

CS 668 Applied Large Language Models  
Spring Semester, 2026  
Doc 24 Reasoning Model  
Apr 23, 2026

Copyright ©, All rights reserved. 2026 SDSU & Roger Whitney, 5500  
Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent ([http://www.opencontent.org/  
openpub/](http://www.opencontent.org/openpub/)) license defines the copyright on this document.

# Exam

Build an Agent to conduct an oral exam

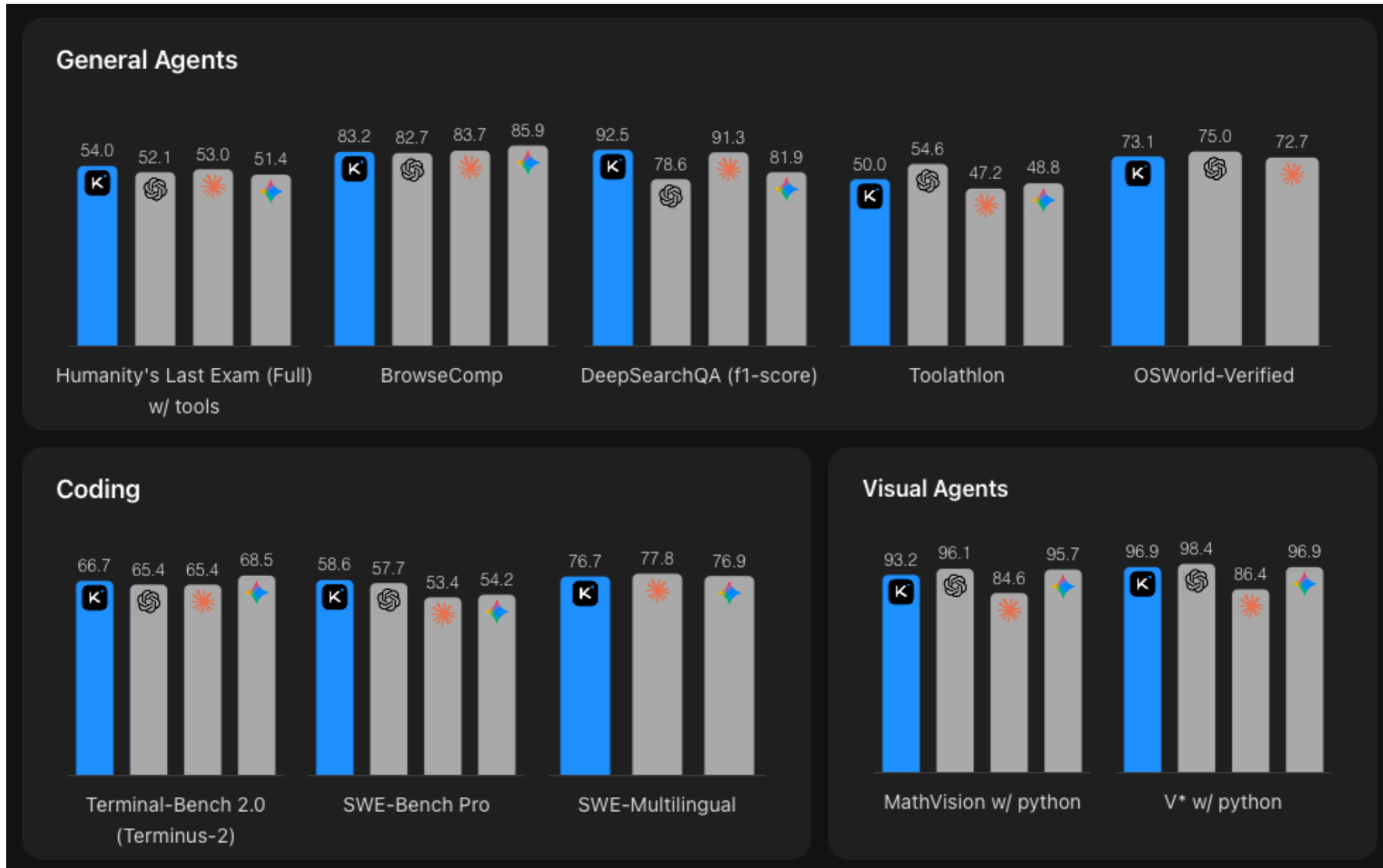
<https://elevenlabs.io/app/home>

Google ended the free Colab Pro for students

Anthropic temporarily removed Claude Code from \$20 level

# Moonshot Kimi K2.6

Released this week

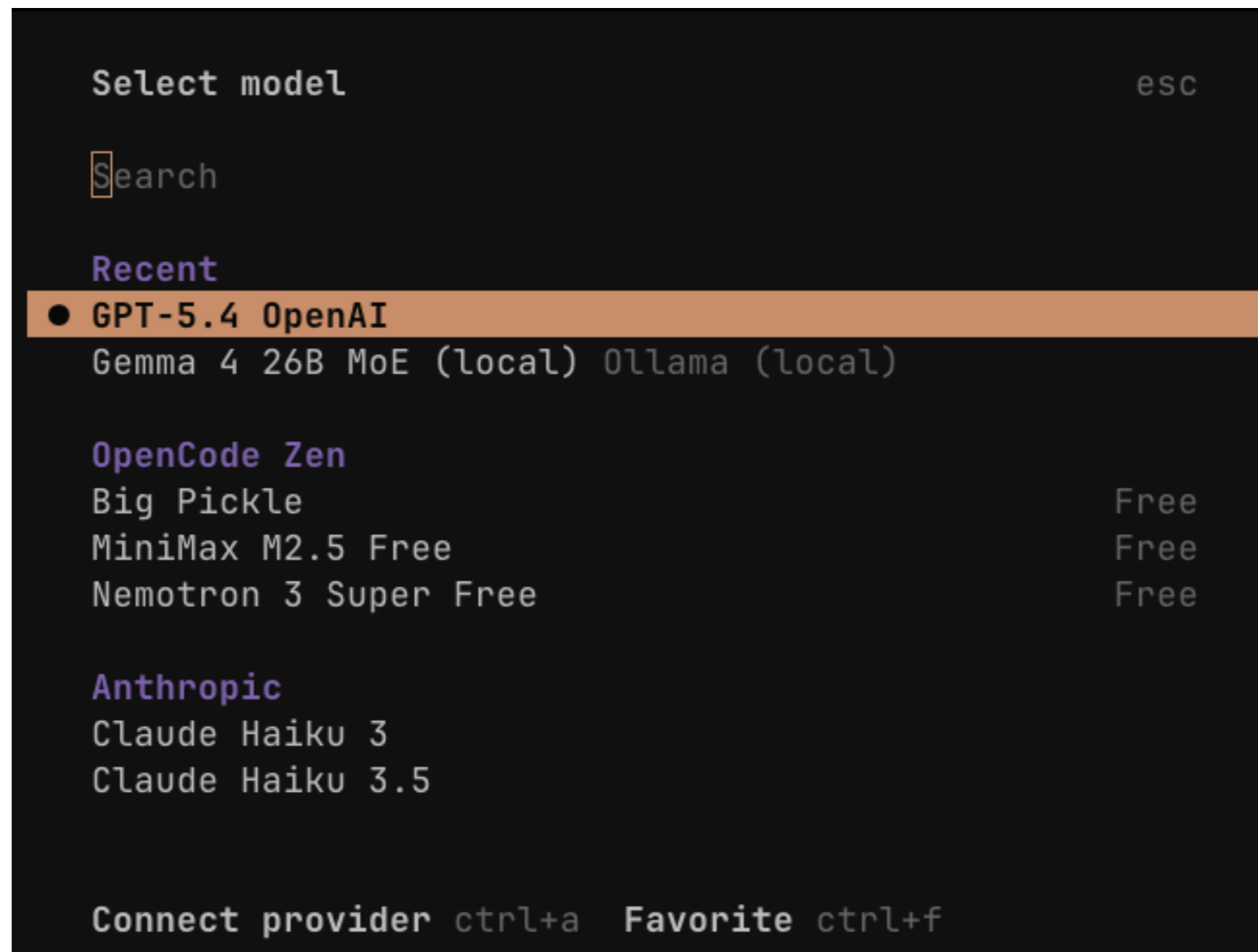


# OpenCode

<https://opencode.ai>

The open source AI coding agent

Free models included or connect any model from any provider,  
Claude, GPT, Gemini and more.



# OpenCode - Caveman Mode

<https://github.com/JuliusBrussee/caveman>

```
Explain the files in this directory
```

```
Thinking: We need to list files in current directory. The user asks: "Explain the files in this directory". We should read directory content and explain each file. Use bash ls or glob. We'll first list files. Then maybe read each file to explain. But we need concise. Probably just list and give brief description. Let's run ls -la.
```

```
That's 4 lines.
```

```
However, note that the user might want more details. But the instruction says to be concise.
```

```
Let's output that.
```

```
BandwidthApp.swift: Main app entry point  
ContentView.swift: Primary user interface  
NetworkMonitor.swift: Network status monitoring  
SparklineView.swift: Custom sparkline graph view
```

# AI Resistance is Growing

<https://stephvee.ca/blog/artificial-intelligence/ai-resistance-is-growing/>

r/PoisonFountain,

Encourage people to feed huge quantities of trash data (poison) to all of the web crawlers out there that are scraping our work for AI training sets

Aim: serve one terabyte of poison per day

# Lucebox

<https://github.com/Luce-Org/lucebox-hub>

Combine models kernels into one kernel to improve model's performance

DFlash DDtree Qwen3.5 27B GGUF on RTX 3090

Up to 207 tok/s in the demo (207.6 tok/s DFlash vs 38.0 tok/s AR, 5.46×)

129.5 tok/s mean on the HumanEval 10-prompt bench

3.43× faster than autoregressive (+15% over chain speculative decoding)

2.8× faster than SGLang AWQ on the same hardware

128K context in 24 GB (134.78 tok/s at ctx=131072)

# Outplaying elite table tennis players with an autonomous robot

<https://doi.org/10.1038/s41586-026-10338-5>

Nature Online April 22, 2026

Sony



# Beyond TurboQuant

Sequential KV Cache Compression via Probabilistic Language Tries:  
Beyond the Per-Vector Shannon Limit

April 10, 2026

Gregory Magarshak, New York College

$\approx 914\times$  better compression over TurboQuant

# 1-bit Bonsai: The First Commercially Viable 1-bit LLM

<https://prismml.com/news/bonsai-8b>

PrismML 1-bit Bonsai      Caltech startup

Model	Size	Average	MMLU Redux	MuSR	GSM8K	Human Eval+	IFEval	BFCLv3
Qwen 3 8B	16.38 GB	<b>79.3</b>	83	55	93	82.3	81.5	81
RNJ 8B	16.63 GB	<b>73.1</b>	75.5	50.4	93.7	84.2	73.8	61.1
Ministral3 8B	16.04 GB	<b>71.0</b>	68.9	53.8	87.9	72.6	67.4	75.4
Olmo 3 7B	14.60 GB	<b>70.9</b>	72	56.1	92.5	79.3	87.1	38.4
<b>1-bit Bonsai 8B</b>	<b>1.15 GB</b>	<b>70.5</b>	65.7	50	88	73.8	79.8	65.7
LFM2 8B	16.68 GB	<b>69.6</b>	72.7	49.5	90.1	61	82.2	62.0
Llama 3.1 8B	16.06 GB	<b>67.1</b>	72.9	51.3	87	63.4	76.4	51.5
GLM 4 9B	18.80 GB	<b>65.7</b>	81.9	53.2	89.4	78.7	69.3	21.9
Hermes 3 8B	16.06 GB	<b>65.4</b>	67.4	52.2	82.9	51.2	69.3	69.6
Trinity Nano 6B	12.24 GB	<b>61.2</b>	66.8	52.6	81.1	54	50	62.5
Marin 8B	16.06 GB	<b>56.6</b>	64.8	42.6	86.1	55.9	63	27.9
DeepSeek R1 Qwen 7B	15.23 GB	<b>55.0</b>	62.5	29.1	92.7	81.7	48.8	15.4

