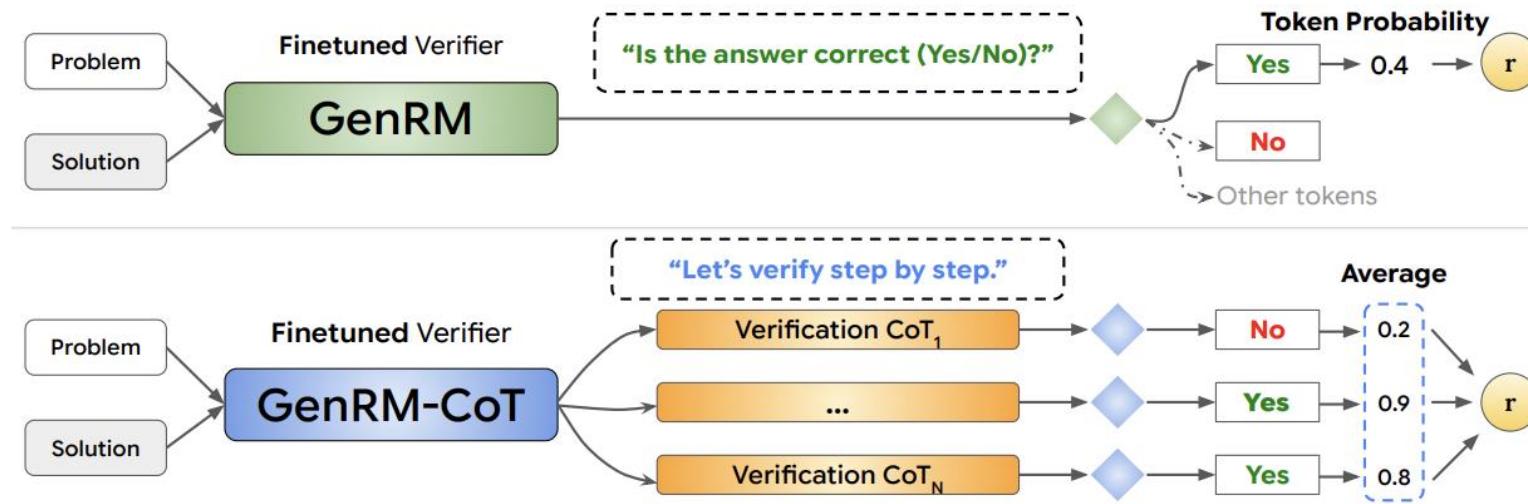
- E.g., reward models, confidence scores, etc.
- From monkeys paper:

