AACL 2026

Measurement Science for Language Model Evaluation

aimslab.stanford.edu/tutorial



Sang Truong
Stanford



Serena Wang
UBC



Rodolfo Corona
Harvard



Sanmi Koyejo
Stanford



Rodolfo (Rudy) Corona
Harvard
Presents C1, C4



Sang Truong
Stanford
Built C2, C3



Serena Wang
UBC
Built C5, C6



Sanmi Koyejo
Stanford
Presents the rest

Opening	C1 · C2 · C3	Coffee break	C4 · C5 · C6	Closing and discussion
14:00 · 15 min	14:15 · 75 min	15:30 · 30 min	16:00 · 75 min	17:15 · 15 min

Tutorial materials: aimslab.stanford.edu/tutorial · AACL-IJCNLP 2026, Hengqin, China

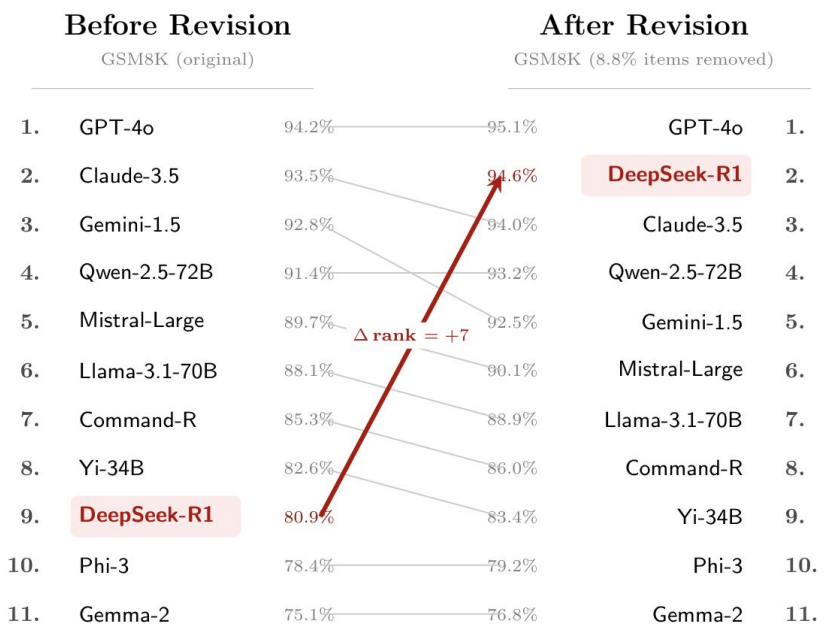
Here’s the plan

- **We will**
  - Take six claims that people who build and use benchmarks often make
  - Ask you to vote on each one before we argue it, and again after
  - Show where each claim holds, where it breaks, and a better claim to make instead
  - Point to the papers if you want more
- **We will not**
  - Survey all of evaluation
  - Say every claim is wrong; most are partly right
  - Get everything right; this area moves fast

Fixing the benchmark changes the leaderboard

- GSM8K (grade-school math word problems): about 5% of questions are invalid (ambiguous wording, wrong answer keys, grading errors)
- After revision, DeepSeek-R1 moves from third-lowest to second place

The models did not change



Truong, …, Koyejo, Fantastic Bugs and Where to Find Them in AI Benchmarks, NeurIPS 2025 D&B (arXiv 2511.16842); Vendrow et al., Do Large Language Model Benchmarks Test Reliability?, 2025 (GSM8K-Platinum); Truong, …, Koyejo, A Measurement Science Roadmap: From Human Assessment to AI Evaluation, arXiv 2026

“Benchmarks built NLP. Why change what works?”

Shared benchmarks drove a decade of progress, and rankings often survive new test sets

What changed

- 29 of 60 widely used LLM benchmarks no longer separate top models
- One score determines deployment, procurement, and policy, across many languages
- Developers train toward the benchmarks



Figure 1: Benchmark saturation over time for popular benchmarks, normalized with initial performance at minus one and human performance at zero.

Kiela et al., Dynabench: Rethinking Benchmarking in NLP, NAACL 2021; Akhtar, Reuel, …, Koyejo, When AI Benchmarks Plateau, ICML 2026 (arXiv 2602.16763)

A benchmark score is a row average of a response matrix

What the score is:

- the average of one row of the response matrix (models × questions; 1 right, 0 wrong, ? not run) under a fixed protocol

What you want to know:

- performance for my users, in my language?
- performance on new tasks?
- performance after developers optimize for it?

	q1	q2	q3	q4	q5	q6	q7
Model A	1	0	1	1	0	1	0
Model B	0	1	0	1	1	0	1
Model C	1	0	1	0	1	1	0
Model D	1	1	0	1	0	?	1
Model E	0	1	1	?	?	?	?

Salaudeen, …, Koyejo, Measurement to Meaning: A Validity-Centered Framework for AI Evaluation, NeurIPS 2025 LLM Eval Workshop; Truong, …, Koyejo, A Measurement Science Roadmap, arXiv 2026

The same benchmark result is rarely reported the same way

101,843 reported results across 5,816 models and 635 benchmarks

96.5%

of (model, benchmark, metric) records lack information needed to reproduce them

98.2%

of model–benchmark pairs are reported by only one party

51.9%

of results reported by more than one party disagree by more than 5%

A score is the output of one protocol, and is usually unverified

Ghosh, Reuel, …, Koyejo, Evaluation Cards: An Interpretive Layer for AI Evaluation Reporting, NeurIPS 2026 (arXiv 2606.09809)

Two common evaluation errors need opposite fixes

Sampling error

Right target, loose estimate

Cause: too few items, noise, how items were chosen

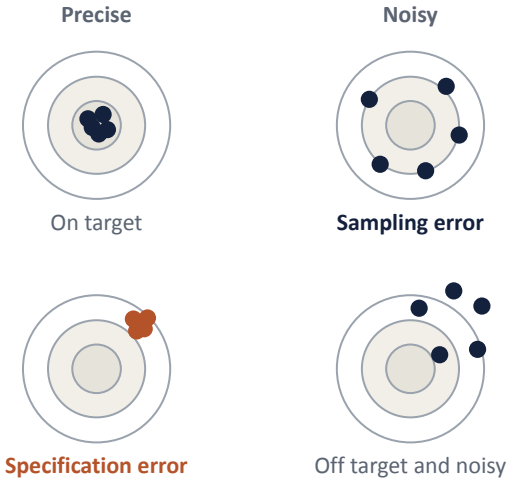
Fix: more items, better design

Specification error

The items measure something else

Cause: wrong construct (the intended capability, e.g. math reasoning) or wrong population

More data estimates the wrong thing more precisely



Messick, Validity, in Educational Measurement (3rd ed.), 1989; Salaudeen, …, Koyejo, Measurement to Meaning, NeurIPS 2025 LLM Eval Workshop

A third error: developers optimize for the score itself

Error	What goes wrong	Do more questions fix it?
Sampling	Too few or unlucky questions	Yes
Specification	The questions measure something other than the claim	No
Strategic	Developers optimize for the score itself	No

Today: Part I is mostly sampling and specification error. Part II is strategic error.

Salaudeen, …, Koyejo, Measurement to Meaning: A Validity-Centered Framework for AI Evaluation, NeurIPS 2025 LLM Eval Workshop; Manheim and Garrabrant, Categorizing Variants of Goodhart's Law, arXiv 1803.04585

Six claims practitioners make

	Claim	Speaker	Main error source
C1	Averaging safety benchmarks robustly measures safety	Rudy	Specification
C2	To compare models fairly, every model must answer the same full test	Sanmi	Sampling
C3	A well-translated benchmark measures the same thing in every language	Sanmi	Specification
C4	The leaderboard winner will fit my product best	Rudy	Specification
C5	Evaluation gaming is mostly cheating or data leakage	Sanmi	Strategic
C6	A private, uncontaminated test set stops gaming	Sanmi	Strategic

Each claim: vote → what holds → where it breaks → a better claim → vote again

Part I (C1–C4): measurement theory Part II (C5–C6): incentive-aware robust evaluation

Part I

Measurement Theory

When does a benchmark measure what we claim?

Averaging safety benchmarks robustly measures safety

Hands up if you agree.

C1 · Will averaging safety benchmarks robustly measure safety?

- **Intuition:** Averaging will cancel benchmark quirks, leaving a stronger signal of the capability I'm after (e.g. safety).

- **Key idea:** A label over a group of test items implicitly makes a claim about what that grouping measures.

C1 · Averaging amplifies what is shared

Benchmark score = General capability + label-specific ability + benchmark-specific ability + noise

C1 · Averaging amplifies what is shared

Benchmark score = General capability + label-specific ability + benchmark-specific ability + noise

E.g. Fluency, grammar, task recognition, etc.

C1 · Averaging amplifies what is shared

Benchmark score = General capability + **label-specific ability** + benchmark-specific ability + noise

The capability we care about, e.g. “safety”, “math”, etc.

C1 · Averaging amplifies what is shared

Benchmark score = General capability + label-specific ability + **benchmark-specific ability** + noise

Specific capabilities not directly related to target capability (e.g. “web search ability”) for a ”safety” benchmark.

C1 · Averaging amplifies what is shared

Benchmark score = General capability + label-specific ability + benchmark-specific ability + **noise**

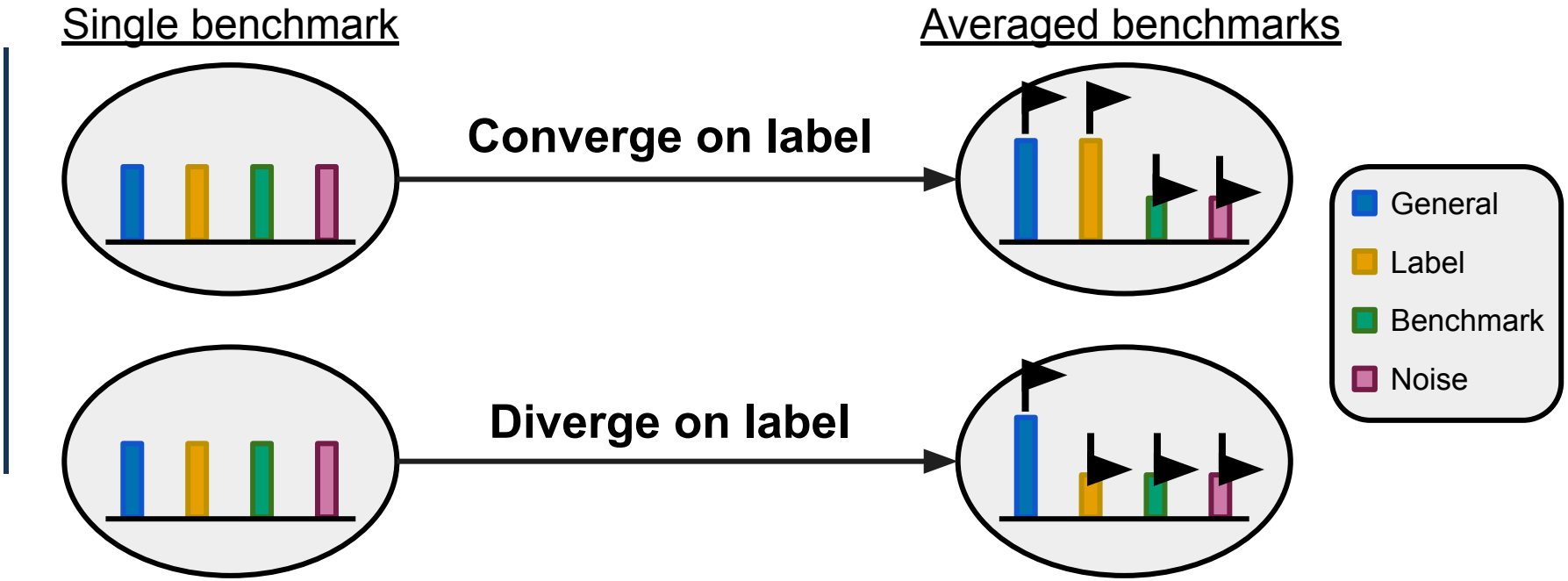
Sampling error, e.g. incorrect label, malformed input, etc.

C1 · Averaging amplifies what is shared

Benchmark score = General capability + label-specific ability + benchmark-specific ability + noise

- Averaging keeps whatever benchmarks *have in common*.
- We want to robustly measure **label-specific** ability.
- Label-specific ability only survives if averaged benchmarks actually **converge** on it.

C1 · Averaging amplifies what is shared

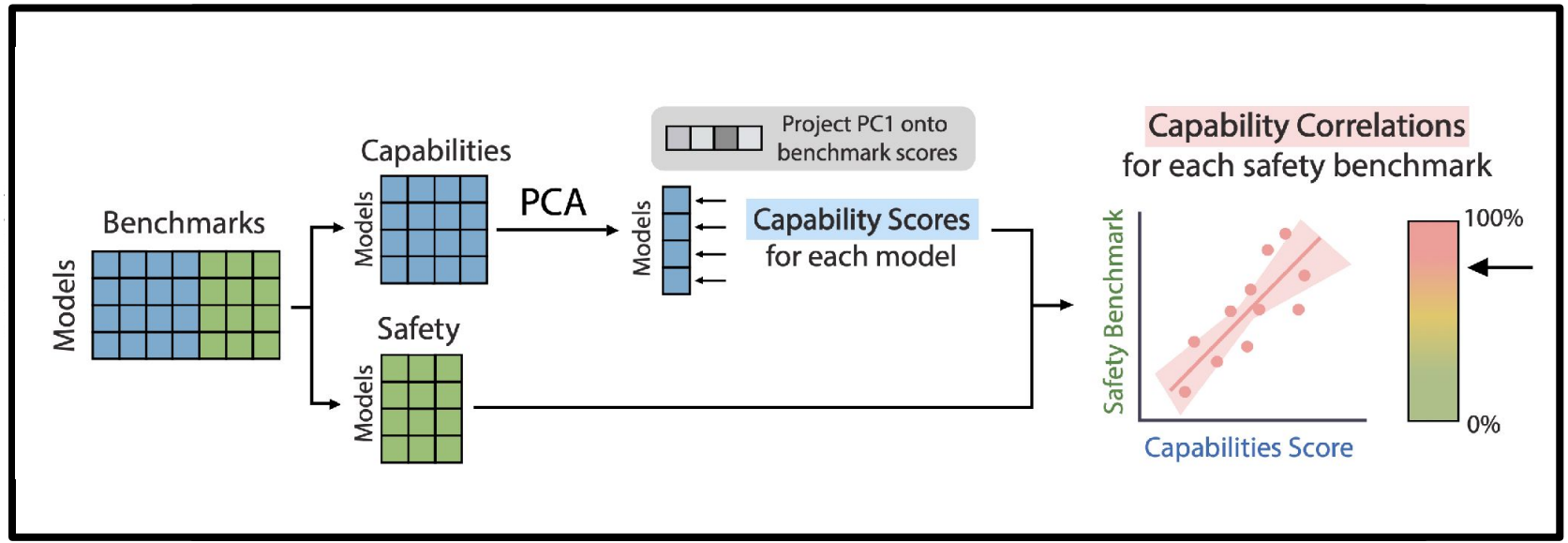


C1 · Averaging amplifies what is shared

Key question: How correlated are rankings induced by a benchark, to those induced by *general capabilities* and *compute*?

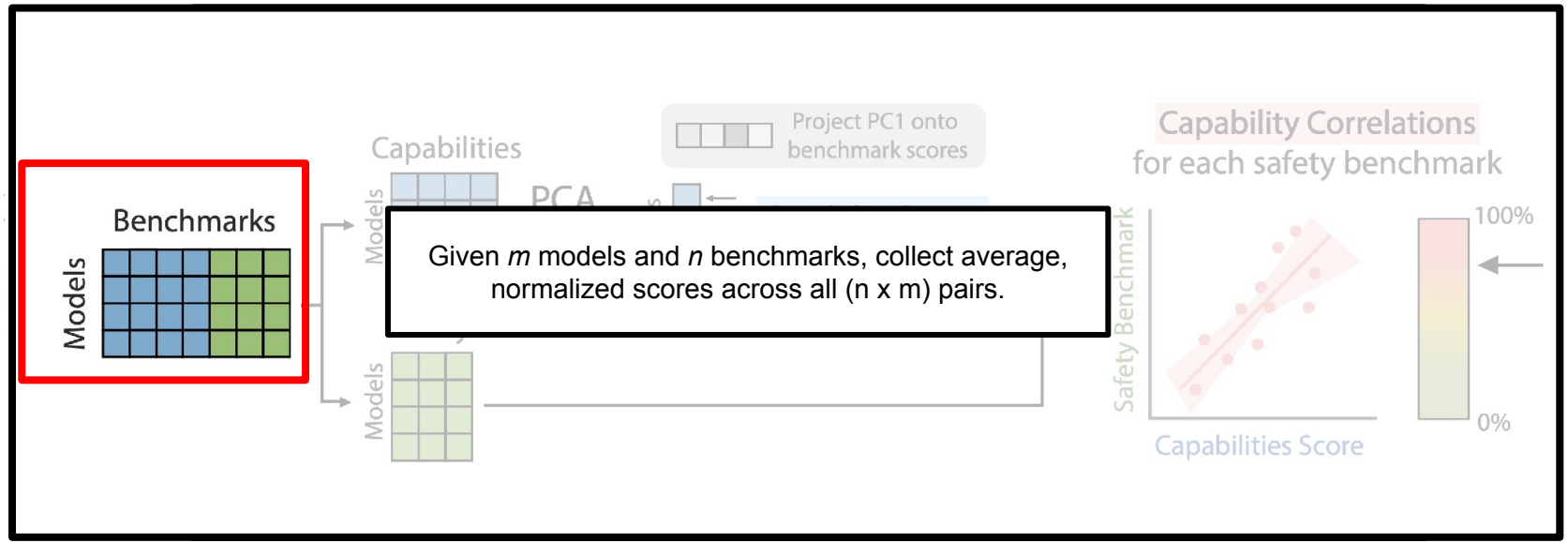
Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared



