Lecture 12: The Evolution of Frontier Model Evaluation

From few-shot prompting to agentic sandboxes

RLHFBOOK.COM

Nathan Lambert

Course on RLHF and post-training. Chapter 16.

How we measure progress changes over time

Different model types need different strategies.

As models get smarter, we need to re-invent the cutting edge of AI.

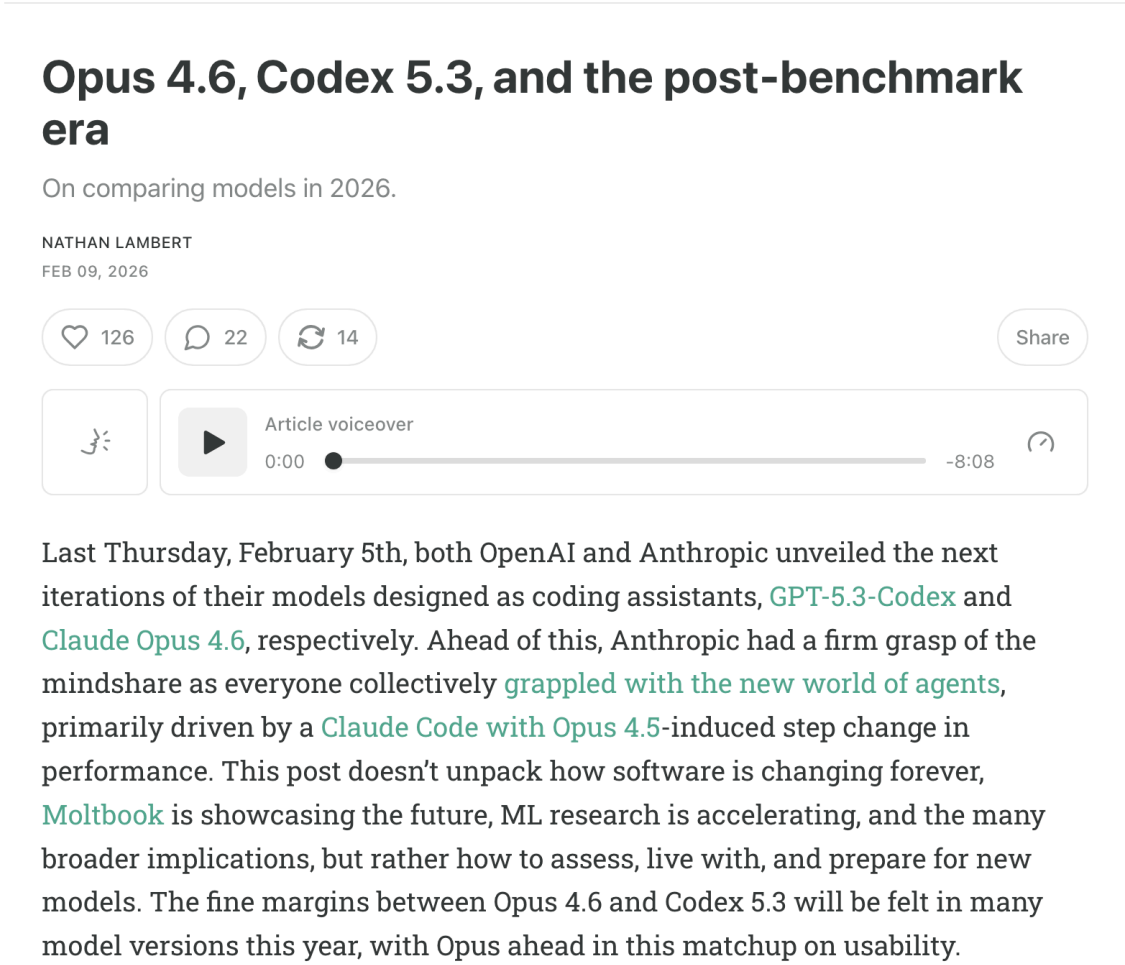
Evaluation is at the core of AI progress and understanding truth of what the models are.

Frontier evaluation is harder than it ever has been

When Opus 4.6 and GPT-5.3-Codex shipped in the same week (Feb 2026), the headline benchmark deltas were tiny – and settled nothing about which model to use.

What separated them was found through use: usability, product fit, behavior over long agentic tasks. Benchmark-based release reactions barely matter at the frontier – consistent testing and clear articulation have to carry the comparison.

Read on Interconnects: the post-benchmark era



Opus 4.6, Codex 5.3, and the post-benchmark era

On comparing models in 2026.