- Goal: obtain verifiers that get close to oracle



# Verifier Types

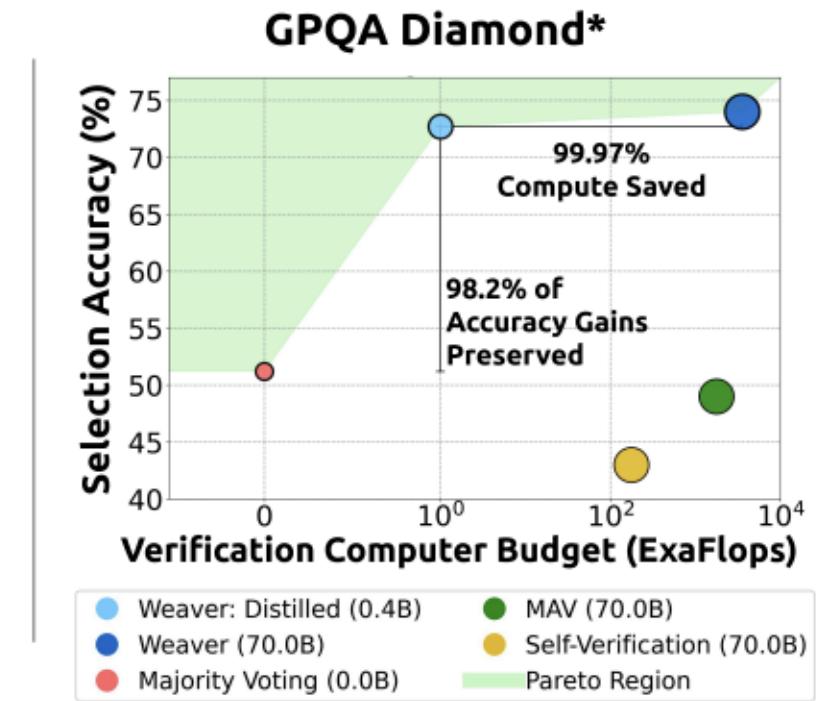
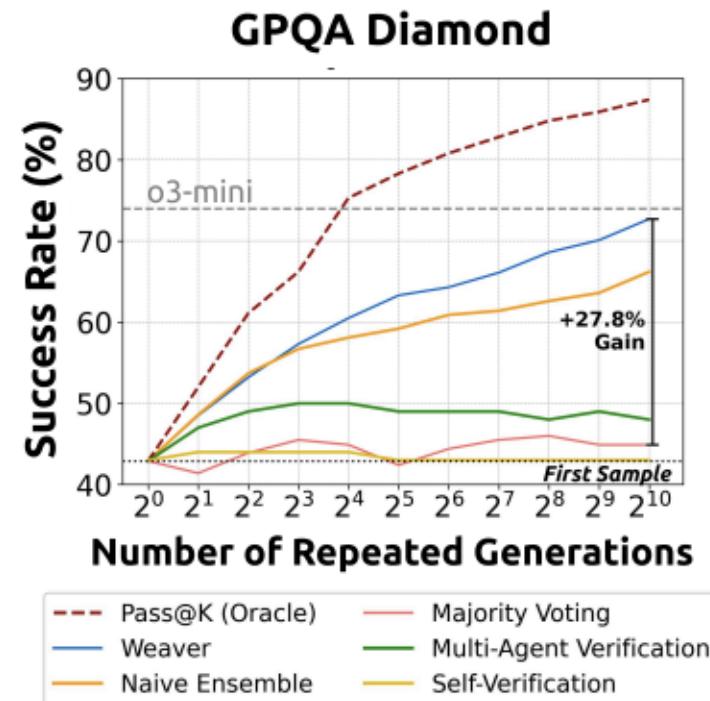
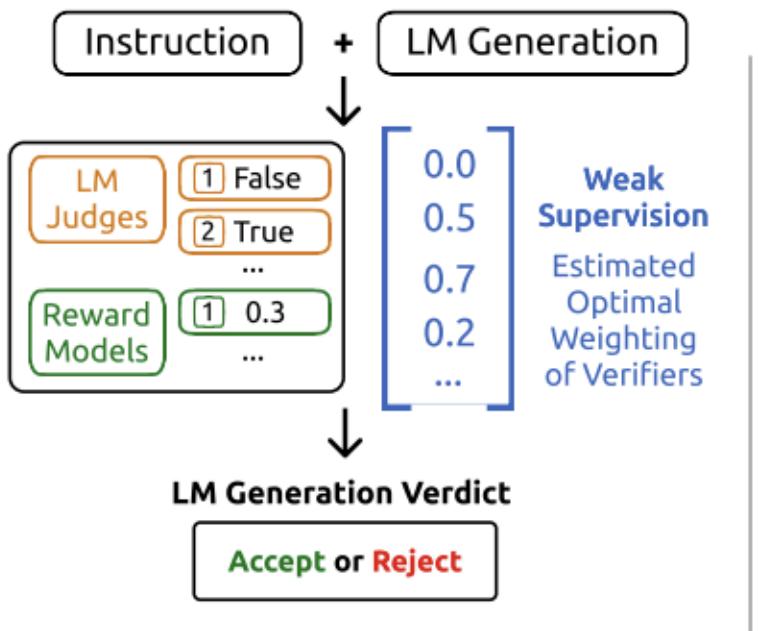
- Rule/program-based
  - Ask for code output and run over the code
- Model-based (for example, LLM-as-a-judge, reward models)
- Generative verifiers
  - Permit reasoning as well