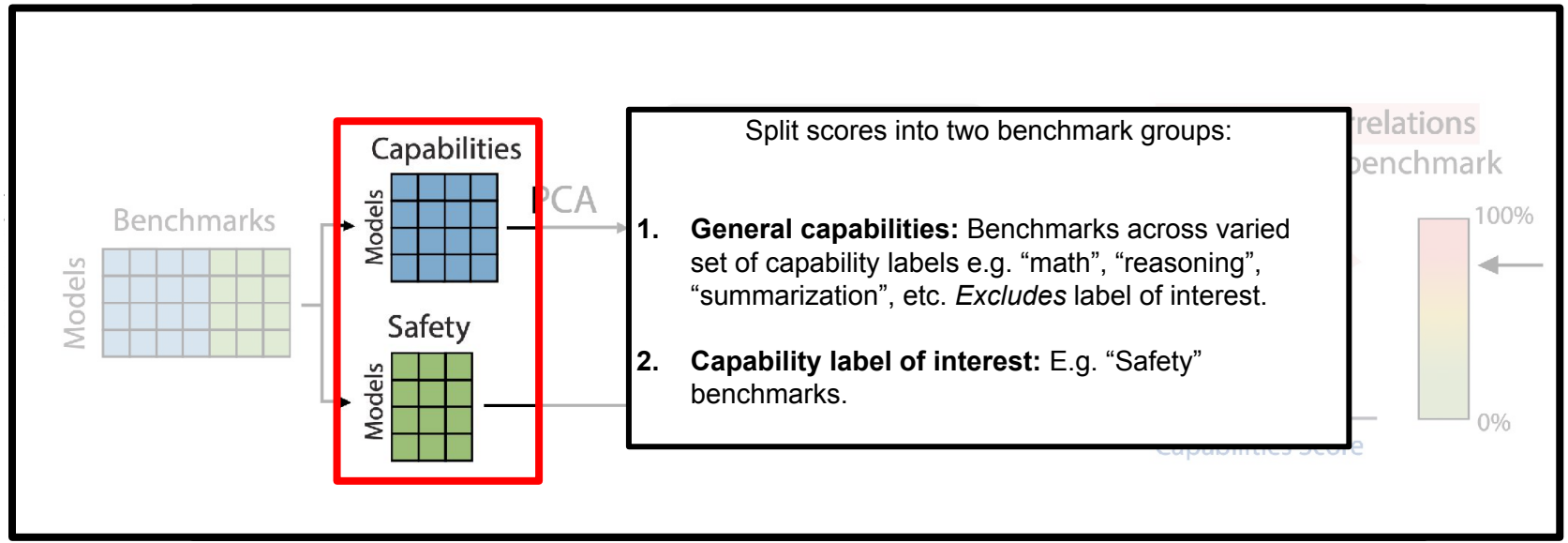
Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared



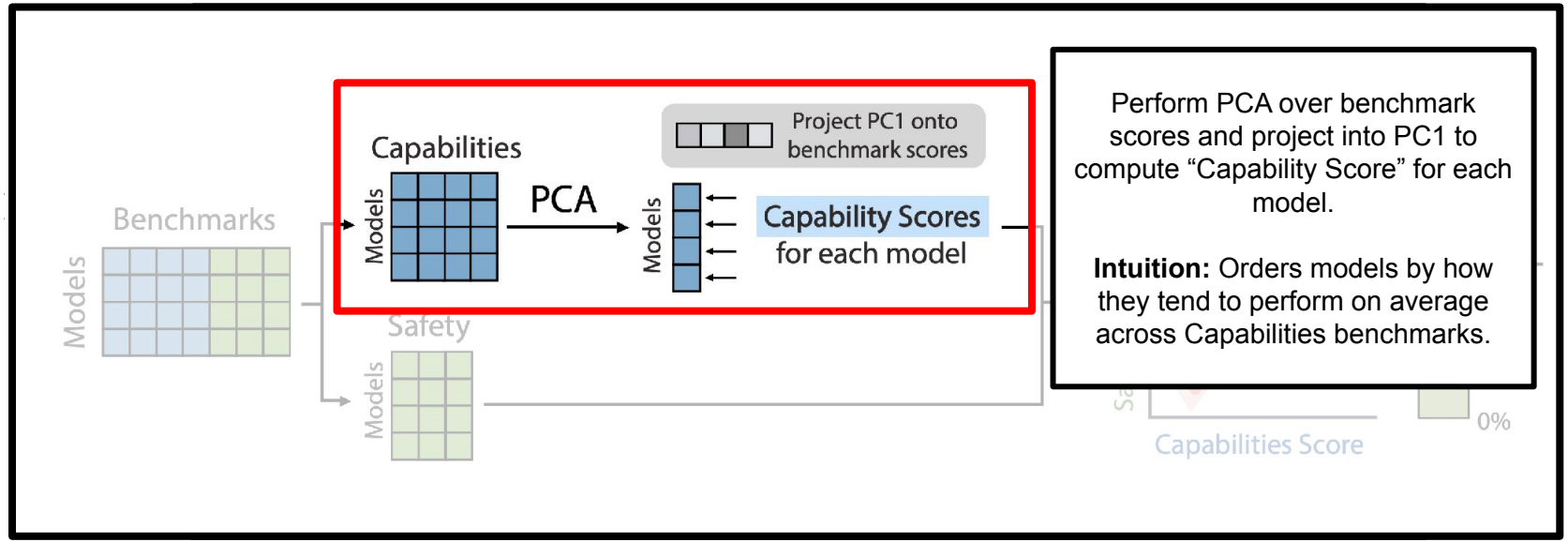
Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared



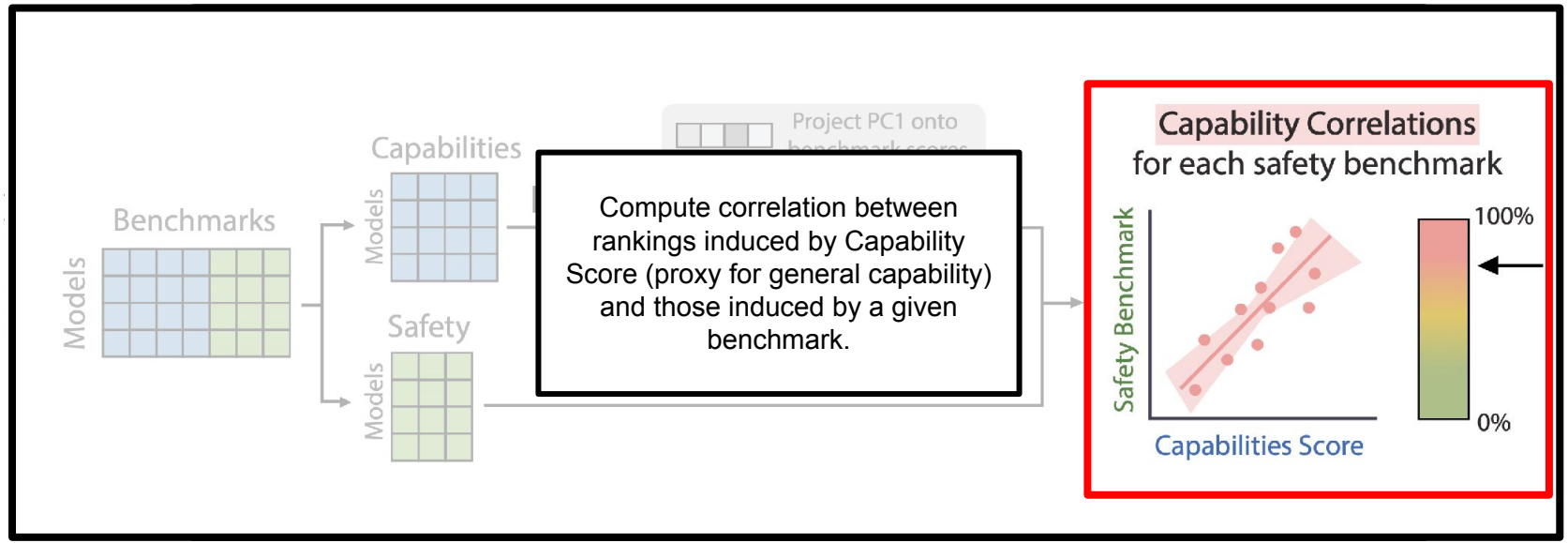
Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared



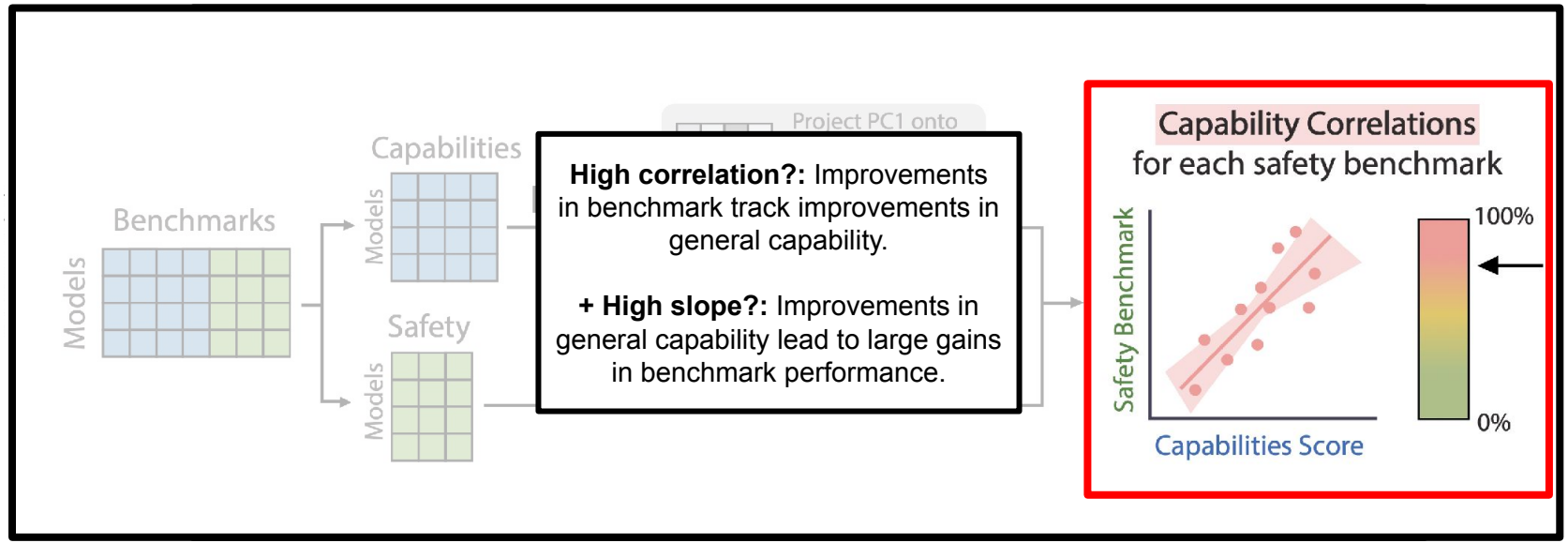
Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared



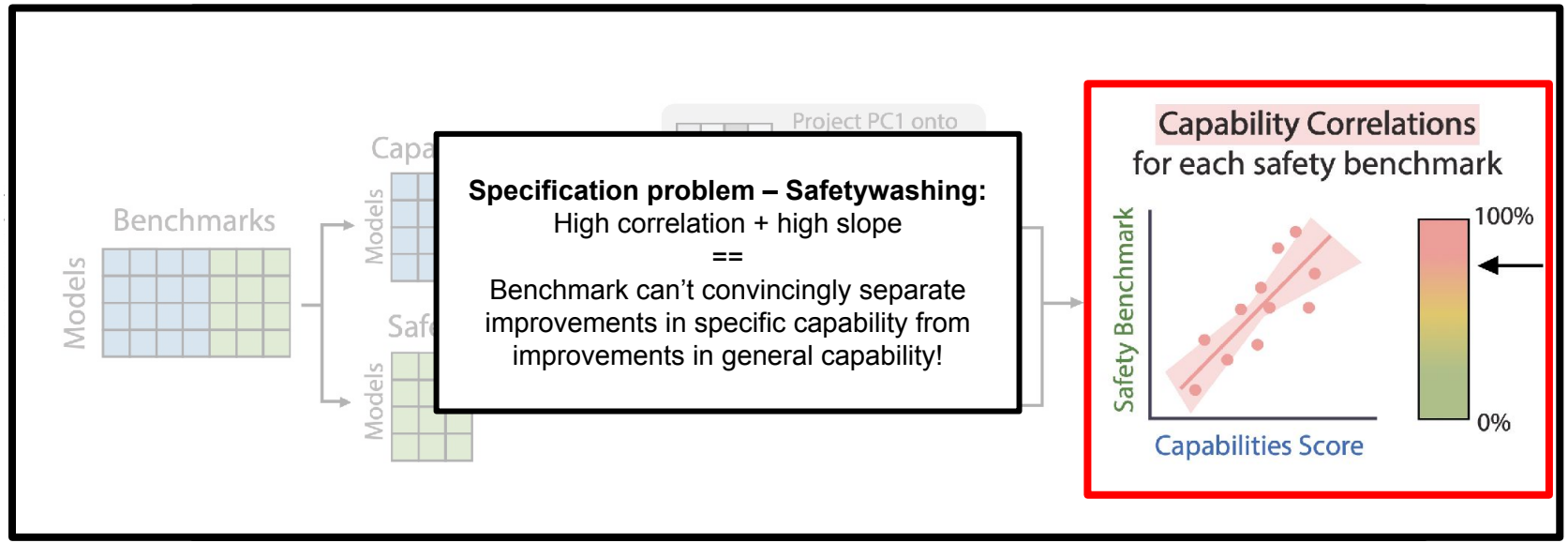
Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared



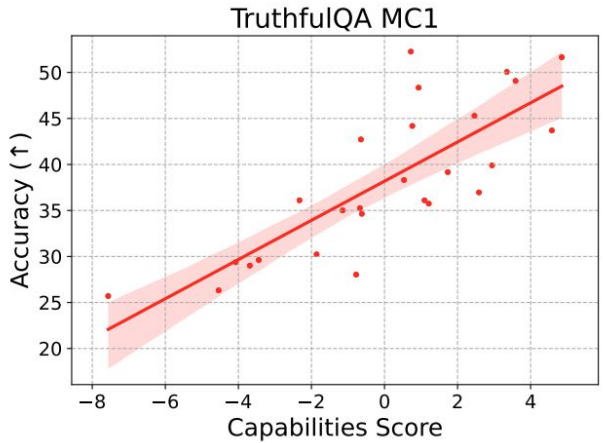
Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared



Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared

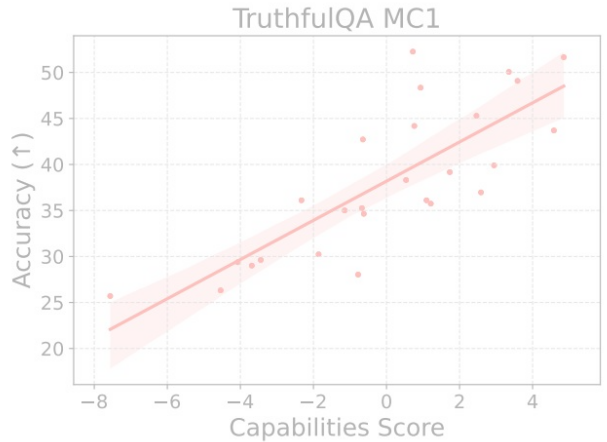


Scalable Oversight Evaluation	Capabilities Correlation	Compute Correlation
GPQA	77.7%	77.2%
QuALITY	88.8%	91.6%

Some benchmarks, e.g. in “Safety”, highly correlated with **general capability**.

Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared

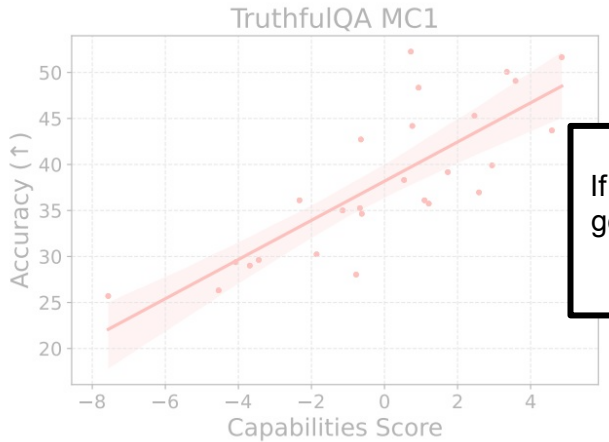


Scalable Oversight Evaluation	Capabilities Correlation	**Compute Correlation**
GPQA	77.7%	77.2%
QuALITY	88.8%	91.6%

Also correlated with rankings induced by compute (FLOPs), another proxy for *general capability*.

Ren et al (NeurIPS 2024)

C1 · Averaging amplifies what is shared



If I average these benchmarks, do I get more robust reading of “safety”, or “general capability”?

Scalable Oversight Evaluation	Capabilities Correlation	Compute Correlation
	77.7%	77.2%
	88.8%	91.6%

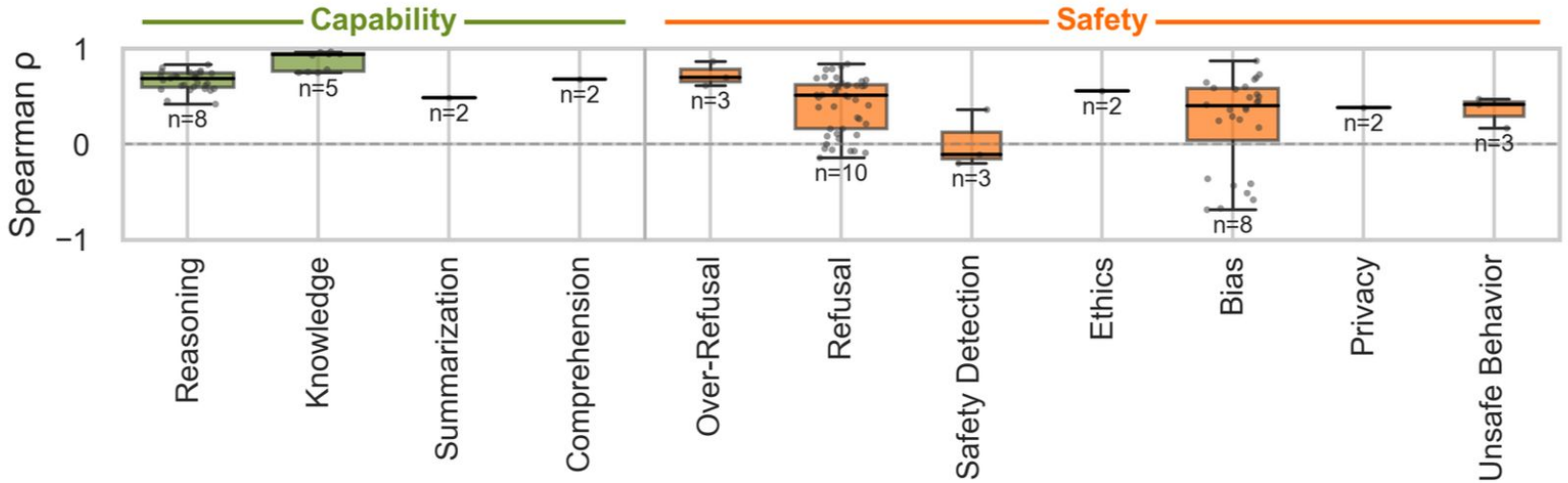
Ren et al (NeurIPS 2024)

C1 · When does averaging give me the label?