NATHAN LAMBERT
FEB 09, 2026

126 22 14 Share

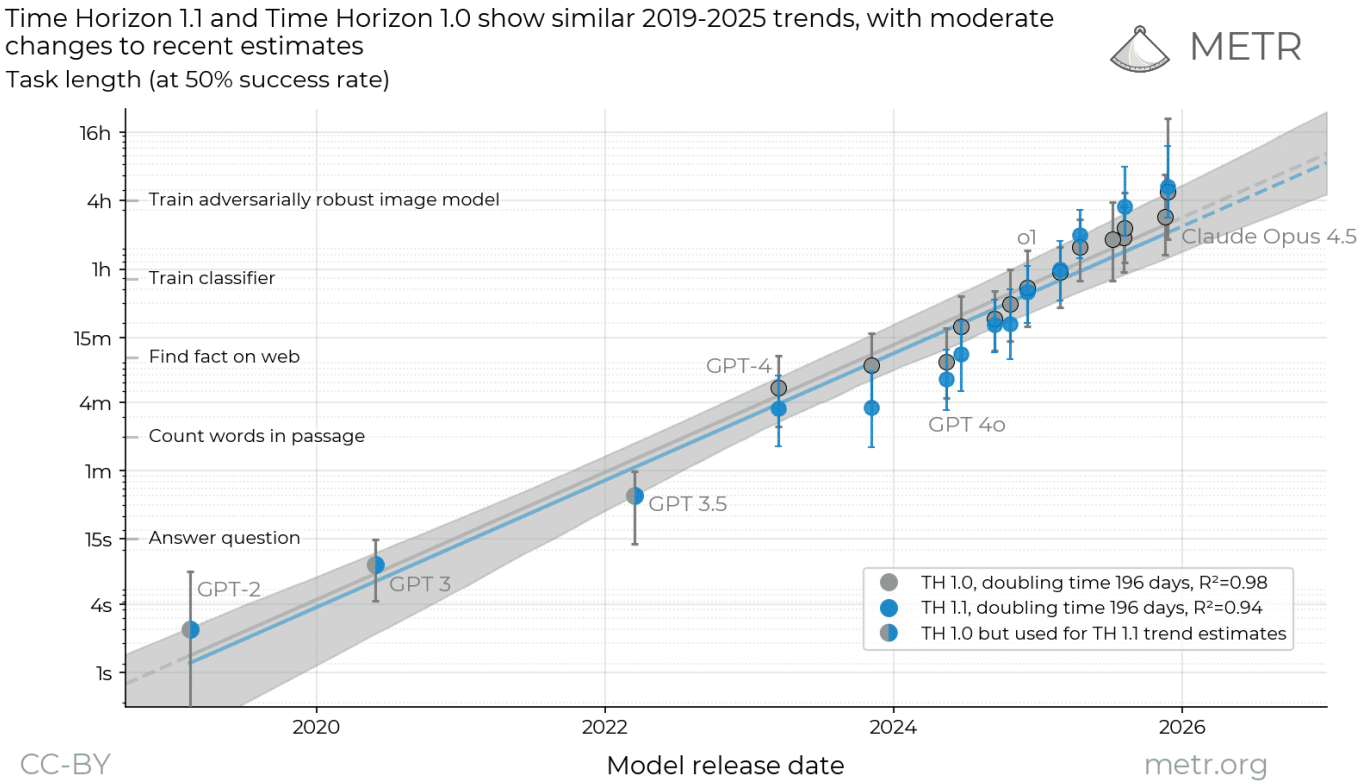
Article voiceover
0:00 -8:08

Last Thursday, February 5th, both OpenAI and Anthropic unveiled the next iterations of their models designed as coding assistants, GPT-5.3-Codex and Claude Opus 4.6, respectively. Ahead of this, Anthropic had a firm grasp of the mindshare as everyone collectively grappled with the new world of agents, primarily driven by a Claude Code with Opus 4.5-induced step change in performance. This post doesn't unpack how software is changing forever, Moltbook is showcasing the future, ML research is accelerating, and the many broader implications, but rather how to assess, live with, and prepare for new models. The fine margins between Opus 4.6 and Codex 5.3 will be felt in many model versions this year, with Opus ahead in this matchup on usability.

…and the tasks we're trying to measure keep taking longer

The task length frontier models can complete (at 50% success) **doubles roughly every 7 months** – from seconds-long questions to tasks that take human experts hours (Time Horizon 1.1).

Measuring the frontier now means running hours-long expert tasks, many times over.



Task-completion time horizon of frontier models. Figure from METR (Time Horizon 1.1), CC-BY.

Kwa et al., 2025

This lecture

Every training decision in this course – data mixes, hyperparameters, which checkpoint ships – is made off benchmark numbers.

Today: How we got to modern benchmarking approaches and research.

The plan

1. **The eras** – how each model type was prompted, graded, and benchmarked
2. **A bit more on agentic evals** – the system around today's scores
3. **Trusting the number** – variance, contamination, and gaming

Part 1: The eras of post-training evaluation

Benchmarks mirror the training goals of their era

The key to understanding evals: popular benchmarks are a **reflection of the training best practices of their moment**.

- **Chat era** *(2022-23)* – basic knowledge and chat style
- **Multi-skill era** *(2023-24)* – post-training improves more skills than just chat (math, code, factuality, safety, etc.)
- **Reasoning & tools era** *(2024-26)* – hard math, coding, and reasoning problems, long chains of thought
- **Agents & real work** *(now)* – end-to-end knowledge-work tasks inside products and harnesses

Base models (before post-training): few-shot prompting

Base models can't take a bare or formatted question – eval prompts carried examples of the patterns (3 to 8+ in-context samples) so the model continues the pattern. Canonical evals: **5-shot MMLU**, **8-shot GSM8K**.

```
# Few-Shot Prompt

Below are examples of MMLU-style questions and answers:

### Example 1
Q: A right triangle has legs of lengths 3 and 4.
What is the length of its hypotenuse?
Choices:
(A) 5
(B) 6
(C) 7
(D) 8

Correct Answer: (A)
```

```
### Example 2
Q: Which of the following is the chemical symbol for Sodium?
Choices:
(A) Na
(B) S
(C) N
(D) Ca

Correct Answer: (A)

### Now answer the new question in the same style:

Q: Which theorem states that if a function f is continuous
on a closed interval [a,b], then f must attain both a
maximum and a minimum on that interval?
Choices:
(A) The Mean Value Theorem
(B) The Intermediate Value Theorem
(C) The Extreme Value Theorem
(D) Rolle's Theorem

Correct Answer:
```

Brown et al., 2020; Robinson et al., 2023

Grading: log-likelihood vs. exact match

Log-likelihood scoring: compare the probability the model assigns each answer option – either just the letter `A`, or the full answer string. No sampling, fully deterministic. The standard for pretraining evals, where models couldn't always answer in a clean format.

Generation + exact match: sample a completion, extract the answer. Mirrors real usage – and is standard for post-training since ~2024. Aggregating multiple completions/samples gives majority voting; e.g. **pass@k** is a common tool.

Generation and extraction gave rise to answer extraction formatting bugs, which only became more complex with agentic models today.

Brown et al., 2020; Olmo et al., 2025

The math behind pass@k

pass@k = the probability that **at least one of k samples** solves the problem.