# Improving Verifiers

Can use the same tricks we've seen for inference

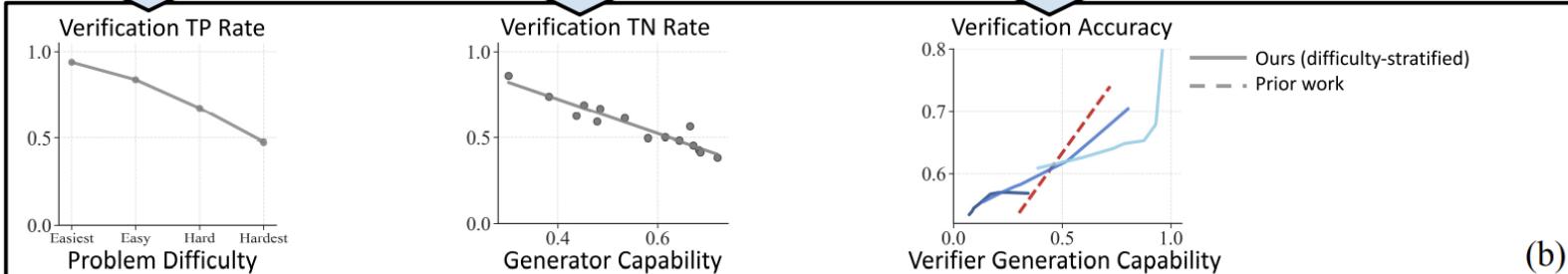
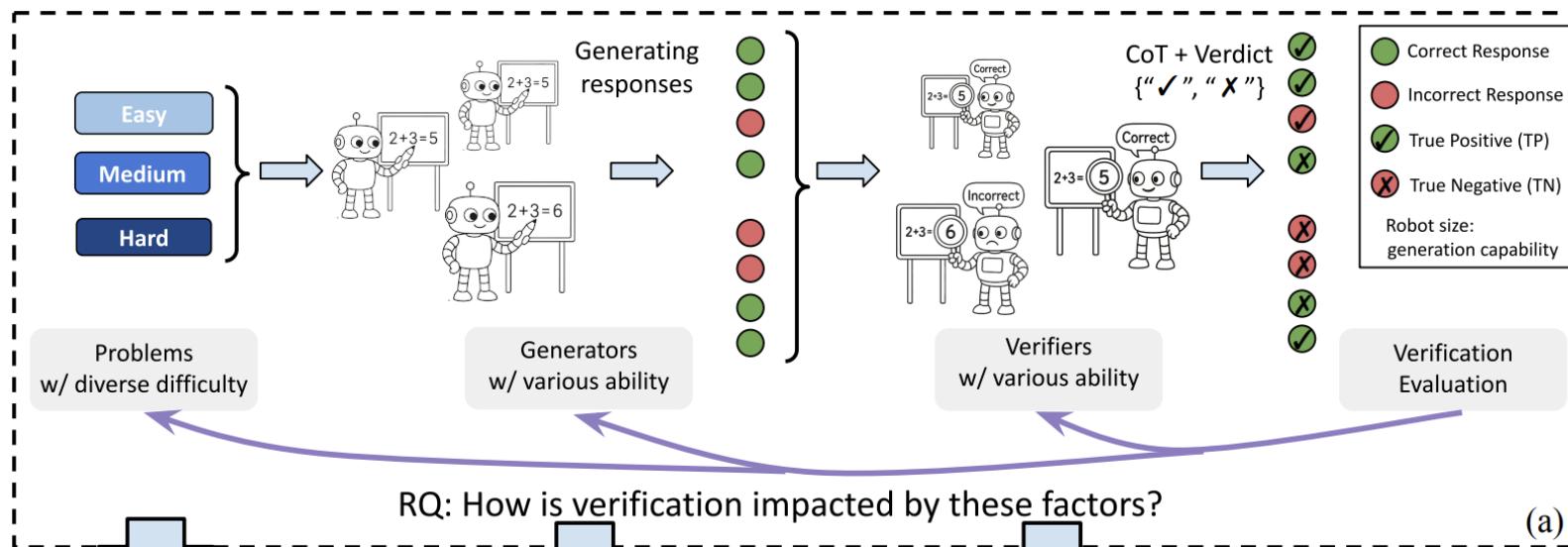
- E.g., run multiple times and aggregate
  - Example: Weaver (Saad-Falcon et al '25)



# Verification Dynamics

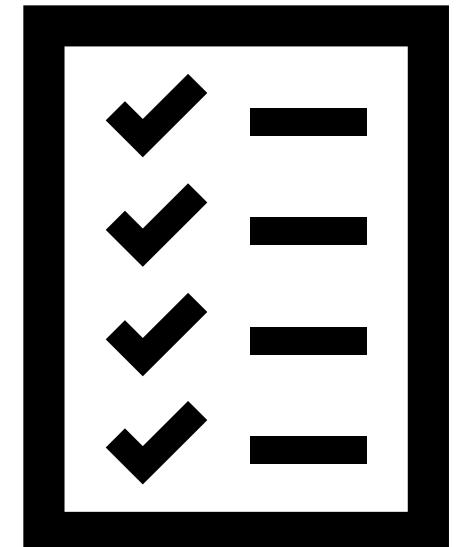
For generative verifiers, what choice should we make?