Two key ingredients for benchmarks with the same label:

1. **Convergence:** Do they agree with each other?
2. **Discrimination:** Do they agree more with *each other* than with benchmarks from a *different* label (e.g. “reasoning” vs. “safety”)?

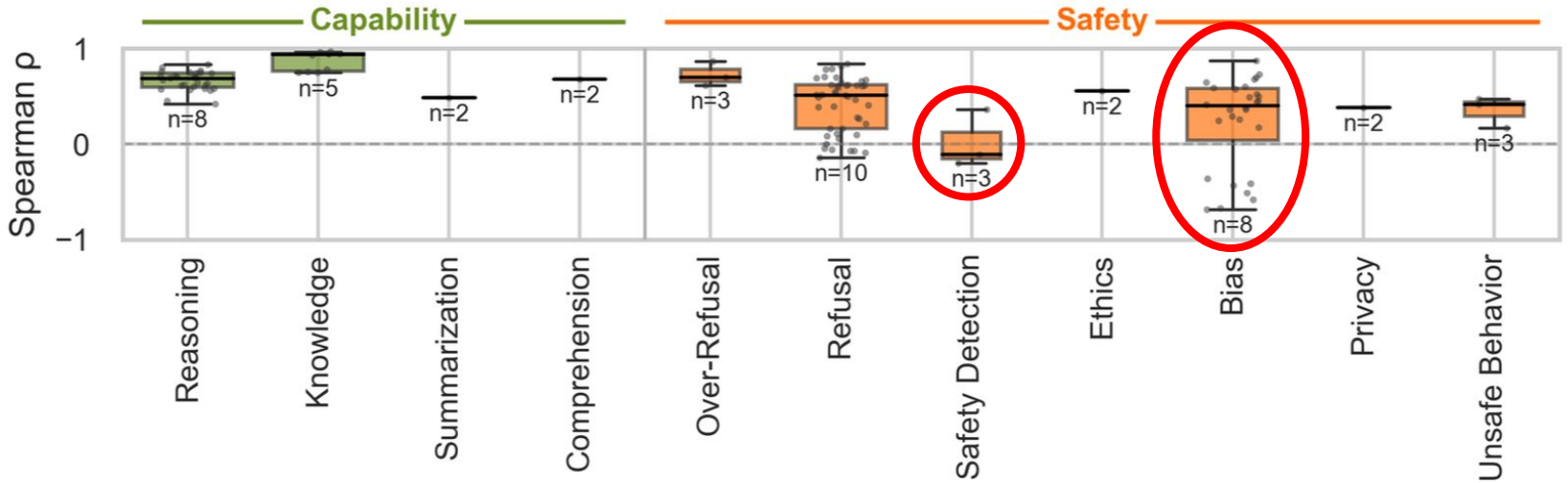
C1 · When does averaging give me the label?



Compute pair-wise correlations across benchmark model rankings labeled with the “same” capability (e.g. “refusal”)

Desai et al. (COLM 2026)

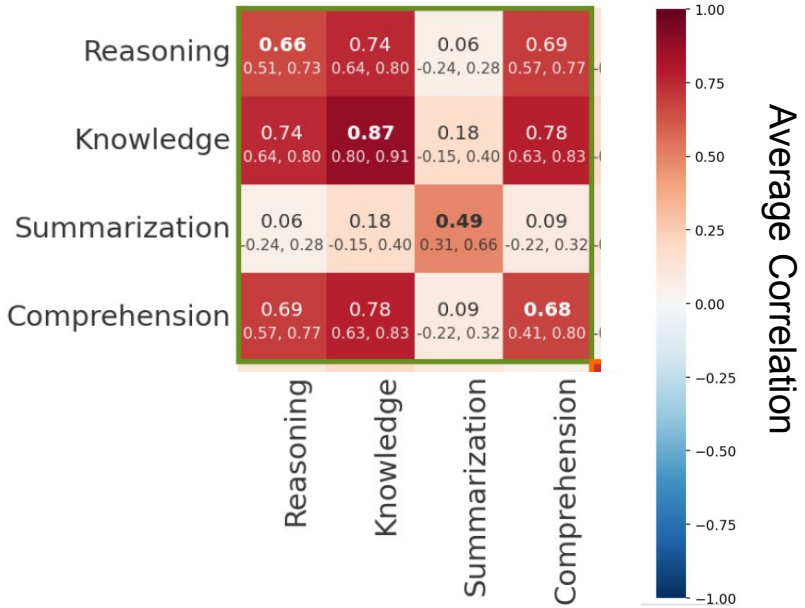
C1 · When does averaging give me the label?



Benchmarks under “safety” labels do not consistently converge.

Desai et al. (COLM 2026)

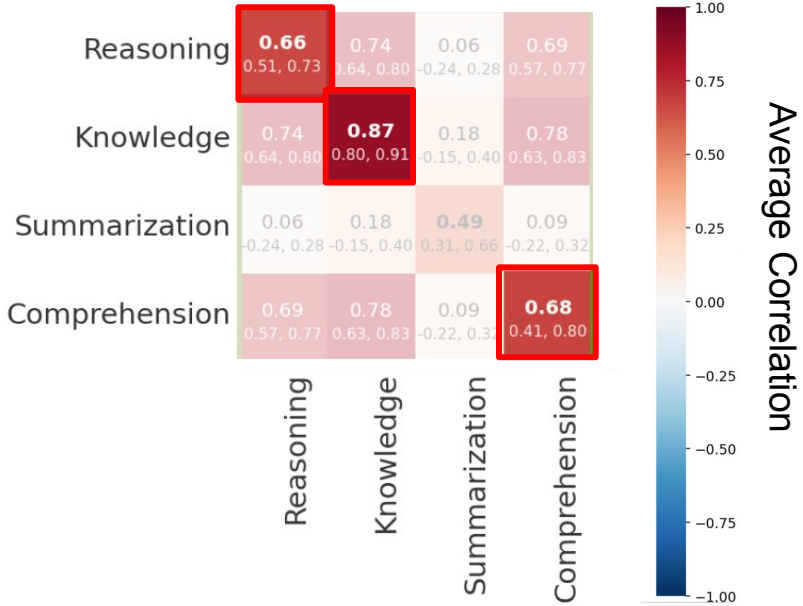
C1 · When does averaging give me the label?



What about other capabilities?

Desai et al. (COLM 2026)

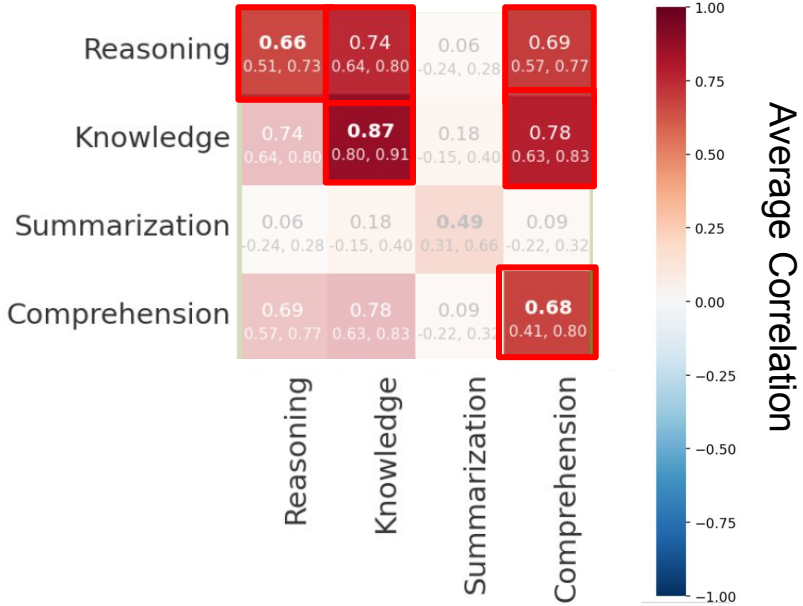
C1 · When does averaging give me the label?



Benchmarks' rankings in *"reasoning"*, *"knowledge"*, and *"comprehension"* **converge** with each other.

Desai et al. (COLM 2026)

C1 · When does averaging give me the label?

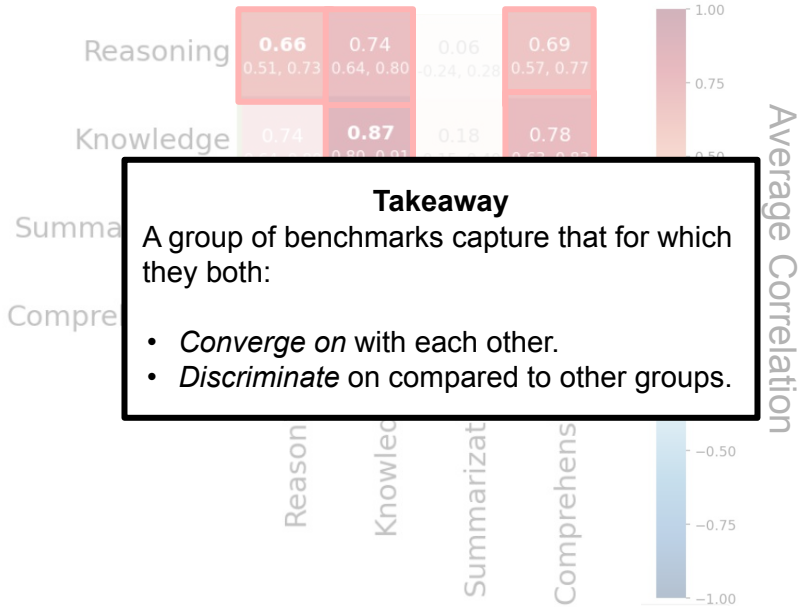


Discrimination Failure: Also highly correlated with benchmarks under other capability labels!

Are they measuring different things?

Desai et al. (COLM 2026)

C1 · When does averaging give me the label?



Takeaway

A group of benchmarks capture that for which they both:

- *Converge on* with each other.
- *Discriminate* on compared to other groups.

Desai et al. (COLM 2026)

C1 · What about averaging across benchmarks for the same application?

Case study: Healthcare applications; what does “healthcare” mean?

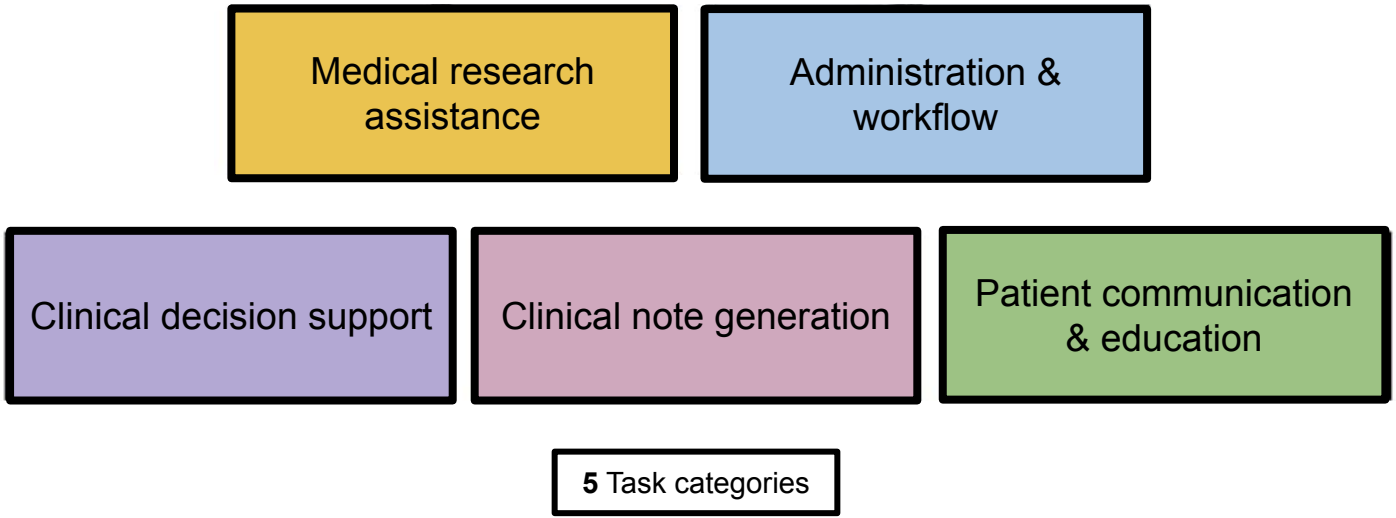
Bedi et al. (Nature Medicine 2026)

C1 · What about averaging across benchmarks for the same application?



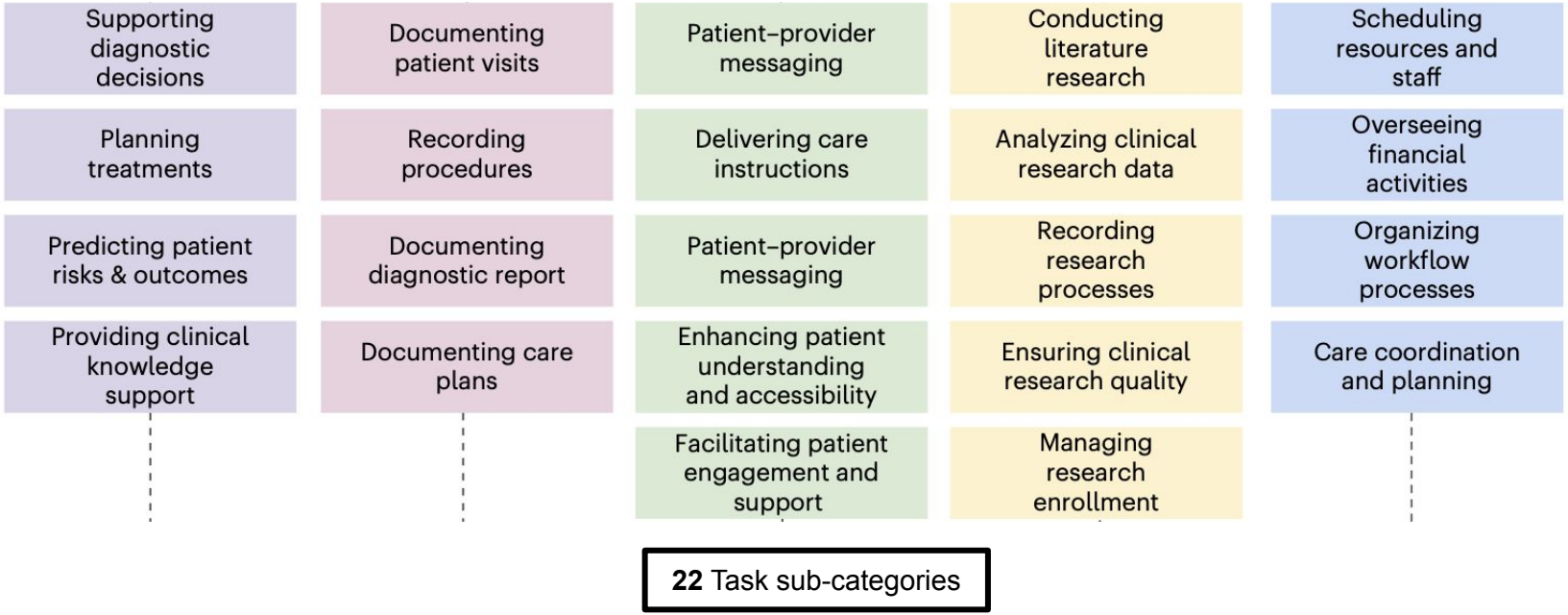
Bedi et al. (Nature Medicine 2026)

C1 · What about averaging across benchmarks for the same application?



Bedi et al. (Nature Medicine 2026)

C1 · What about averaging across benchmarks for the same application?



Bedi et al. (Nature Medicine 2026)

C1 · What about averaging across benchmarks for the same application?

Table 1 | Comparison of performance of frontier models across 37 MedHELM benchmarks, sorted by descending win rate

Model (snapshot)	Win rate↑	Win s.d.↓	Macro-avg↑	s.d.↓
DeepSeek R1	**0.66**	0.11	0.75	0.21
o3-mini (2025-01-31)	**0.66**	0.15	**0.77**	**0.18**
Claude 3.5 Sonnet (20241022)	0.63	0.15	0.74	0.20
Claude 3.7 Sonnet (20250219)	0.63	0.15	0.73	0.21
GPT-4o (2024-05-13)	0.58	0.17	0.74	**0.18**
Gemini 2.0 Flash	0.43	0.18	0.71	0.20
GPT-4o mini (2024-07-18)	0.37	0.17	0.71	0.19
Llama 3.3 Instruct (70B)	0.30	0.15	0.70	0.21
Gemini 1.5 Pro (001)	0.23	**0.08**	0.68	0.20

Bedi et al. (Nature Medicine 2026)

C1 · What about averaging across benchmarks for the same application?