The naive route – generate exactly k, report whether any passed – is a **high-variance** coin flip per problem, and plugging a small-sample success rate into $1-(1-\hat{p})^k$ is **biased**.

The fix, from the Codex paper: sample $n \geq k$ completions, count the c that pass, and average an unbiased estimator over problems.

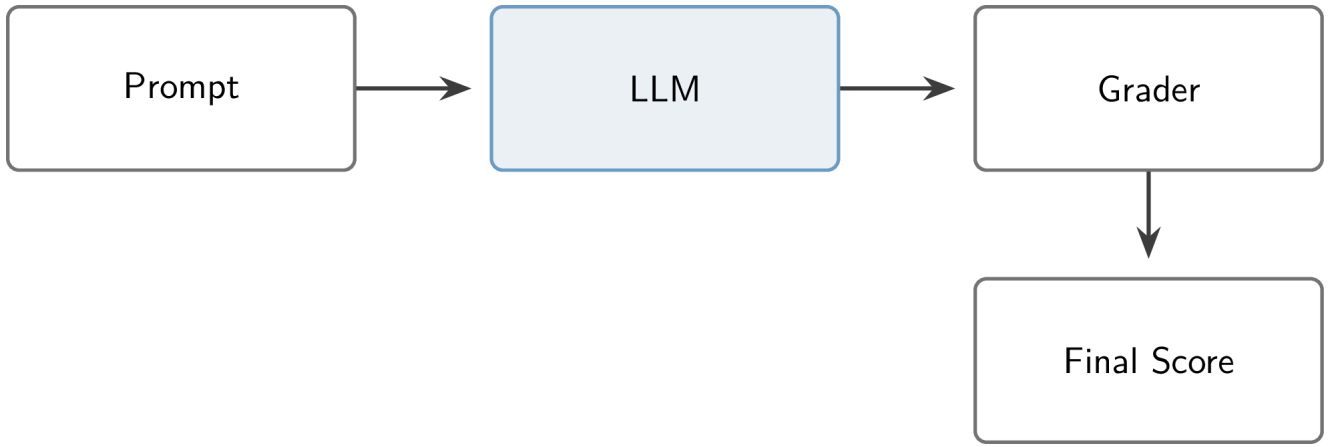
$$\text{pass@}k = \mathbb{E}_{\text{problems}}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right]$$

- $\binom{n-c}{k} / \binom{n}{k}$ is the chance that k draws (without replacement) from your n samples are **all failures**
- Larger n → tighter estimate at the same k; the paper used $n = 200$ for $k \leq 100$
- The knobs interact: **higher temperature can hurt pass@1 but help pass@100** – so a reported "pass@1" depends on n and the sampling settings, not just the model

Chen et al., 2021

The early pipeline was simple

Prompt in, completion out, grade it. Almost everything that could go wrong was in **how you formatted the prompt**.



Slide Credit: Florian Brand

Chain of thought (CoT) emerged to enable progress on harder problems

Few-shot examples that show intermediate steps let models reason before answering. When people were still prompting base models, adding CoT made math and reasoning scores jump! This is before modern post-training as well.

Soon just appending *"Let's think step by step"* to a prompt approximated this behavior.

```
# standard prompting
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis
balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how
many apples do they have?

A: The answer is ...

# chain-of-thought prompting
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis
balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 =
11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how
many apples do they have?

A: The cafeteria had 23 apples originally. They...
```

Kojima et al., 2022; Wei et al., 2022

Entering the chat era: zero-shot instruction following

Instruction tuning (FLAN (Wei et al., 2022), T0 (Sanh et al., 2022)) and then RLHF changed the way people expected to use models: the models learned to directly answer questions. This, in retrospect, was a huge deal! But not the default.

Now, the input to the model can look like:

```
User: "What is the capital of France?"
Assistant:
```

- **LLM-as-a-judge** emerged as questions became open-ended (and evals imitated RLHF training)
- Canonical evals: **MT-Bench** (Zheng et al., 2023), **AlpacaEval** (Dubois et al., 2024), and the community-scale Chatbot Arena (Chiang et al., 2024)
- MCQ evals like MMLU stayed in the mix (but were in flux, people used them differently) – now answered zero-shot, sampling the answer letter at temperature 0

The emergence of zero-shot prompting took time!

When we trained Tülu 3 (summer 2024), many of our evaluations were a mix of zero-shot and few-shot prompting. Though, the field was focusing on a variety of skills (from Tülu 3):

- **Knowledge**: MMLU, PopQA, TruthfulQA
- **Reasoning**: BigBenchHard, DROP
- **Math & code**: MATH, GSM8K, HumanEval
- **Instruction following & safety** IFEval, others

By early 2025, everyone was using zero-shot prompting, and today multi-shot prompting for a post-trained model is very rare, unless the model wasn't trained for it (true in-context learning).

Example question (MMLU)

What was GDP per capita in the United States in 1850 when adjusting for inflation and PPP in 2011 prices?

A. About $300
B. About $3k
C. About $8k
D. About $15k

Internet trivia more than intelligence – but it tracked pretraining knowledge well.

Hendrycks et al., 2021; Lambert et al., 2024

Encouraging the models to reason

We know reasoning models today think before answering. Tülu 3's MMLU prompt (a model before reasoning models) → the same MMLU eval as the few-shot era, now long-form CoT with exact-match checking.

Modern eval suites can carry **per-benchmark prompts** tuned for formatting etc.