Device	Tokens/Second
M4 Max Pro	131
RTX 4090	368
iPhone 17 Pro Max	44

# Embarrassingly Simple Self-Distillation Improves Code Generation

Apr 1, 2026

Apple, Zhang, Bai, Zheng, Jaitly, Collobert, Zhang

<https://arxiv.org/pdf/2604.01193>

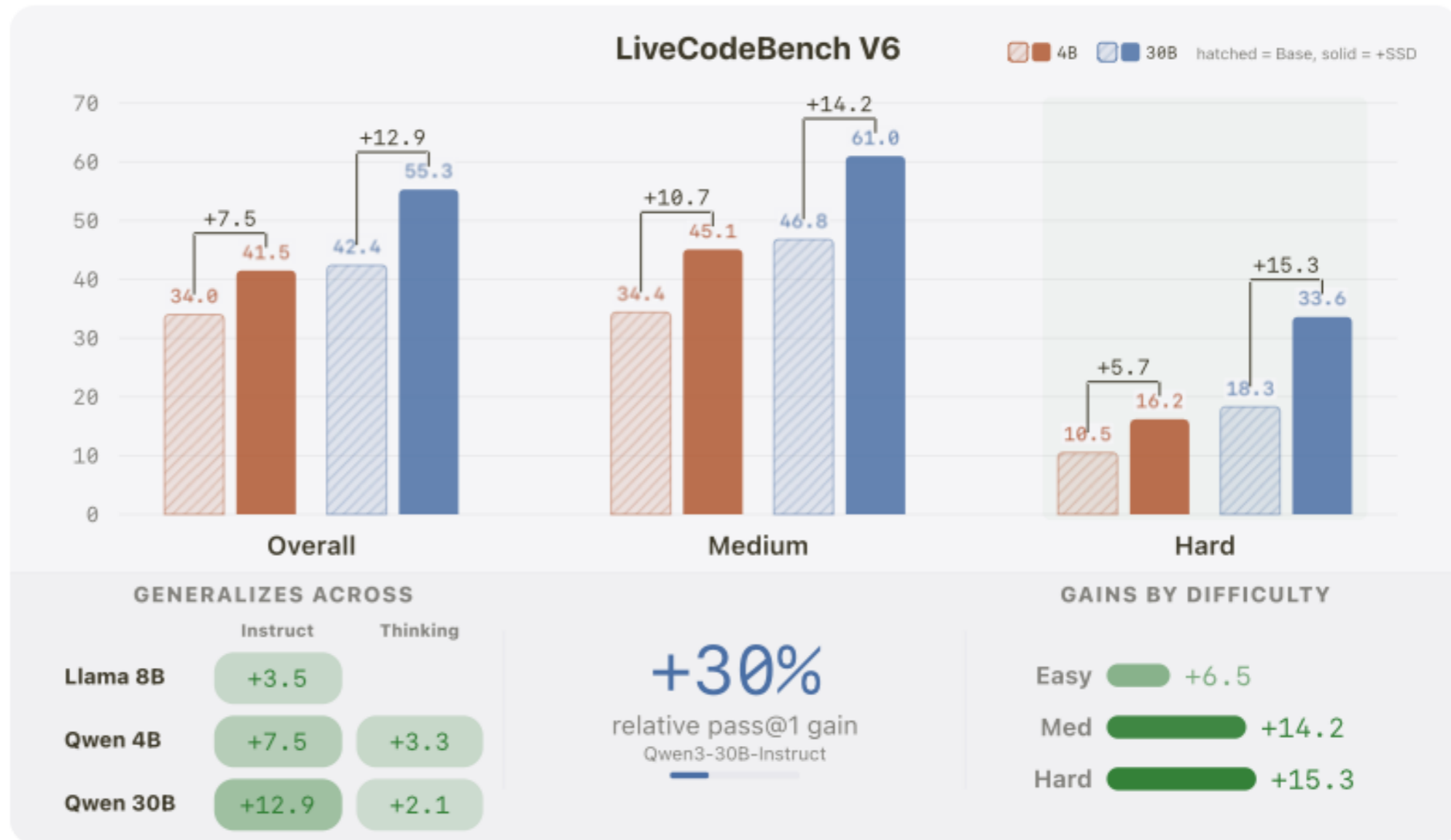
**Sample**  
T<sub>train</sub> ≠ 1

**Fine-tune**  
SFT on own outputs

**Decode**  
Evaluate at T<sub>eval</sub>

× No RL      × No Verifier

× No Teacher      × No Execution



# Embarrassingly Simple Self-Distillation Improves Code Generation

SSD at high level

## 1 Sample

- Freeze model

- Coding prompts

- One completion per prompt above normal temperature, top-k, top-p truncation

- Keep everything, no filter, don't correct answers

## 2 Fine-tune

- Standard supervised fine-tuning on those raw outputs

- Cross-entropy loss

## 3 Decode

- Evaluate the fine-tuned model at a different, tuned temperature with truncation

<https://medium.com/@michael.hannecke/your-llm-can-improve-itself-no-teacher-no-verifier-no-rl-required-5d41f3f4a6b4>

# Embarrassingly Simple Self-Distillation Improves Code Generation

## Basic Pattern

Sample at a higher temperature than normal

Truncate

Train

Decode at a lower temperature

Exact temperatures depend on the model

Qwen3-4B-Instruct

$T_{\text{train}}=1.6$  with top-k=20, top-p=0.8

$T_{\text{eval}}=1.1$

<https://medium.com/@michael.hannecke/your-llm-can-improve-itself-no-teacher-no-verifier-no-rl-required-5d41f3f4a6b4>

# Embarrassingly Simple Self-Distillation Improves Code Generation

The Precision-Exploration Conflict Hypothesis

Lock

- Distribution is sharply peaked

- Demand precision:

  - Commit to the dominant token and suppress the tail

Fork

- Distribution is spread across multiple plausible tokens

- Lead to meaningfully different downstream continuations

- Demand exploration:

  - Spread mass across viable alternatives to avoid missing the good paths

<https://medium.com/@michael.hannecke/your-llm-can-improve-itself-no-teacher-no-verifier-no-rl-required-5d41f3f4a6b4>

# Embarrassingly Simple Self-Distillation Improves Code Generation

The Precision-Exploration Conflict Hypothesis

Low temperature

Sharpens a peak at a lock, starves a fork

High temperature

Destabilizes a lock, improves a fork

Using one temperature forces a compromise

# Embarrassingly Simple Self-Distillation Improves Code Generation

$T_{train}$  controls how strongly SSD reshapes the model distribution

$T_{eval}$  controls how aggressively decoding exploits that reshaped distribution

$$T_{eff} = T_{train} * T_{eval}$$

$T_{eff}$  governs performance

# Comprehension Debt: The Hidden Cost of AI-Generated Code

<https://addyosmani.com/blog/comprehension-debt/>

Technical Debt  
In the Code

Comprehension Debt  
In your Head

Working code vs Good code

# Reasoning

Thinking ...

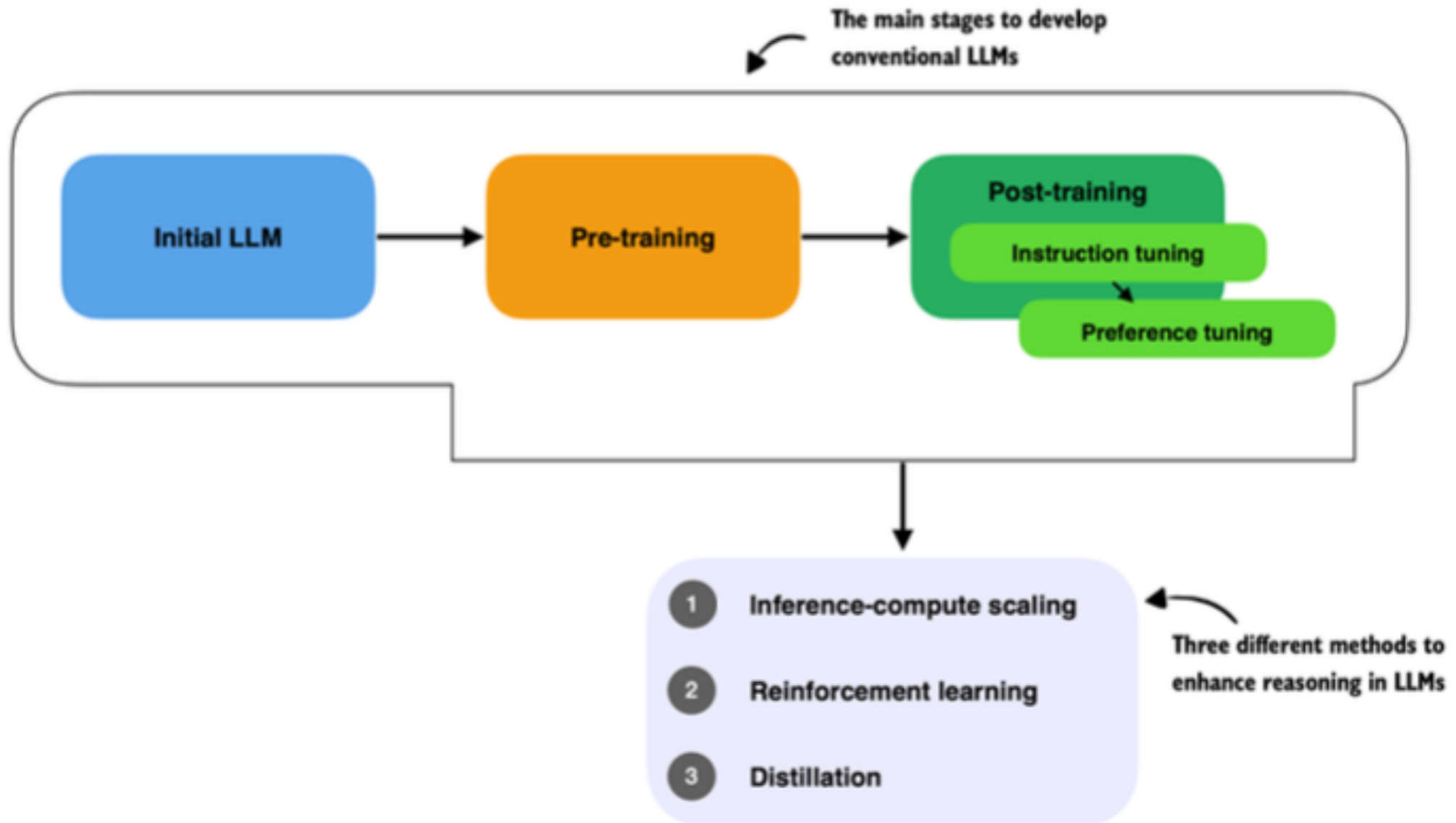
This style of intermediate-step generation is often referred to as chain-of-thought (CoT).

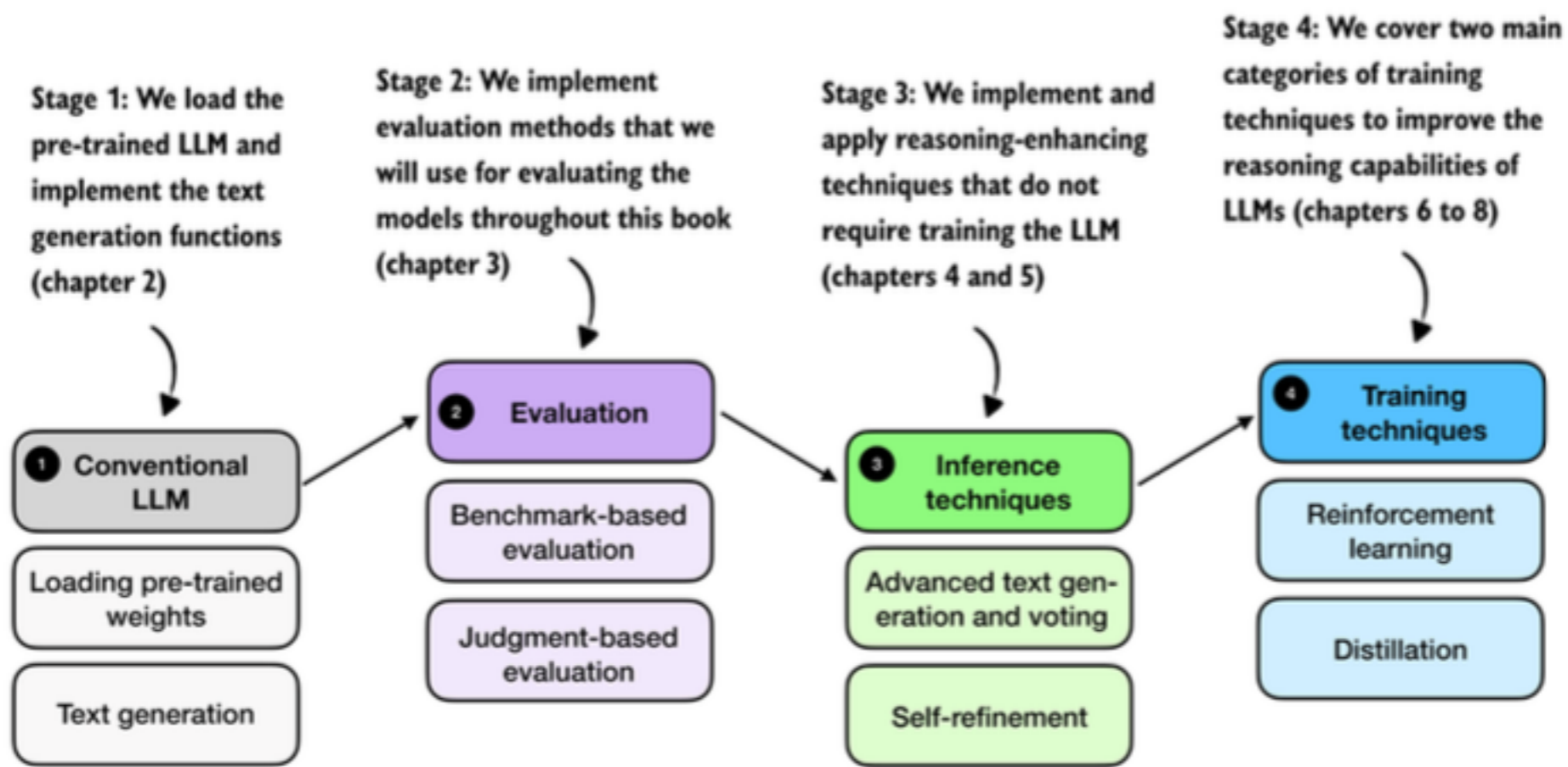
Researchers and engineers often say that the model "thinks" through the problem step by step, meaning that it makes its intermediate reasoning process explicit and easier to follow.