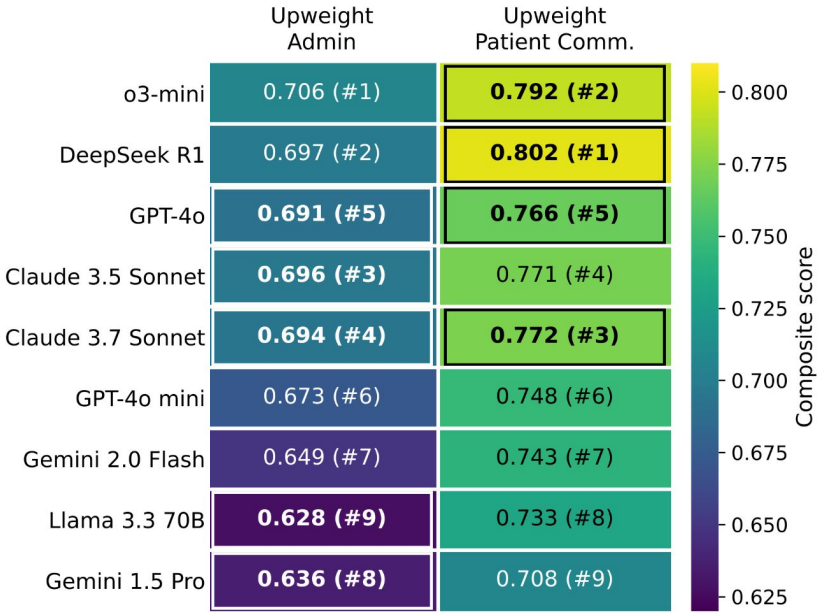
Table 1 | Comparison of performance of frontier models across 37 MedHELM benchmarks, sorted by descending win rate

Model (snapshot)	Win rate↑	Win s.d.↓	Macro-avg↑	s.d.↓
DeepSeek R1	**0.66**	0.11	0.75	0.21
o3-mini (2025-01-31)	**0.66**	0.15	**0.77**	**0.18**
Claude 3.5 Sonnet (20241022)	0.63	0.15	0.74	0.20
Claude 3.7 Sonnet (20250219)	0.63	0.15	0.73	0.21
GPT-4o (2024-05-13)	0.58	0.17	0.74	**0.18**
Gemini 2.0 Flash	0.43	0.18	0.71	0.20
GPT-4o mini (2024-07-18)	0.37	0.17	0.71	0.19
Llama 3.3 Instruct (70B)	0.30	0.15	0.70	0.21
Gemini 1.5 Pro (001)	0.23	**0.08**	0.68	0.20

Is this “health” performance?

Bedi et al. (Nature Medicine 2026)

C1 · What about averaging across benchmarks for the same application?

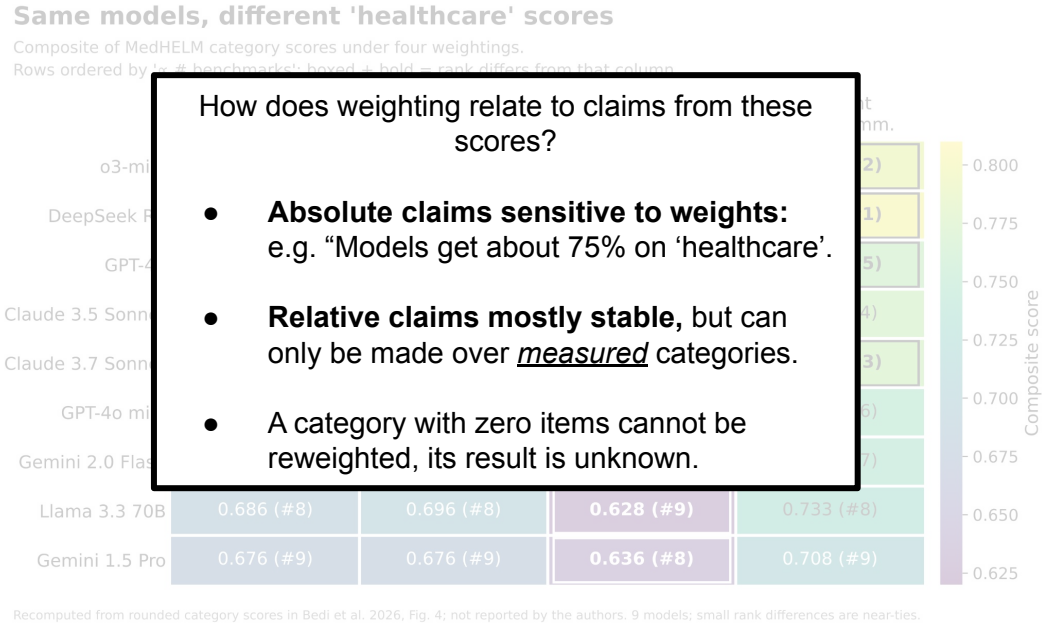


target category weighted 4× (50%), others 12.5% each.

Weighting decisions define which items matter most, and can change score properties!

Recomputed from Bedi et al. (Nature Medicine 2026) Figure 4.

C1 · What about averaging across benchmarks for the same application?

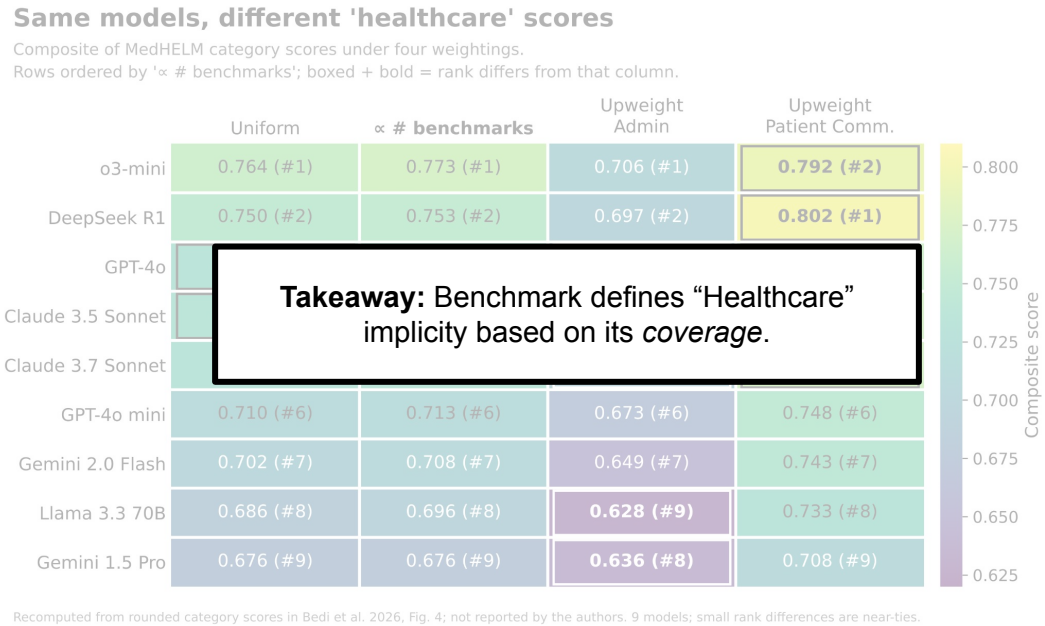


How does weighting relate to claims from these scores?

- **Absolute claims sensitive to weights:** e.g. “Models get about 75% on ‘healthcare’.
- **Relative claims mostly stable,** but can only be made over ***measured*** categories.
- A category with zero items cannot be reweighted, its result is unknown.

Bedi et al. (Nature Medicine 2026)

C1 · What about averaging across benchmarks for the same application?



Bedi et al. (Nature Medicine 2026)

C1 · So what can averaging give me?

Original claim: Averaging benchmarks labeled for capability X will give me a more robust reading of that capability.

What holds: Averaging benchmarks provides a more reliable reading of what they **share**.

Breaks when: Label *underspecified*, benchmarks either don't converge/discriminate on capability label or they fail to cover the intended application.

Better claim – averaging benchmarks gives a more robust read of:

- The *labeled capability* if the benchmarks for that label can converge with each other and discriminate with benchmarks of different labels.
- The *labeled application* when they cover it.

C1 · Averaging safety benchmarks robustly measures safety

Vote again: hands up if you still agree.

C2 · Setup: Recall the response matrix

	q_1	q_2	q_3	q_4	q_5	q_6	q_7
Model A	1	0	1	?	?	?	?
Model B	?	?	?	1	0	1	?
Model C	?	1	?	0	?	1	?
Model D	1	?	?	?	1	?	0
Model E	?	?	1	?	?	0	1

Each answer can be costly: long multi-step runs, human graders, paid APIs

So each model answers a different few questions

One short shared test, or different questions per model?

Illustration · the response matrix from the opening · rows = models, columns = questions · 1 right, 0 wrong, ? not run

To compare models fairly, every model must answer the same full test

Hands up if you agree.

C2 · To compare models fairly, every model must answer the same full test

Why people believe it: Different questions have different difficulty, so averages over different questions are not comparable.

What we'll show: Put models and questions on one scale. Then different, fewer questions can still compare models. Adaptive exams like the GRE already do this: each test-taker sees different questions, and the scores stay comparable.

C2 · One model, two kinds of test

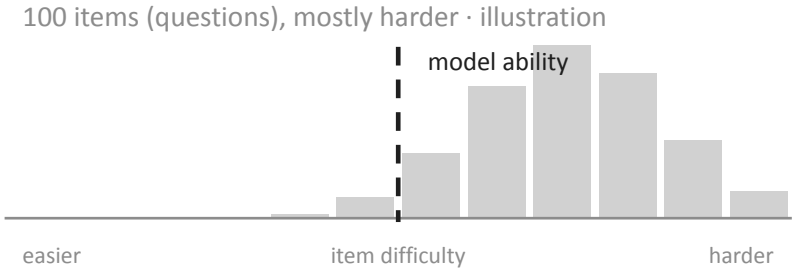
Easier questions



Raw average ≈ 75%

Ability estimate: unchanged

Harder questions



Raw average ≈ 25%

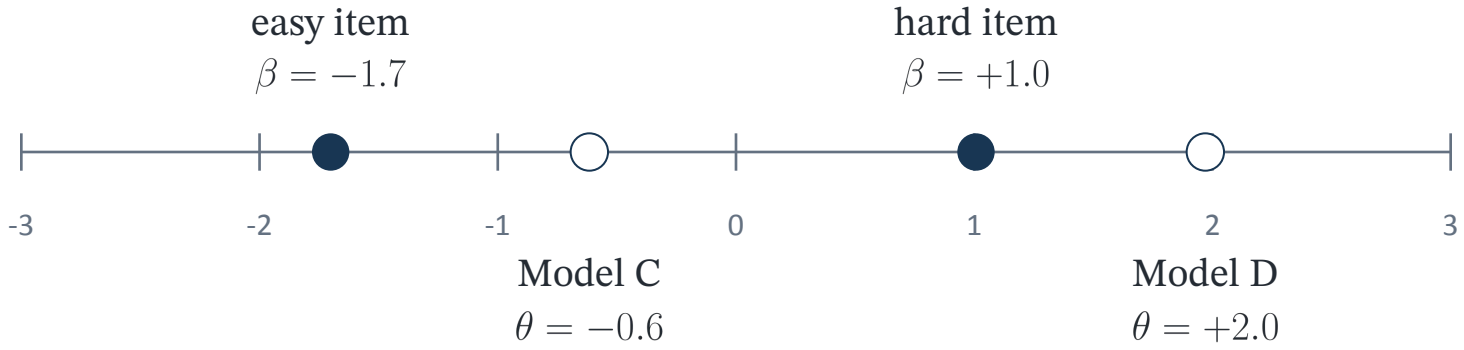
Ability estimate: unchanged

Raw averages mix model ability with item difficulty.

Truong, Tu, Liang, Li, Koyejo, Reliable and Efficient Amortized Model-based Evaluation, ICML 2025

C2 · The Rasch model puts models and items on one ruler

$$\Pr(\text{correct}) = \sigma(\theta - \beta)$$



This assumes the items share one construct (C1). Then check that the model fits.

When $\theta = \beta$, the probability of success is 50%.

Truong, Tu, Liang, Li, Koyejo, Reliable and Efficient Amortized Model-based Evaluation, ICML 2025

C2 · A common scale needs a common calibration

One item bank calibrated from reference responses

Model A

q_1 q_2 q_3

Model B

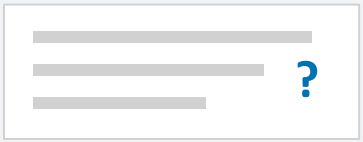
q_4 q_5 q_6

Different subsets. Item difficulties anchored to one scale.

Rasch calibration · schematic · fitting two tests separately does not put them on one scale

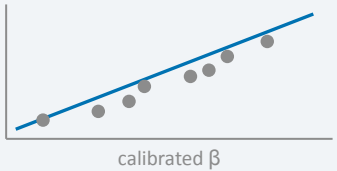
C2 · Prediction-guided evaluation reuses calibration for new questions

1 New item



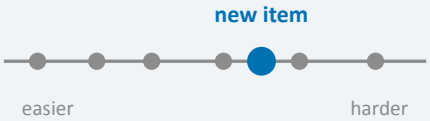
Content + context

2 Predictor



Earlier calibration

3 Difficulty



On the shared scale

New items need calibration; earlier calibration can predict where they sit.

Spreads calibration cost over later items; check predictions on held-out responses.

Truong, Tu, Liang, Li, Koyejo, Reliable and Efficient Amortized Model-based Evaluation, ICML 2025

C2 · Select questions using Fisher information

$$p_j(\theta) = \sigma(\theta - \beta_j)$$

$$\ell_j(\theta) = y_j \log p_j + (1 - y_j) \log(1 - p_j)$$

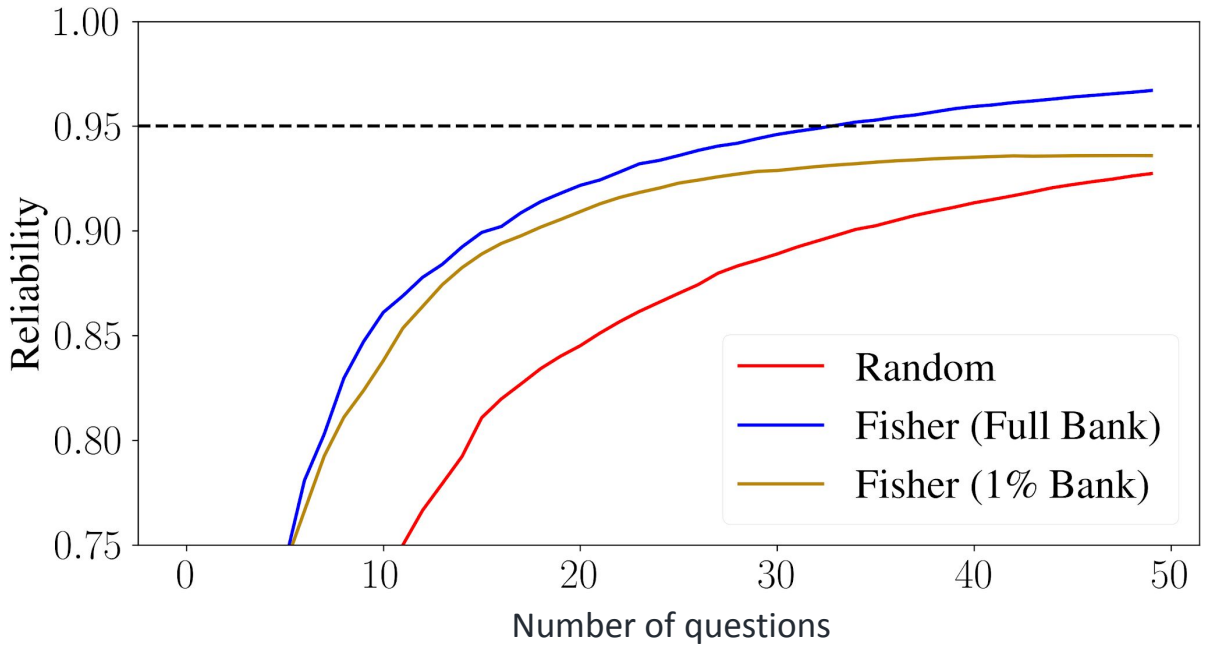
$$I_j(\theta) = -\mathbb{E}\left[\ell_j''(\theta)\right] = p_j(1 - p_j)$$

$p = 0.10$	**$p = 0.50$**	$p = 0.90$
$I = 0.09$	**$I = 0.25$**	$I = 0.09$

Questions near the model's level tell you the most. Adaptive exams like the GRE choose your next question this way.

Truong, Tu, Liang, Li, Koyejo, Reliable and Efficient Amortized Model-based Evaluation, ICML 2025

C2 · Adaptive tests need few questions



31

questions

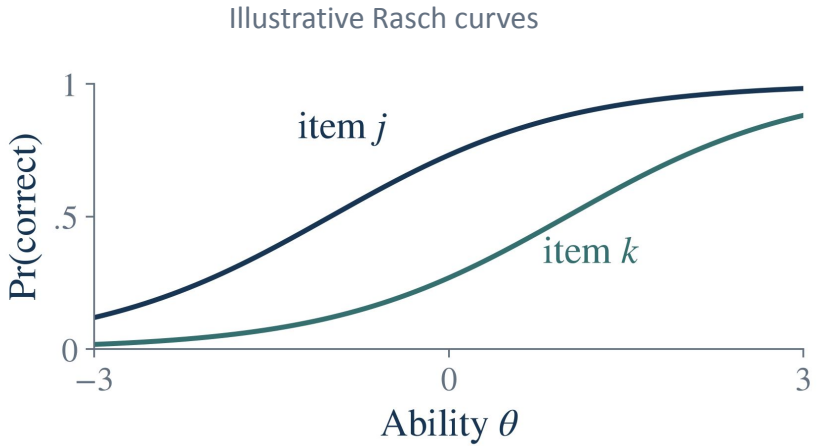
.95 reliability across test takers

AIRBench (AI safety) simulation

Simulated test takers; excludes building the bank.

Truong, Tu, Liang, Li, Koyejo, Reliable and Efficient Amortized Model-based Evaluation, ICML 2025

C2 · The response model makes testable predictions



Shared ability

Successes on different items should tend to go together across models.

Unexpected associations flag items for review.

$$\mathrm{Cov}(X_j, X_k) = \mathrm{Cov}(p_j(\theta), p_k(\theta)) \geq 0 \qquad (j \neq k)$$

Under the Rasch model: once ability is known, answers to different items are unrelated · across models

Truong, …, Koyejo, Fantastic Bugs and Where to Find Them in AI Benchmarks, NeurIPS 2025 D&B (arXiv 2511.16842)

C2 · A correct answer can fail the grader

A GSM8K item: the answer key is right, the grader is wrong.