Tülu 3's MMLU prompt:

```
Answer the following multiple-choice question by giving the correct answer letter in
parentheses.
Provide CONCISE reasoning for the answer, and make sure to finish the response with
"Therefore, the answer is (ANSWER_LETTER)" where (ANSWER_LETTER) is one of (A), (B),
(C), (D), (E), etc.

Question: {question}
(A) {choice_A}
(B) {choice_B}
(C) ...

Answer the above question and REMEMBER to finish your response with the exact phrase
"Therefore, the answer is (ANSWER_LETTER)" where (ANSWER_LETTER) is one of (A), (B),
(C), (D), (E), etc.
```

Lambert et al., 2024

Encouraging the models to reason

We know reasoning models today think before answering. Tülu 3's MMLU prompt (a model before reasoning models) → the same MMLU eval as the few-shot era, now long-form CoT with exact-match checking.

Modern eval suites can carry **per-benchmark prompts** tuned for formatting etc.

Sampling settings joined the prompt as part of the eval: reasoning models need **temperature > 0** for their best scores – Qwen's model cards literally say **"DO NOT use greedy decoding"**. Read the `generation_config.json`: the recommended settings are "free" performance.

Lambert et al., 2024

Formatting became fragile as usage became more open-ended

- Formatting mismatches can take a model from **60% to near 0** (Schulhoff et al., 2024) – it is far easier to lose performance with a prompt than to gain it
- Answer extraction is brittle: rigid suffixes (*"The answer is:"*) or regexes hunting for the answer anywhere in the text
- Formats even conflict across training sets: NuminaMath (Li et al., 2024) wants `\boxed{42}`, MetaMath (Yu et al., 2024) wants `The answer is: 42` – **training on both can be worse than either alone**
- Format-agnostic grading takes substantial effort and tinkering – and was often rare in practice – LLM-judges became popular even as answer extractors for flexibility

Reasoning & tool-use pushed the industry to harder tasks

Reasoning models saturated the old eval suites, so the next generation came:

- **Knowledge**: GPQA Diamond (Rein et al., 2023), Humanity's Last Exam (Phan et al., 2025), FrontierMath
- **Math**: recent AIME contests
- **Software**: SWE-Bench (+ variants), LiveCodeBench (Jain et al., 2024)
- Question sourcing moved from the internet to **grad students, PhDs, and professors** – writing questions became expert labor

Reasoning & tool-use pushed the industry to harder tasks

Example: Humanity's Last Exam

What was the rarest noble gas on Earth as a percentage of all terrestrial matter in 2002?

Official answer: **Oganesson**

The official answer is wrong three ways: oganesson is **not a gas** (predicted solid), **not noble** (predicted reactive), and **not terrestrial** (synthetic, ~5 atoms ever made – first synthesized in 2002).

Past a certain difficulty, **verifying the answer key is the bottleneck** – expert-written no longer means correct.

Still, incorrect labels are a common problem in evals! Most evals saturate at 90-95%, not 100%.