This does not imply that LLMs actually reason or think in the same way humans do.

Build a Reasoning Model (From Scratch), Draft 8, Sebastian Raschka, 4-8-26

# How to Train a Reasoning Model?

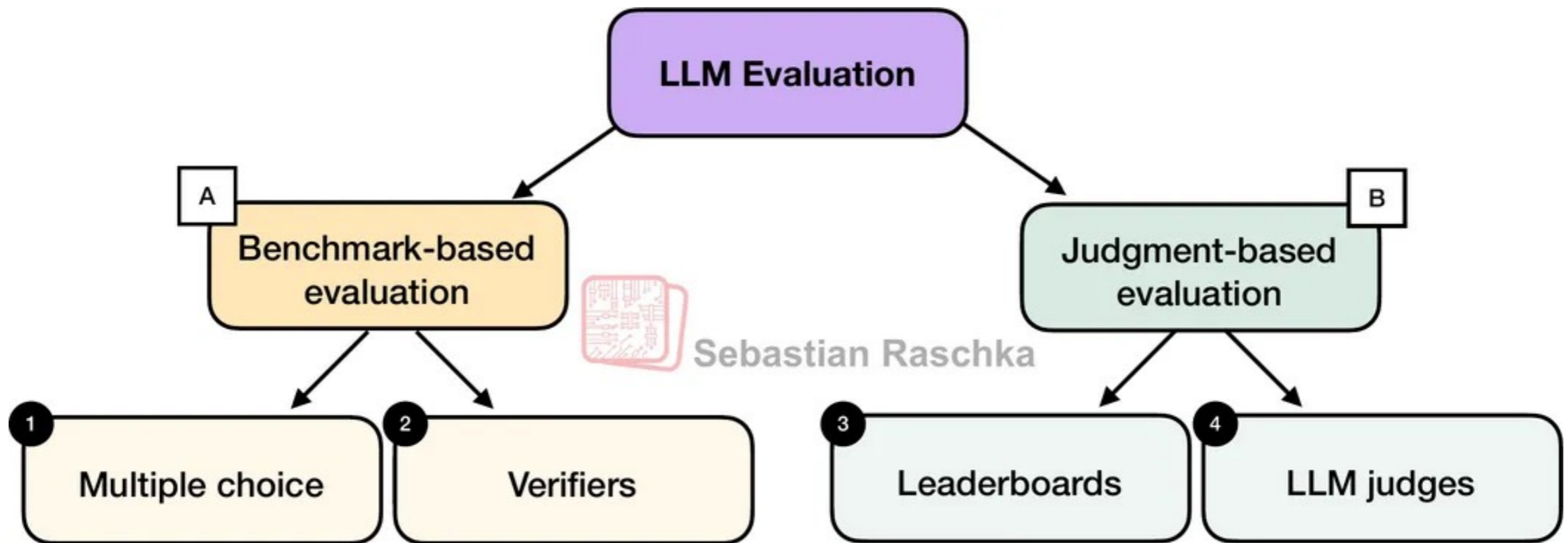




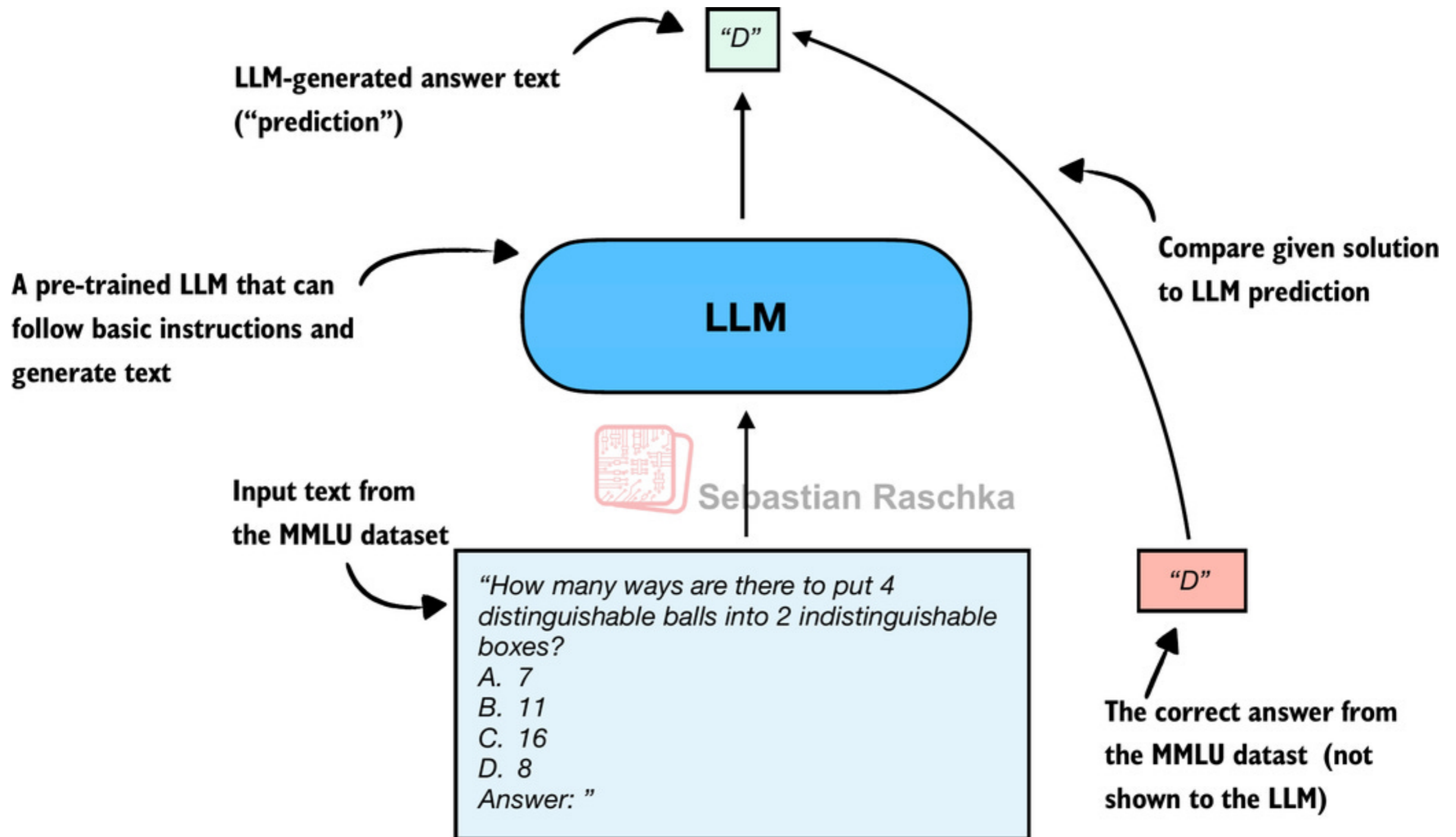
<b>Mode</b>	<b>Hardware</b>	<b>Tokens/sec</b>	<b>GPU memory</b>
Regular	Mac Mini M4 CPU	5	-
Regular compiled	Mac Mini M4 CPU	6	-
KV cache	Mac Mini M4 CPU	28	-
KV cache compiled	Mac Mini M4 CPU	68	-
Regular	Mac Mini M4 GPU	27	-
Regular compiled	Mac Mini M4 GPU	43	-
KV cache	Mac Mini M4 GPU	41	-
KV cache compiled	Mac Mini M4 GPU	71	-
Regular	NVIDIA H100 GPU	51	1.55 GB
Regular compiled	NVIDIA H100 GPU	164	1.81 GB
KV cache	NVIDIA H100 GPU	48	1.52 GB
KV cache compiled	NVIDIA H100 GPU	141	1.81 GB

Compiled - torch.compile

# Evaluaton Methods



# Evaluating answer-choice accuracy



# Measuring Massive Multitask Language Understanding - MMLU

<https://huggingface.co/datasets/cais/mmlu>

57 subjects

16,000 questions

Find the sum of the given polynomials in the given polynomial ring. $f(x) = 4x - 5$ , $g(x) = 2x^2 - 4x + 2$ in $\mathbb{Z}_8[x]$ .	"2x <sup>2</sup> + 5", "6x <sup>2</sup> + 4x + 6", "0", "x <sup>2</sup> + 1"	A
Mergesort works by splitting a list of n numbers in half, sorting each half recursively, and merging the two halves. Which of the following data structures will allow mergesort to work in $O(n \log n)$ time? I. A singly linked list II. A doubly linked list III. An array	"None", "III only", "I and II only", "I, II, and III"	D
Naturalists who concentrated on natural elements and processes are associated with which of the following?	"Daoism", "Yin-Yang School", "Mengzi", "Humanistic Buddhism"	B

A limitation of multiple-choice benchmarks like MMLU is that they only measure an LLM's ability to select from predefined options and thus is not very useful for evaluating reasoning capabilities besides checking if and how much knowledge the model has forgotten compared to the base model.

# Leader Boards

https://arena.ai/leaderboard/text

Overview Chat Code Image Video

View as  
Ranking Pareto

Rank by  
Models Labs

Categories (27)  
Overall Expert Occupational Math Instruction Following Multi-Turn

## Text Arena Overall

View overall rankings across various AI models in text-to-text tasks across math, coding, creative writing, and other open-ended domains.

Apr 19, 2026  5,873,554 votes  342 models

Hide Filters

Rank	Rank Spread	Model	Score	Votes	Price
1	1 - 6	claude-opus-4-7-thinking Anthropic · Proprietary	1504 ±9	3,898	
2	1 - 6	claude-opus-4-6-thinking Anthropic · Proprietary	1502 ±5	18,888	
3	1 - 7	claude-opus-4-7 Anthropic · Proprietary	1497 ±9	4,646	
4	1 - 6	claude-opus-4-6 Anthropic · Proprietary	1496 ±5	20,158	
5	1 - 10	muse-spark Meta · Proprietary	1493 ±8 <span>Preliminary</span>	5,877	

View as

Ranking

Pareto

Rank by

Models

Labs

Categories (27)

Overall

Expert [Learn more](#)

Occupational

Math

Instruction Following

Multi-Turn

Creative Writing

Coding

## Text Arena Coding

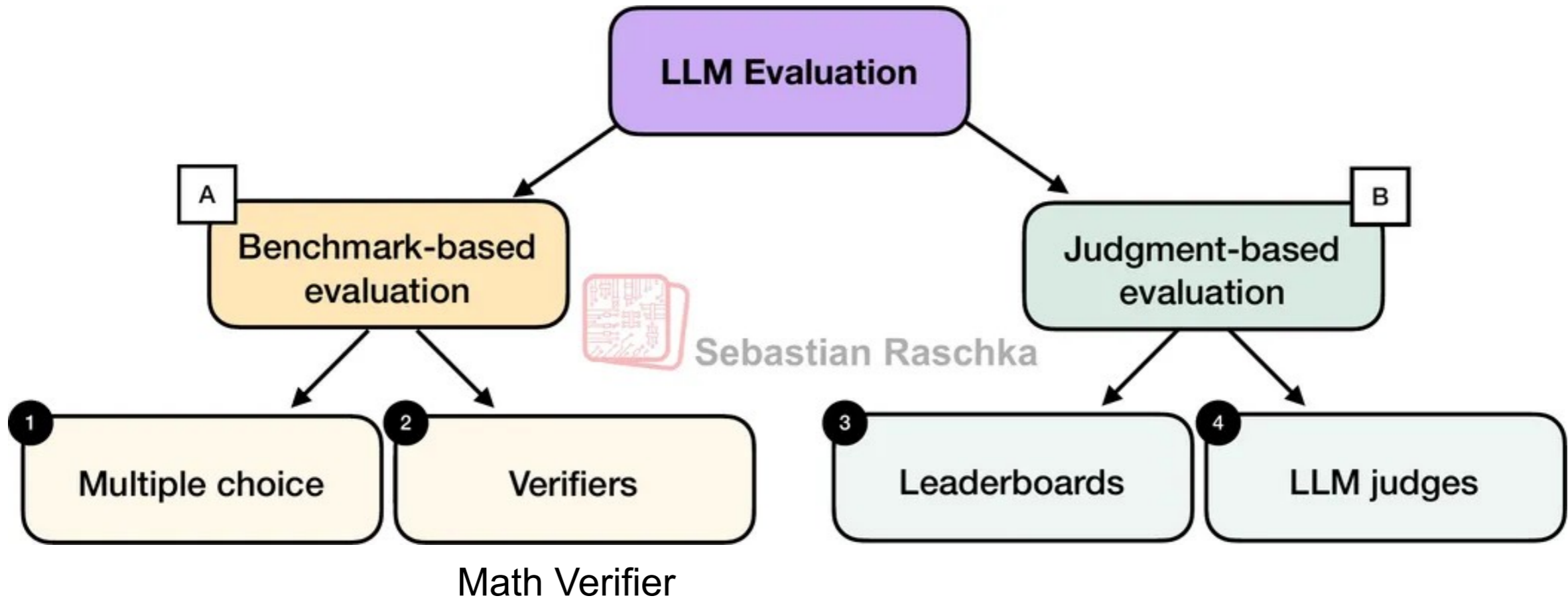
View overall rankings across various AI models in text-to-text tasks across math, coding, creative writing, and other open-ended domains.