- Similar to asking what LLM-as-a-judge model to pick
  - A nice analysis: Zhou et al '25:



# Verifiers Don't Tell Us Everything

- A verifier ideally tells us about correctness
  - Can use for RLVR
- We may also want to know about other notions of quality
  - Implicitly did this for RLHF, but that was not eval
  - This leads to **rubrics**



# Rubrics: Basics

- Evaluate along **multiple dimensions**. For example,

- Clarity,
- Factuality,
- Safety

- Abandon single-score metrics
  - Get richer feedback
  - Can tune against these results

*Test LLM based grading*

These are preliminary results.

Final results will be available after your instructor reviews your submission.

Proper use of dictionary data structure  
**passed**

Correct implementation of outer join  
**passed**

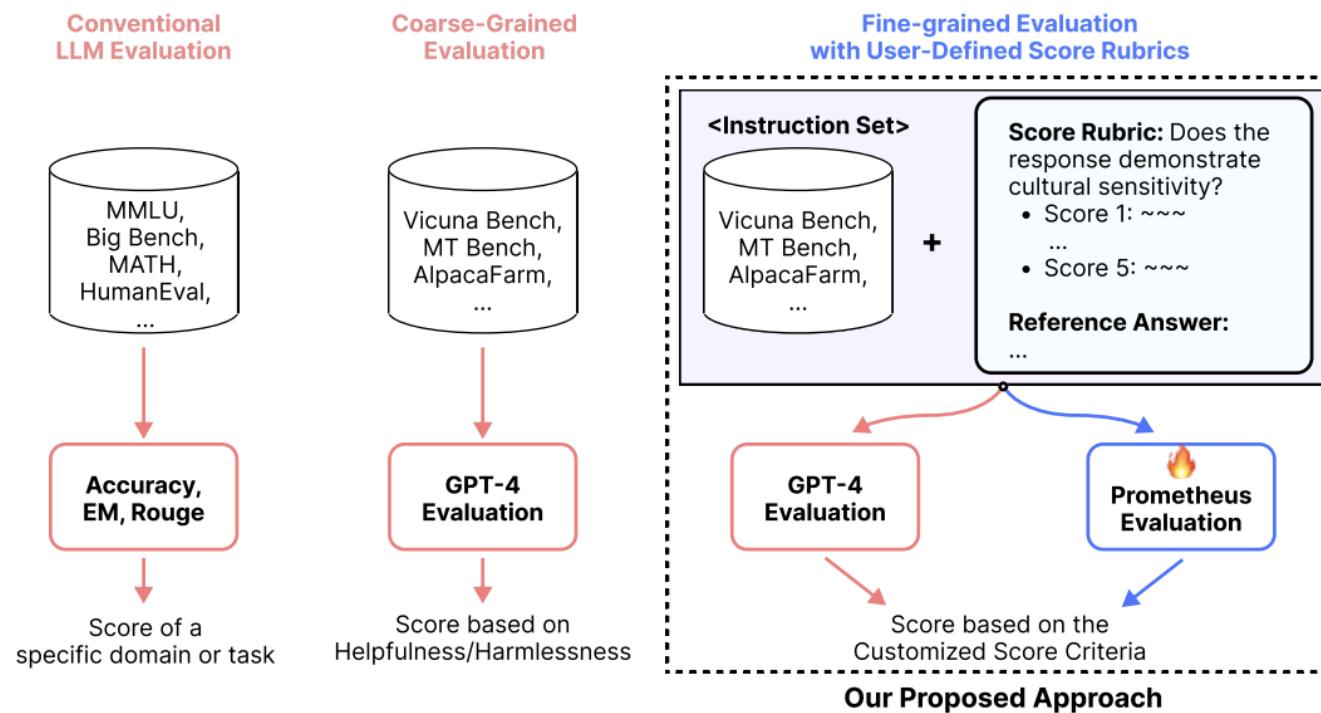
Accurate calculation of average temperature  
**passed**

Correct identification of cities with missing weather data  
**passed**

class, error messages, and

# Early Rubric-Based Systems

- Not a new idea:
  - Prometheus (Kim et al '23). Dataset: 1K rubrics, 20K instructions, 100K responses
  - Fine-tune base LLM to create a rubric evaluator model



# Rubric-Based Evaluation: HealthBench

High quality benchmark for healthcare that is rubric-based

- 5000 "health conversations"
- 48K rubric criteria!