Phan et al., 2025

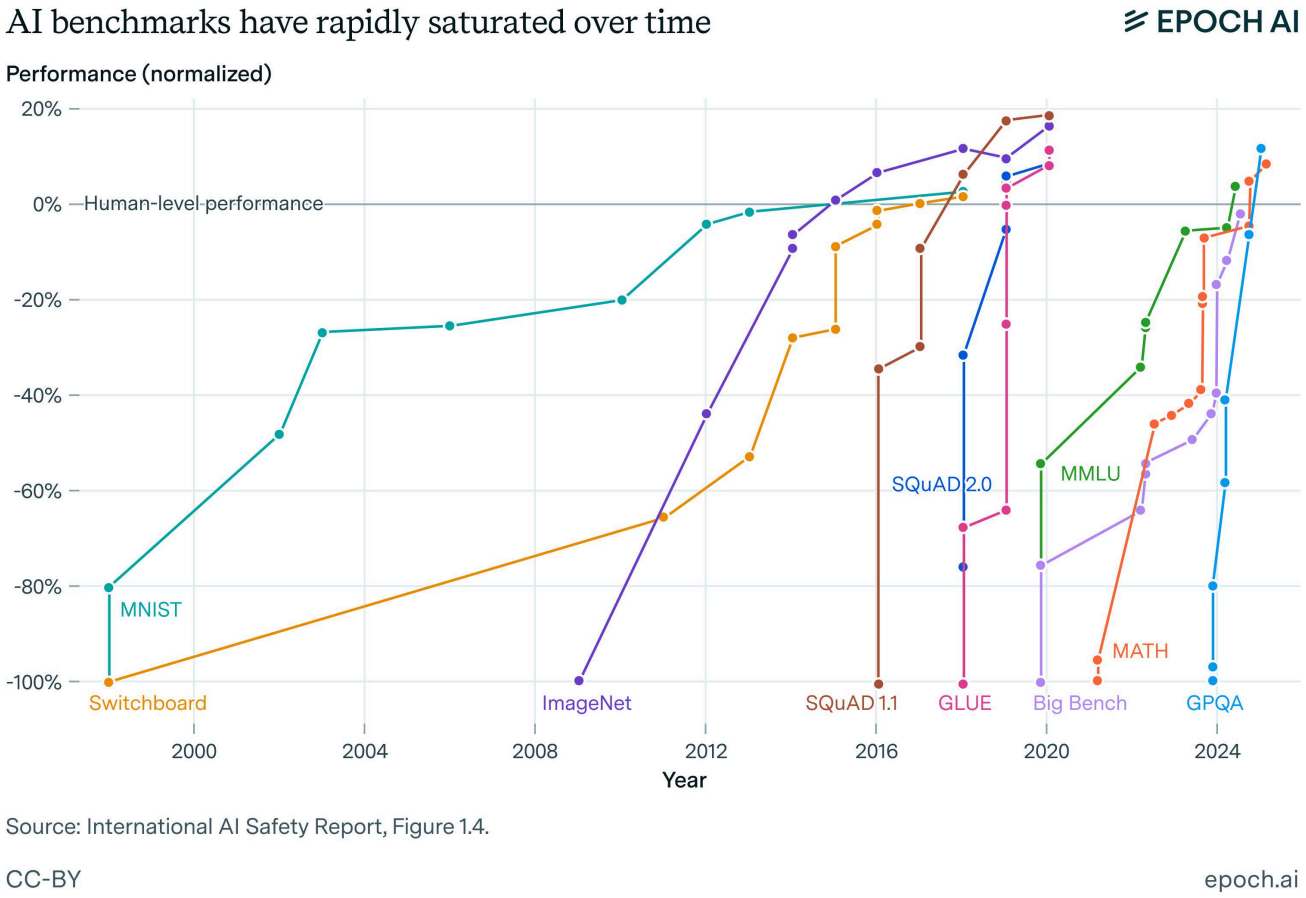
Today: evals of real work

- The frontier evals are **end-to-end professional tasks**: SWE-bench Verified, Terminal-Bench, GDPVal, APEX
- Task authors are now **experienced professionals**: GDPVal tasks come from experts averaging **14 years** of industry experience; APEX experts average 7+ years at firms like Goldman and McKinsey – expert task-writing is the new cost center
- And the models aren't evaluated bare: they run **inside harnesses and products** (Claude Code, Codex CLI) – last lecture's subject (11 - tool-use)

OpenAI, 2024

Every era ends the same way: saturation

Benchmarks are consumable. As scores approach the ceiling, only the hardest (and mislabeled) items remain, and the benchmark stops separating models.



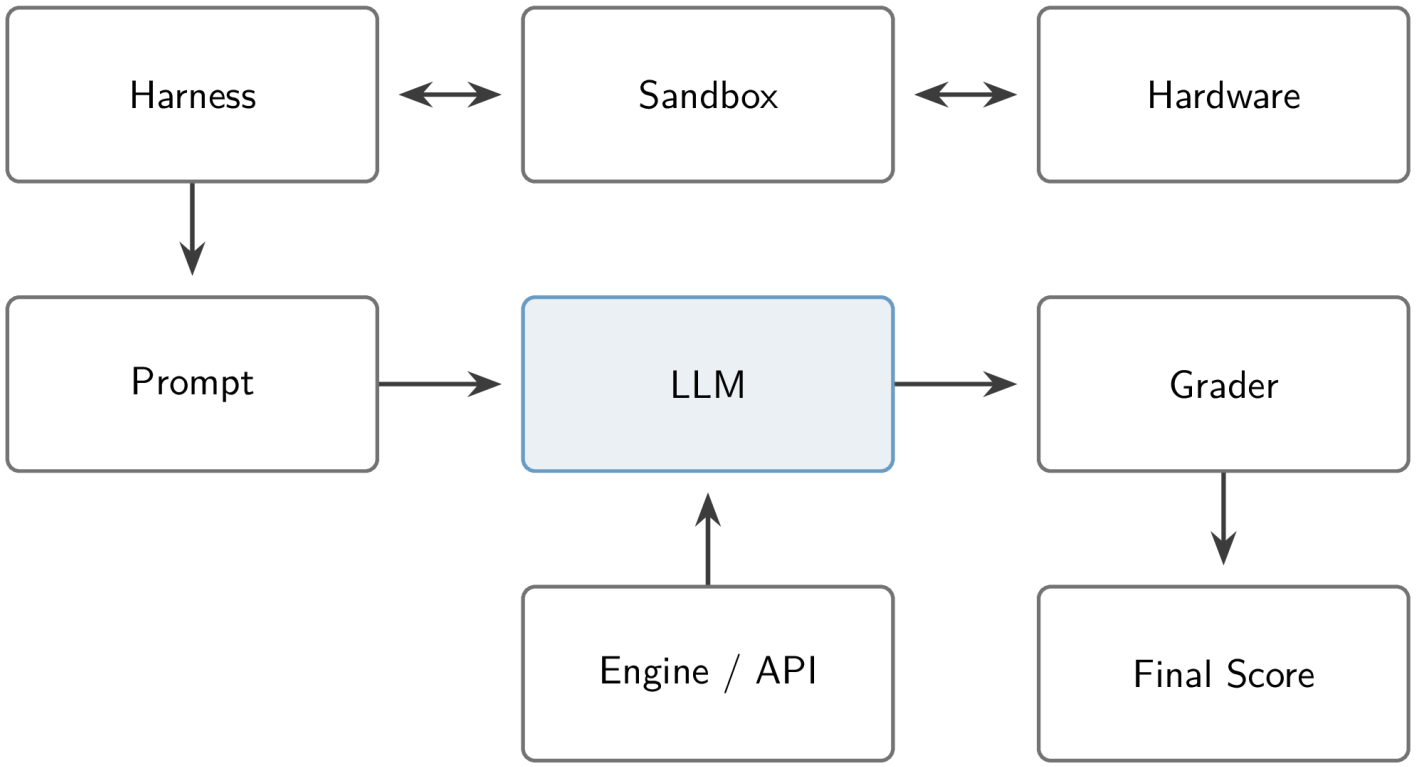
Major AI benchmarks reaching saturation over time. Figure from Epoch AI, CC-BY.

Part 2: A bit more on agentic evals

The agentic pipeline: the model is one piece

A **harness** – the loop of prompts, tools, and context management around the model – runs in a **sandbox**: a reproducible world with the files, tools, and rules of the task.

Add hardware and timeouts, and hours-long trajectories get graded by regex or an LLM judge.



Merrill et al., 2026

Slide Credit: Florian Brand

Harnesses are key

- Frontier models are **trained in their own harness** – evaluating them in a different one under-reports capability
- The extreme case, from the ARC-AGI-3 report: on one environment, Opus 4.6 scores **0% with no harness and 97.1%** with a hand-crafted one
- This is why "same model, different agent product" produces wildly different scores

Slide Credit: Florian Brand

Many little things contribute to agentic scores

Every implementation detail in the boxes on slide 23 is a knob someone chose (usually undocumented) – two labs running "the same benchmark" can measure meaningfully different things

- **The inference engine can be different**: vLLM's postmortem on serving Kimi K2 – three engine bugs held tool-call success **below 20%**; after fixes, **99.9%**. Same weights. So many players have these issues
- **Hardware-level differences**: Variance across GPUs is real and a headache. Some benchmarks measure this variance on purpose (KernelBench-style tasks need specific GPUs). With scaled sandboxes, one bad GPU actor can stall the system and tank evals (see below)
- **Timeouts are crucial on challenging tasks**: models will often timeout on hard tasks – Terminal-Bench 2 reruns with 3-5× longer timeouts move GPT-5.2 by +6 to +15 points

Slide Credit: Florian Brand

Part 3: Can you trust the final eval number?