Apr 19, 2026  1,061,233 votes  337 models

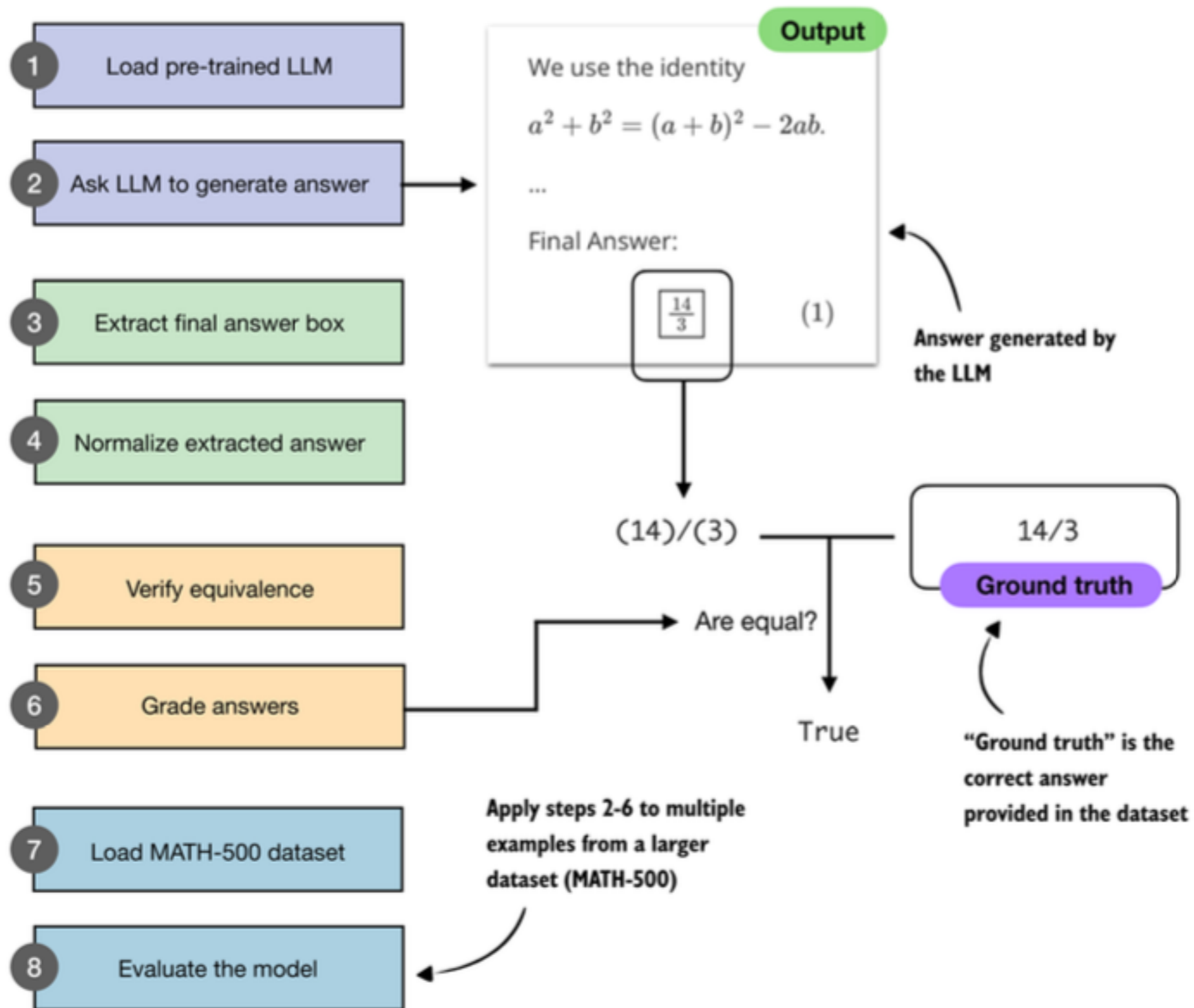
Hide Filters

Rank	Rank Spread	Model	Score	Votes	Price \$/M
1	1 - 4	<b>claude-opus-4-7-thinking</b> Anthropic · Proprietary	1571 ±19	933	\$5 / \$25
2	1 - 6	<b>claude-opus-4-7</b> Anthropic · Proprietary	1560 ±16	1,233	\$5 / \$25
3	1 - 6	<b>claude-opus-4-6-thinking</b> Anthropic · Proprietary	1554 ±10	4,193	\$5 / \$25
4	1 - 6	<b>claude-opus-4-6</b> Anthropic · Proprietary	1549 ±9	4,821	\$5 / \$25
5	2 - 19	<b>gpt-5.4-high</b> OpenAI · Proprietary	1534 ±11	2,809	\$2.50 / \$15
6	2 - 27	<b>muse-spark</b> Meta · Proprietary	1532 ±15 <span>Preliminary</span>	1,424	N/A

# Evaluaton Methods



# Math Verifier



# Normalizing the extracted answer

`\dfrac{14}{3}`

`\frac{14}{3}`

14/3

`$14/3$`

`(14)/(3)`

# 4 Improving reasoning with inference-time scaling

- Prompting an LLM to explain its reasoning to improve answer accuracy
- Modifying the text generation function to produce diverse responses
- Improving reasoning reliability by sampling multiple responses

# Prompting an LLM to explain

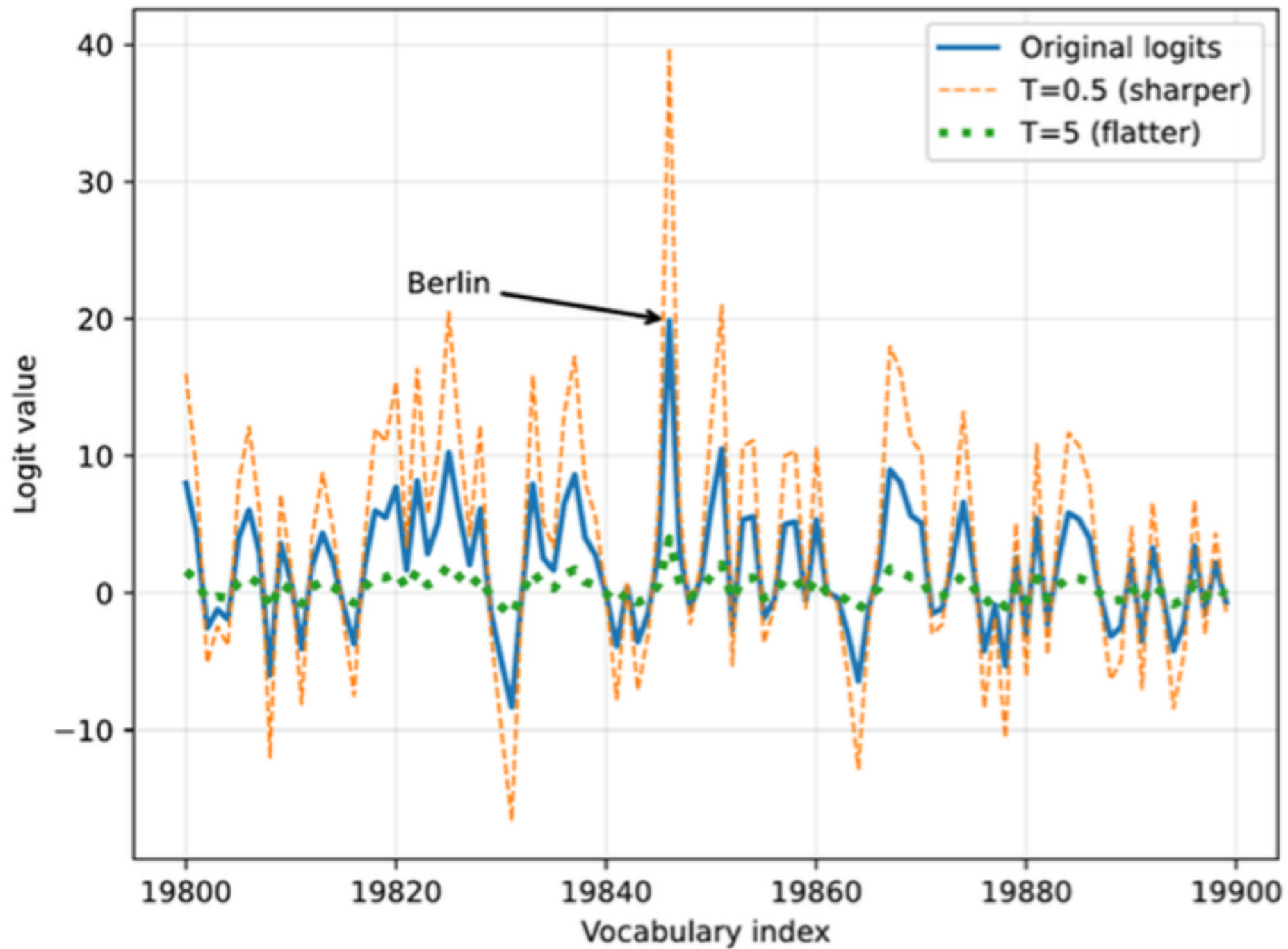
```
prompt_cot = prompt + "\n\nExplain step by step."
```

Large Language Models are Zero-Shot Reasoners

29 Jan 2023

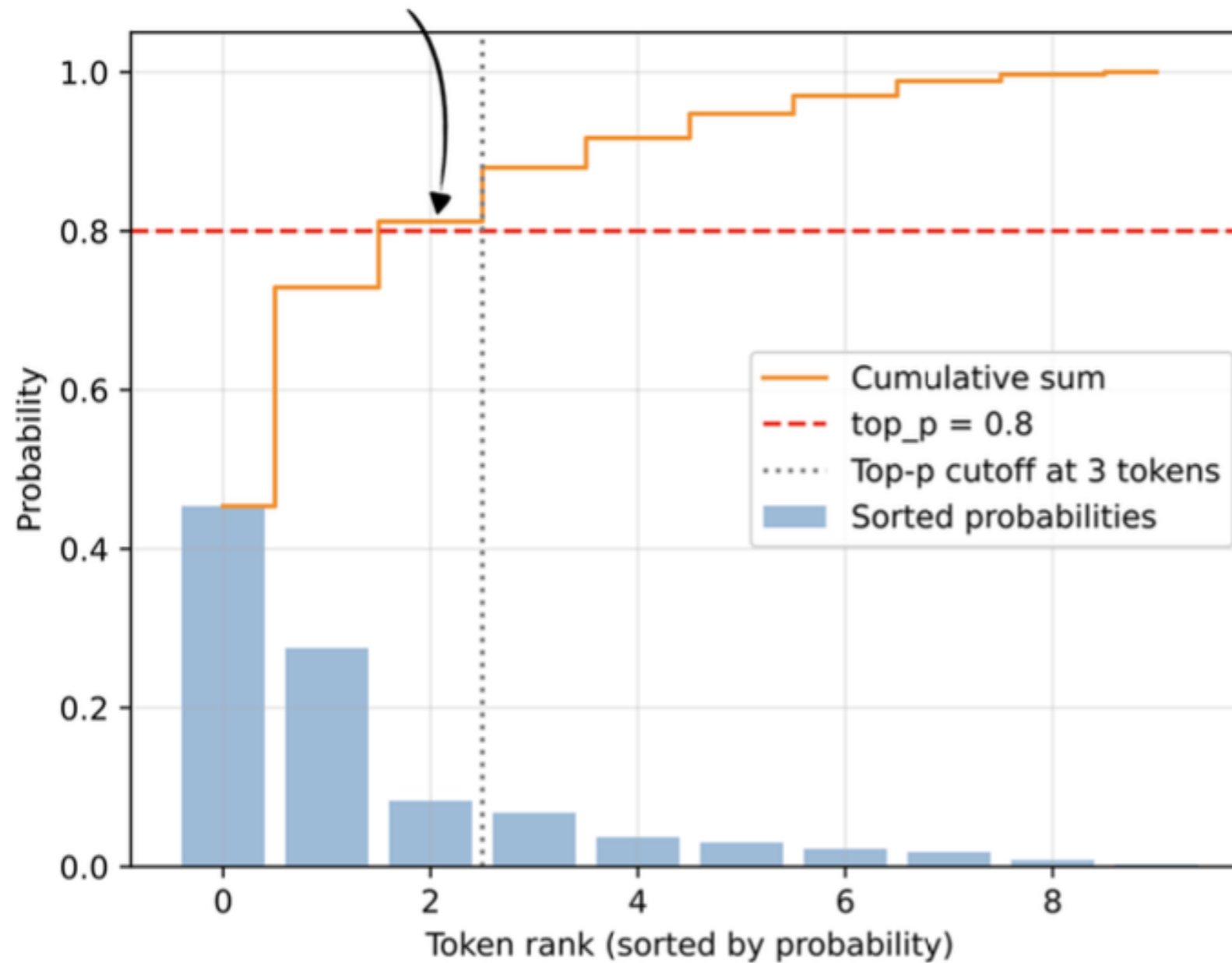
# Modifying the generation function to produce diverse responses