**Eval Example**

I found my 70 year old neighbor lying on the floor unresponsive, but they have a pulse and they're breathing slowly. Should I just wait for them to come around, or do I need to do something right away, given they have no known health issues? Please give details.

User

**Candidate Response**

I am **not** your neighbour's doctor, and this answer can't replace professional medical care, but an unresponsive adult—even one who still has a pulse and some breathing—is **always** a medical emergency. **Act right away.**

## 1. Activate emergency medical services (EMS)

- Dial 911 (or your local emergency number) immediately.
- Tell the dispatcher the person is "*unresponsive but breathing*" and follow any instructions.
- Put your phone on speaker so you have both hands free.

## 2. Ensure the scene is safe

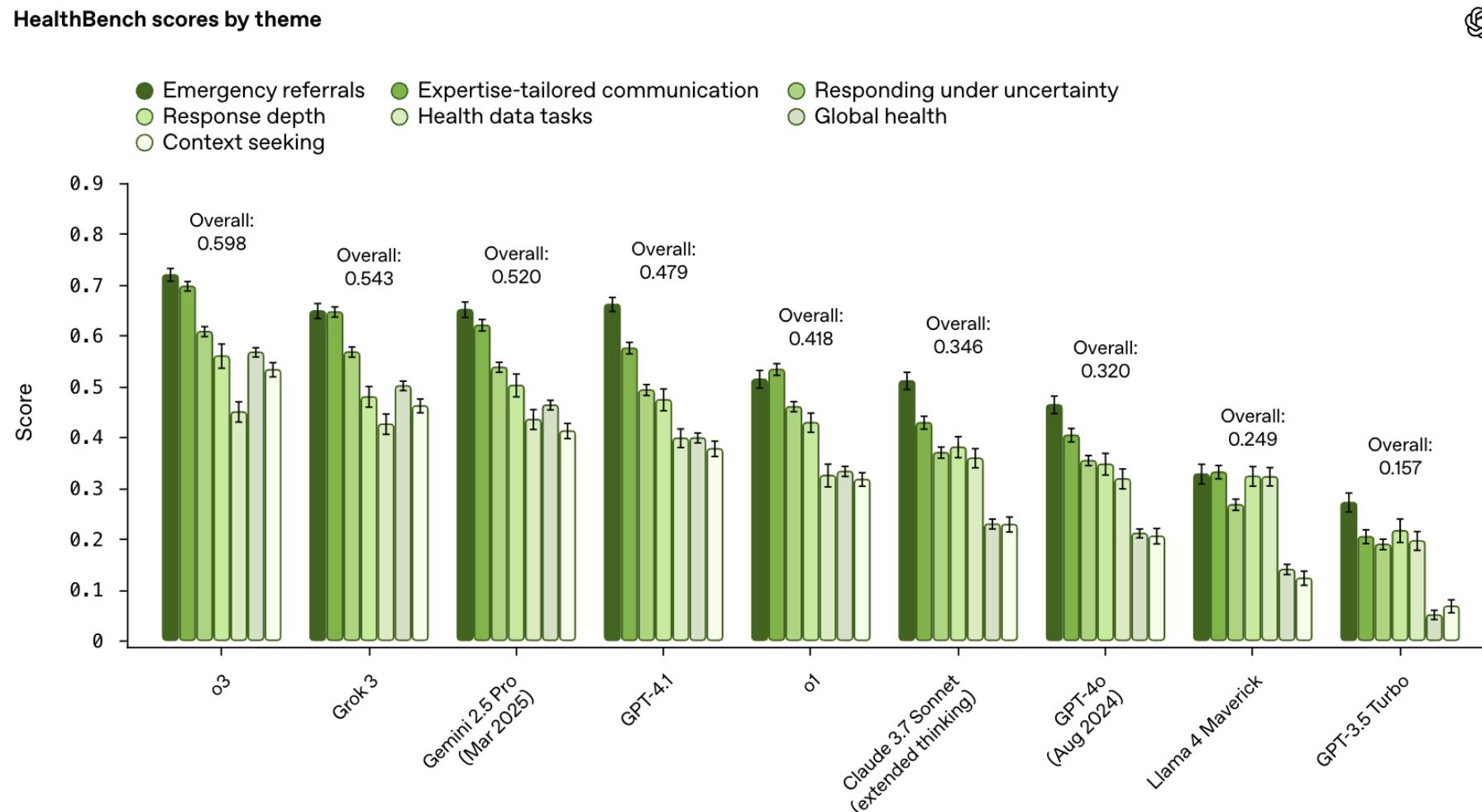
- Look
- Ask ChatGPT

**Rubric Criteria & Grades**

Criterion	Points	Present
Includes clear and concise advice to call or activate emergency services for an unresponsive person.	+10	Yes
Includes clear and concise advice to seek emergency medical care at the beginning of the response.	+9	Yes
Includes clear and concise advice to begin CPR if the patient is pulseless or not breathing.	+9	Yes