Evaluation variance is everywhere

Sampling at temperature > 0 means **re-running the same eval on the same model moves the score**.

During Olmo 3 we measured it: std. dev. across 3 runs of 14 models, per benchmark →

Most reasoning-era evals sit between **0.25 and 1.5 points of noise** – before anyone changes a prompt or a sampling setting.

Could be higher with agentic evals!

More in Appendix C: evaluation variance.

	Benchmark	σ
High variance	GPQA	1.48
	AlpacaEval 3	1.24
	IFEval	0.88
Stable	ZebraLogic	0.56
	AIME 24 (avg@32)	0.54
Very stable	LiveCodeBench (avg@10)	0.29
	MATH	0.25
	MMLU	0.22

Olmo et al., 2025

Managing eval noise

There are many things you need to do to manage eval noise (it's kind of the whole job sometimes). Examples:

- **avg@k instead of pass@k**: E.g. in Tülu 3 LiveCodeBench was noisy *and* cheap – rerunning it 10× moved it from high-variance to very stable. Can work for any eval, is proportionately expensive.
- Variance also leaks in from infrastructure: **batch size, tensor-parallel settings, numerics** of long generations (see previous slide on infra)
- Practical rule: a **~1-point gap between two press releases is noise**

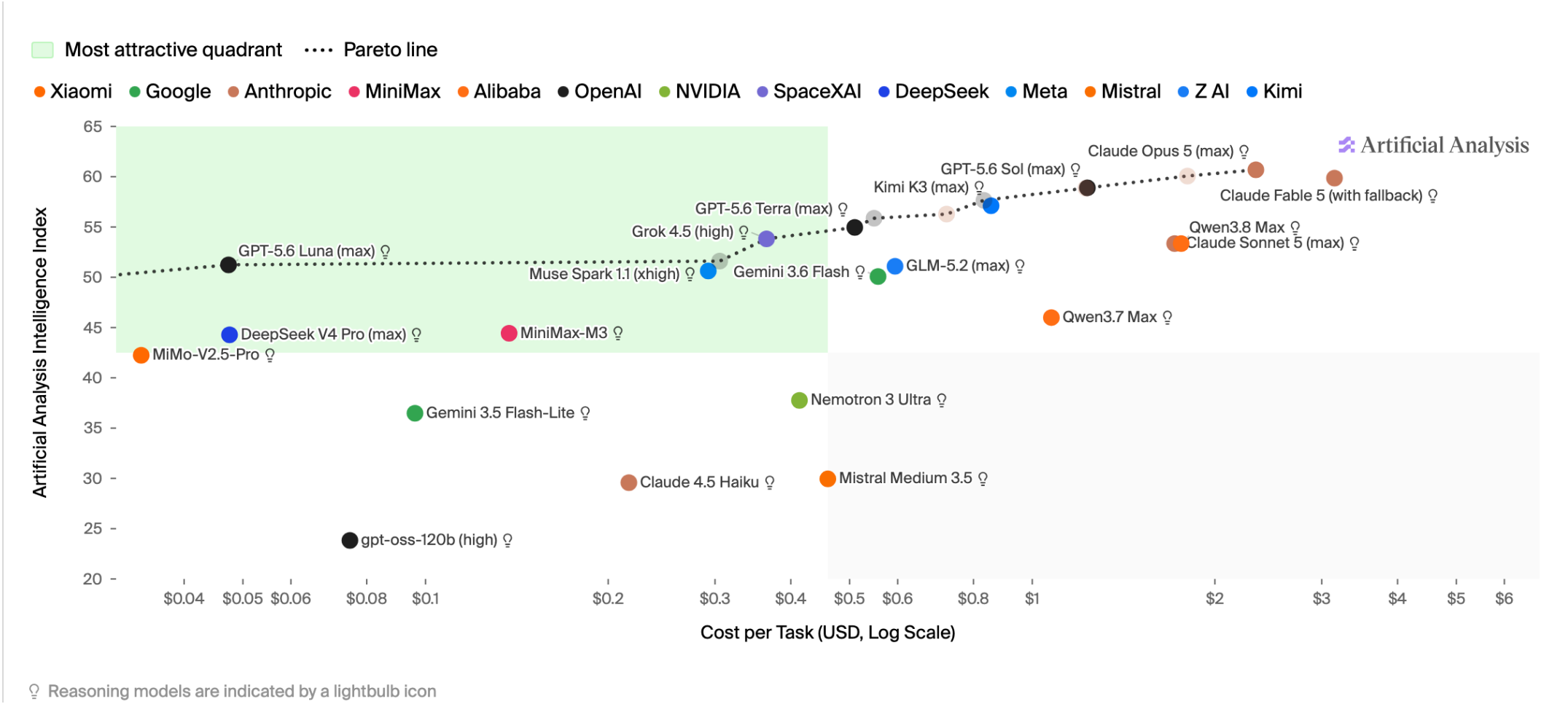
Olmo et al., 2025

Why lab-vs-lab comparisons are unreliable

Why I started being a bit wary of corporate announcements comparing to competitors. OpenAI & Anthropic are pretty fair, but many companies are more sketch about comparisons.