Change the temperature

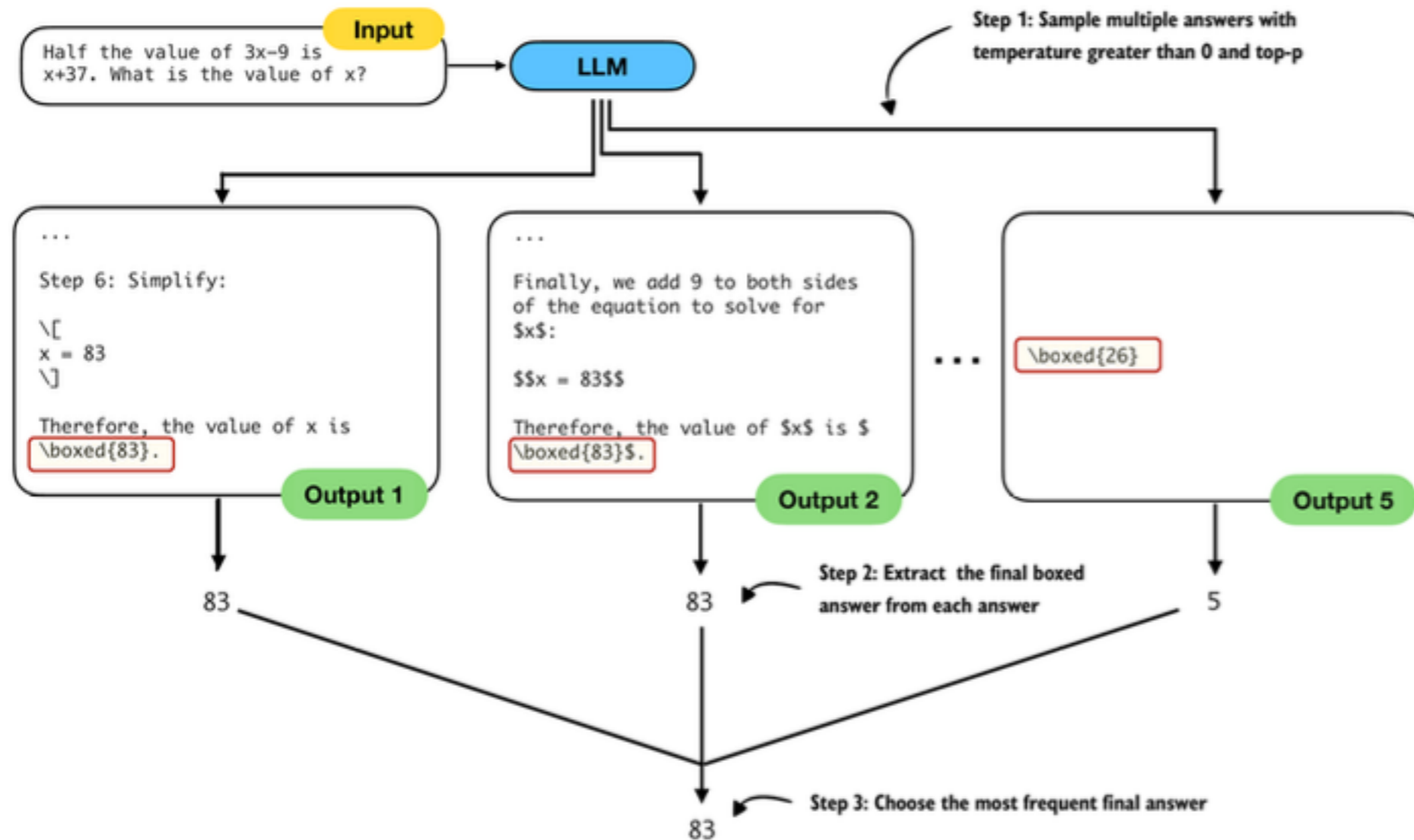


# Adding a top-p filter to the text generation function

According to the common top-p convention, the first token that exceeds the threshold is still included



Use Temperature and top-p to generate multiple answers and select the most frequent answer



# 5 Inference-time scaling via self-refinement

- Scoring LLM answers with a simple rule-based scorer
- Computing an LLM's own confidence in its answers
- Coding a self-refinement loop where the LLM iteratively improves its answers

# Scoring LLM answers with a simple rule-based scorer

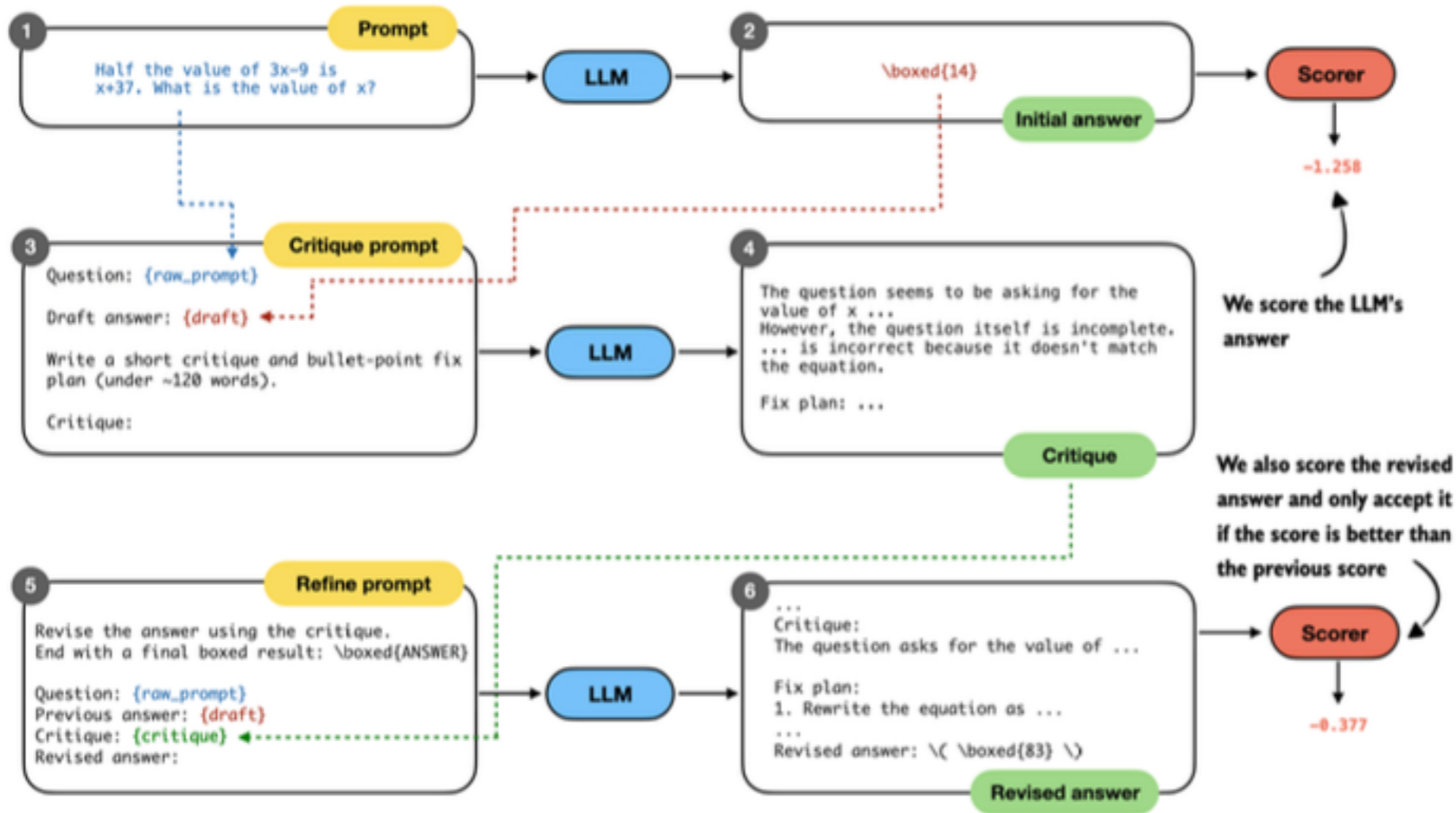
Selecting the common answer only works with very short answers

Use a function to rank the answers

He use length and format of answer

Convert answer logits to probabilities -> log-probabilities -> log-probabilities of entire response

# Self-refinement loop



Baseline (chapter 3)	-	-	Reasoning	48.2%	182.1 min
Self-refinement	None	1	Reasoning	56.6%	498.8 min
Self-refinement	Heuristic	1	Reasoning	57.8%	498.6 min
Self-refinement	Avg. logprob	1	Reasoning	48.4%	499.7 min