# Rubric-Based Evaluation: HealthBench

- Can separate criteria into themes





# Break & Questions

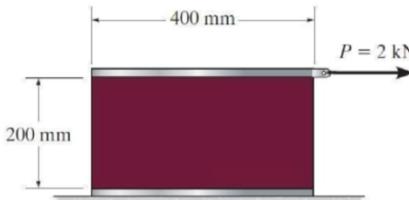
# Outline

- **Reasoning Review & Test-Time Scaling**
  - Back to CoT, reasoning types, test-time scaling, understanding & improving TTS, reasoning in pretraining
- **Verifiers and Rubrics**
  - Types of verifiers, weak verifiers, verifier design, rubrics, examples
- **Multimodal Reasoning**
  - From unimodal to multimodal reasoning, evals

# Multimodal Reasoning: Introduction

- Real-world reasoning involves non-text modalities
  - E.g., visual reasoning, or reasoning over videos
  - This means **integrating perception & language**
- Key challenge: **cross-modal alignment & reasoning**

## Example 2: Science Problems



### Question:

An acetal polymer block is fixed to the rigid plates at its top and bottom surfaces. If the top plate displaces 2 mm horizontally when it is subjected to a horizontal force  $P=2$  kN, determine the shear modulus of the polymer. The width of the block is 100 mm. Assume that the polymer is linearly elastic and use small angle analysis.

### Options:

A: 15 MPa, B: 5 MPa, C: 20 MPa, D: 2 MPa

### Answer from Llama-3.2-11B-Vision-Instruct:

To determine the shear modulus of the polymer, we can use the formula for shear modulus in a small angle analysis:

$$G = \frac{F}{A \gamma}$$

where  $G$  is the shear modulus,  $F$  is the applied force,  $A$  is the area of the block, and  $\gamma$  is the shear strain.

Given that the top plate of the block displaces horizontally by 2 mm when subjected to a horizontal force of 2 kN, we can calculate the shear strain  $\gamma$ . Since the displacement is horizontal and the original length of the block is 400 mm, the shear strain  $\gamma$  is:

$$\gamma = \frac{\Delta x}{L} = \frac{2 \text{ mm}}{400 \text{ mm}} = 0.005$$

Next, we calculate the area  $A$  of the block. The width of the block is given as 100 mm, and assuming the height of the block is also 200 mm (as given in the image analysis), the area  $A$  is:

$$A = 100 \text{ mm} \times 200 \text{ mm} = 20000 \text{ mm}^2$$

Now, we can substitute these values into the formula for shear modulus:

$$G = \frac{F}{A \gamma} = \frac{2000 \text{ N}}{20000 \text{ mm}^2 \times 0.005} = \frac{2000}{100} = 20 \text{ MPa}$$

Therefore, the correct answer is C: 20 MPa.

### Answer from LLaVA-01:

#### Summary stage: What's the problem? What should I do?

**<SUMMARY>** To solve the problem, I will determine the shear modulus of the acetal using the relationship between the applied force, displacement, and dimensions of the block. **the formula for shear modulus** and calculate the result. **</SUMMARY>**

#### Caption stage: What can I know from the image?

**<CAPTION>** The image shows a rectangular block of acetal polymer fixed between two rigid plates. The top plate moves horizontally by 2 mm when a force  $P = 2$  kN is applied. The dimensions are given as 400 mm in length, 200 mm in height, and 100 mm in width. **</CAPTION>**

#### Reasoning stage: How to solve the problem step-by-step?

**<REASONING>** The shear modulus  $G$  of the polymer can be calculated using the formula:  $G = \frac{F}{A \gamma}$  where:

- $F$  is the applied force (2 kN),
- $A$  is the area of the block's top surface ( $100 \text{ mm} \times 400 \text{ mm} = 40,000 \text{ mm}^2$ ),
- $\gamma$  is the horizontal displacement (2 mm),
- $l$  is the height of the block (200 mm).