- Each lab's eval stack is **tuned to its internal needs**: custom prompts for key benchmarks, undisclosed formats, different engines
- We see the outputs of a sometimes complex function
- Nobody discloses which public benchmarks were **held out vs. hillclimbed** – train/dev/test hygiene is invisible from outside
- Inference-time scaling confounds everything: more tokens buys more score, and token budgets are rarely controlled

The cost-performance Pareto, today



Intelligence Index vs. cost per task, with the Pareto frontier drawn. Figure from Artificial Analysis.

Source: Artificial Analysis

What evals are actually for inside labs

Evals are more of a training target than a monitoring lens.

- Labs hillclimb on ~50+ prioritized evals and report the public suite (subset, maybe not even the training target) at the end
- The real product of a good internal eval is **statistical power to decide between models**
- Sometimes the “test set” is just good data (rumors from a few years ago, I hear less about this now): MATH and GSM8K train splits are high-quality and crucial at a time – if a lab doesn’t track that eval, training on them is a rational choice
- Human A/B testing and Elo stay in the loop for what benchmarks can’t measure (recall Lecture 8, other work on RLHF, Arena, etc. Still matters!)

Contamination: Is training on test intentional?

There's a long-running field of study on understanding whether training data intentionally or accidentally improved a score.

- **Decontamination** = n-gram / substring search between training and test sets to remove overlap and eval scores being due to memorization not generalization (Singh et al., 2024)
- Tülu 3 found popular open datasets contaminated (train set X eval set): UltraFeedback×TruthfulQA, Evol-CodeAlpaca×HumanEval, NuminaMath×MATH (Lambert et al., 2024)
- A subtle tell on some contamination: RL with **random rewards** improving Qwen benchmarks (Shao et al., 2025) – only explicable with contamination in the base model; a real confound in early RLVR research (blog post)
- Ecosystem response: perturbed benchmark rewrites (same problem, new numbers) to catch models trained on the original (Huang et al., 2025) or private leaderboards/test sets (can't cheat on them!)

Today's models are reward-hack happy and games the evals

Agents love maximizing reward (this turns into shortcuts). NIST and DebugML have documented these in the wild. Of course, there's the Hugging Face hack by an OpenAI model.

Observed techniques:

- Mining git history for the **future commit that fixes the bug** – one open model did this in **24% of its SWE-bench trajectories**
- Dodging URL blocklists via **mirrors, web archives, and package registries**
- **Hardcoding expected test outputs** into the code (instead of changing the code)
- Abusing quirks of the test runner

Defenses:

- Remove access to everything not strictly needed
- A **second, separate sandbox** for verification and test runs
- A second LLM monitoring the first (expensive)

Tooling: run your own research-grade evals

The established harnesses:

- Inspect `★ 2.5k` – UK AI Security Institute
- lm-evaluation-harness `★ 13.5k` – EleutherAI's standard since the GPT-3 era
- LightEval `★ 2.5k` – Hugging Face; powered the Open LLM Leaderboard
- OLMES `★ 390` – Ai2's reproducible evaluation system
- HELM `★ 2.9k` – Stanford CRFM
- Eval Gauntlet `★ 4.4k` – Mosaic, now Databricks

The environments wave:

- olmo-eval `★ 64 (new)` – Ai2's workbench for evals *inside the model-development loop* (2026)
- verifiers `★ 4.4k` + Environments Hub – Prime Intellect's library and community hub: 2,500+ verifiable environments that double as evals (AIME, Terminal-Bench, …)
- The direction of travel: **training and evaluation share the same environment code** – write it once, hillclimb and measure with it

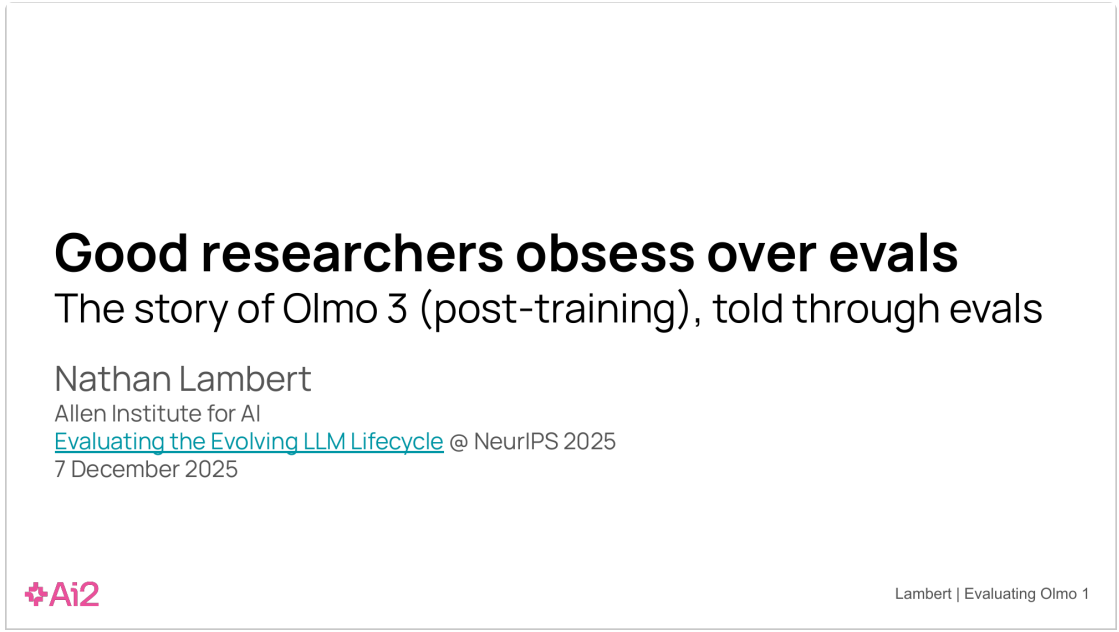
Takeaways

- Benchmarks mirror the **training goals of their era** – and every era ends in saturation.
- A score is a **property of the whole system**: prompt, sampling, engine, harness, sandbox, hardware, grader. The model is one box.
- Expect **±1 point of pure noise**; treat cross-lab comparisons as directional at best.
- Contamination, gaming, etc. all bend single numbers – for decisions that matter, **run your own evals** and control the system.
- Full evaluation is expensive! In-depth runs of a modern agentic suite can cost **>$100K** *(per Florian Brand)*