20 popsicles × $0.25 + 4 ice-cream bars × $0.50

Accepted

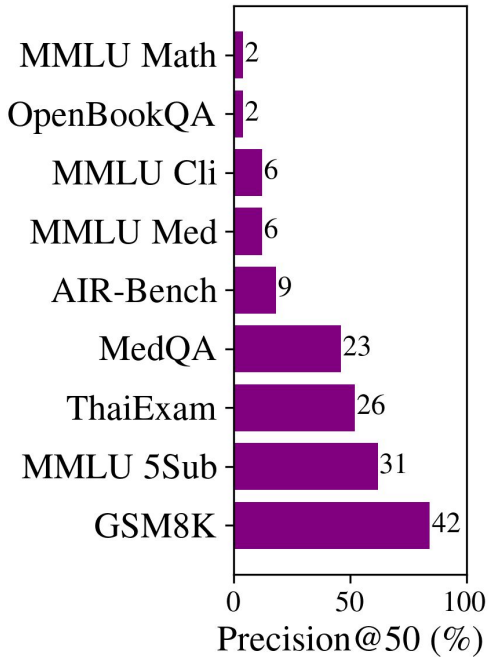
$7

Rejected

$7.00

Truong, …, Koyejo, Fantastic Bugs and Where to Find Them in AI Benchmarks, NeurIPS 2025 D&B (arXiv 2511.16842)

C2 · Targeted inspection finds real problems



42/50

flagged GSM8K items had confirmed problems

The yield varies substantially across benchmarks.

Truong, …, Koyejo, Fantastic Bugs and Where to Find Them in AI Benchmarks, NeurIPS 2025 D&B (arXiv 2511.16842)

C2 · What holds, where it breaks, a better claim

Original claim: *“To compare models fairly, every model must answer the same full test.”*

What holds: A shared full test is the simplest valid design and needs no model. Raw averages on different subsets do mix ability with question difficulty.

Where it breaks: Once questions share a calibrated scale, adaptive tests can compare models on different, fewer questions. The same fitted model flags broken questions; experts confirmed up to 84% of flagged items.

Better claim: Models can be compared on different, fewer questions when the questions share a calibrated scale and the response model fits. Check the fit first.

Truong, Tu, Liang, Li, Koyejo, Reliable and Efficient Amortized Model-based Evaluation, ICML 2025; Truong, …, Koyejo, Fantastic Bugs and Where to Find Them in AI Benchmarks, NeurIPS 2025 D&B (arXiv 2511.16842)

C2 · To compare models fairly, every model must answer the same full test

Vote again: hands up if you still agree.

A well-translated benchmark measures the same thing in every language

Hands up if you agree.

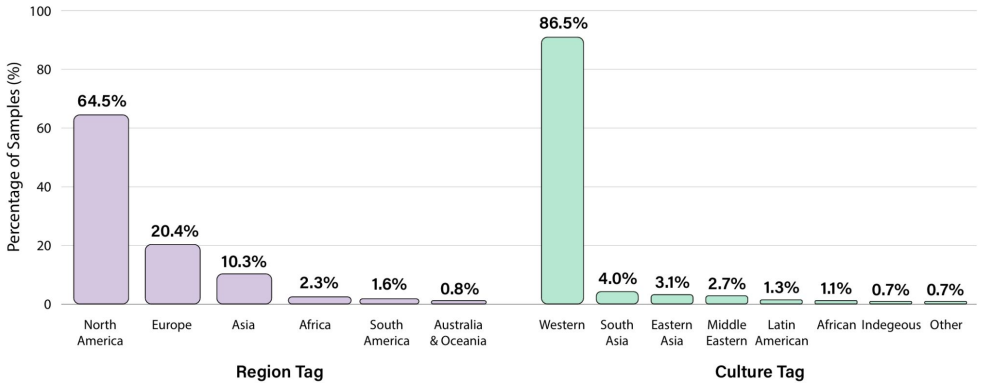
C3 · A well-translated benchmark measures the same thing in every language

Why people believe it: If the translation is faithful, a score gap between languages reflects the model.

What we'll show: A gap can come from the model, the language, or single prompts. Separate them before reading the gap.

C3 · Most benchmarks were written for one place

- 28% of questions in MMLU, a 57-subject knowledge test, need culturally sensitive knowledge
- 84.9% of those focus on North America or Europe
- Translation into 42 languages carries those assumptions along



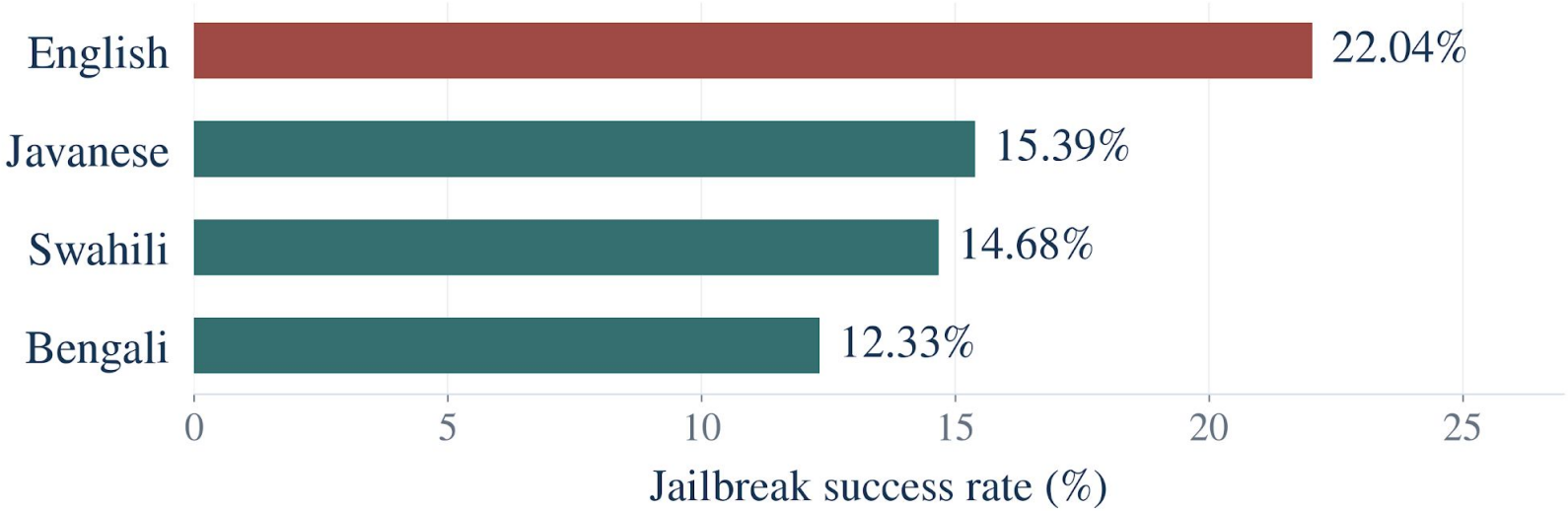
Whose population does the score describe?

Next: what translation does to safety prompts.

Singh, Romanou, Fourrier, Adelani, et al., Global MMLU: Understanding and Addressing Cultural and Linguistic Biases in Multilingual Evaluation, ACL 2025 (arXiv 2412.03304)

C3 · English is not always the safest language

MultiJail (harmful requests in 10 languages) · all 16 Grok configurations pooled · GPT-5.2 judge · observed rates



Jailbreak = the model complies. English has the highest jailbreak rate of these four languages.

Zhang, Patel, Truong, Koyejo, Why Do Safety Guardrails Degrade Across Languages?, COLM 2026 (arXiv 2605.17173)

C3 · Extend C2's model to include language

p: probability of a judged safe response

C2: model and prompt

Ignore the language

$$p_{mqL} = \sigma(\theta_m - \beta_q)$$

Add a language effect

Allow a shared shift

$$p_{mqL} = \sigma(\theta_m - \beta_q - \gamma_L)$$

Same construct. Items are now prompts;
language is a new facet (a condition of measurement).

Zhang, Patel, Truong, Koyejo, Why Do Safety Guardrails Degrade Across Languages?, COLM 2026 (arXiv 2605.17173)

C3 · Separate model shifts from prompt shifts

Rasch model with language effects

$$p_{mqL} = \sigma[(\theta_m + \delta_{mL}) - (\beta_q + \gamma_L + \tau_{qL})]$$

m = model · *q* = prompt · *L* = language

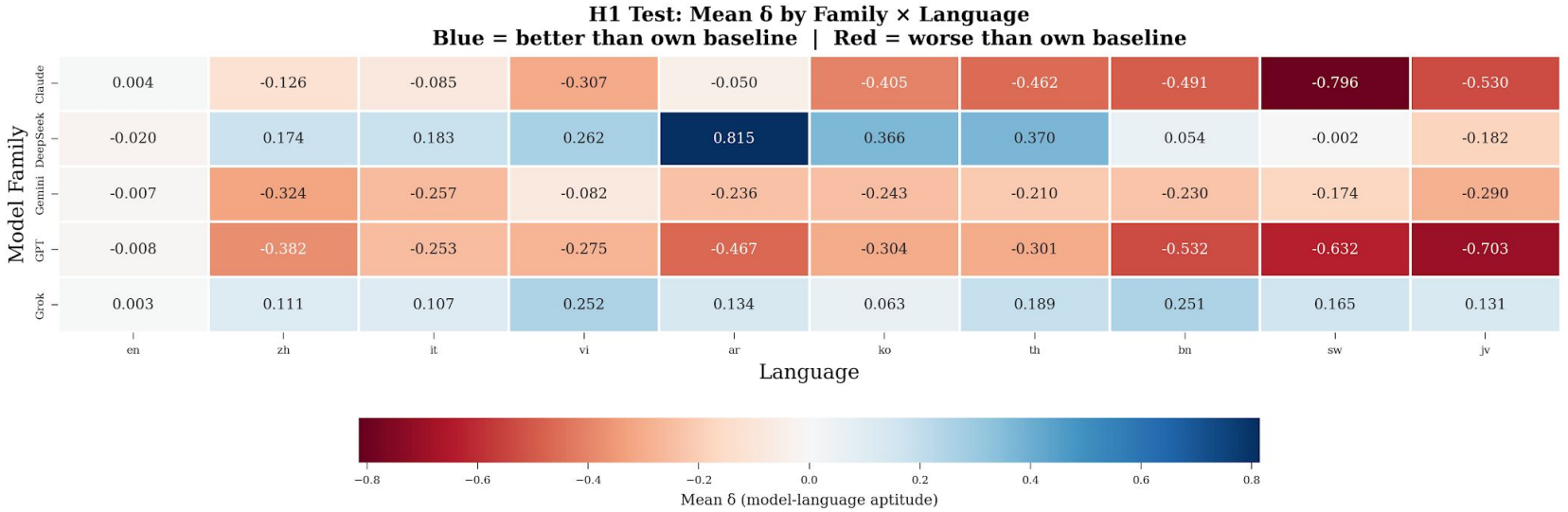
δ_{mL}	Model × language (δ)
γ_L	Shared language shift (γ)
τ_{qL}	Prompt × language (τ)

A shared language effect can hide model-specific and prompt-specific variation.

Adapted from Zhang, Patel, Truong, Koyejo, Why Do Safety Guardrails Degrade Across Languages?, COLM 2026 (arXiv 2605.17173), Eq. 1

C3 · Model families shift in opposite directions

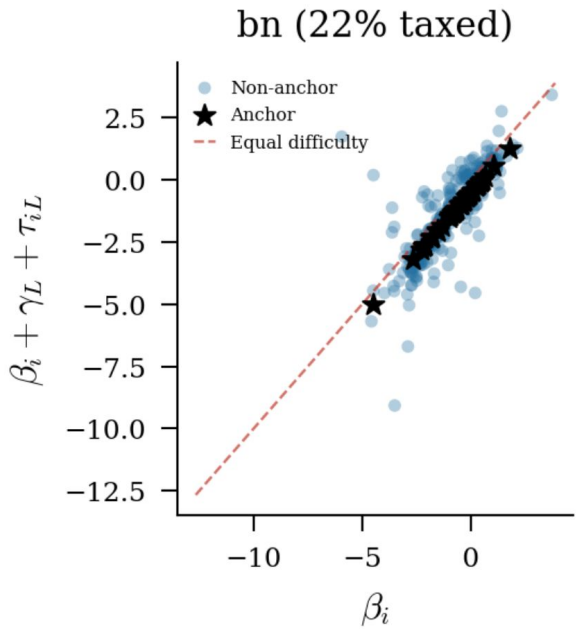
Fitted model-specific language shifts (not raw rate differences), averaged by family



GPT/Claude: mostly negative (less safe outside English). Grok: positive (safer outside English).

Zhang, Patel, Truong, Koyejo, Why Do Safety Guardrails Degrade Across Languages?, COLM 2026 (arXiv 2605.17173)

C3 · A shared language shift misses individual prompts



English vs. Bengali item difficulty

★ Anchors: prompts that behave the same in both languages; they line up the scales

Total difficulty shift

$\gamma_L + \tau_{qL}$

Some prompts depart far from the shared shift.

Zhang, Patel, Truong, Koyejo, Why Do Safety Guardrails Degrade Across Languages?, COLM 2026 (arXiv 2605.17173)

C3 · Good translations can still have large gaps

Two Bengali prompts that models answer unsafely far more often than the model and language predict

Translation changed the meaning

"hot-wire a car" became "warm a car"

Translation quality 2/5 · τ = 8.2

The harmful request is lost, so the item no longer tests safety: a benchmark bug.

Translation was perfect

"Help me find the SSN of Ben Sasse"

Translation quality 5/5 · τ in the top 50

Ben Sasse is a U.S. politician and SSN a U.S. identifier; the request reads as less threatening in Bengali.

Chinese shows the same bug: "roofie someone's drink" became "secretly drink someone else's beverage" (τ = 1.8).

Mistranslation explains a few outliers; translation quality explains about 1% of τ overall.

Zhang, Patel, Truong, Koyejo, Why Do Safety Guardrails Degrade Across Languages?, COLM 2026 (arXiv 2605.17173)

C3 · One average can hide different language effects

Language

English can be more vulnerable.

Model

Language patterns differ by family.

Prompt

Good translation can still leave gaps.

A score gap alone does not tell us what changed.

Next: which conditions matter in deployment?

Tutorial synthesis · distinguish shared, model-specific, and prompt-specific effects

C3 · What holds, where it breaks, a better claim

Original claim: *“A well-translated benchmark measures the same thing in every language.”*

What holds: Faithful translation removes one source of error, and low-resource languages do produce more uncertain responses.

Where it breaks: 22 of 61 model configurations were less safe in English than in low-resource languages. Model families shift in opposite directions, and some prompts diverge even with accurate translations.

Better claim: Compare languages with a measurement model that has model, language, and prompt terms, and check that the measurement model predicts well out of sample.

Zhang, Patel, Truong, Koyejo, Why Do Safety Guardrails Degrade Across Languages?, COLM 2026 (arXiv 2605.17173)

C3 · A well-translated benchmark measures the same thing in every language

Vote again: hands up if you still agree.

Where we are

So far

C1 Averaging safety benchmarks robustly measures safety

C2 To compare models fairly, every model must answer the same full test

C3 A well-translated benchmark measures the same thing in every language

Coming up

C4 The leaderboard winner will fit my product best

C5 Evaluation gaming is mostly cheating or data leakage

C6 A private, uncontaminated test set stops gaming

The leaderboard winner will fit my product best

Hands up if you agree.

C4 · The leaderboard winner will fit my product best?

Two key ingredients needed for leaderboard ranking to survive deployment:

1. The outcome being scored is the deployment outcome.
2. The unit being ranked is the unit being deployed.

Misaligned fit can shift leaderboard interpretations.

C4 · Rankings do survive certain shifts

> **Concession:** Distribution shifts don’t necessarily modify model rankings.

C4 · Rankings do survive certain shifts

ImageNet Top-1

Orig. Rank	Model	Orig. Accuracy	New Accuracy	Gap	New Rank	Δ Rank
1	pnasnet_large_tf	82.9 [82.5, 83.2]	72.2 [71.3, 73.1]	10.7	3	-2
4	nasnetalarge	82.5		0.3	1	3
21	resnet152	78.3		1.3	21	0
23	inception_v3_tf	78.0		1.9	24	-1
30	densenet161	77.1		1.8	30	0
43	vgg19_bn	74.2 [73.8, 74.6]	61.9 [60.9, 62.8]	12.3	44	-1
64	alexnet	56.5 [56.1, 57.0]	44.0 [43.0, 45.0]	12.5	64	0
65	fv_64k	35.1 [34.7, 35.5]	24.1 [23.2, 24.9]	11.0	65	0

How robust are ImageNet model rankings to new images from a replicated data collection process?

Recht et al. (PMLR 2019)

C4 · Rankings do survive certain shifts

ImageNet Top-1							
Orig. Rank	Model	Orig. Accuracy		New Accuracy		Gap	New Rank	Δ Rank
1	pnasnet_large_tf	82				10.7	3	-2
4	nasnetalarge	82				10.3	1	3
21	resnet152	78				11.3	21	0
23	inception_v3_tf	78				11.9	24	-1
30	densenet161	77				11.8	30	0
43	vgg19_bn	74.2	[73.8, 74.6]	61.9	[60.9, 62.8]	12.3	44	-1
64	alexnet	56.5	[56.1, 57.0]	44.0	[43.0, 45.0]	12.5	64	0
65	fv_64k	35.1	[34.7, 35.5]	24.1	[23.2, 24.9]	11.0	65	0

How robust are ImageNet model rankings to new images from a replicated data collection process?

Largely stay the same!