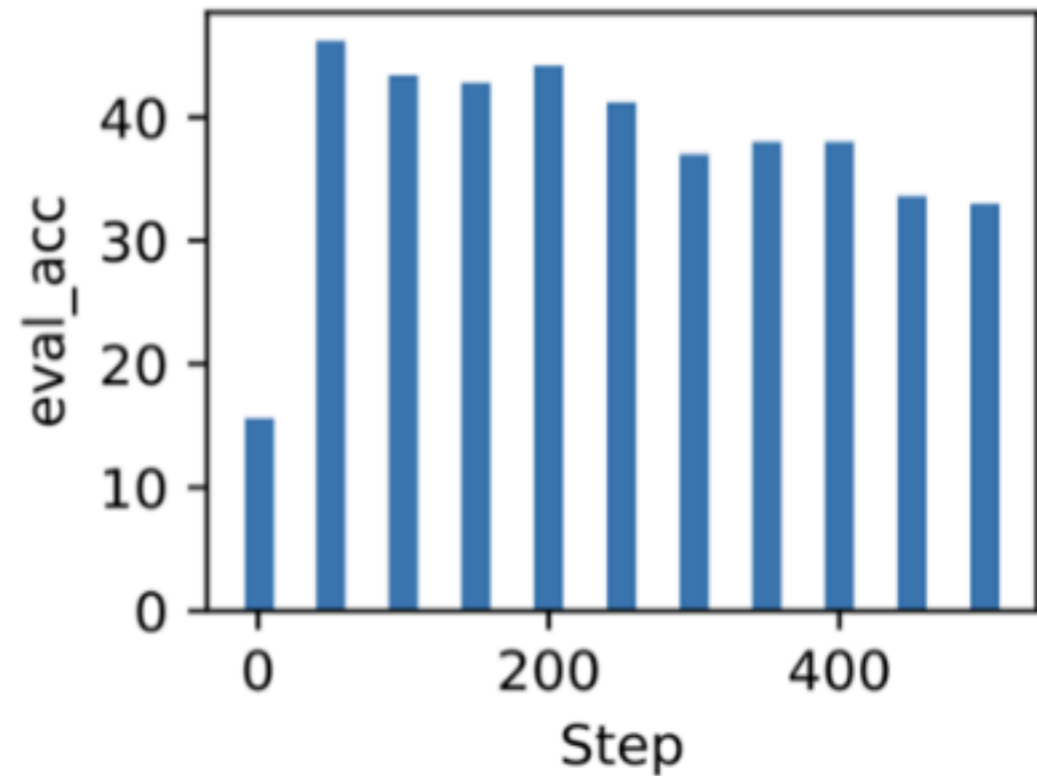
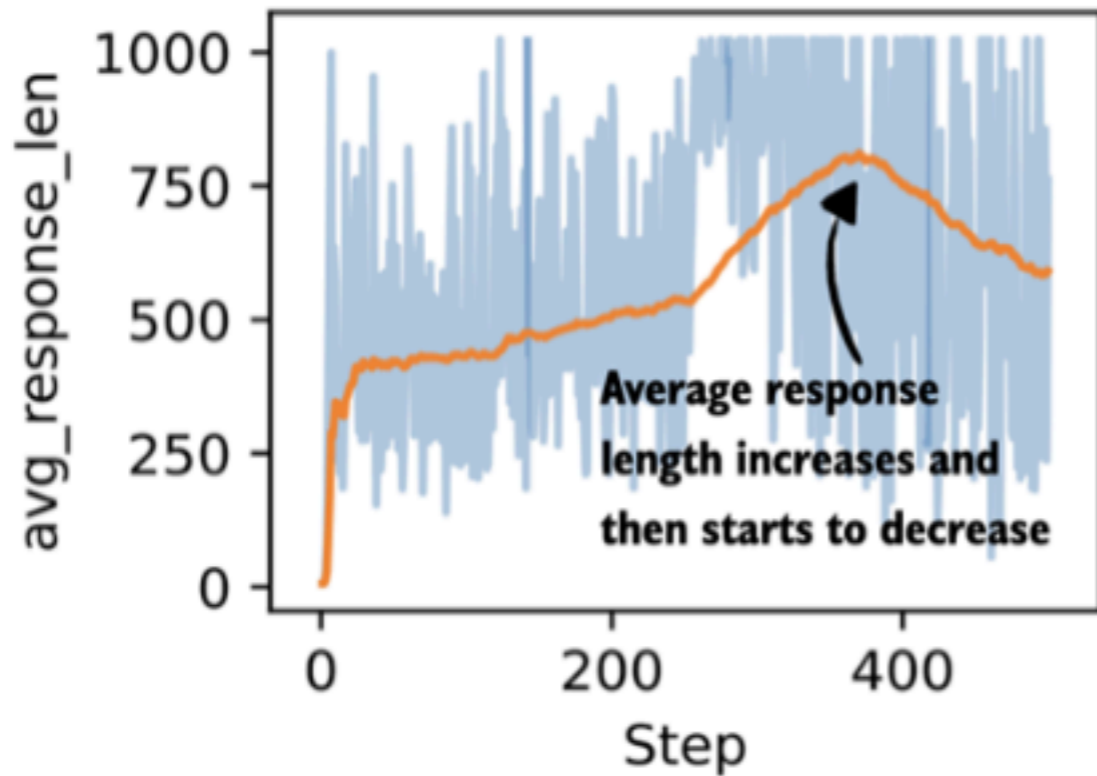
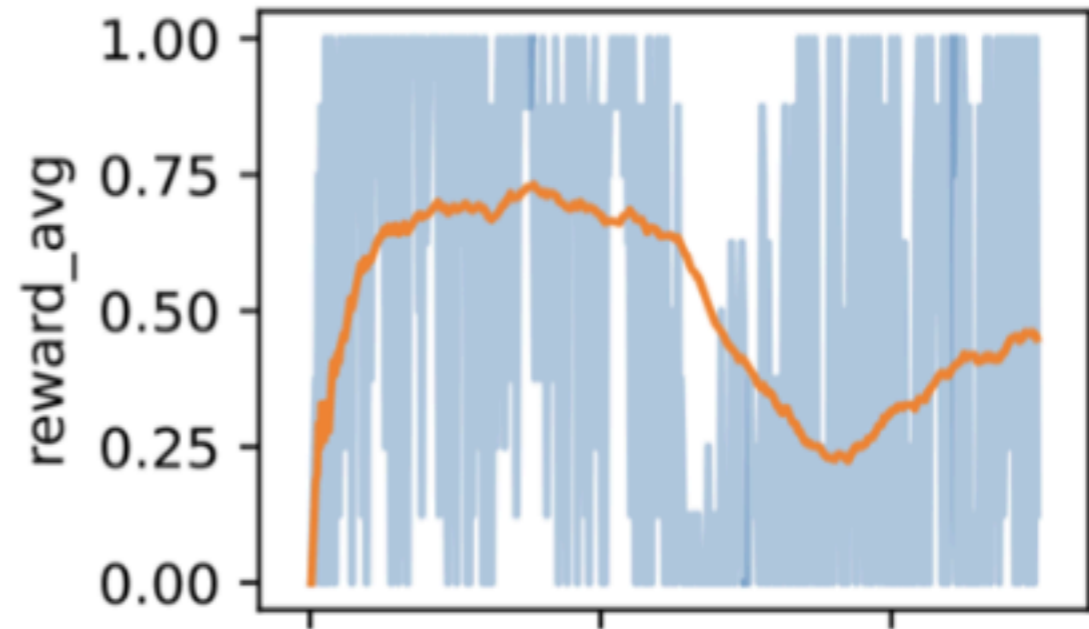
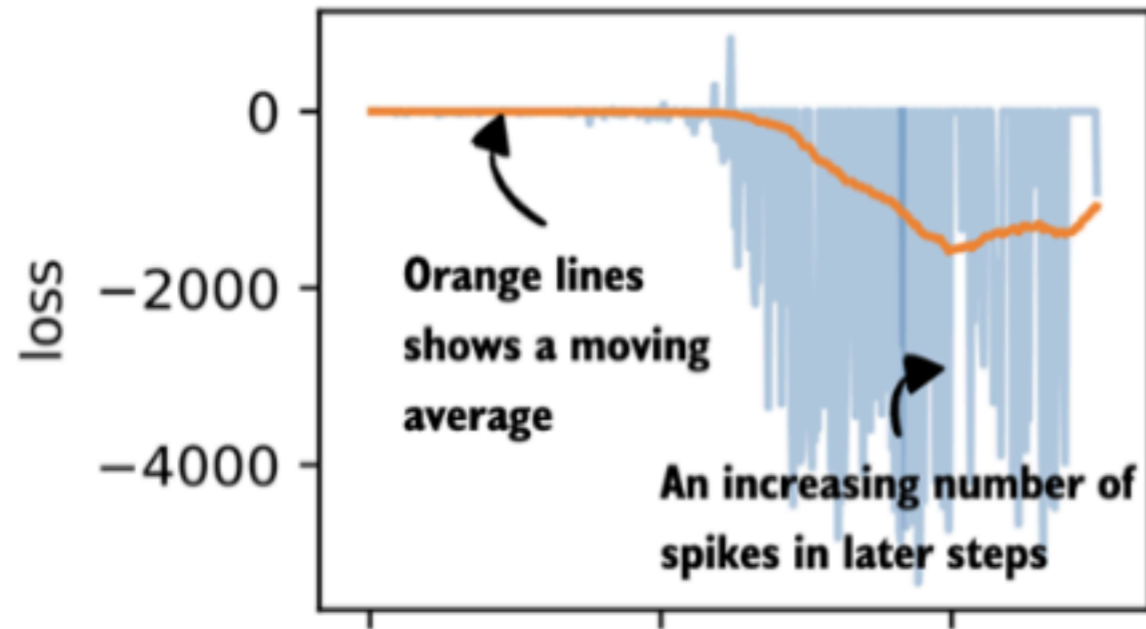
500 samples in the MATH-500 test set using a "cuda" GPU (DGX Spark)

## 6 Training reasoning models with reinforcement learning

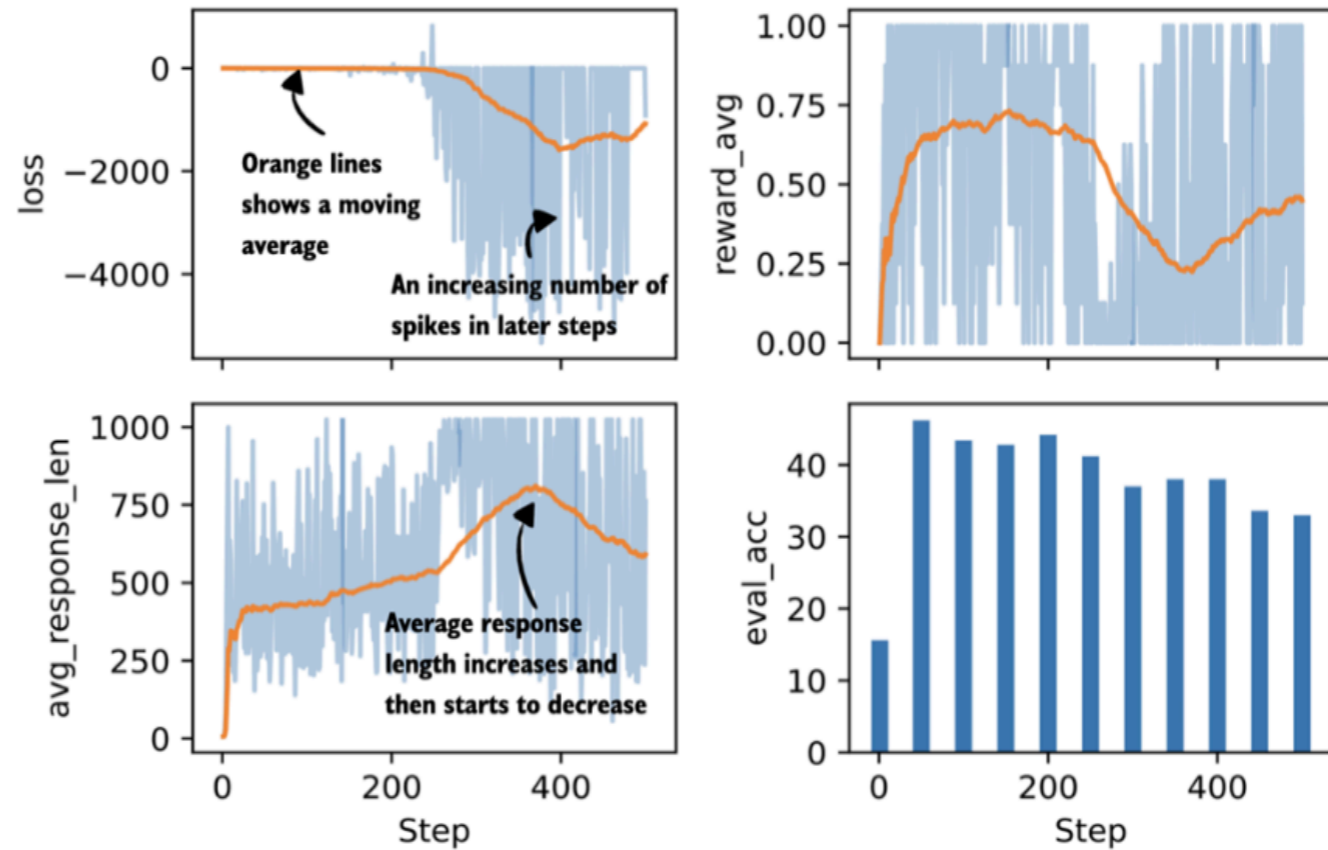
Use GRPO

- 1) A prompt is sampled and multiple outputs are generated.
- 2) Each output is scored using a verifiable reward.
- 3) Group-relative advantages are computed from these rewards.
- 4) The log-probability of each output under the current model is calculated.
- 5) Advantages and log-probabilities are combined to form the policy gradient loss.
- 6) A KL regularization term against a reference model is added, and the resulting total loss is used to update the model parameters.