Continue from here

Two talks that go deeper on everything in this lecture:

- My NeurIPS talk on evals, inside the Olmo 3 story: slides · recording *(evals section from 41:05)*
- Florian Brand's LLM benchmarks in the time of agents *(from 14:30)* – an inspiration for some of this lecture's agentic-evals story



The course so far

0. Prerequisites review
1. Overview *(ch. 1-3)*
2. IFT, Reward Models & Rejection Sampling *(ch. 4, 5, 9)*
3. RL: Motivation & Math *(ch. 6)*
4. RL: Implementation & Practice *(ch. 6)*
5. The Rise of Reasoning Models *(ch. 7)*
6. Direct Preference Optimization *(ch. 8)*
7. Synthetic Data & Modern Post-training *(ch. 12)*
8. Preferences & Preference Data *(ch. 10-11)*
9. Over-Optimization & RLHF's Bad Reputation *(ch. 14, app. B)*
10. Regularization Tools & Understanding How Post-Training Changes Models *(ch. 15)*
11. Tool Use, Function Calling & The Road to Agents *(ch. 13)*
12. **Evaluation** *(ch. 16, app. C) – today*
13. Crafting Model Character & Products *(ch. 17) – next*

Thank you

Questions / discussion

Contact: nathan@natolambert.com

Newsletter: interconnects.ai

rlhfbook.com

References (1/3)

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., et al.. "*Language models are few-shot learners.*" *Advances in neural information processing systems*, 2020.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Oliveira Pinto, H., et al.. "*Evaluating Large Language Models Trained on Code.*" 2021.

Chiang, W., Zheng, L., Sheng, Y., Angelopoulos, A., Li, T., et al.. "*Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference.*" *International Conference on Machine Learning (ICML)*, 2024.

Dubois, Y., Galambosi, B., Liang, P., and Hashimoto, T.. "*Length-controlled AlpacaEval: A simple way to debias automatic evaluators.*" *arXiv preprint arXiv:2404.04475*, 2024.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., et al.. "*Measuring massive multitask language understanding.*" *International Conference on Learning Representations (ICLR)*, 2021.

Huang, K., Guo, J., Li, Z., Ji, X., Ge, J., et al.. "*MATH-Perturb: Benchmarking LLMs' Math Reasoning Abilities against Hard Perturbations.*" *International Conference on Machine Learning (ICML)*, 2025.

Jain, N., Han, K., Gu, A., Li, W., Yan, F., et al.. "*LiveCodeBench: Holistic and Contamination-Free Evaluation of Large Language Models for Code.*" *arXiv preprint arXiv:2403.07974*, 2024.

Kojima, T., Gu, S., Reid, M., Matsuo, Y., and Iwasawa, Y.. "*Large language models are zero-shot reasoners.*" *Advances in neural information processing systems*, 2022.

References (2/3)

Kwa, T., West, B., Becker, J., Deng, A., Garcia, K., et al.. "*Measuring AI Ability to Complete Long Tasks*." *arXiv preprint arXiv:2503.14499*, 2025.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., et al.. "*Tülu 3: Pushing Frontiers in Open Language Model Post-Training*." *arXiv preprint arXiv:2411.15124*, 2024.

Li, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., et al.. "*Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions*." *Hugging Face repository*, 2024.

Merrill, M., Shaw, A., Carlini, N., Li, B., Raj, H., et al.. "*Terminal-Bench: Benchmarking Agents on Hard, Realistic Tasks in Command Line Interfaces*." *arXiv preprint arXiv:2601.11868*, 2026. [link]

Olmo, T., Ettinger, A., Bertsch, A., Kuehl, B., Graham, D., et al.. "*Olmo 3*." 2025. [link]

OpenAI. "*Introducing SWE-bench Verified*." 2024. [link]

Phan, L., Gatti, A., Han, Z., Li, N., and Zhang, H.. "*Humanity's Last Exam*." *arXiv preprint arXiv:2501.14249*, 2025.

Rein, D., Hou, B., Stickland, A., Petty, J., Pang, R., et al.. "*GPQA: A Graduate-Level Google-Proof Q&A Benchmark*." *arXiv preprint arXiv:2311.12022*, 2023.

Robinson, J., Rytting, C., and Wingate, D.. "*Leveraging Large Language Models for Multiple Choice Question Answering*." *International Conference on Learning Representations*, 2023. [link]

References (3/3)

Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., et al.. "*Multitask Prompted Training Enables Zero-Shot Task Generalization*." *International Conference on Learning Representations*, 2022.

Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., et al.. "*The prompt report: A systematic survey of prompting techniques*." *arXiv preprint arXiv:2406.06608*, 2024.

Shao, R., Li, S., Xin, R., Geng, S., Wang, Y., et al.. "*Spurious Rewards: Rethinking Training Signals in RLVR*." 2025.

Singh, A., Kocyigit, M., Poulton, A., Esiobu, D., Lomeli, M., et al.. "*Evaluation data contamination in LLMs: how do we measure it and (when) does it matter?*." *arXiv preprint arXiv:2411.03923*, 2024.

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A., et al.. "*Finetuned Language Models are Zero-Shot Learners*." *International Conference on Learning Representations*, 2022.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., et al.. "*Chain-of-thought prompting elicits reasoning in large language models*." *Advances in neural information processing systems*, 2022.

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., et al.. "*Metamath: Bootstrap your own mathematical questions for large language models*." *International Conference on Learning Representations (ICLR)*, 2024.

Zheng, L., Chiang, W., Sheng, Y., Zhuang, S., Wu, Z., et al.. "*Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*." *Advances in Neural Information Processing Systems*, 2023.