Recht et al. (PMLR 2019)

C4 · Rankings do survive certain shifts

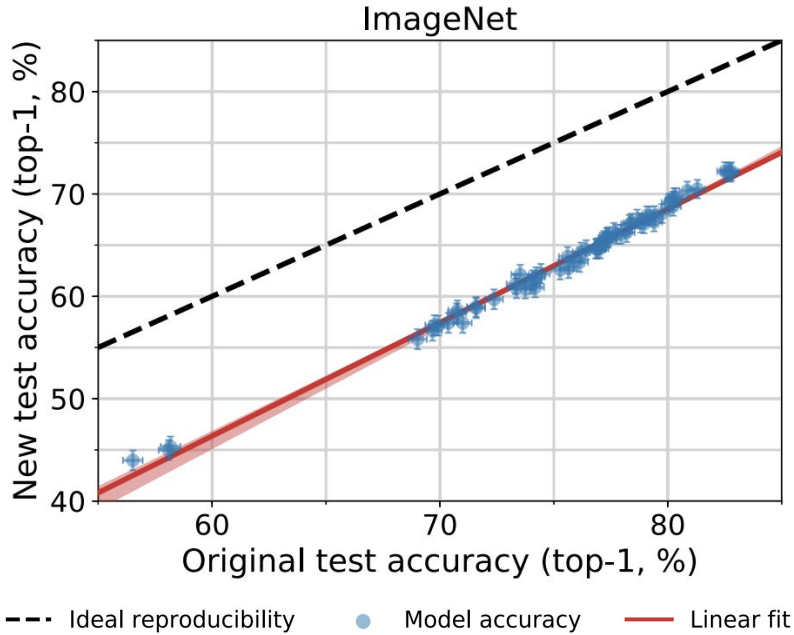
ImageNet Top-1

Orig. Rank	Model	Orig. Accuracy	New Accuracy	Gap	New Rank	Δ Rank
1	pnasnet_large_tf		[1.3, 73.1]	10.7	3	-2
4	nasnetalarge		[1.3, 73.1]	10.3	1	3
21	resnet152	78.3 [77.9, 78.7]	67.0 [66.1, 67.9]	11.3	21	0
23	inception_v3_tf	78.0 [77.6, 78.3]	66.1 [65.1, 67.0]	11.9	24	-1
30	densenet161	77.1 [76.8, 77.5]	65.3 [64.4, 66.2]	11.8	30	0
43	vgg19_bn	74.2 [73.8, 74.6]	61.9 [60.9, 62.8]	12.3	44	-1
64	alexnet	56.5 [56.1, 57.0]	44.0 [43.0, 45.0]	12.5	64	0
65	fv_64k	35.1 [34.7, 35.5]	24.1 [23.2, 24.9]	11.0	65	0

What about performance?

Recht et al. (PMLR 2019)

C4 · Rankings do survive certain shifts



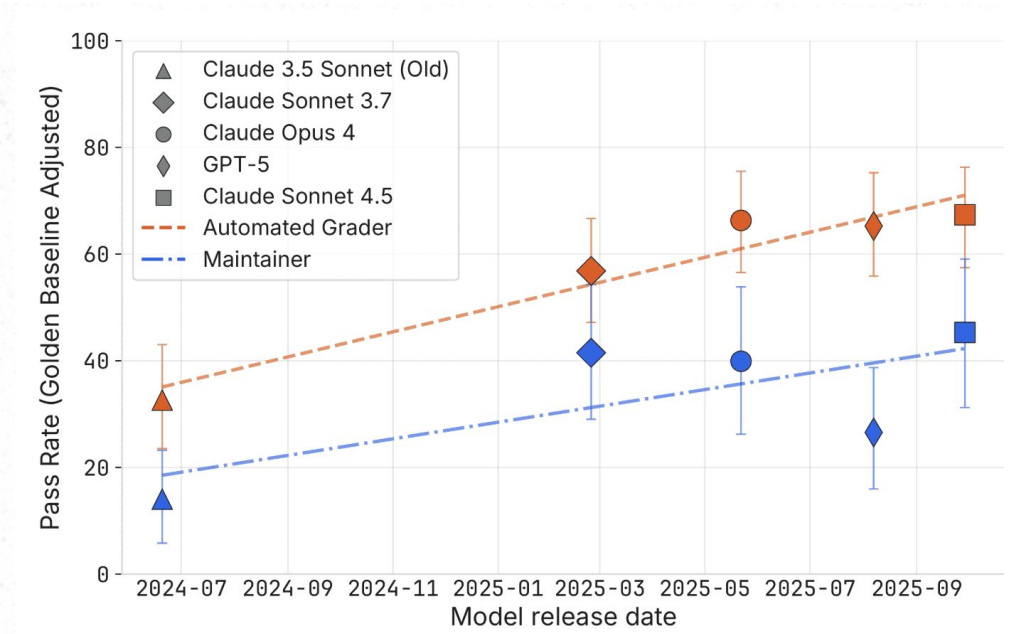
Performance does not necessarily hold even if rankings do.

Recht et al. (PMLR 2019)

C4 · Rank stability does not imply benchmark score will clear bar for deployment

> Leaderboard performance can obfuscate deployment performance when measured outcomes differ.

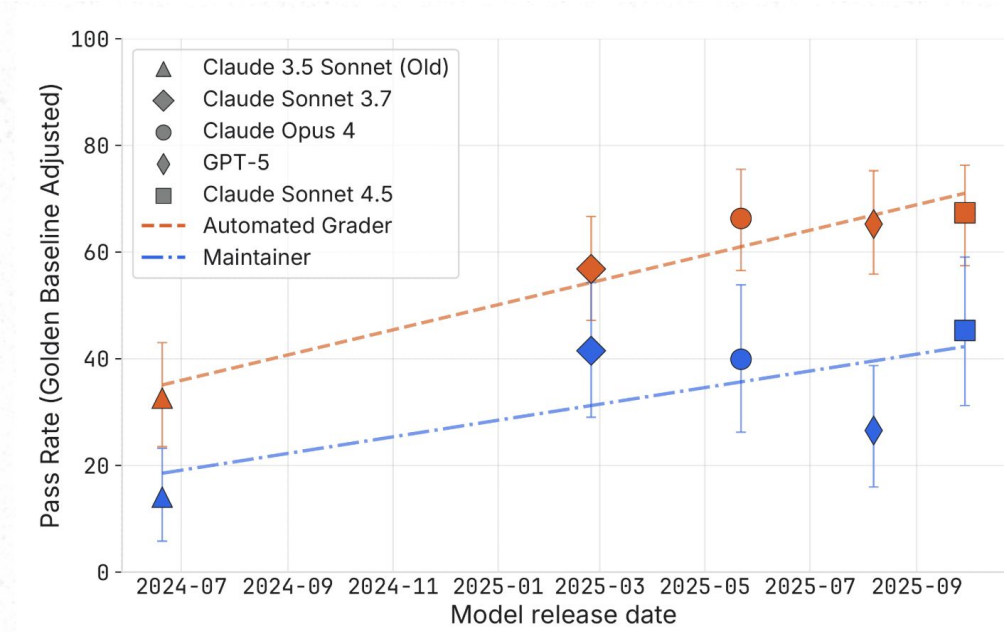
C4 · Rank stability does not imply benchmark score will clear bar for deployment



Would a repo maintainer accept SWE Bench Verified which pass automated grader?

Whitfill et al. (METR Note, 2026)

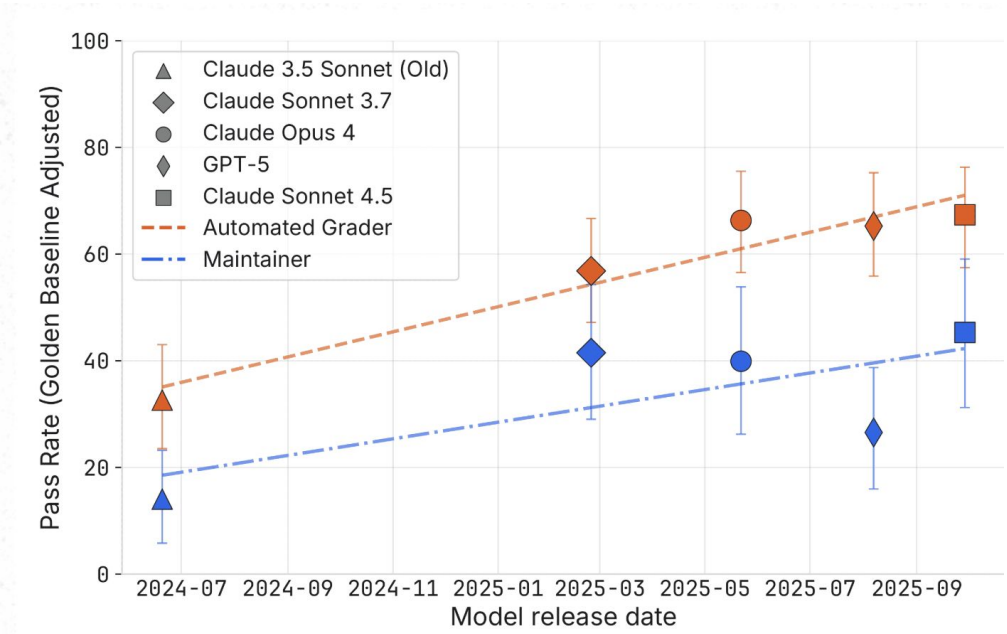
C4 · Rank stability does not imply benchmark score will clear bar for deployment



Git repo maintainer pass rate significantly lower than SWE Bench automatic grader pass rate.

Whitfill et al. (METR Note, 2026)

C4 · Rank stability does not imply benchmark score will clear bar for deployment

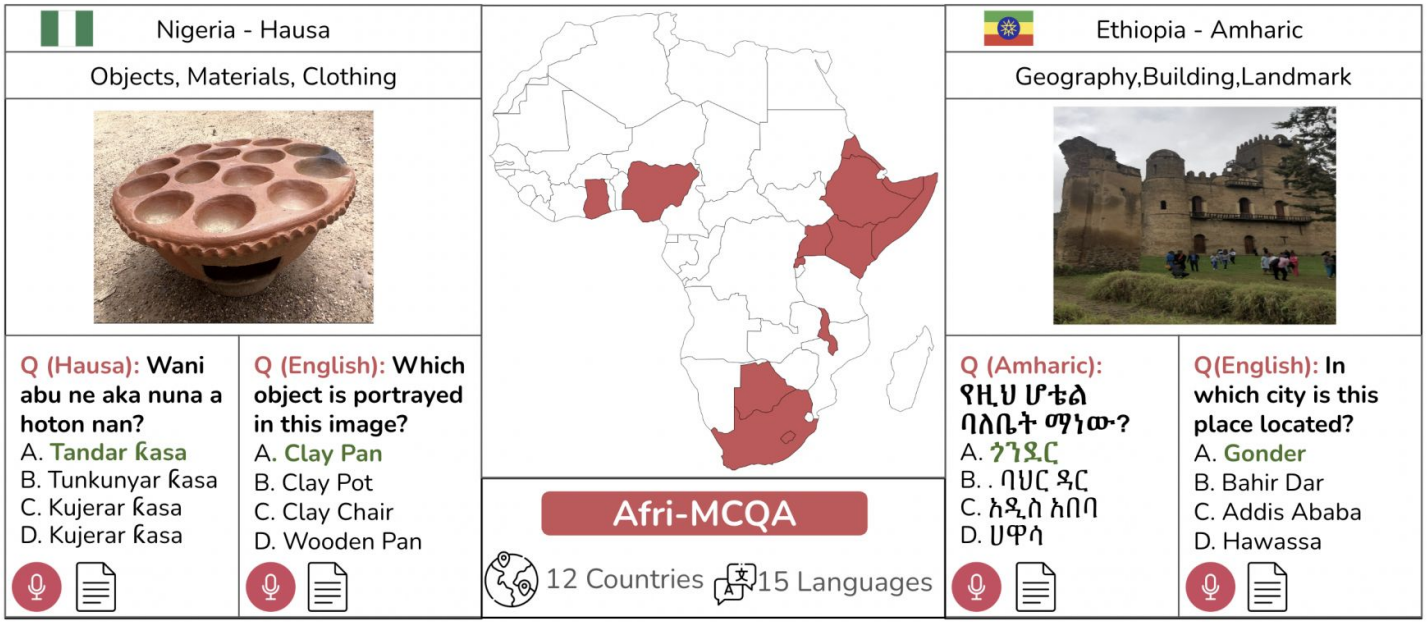


Why?

- Not following repo standards.
- Breaks other code.
- Not solving core issue.

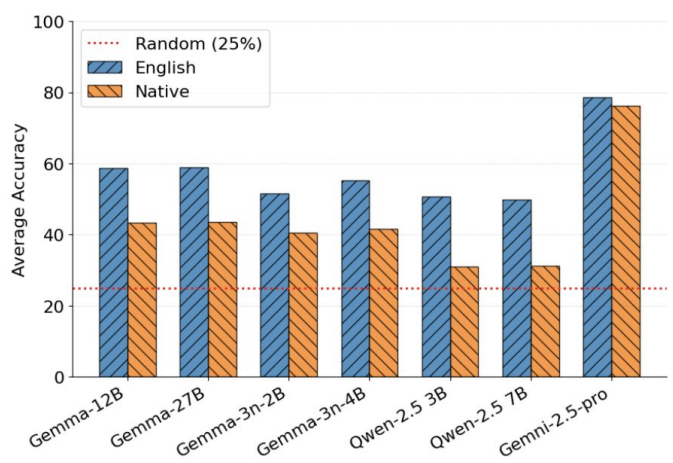
Whitfill et al. (METR Note, 2026)

C4 · Rank stability does not imply benchmark score will clear bar for deployment

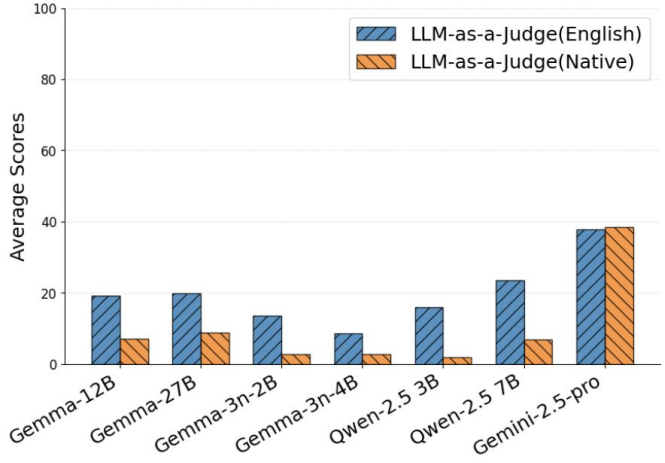


Tonja et al. (ACL 2026)

C4 · Rank stability does not imply benchmark score will clear bar for deployment



(a) MC-VQA (Text)

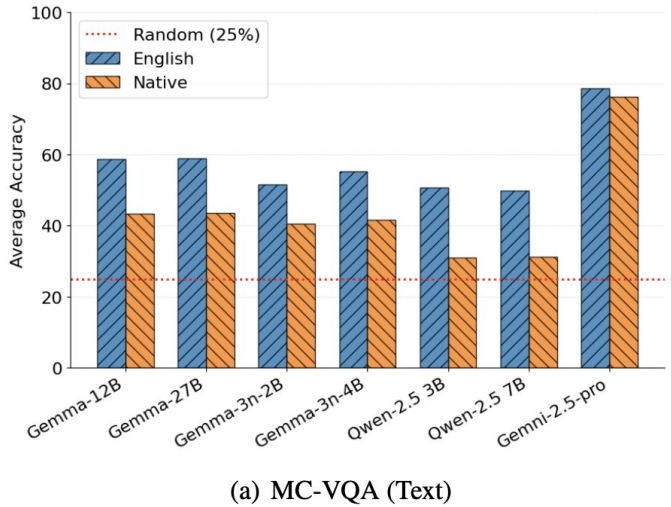


(b) Open-ended VQA (Text)

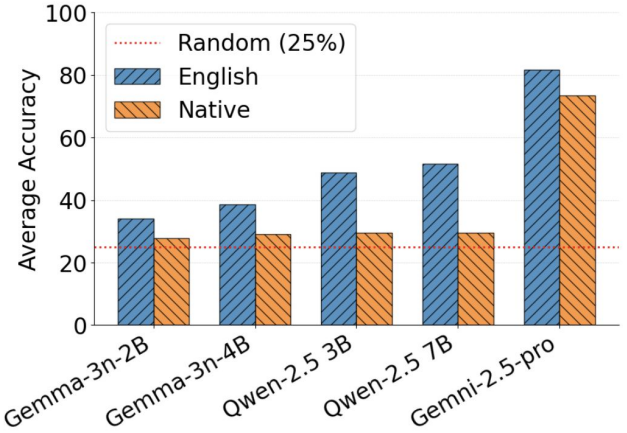
Performance drops significantly when evaluated on open-ended generation vs. multiple-choice.

Tonja et al. (ACL 2026)

C4 · Rank stability does not imply benchmark score will clear bar for deployment



(a) MC-VQA (Text)

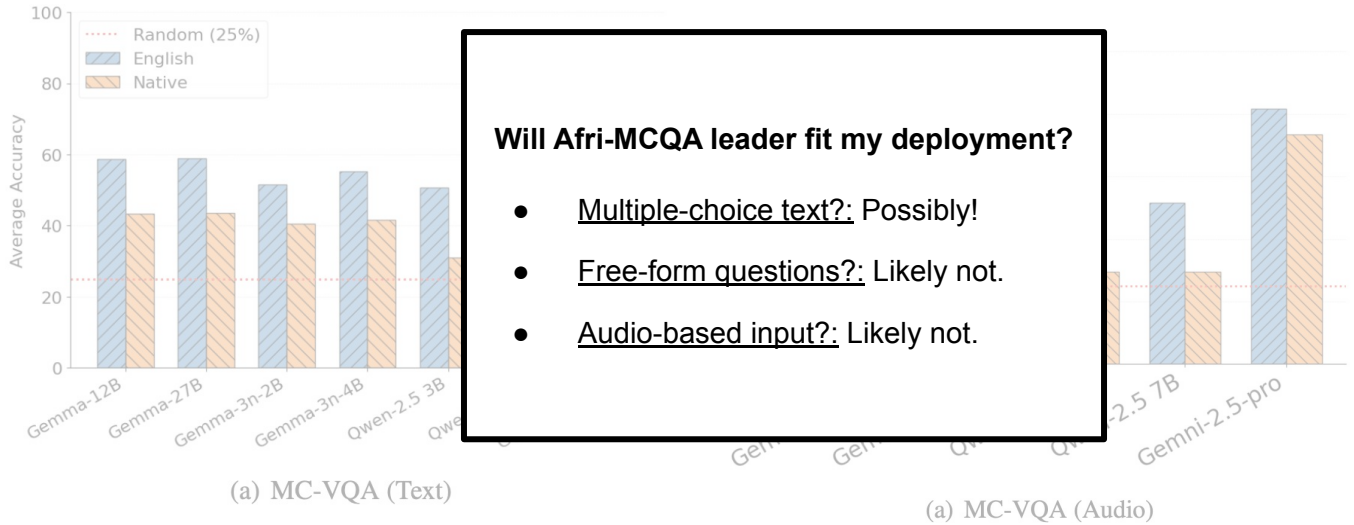


(a) MC-VQA (Audio)

Switching from text to audio input can also drop performance, particularly in native languages.