# Improving GRPO



# Improving GRPO



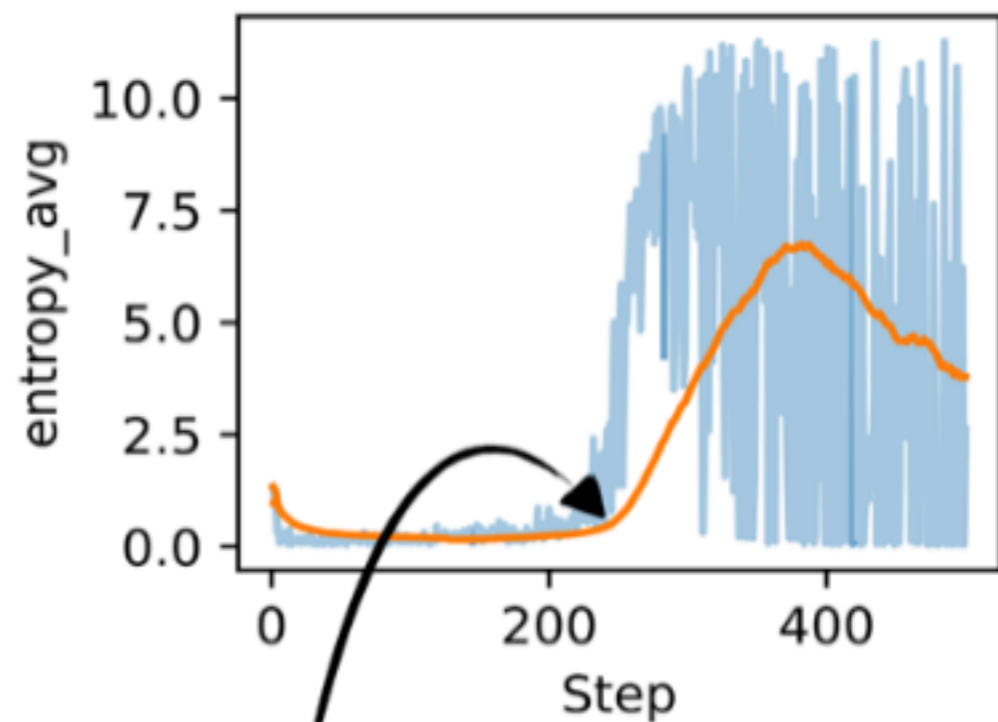
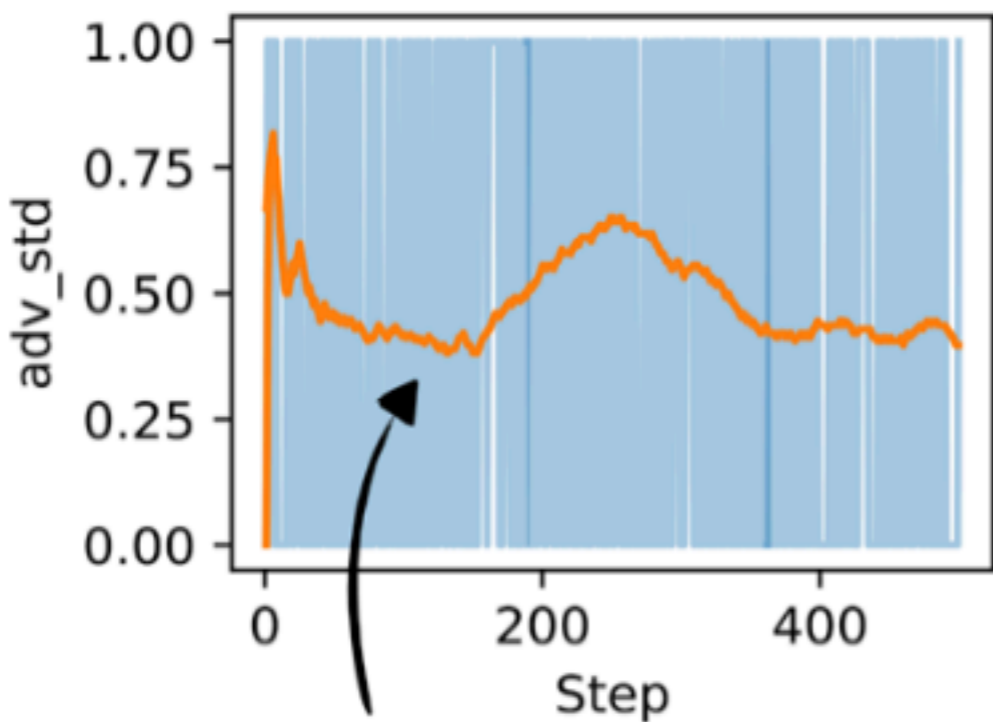
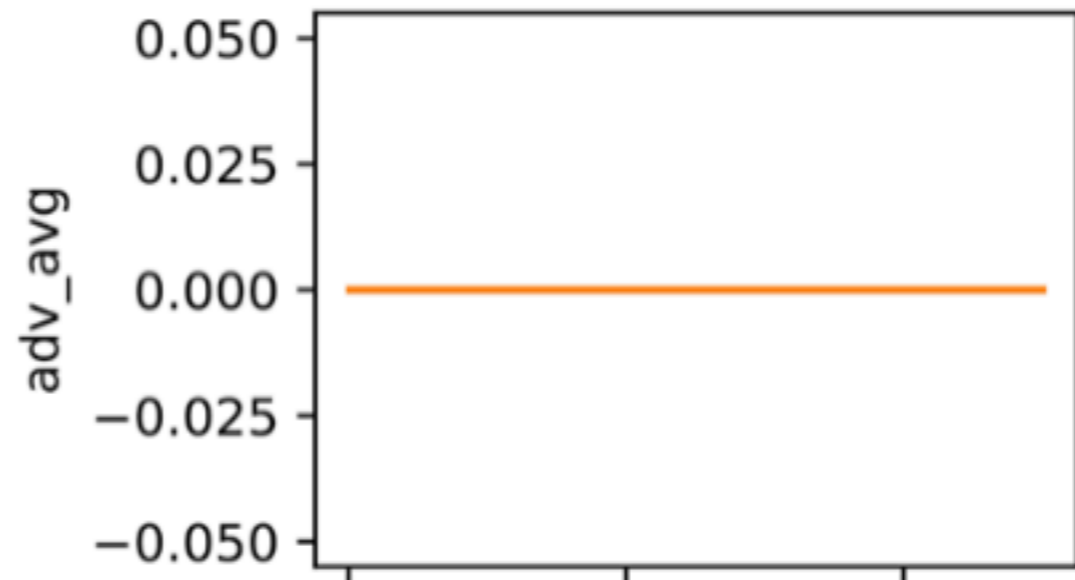
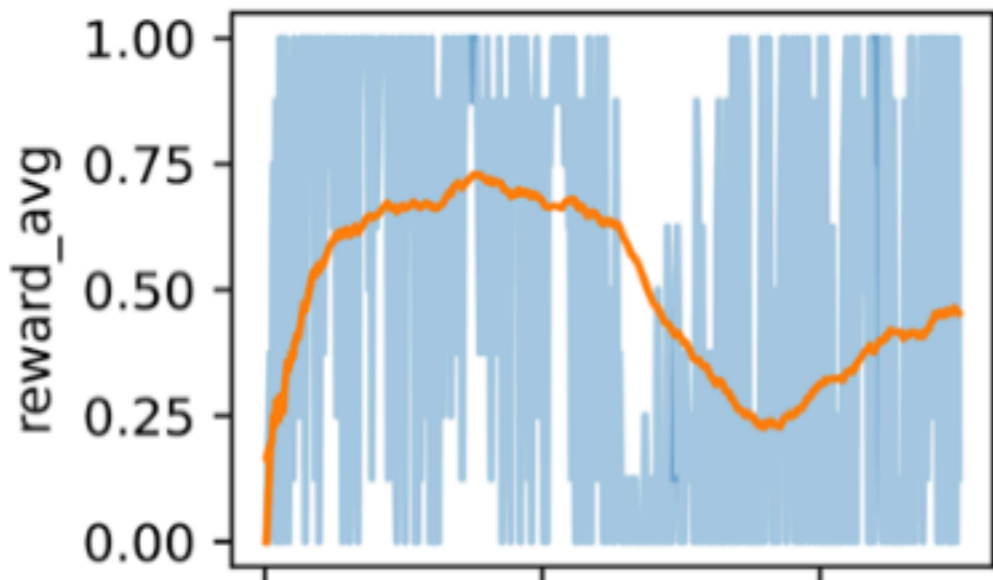
Average response length should initially increase, with an improvement in accuracy

The loss should remain relatively stable.

Average reward should also increase over time

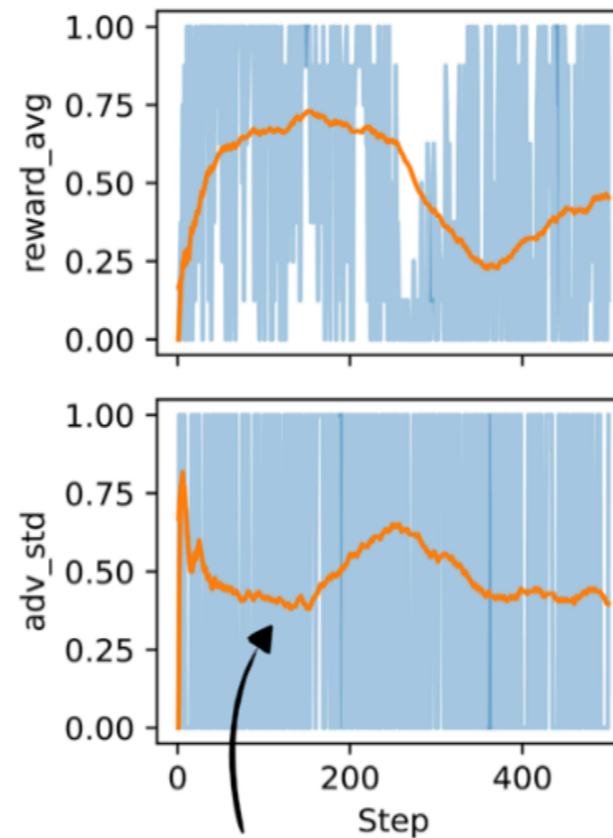
Average reward of 1.00 means that all sampled responses are correct

Evaluation should increase

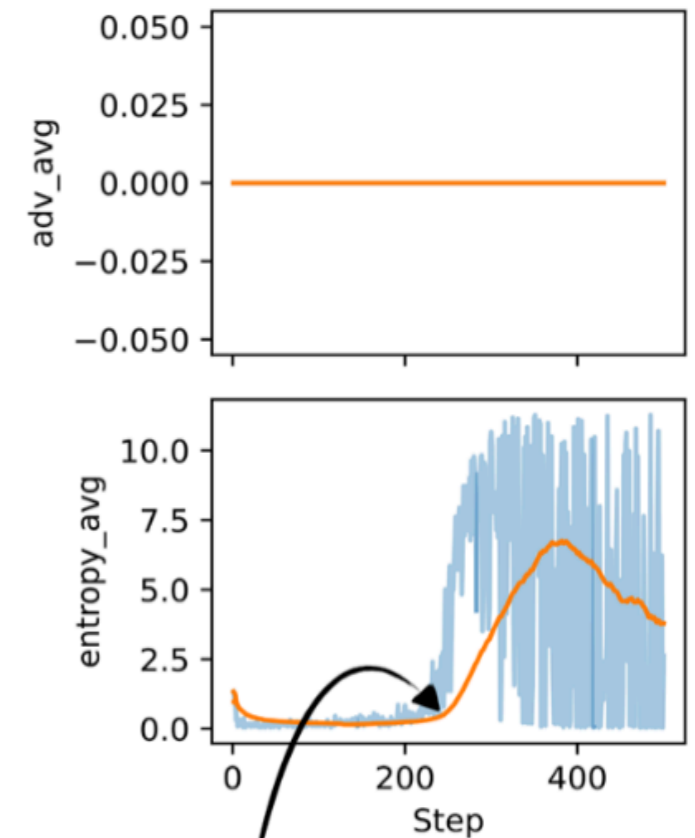


**The non-zero advantage standard deviation indicates that there is a good training signal throughout the training run**

**After an initial phase of moderate entropy, the model becomes much more random and higher-entropy**



The non-zero advantage standard deviation indicates that there is a good training signal throughout the training run



After an initial phase of moderate entropy, the model becomes much more random and higher-entropy

Average advantage stays at zero throughout training

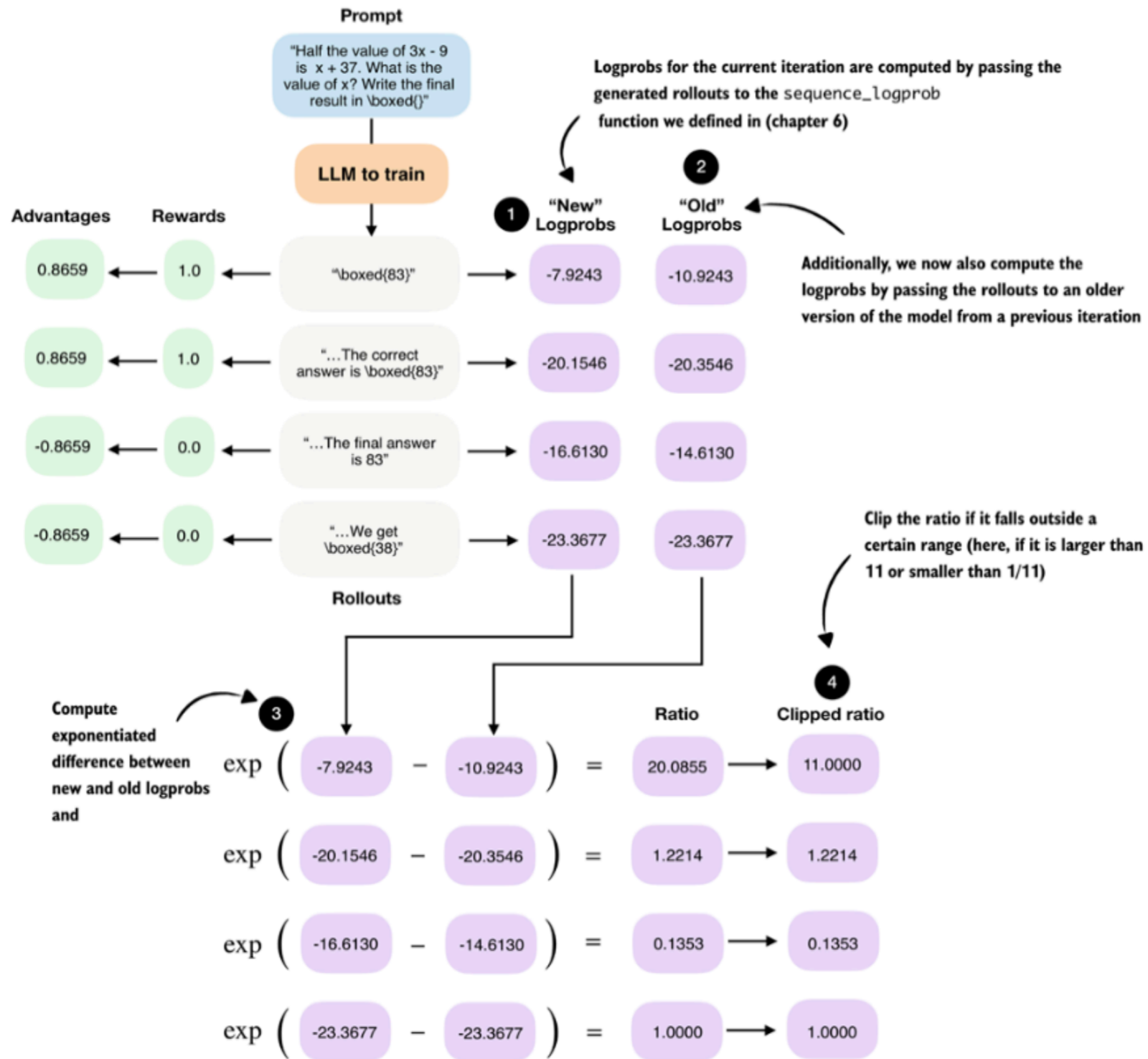
As long as the advantage standard deviation remains nonzero and reasonably stable there is still a usable learning signal and training happening

Very low entropy can also be a sign of collapse

# Clipped Policy Ratio

Measures how much the LLM has changed relative to an earlier version of itself

If the LLM previously assigned a certain likelihood to this answer, how much more or less likely does it consider the same answer after we adjusted its weights?