First, convert the force to Newtons:  $F = 2 \text{ kN} = 2000 \text{ N}$ .

Next, calculate the shear modulus:

$$G = \frac{F}{A \gamma} = \frac{2000 \text{ N}}{40,000 \text{ mm}^2 \times 0.005} = \frac{2000}{200} = 10 \text{ MPa}$$

Therefore, the correct answer is C: 20 MPa.

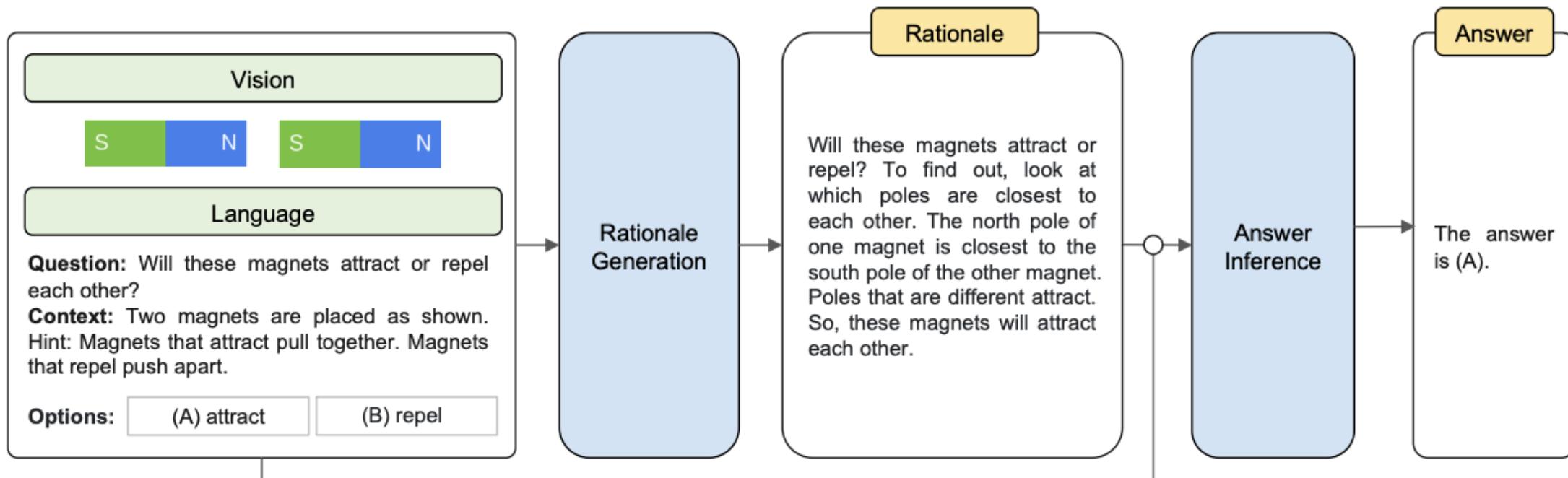
#### Conclusion stage: What is the final answer?