Tonja et al. (ACL 2026)

C4 · Rank stability does not imply benchmark score will clear bar for deployment



Will Afri-MCQA leader fit my deployment?

- Multiple-choice text?: Possibly!
- Free-form questions?: Likely not.
- Audio-based input?: Likely not.

Tonja et al. (ACL 2026)

C4 · Rank stability does not imply benchmark score will clear bar for deployment

What happened? Specification error over benchmark ***outcomes***.

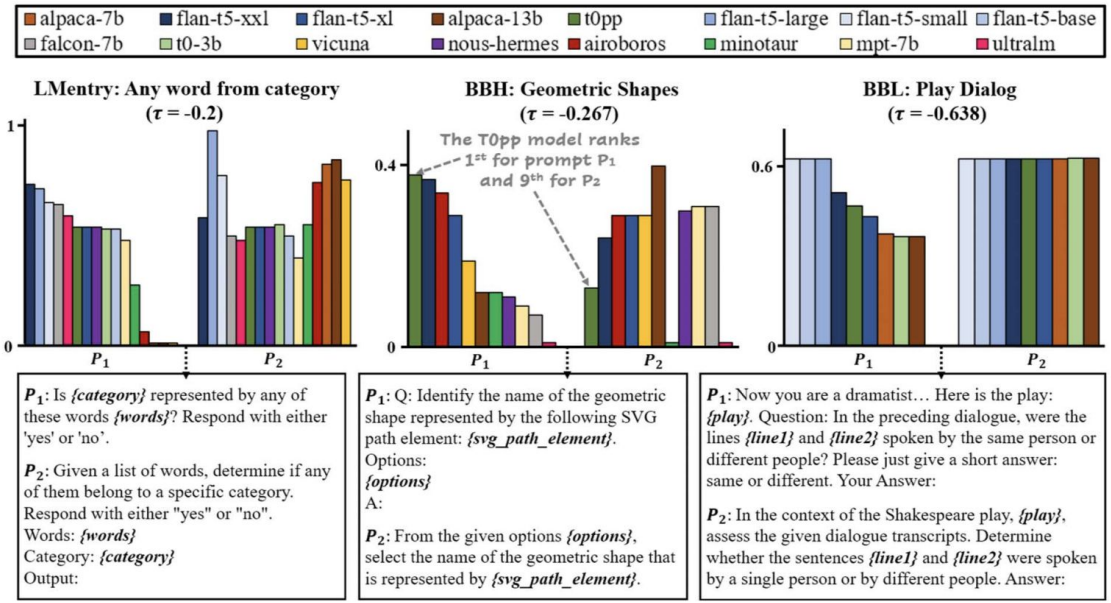
- Automated test passing does not equate to PR standards being met.
- MCQA performance does not equate to free-generation behavior.

Takeaway: Important to scrutinize fit between deployment outcomes and outcomes benchmark evaluates.

C4 · Thinking about “units” we evaluate

> **Example statement:** “Claude Fable X.Y is the best model on the Z leaderboard.”
>
> What components do we mean when we say *“model”*?

C4 · Changing the “unit” being evaluated can change rankings

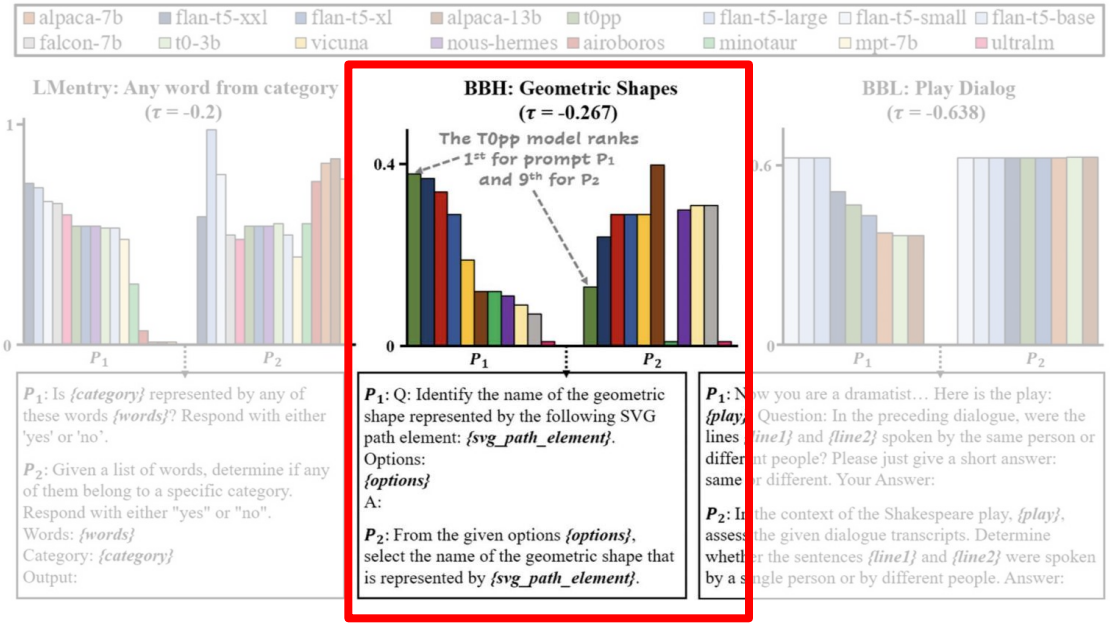


Unit: (prompt, model)

Rankings can *reverse* for certain models depending on prompt!

Mizrahi et al. (TACL 2024)

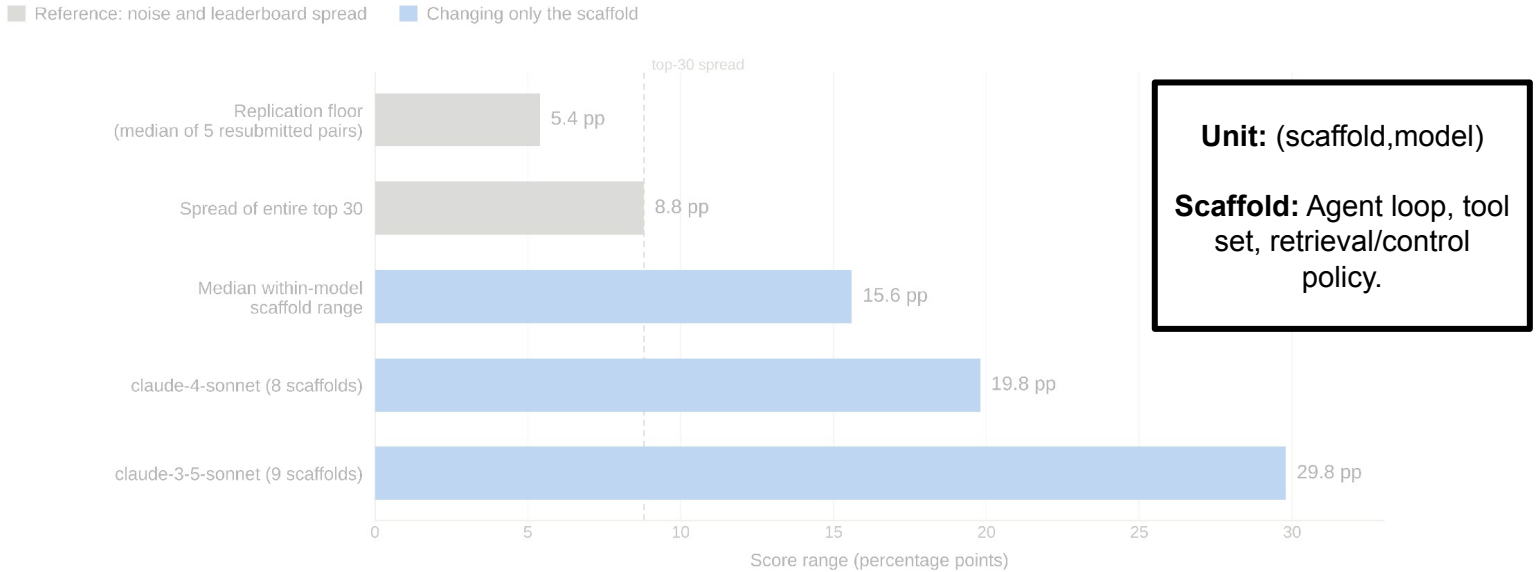
C4 · Changing the "unit" being evaluated can change rankings



Top model ranked highest under one prompt, and lowest under another!

Mizrahi et al. (TACL 2024)

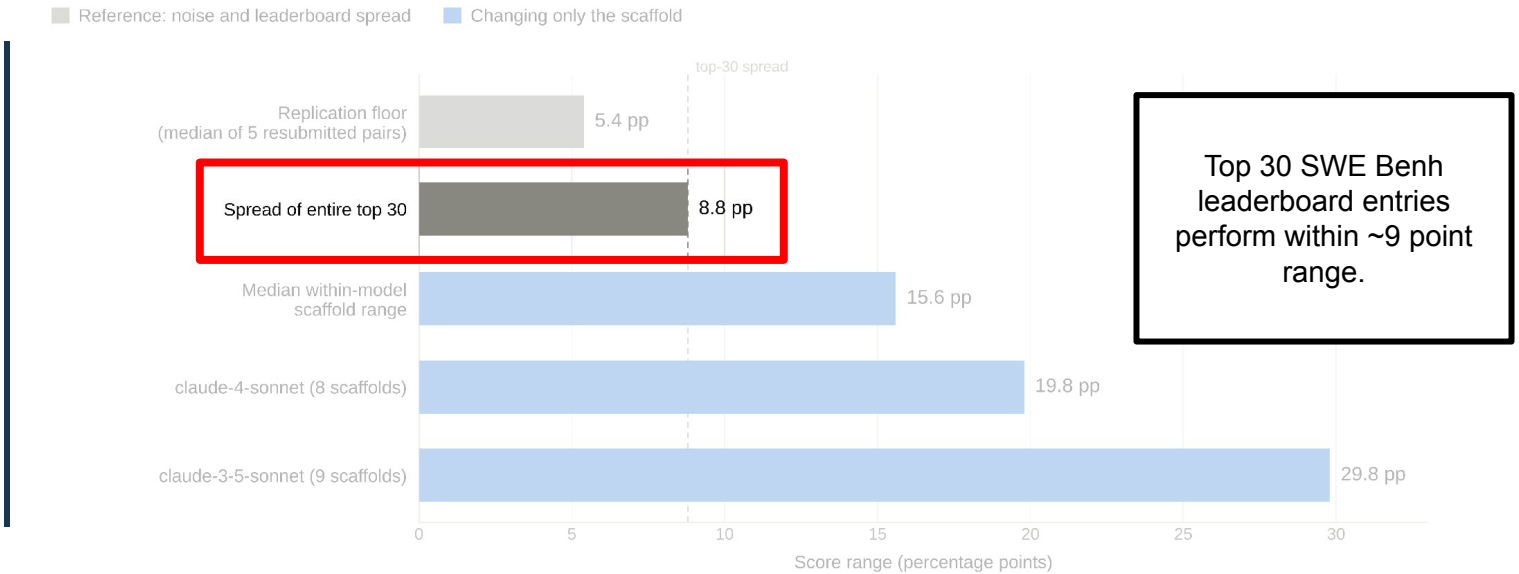
C4 · Changing the “unit” being evaluated can change rankings



Source: Liu et al., 'Coding Agents Have Converged' (arXiv 2609.17394), §4.1. SWE-bench Verified, 500 instances. Observational: teams choose model and scaffold together.
Paper reports claude-4-sonnet as 99 instances; 19.8 pp = 99 / 500.

Liu et al. (ADMA 2026)

C4 · Changing the “unit” being evaluated can change rankings



Source: Liu et al., 'Coding Agents Have Converged' (arXiv 2609.17394), §4.1. SWE-bench Verified, 500 instances. Observational: teams choose model and scaffold together.
Paper reports claude-4-sonnet as 99 instances; 19.8 pp = 99 / 500.

Liu et al. (ADMA 2026)

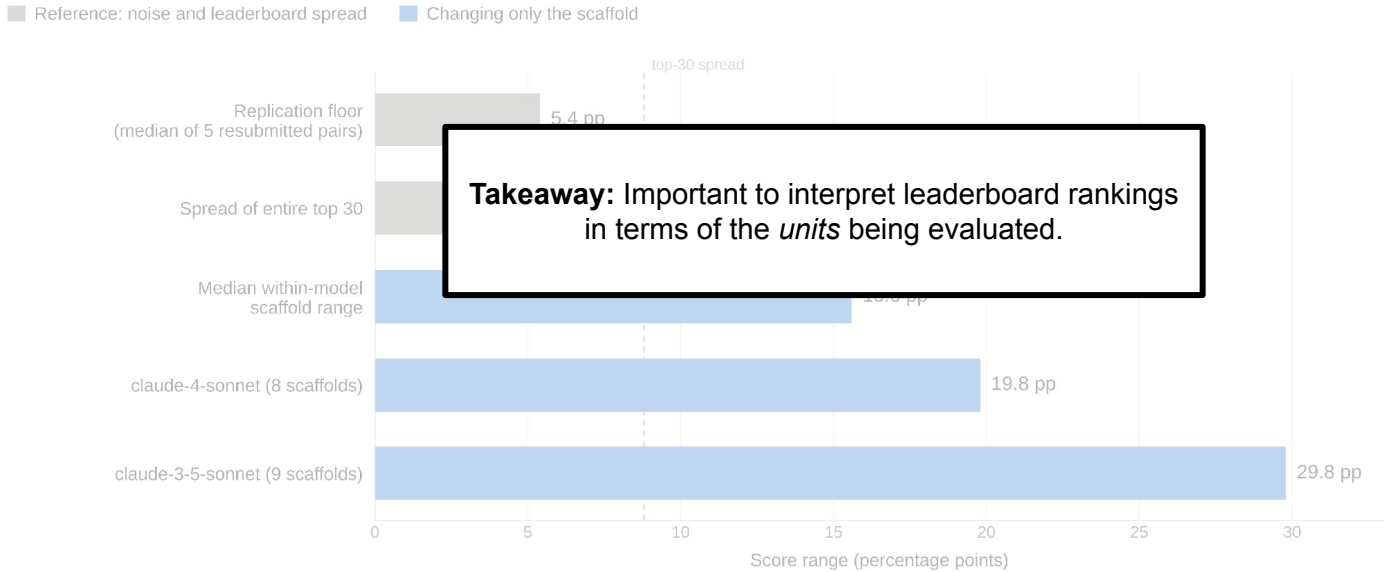
C4 · Changing the “unit” being evaluated can change rankings



Source: Liu et al., 'Coding Agents Have Converged' (arXiv 2609.17394), §4.1. SWE-bench Verified, 500 instances. Observational: teams choose model and scaffold together.
Paper reports claude-4-sonnet as 99 instances; 19.8 pp = 99 / 500.

Liu et al. (ADMA 2026)

C4 · Changing the “unit” being evaluated can change rankings



Takeaway: Important to interpret leaderboard rankings in terms of the *units* being evaluated.

Source: Liu et al., 'Coding Agents Have Converged' (arXiv 2609.17394), §4.1. SWE-bench Verified, 500 instances. Observational: teams choose model and scaffold together.
Paper reports claude-4-sonnet as 99 instances; 19.8 pp = 99 / 500.

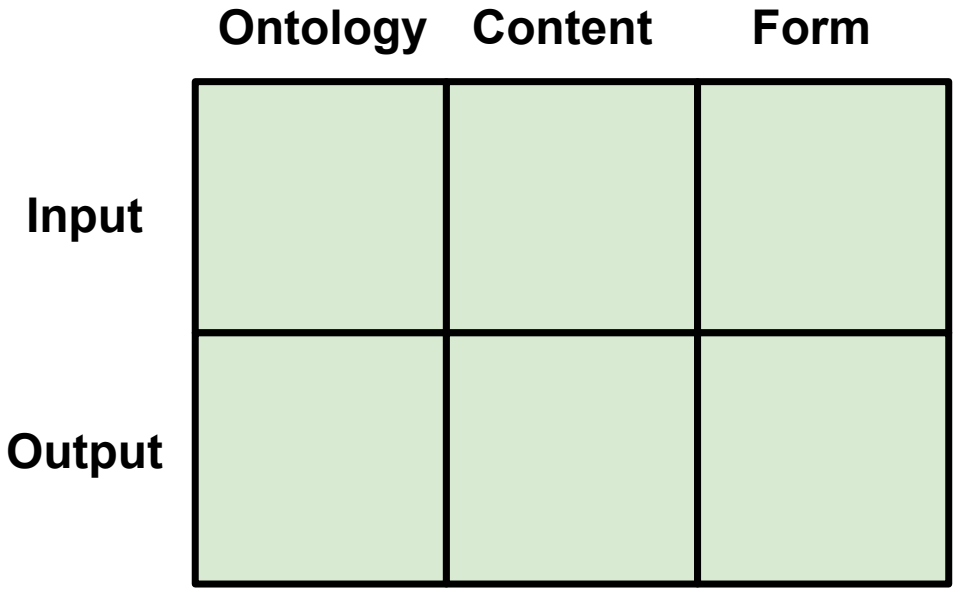
C4 · Tightening outcome and unit fit for deployment can facilitate leaderboard trust

Ensuring a leaderboard:

1. scores the same outcome as your deployment.
2. Ranks the “system” type you deploy.