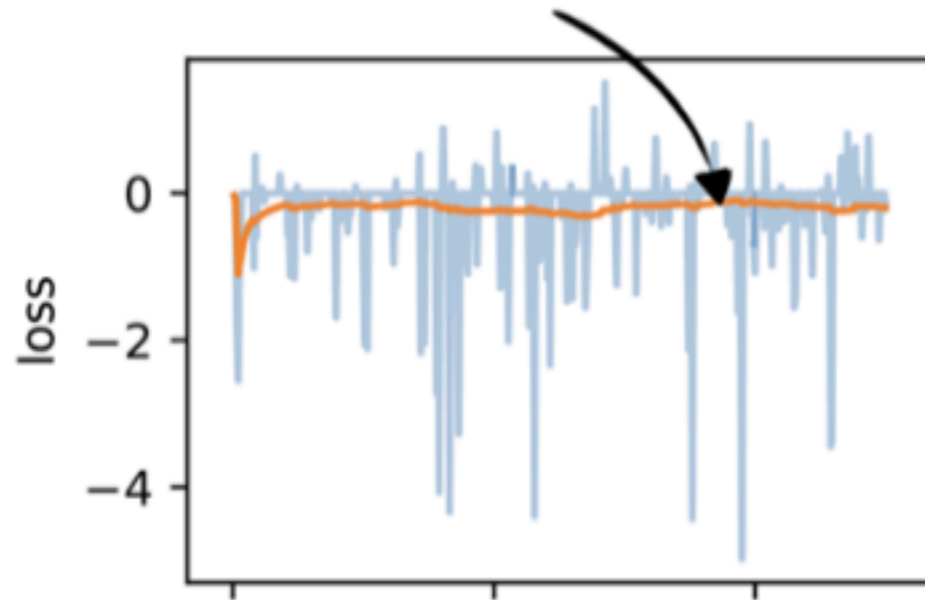


Clamp the ratio to the range  $1 \pm \text{clip\_eps}$ .

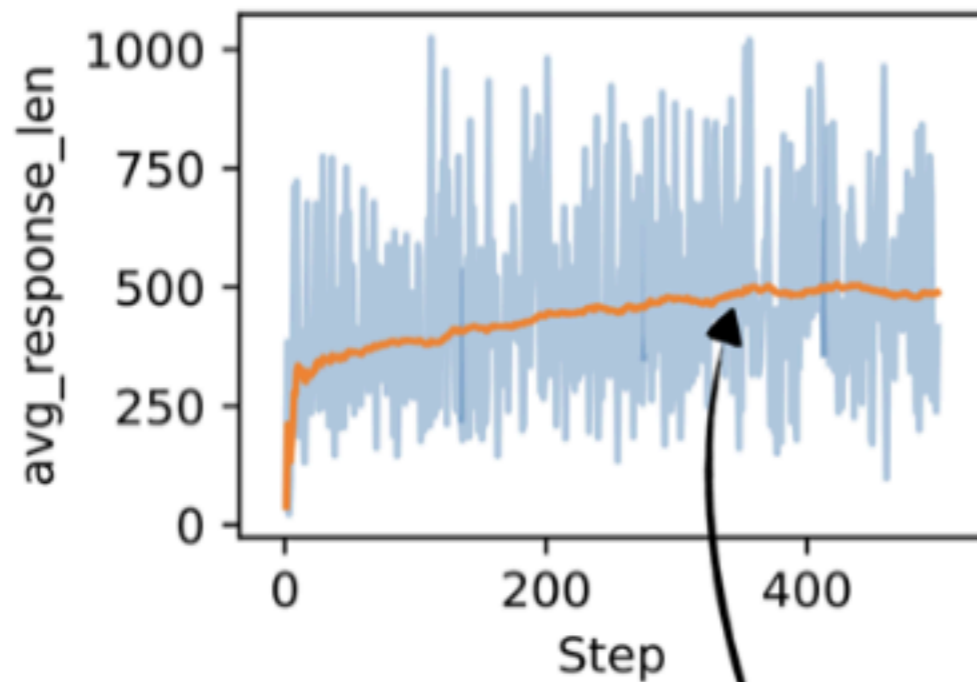
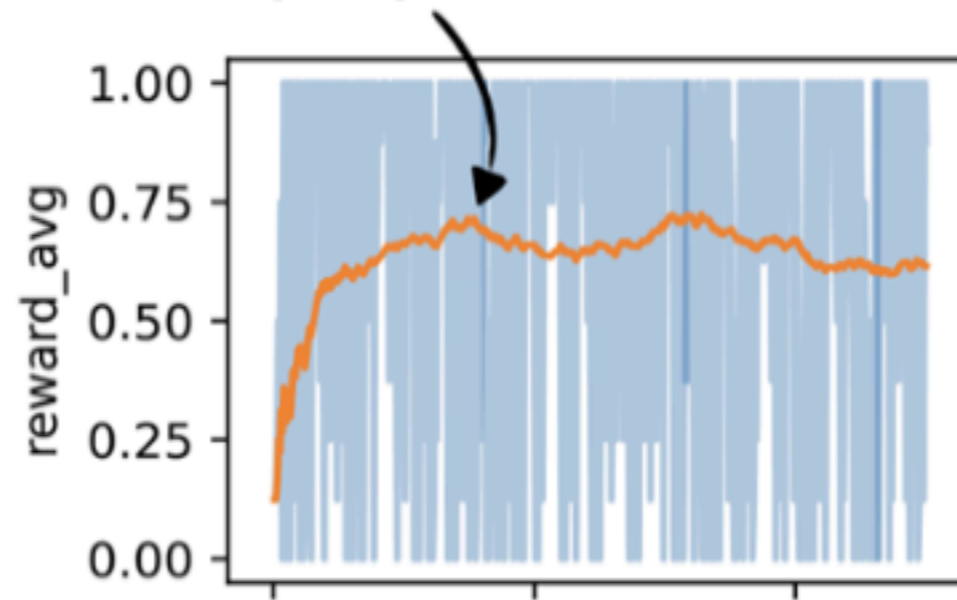
DeepSeek-R1 used `clip_eps = 10`

# Using clipped Policy Ratios

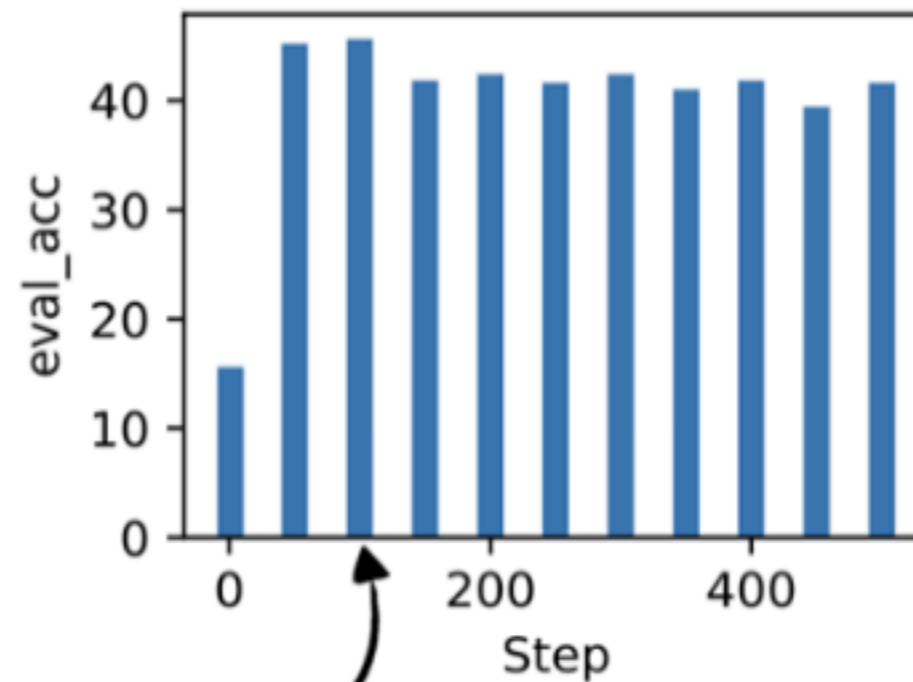
The number of loss spikes does not suddenly increase



The average rewards increase and then stay relatively steady and stable



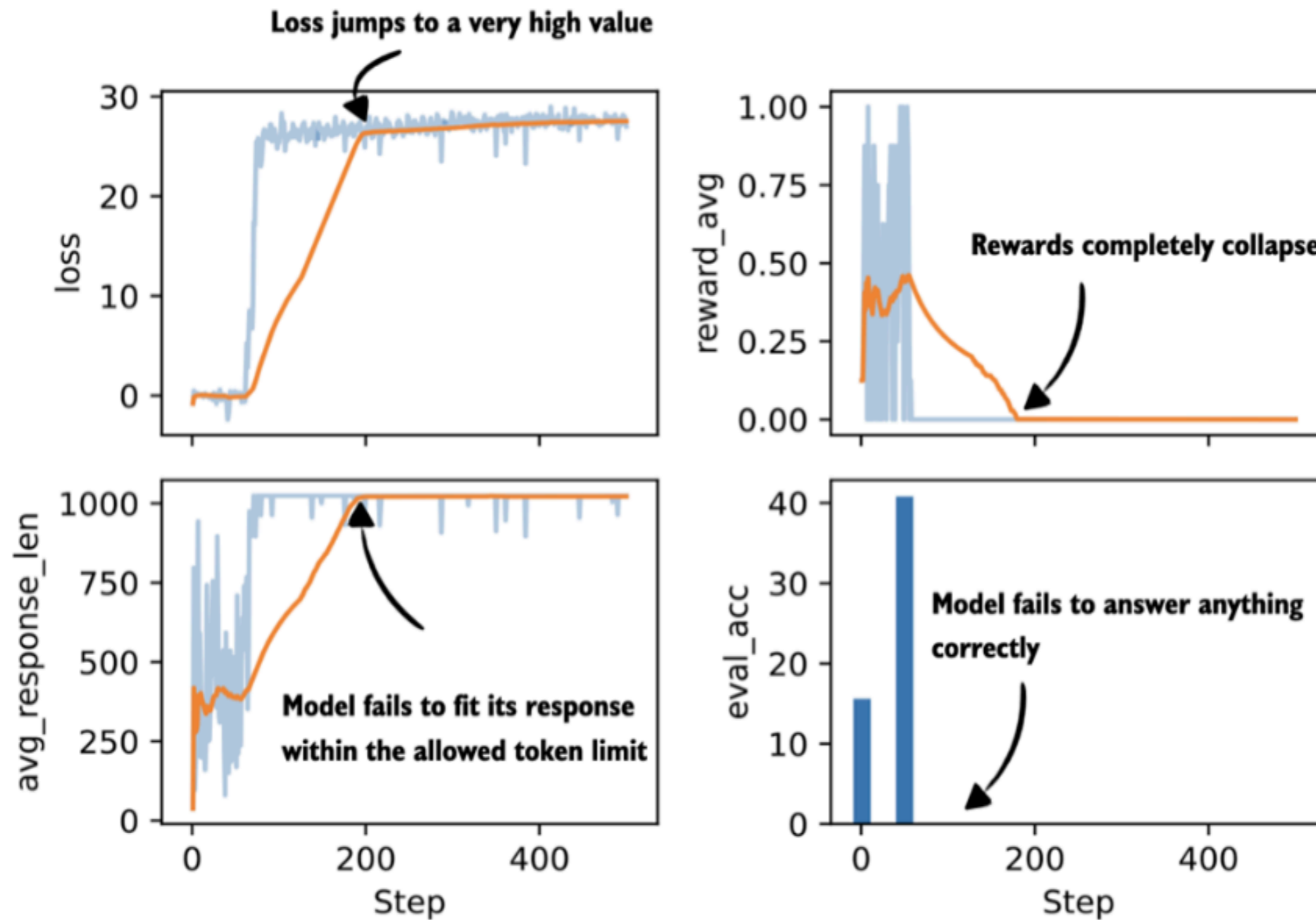
The average response length is now steadily increasing



While the accuracy stops increasing after the first 100 steps, it doesn't suddenly decline

# Kullback–Leibler (KL) divergence

Measures how much the LLM deviates from the original model



Several recent works report that models can train better without it

# Other Possible Improvements

1. Zero gradient signal filtering (DAPO)
2. Active sampling (DAPO)
3. Switch from sequence- to token-level loss (DAPO)
4. No KL loss (DAPO and Dr. GRPO)
5. Clip higher (DAPO)
6. Truncated importance sampling (VERL)
7. No standard deviation normalization (Dr. GRPO)
8. KL tuning with domain-specific KL strengths; zero for math (DeepSeek-V3.2)
9. Reweighted KL (DeepSeek-V3.2)
10. Off-policy sequence masking (DeepSeek-V3.2)
11. Keep sampling mask for top-p / top-k (DeepSeek-V3.2)
12. Keep original GRPO advantage normalization (DeepSeek-V3.2)
13. Per-reward group-wise normalization before aggregation (GDPO)
14. Sequence-level importance sampling and clipping (GSPO)
15. Clip importance-sampling weights rather than token updates (CISPO)

# 8 Distilling reasoning models for efficient reasoning

## Hard Distillation

The student model is trained on the teacher model outputs

## Soft Distillation

The student model is trained on the teacher's probability distribution over the vocabulary by minimizing the KL divergence