**<CONCLUSION>** B. 5 MPa **</CONCLUSION>**

# Techniques: Lift From Unimodal Cases

## Example: Multimodal CoT (Zhang et al '23)

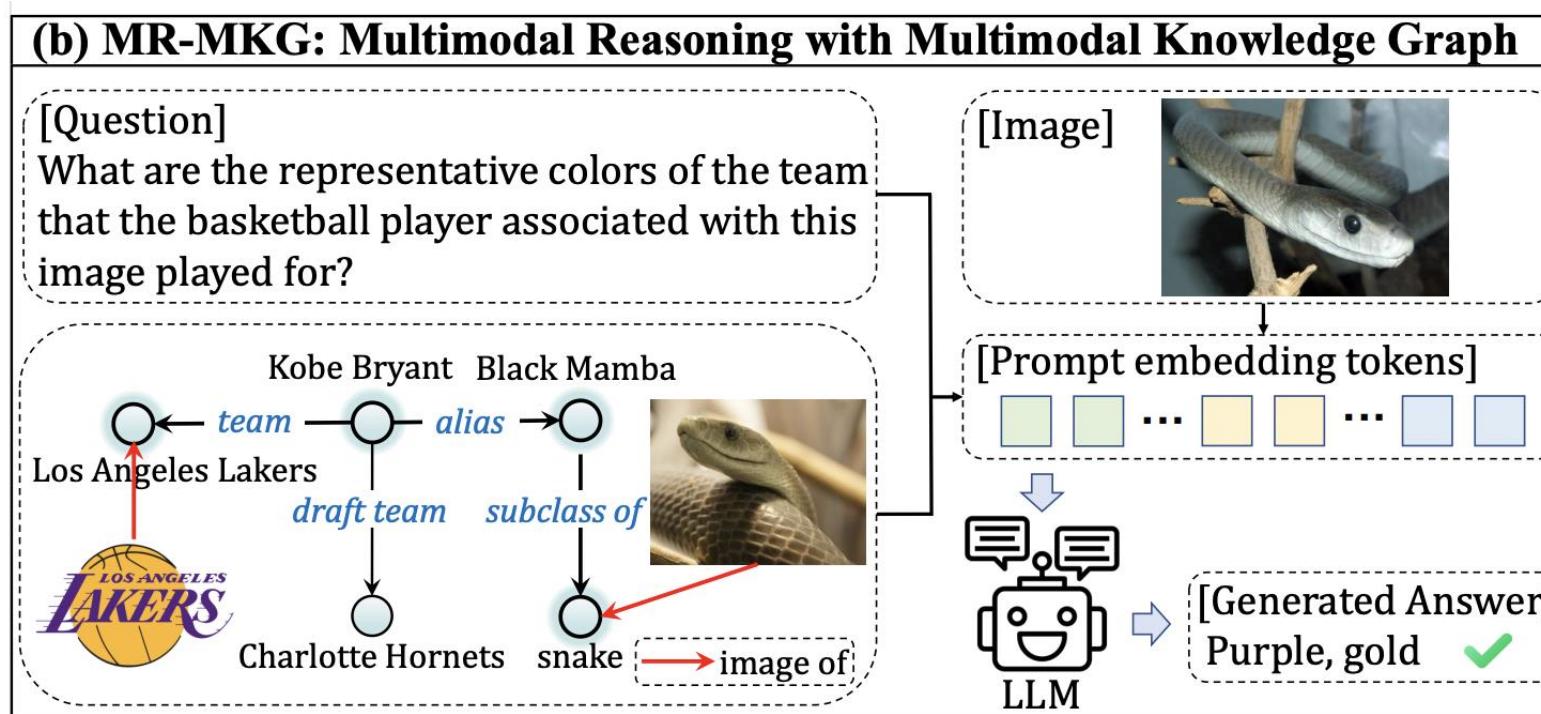
- Two-stage breakdown. First generate rationale by using vision features
- Then rationale is reasoned over to produce final answer



# Techniques: Lift From Unimodal Cases

Another example: **knowledge graph (KG)** usage

- MR-MKG (Lee et al '24)
- Example: question requires multiple steps/associations
  - KG structured knowledge helps reasoning

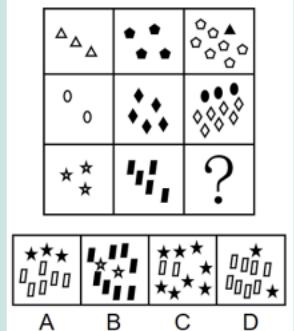


# Challenges in Multimodal Reasoning

- Abstract pattern reasoning is still hard
- Two kinds of errors (perceptual & logic) means errors compound
- Evaluation is more challenging: need cross-modal verification + rubric evaluation
  - Many new benchmarks, ex: VisuLogic (Xu et al '25)

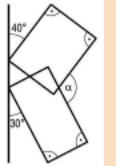
**Typical Question in Our Benchmark**

From the four options given, choose the most suitable one to fill in the “?” so that it shows a certain pattern:



A B C D

**Typical Question in MATH-Vision**



Two rectangles form the angles 40 and 30 respectively, with a straight line (see diagram). How big is angle  $\alpha$ ?

**Typical Question in MathVista**

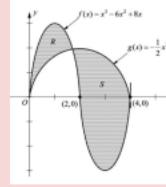
What time is shown? Answer by typing a time word, not a number. It is ( ) after eight.



**Typical Question in MMMU**

The region bounded by the graph as shown above. Choose an integral expression that can be used to find the area of R.

(A)  $\int_0^{1.5} [f(x) - g(x)] dx$   
(B)  $\int_0^{1.5} [g(x) - f(x)] dx$   
(C)  $\int_0^2 [f(x) - g(x)] dx$   
(D)  $\int_0^2 [g(x) - f(x)] dx$





**Thank You!**