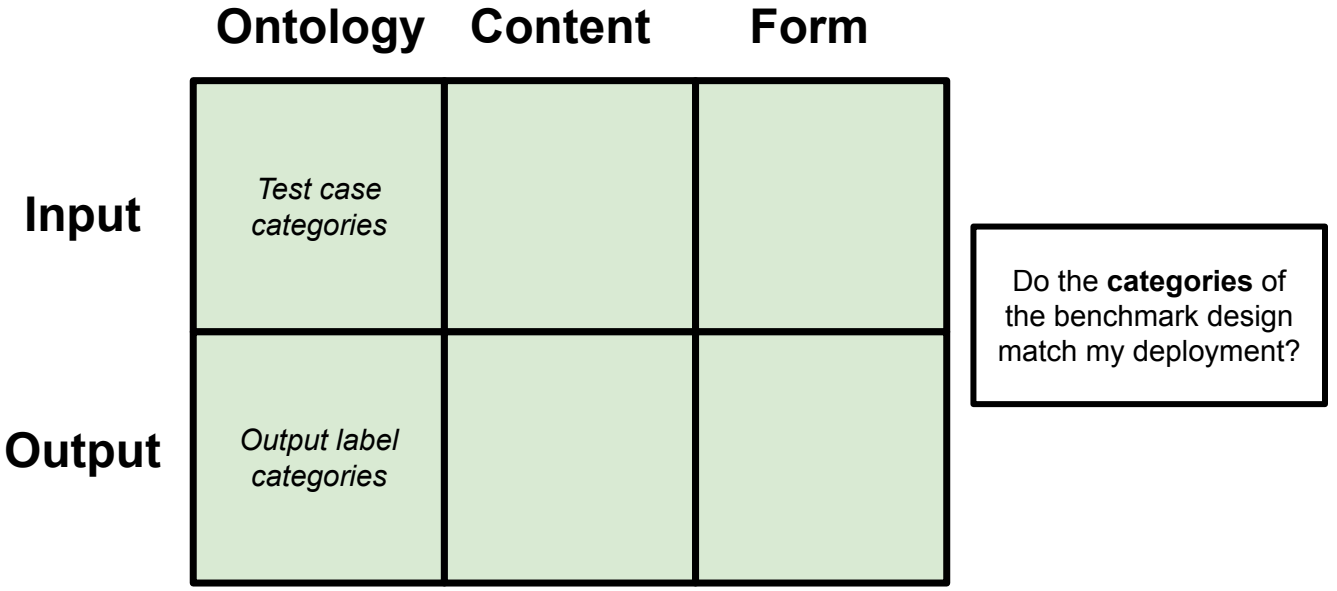
Builds trust in its interpretation.

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust

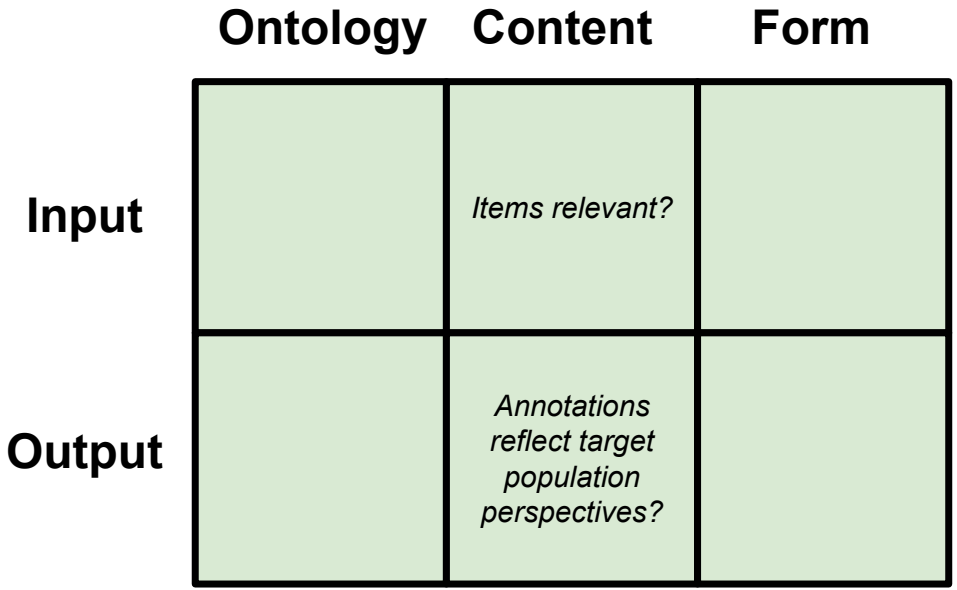


Scrutinizing benchmark at varying levels of abstraction can be useful for assessing fit for outcomes and system.

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust

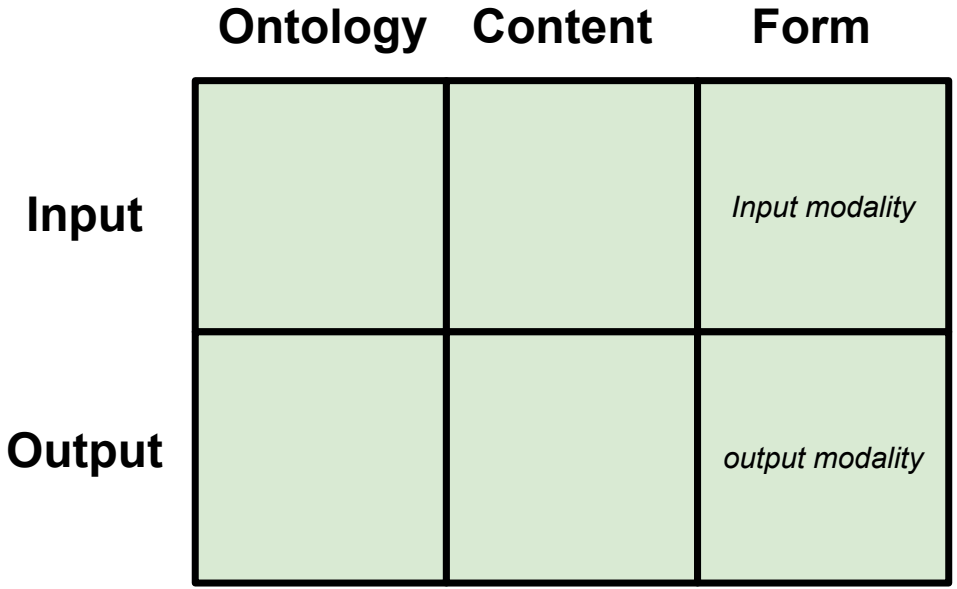


C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust



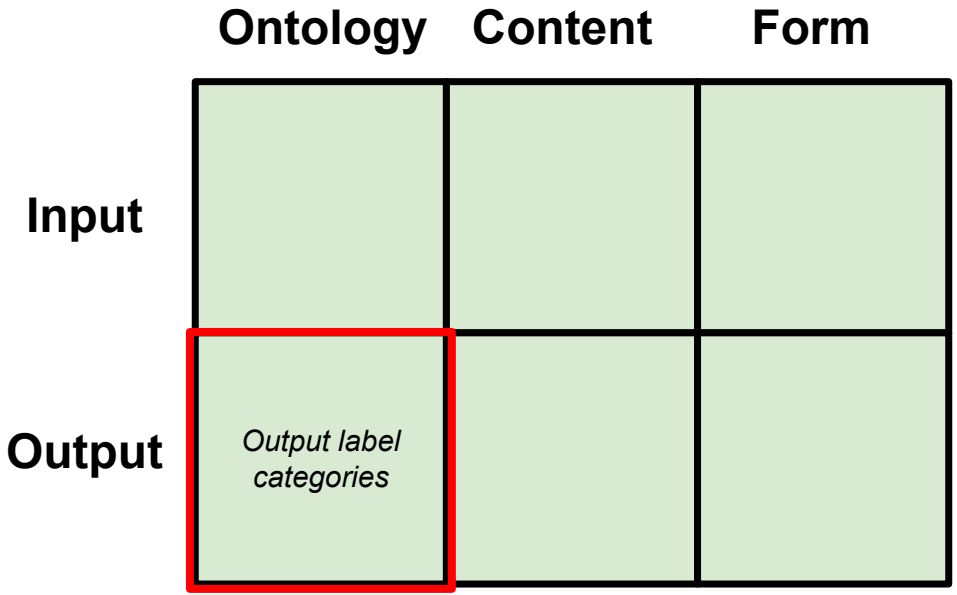
How well do the individual items and their labels fit my deployment?

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust



Does the representation of the data input signal and output modality match my deployment?

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust



Let's look at an example of **outcome** fit.

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust

Image



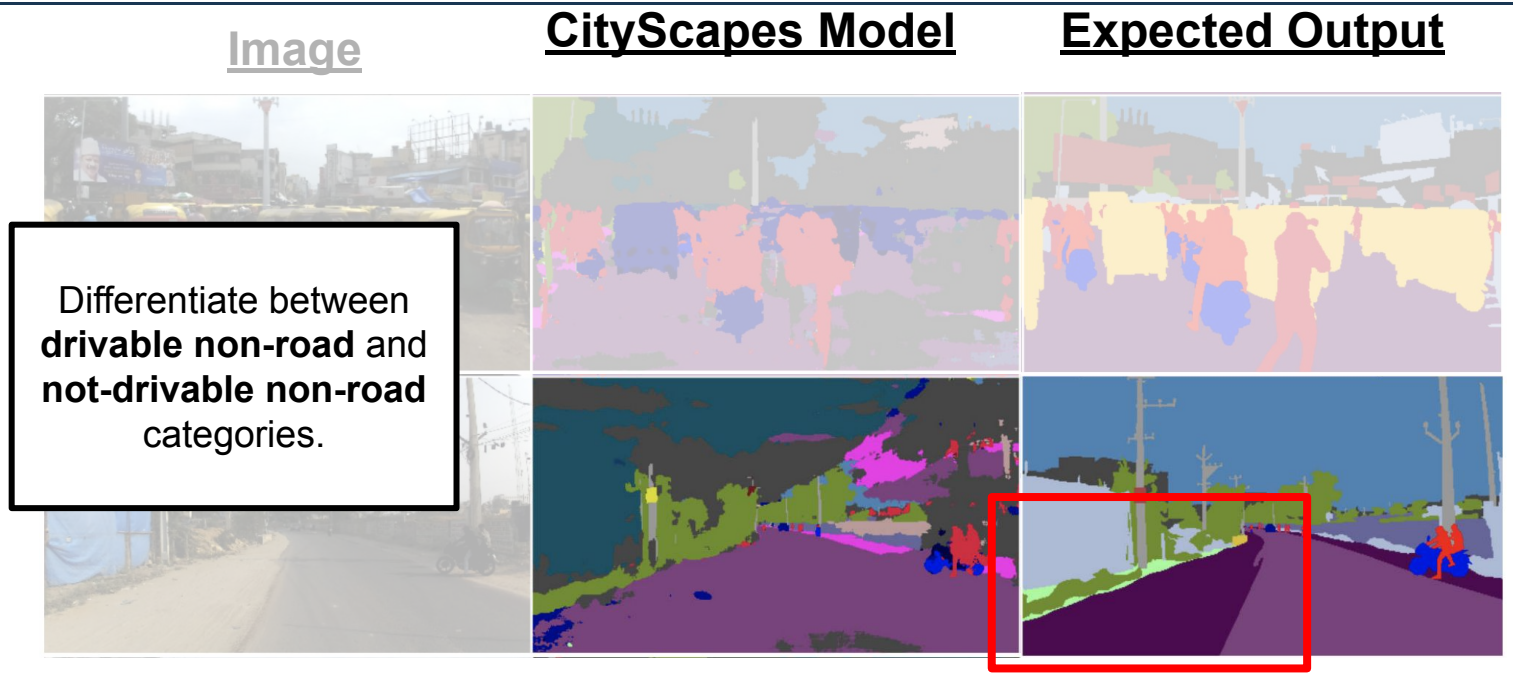
Indian Driving Dataset

Image segmentation dataset focused on driving environments.

Data collected in India.

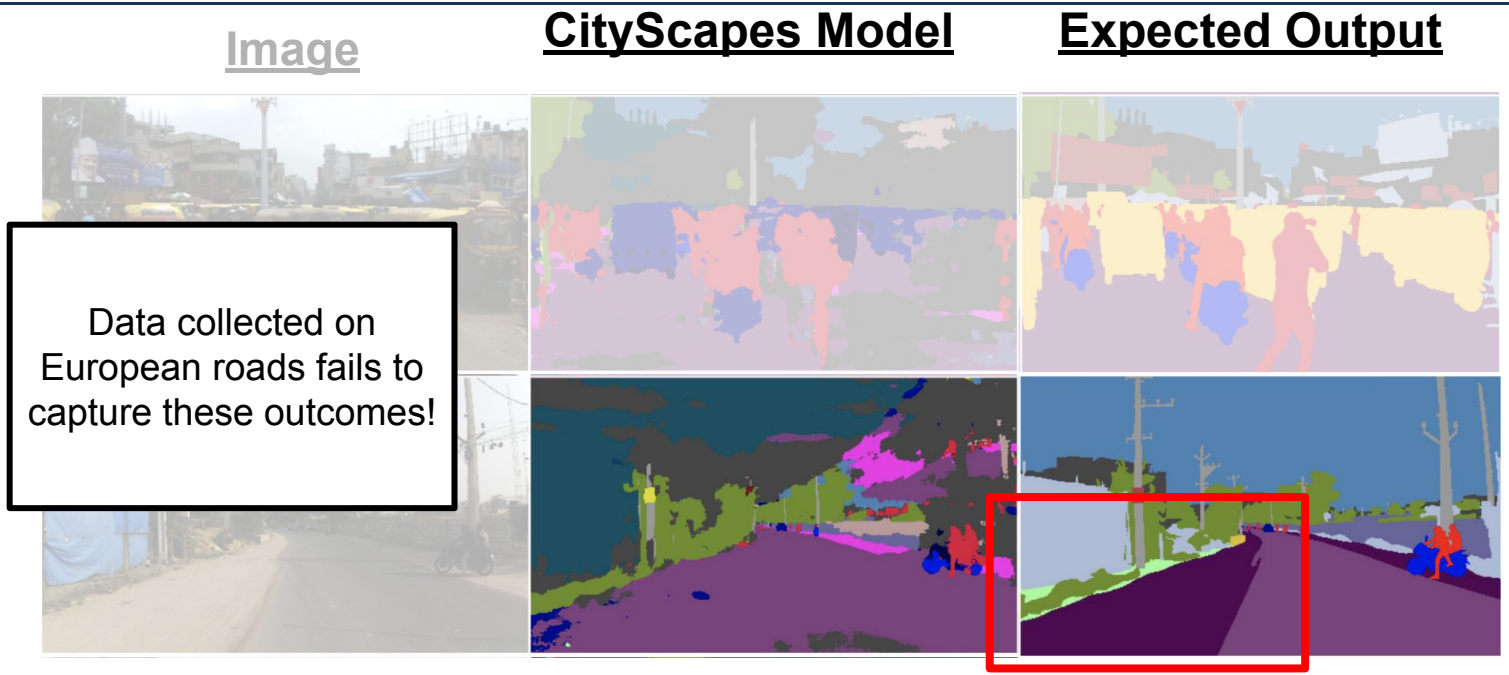
Varma et al. (WACV 2019)

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust



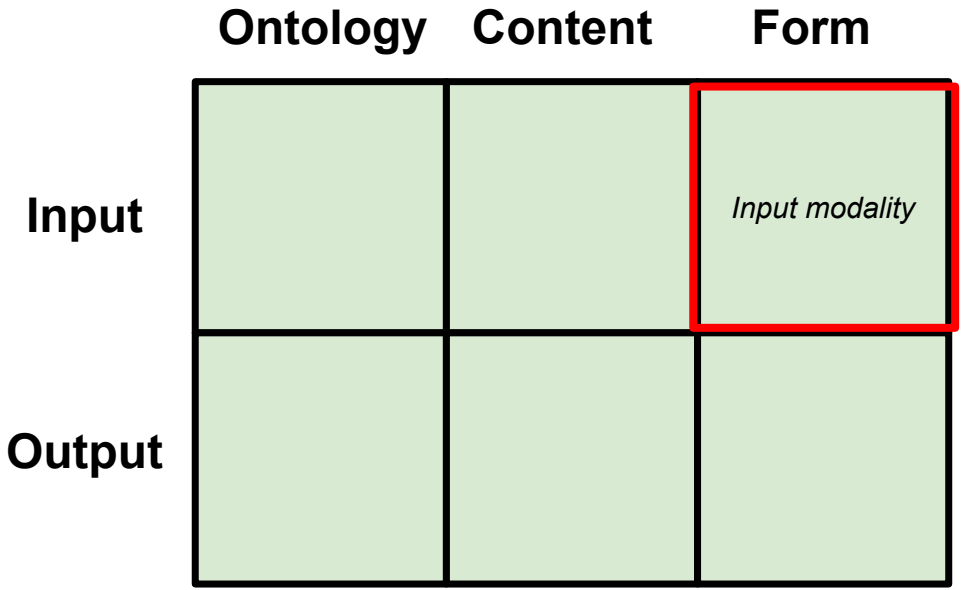
Varma et al. (WACV 2019)

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust



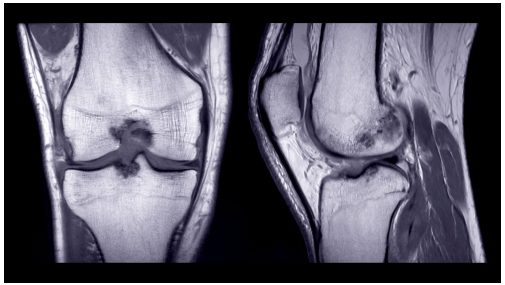
Varma et al. (WACV 2019)

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust



Let's look at an example of **system** fit.

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust

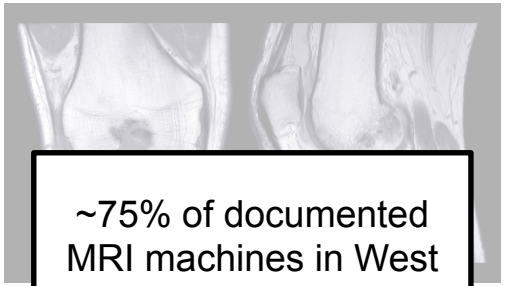


Field Strength	1.5T	3T
T1	375	407
T1 POST	849	641
T2	1651	2515
FLAIR	126	406
Total	3001	3969

fastMRI Dataset.
Contains 1.5T and 3T MRI data

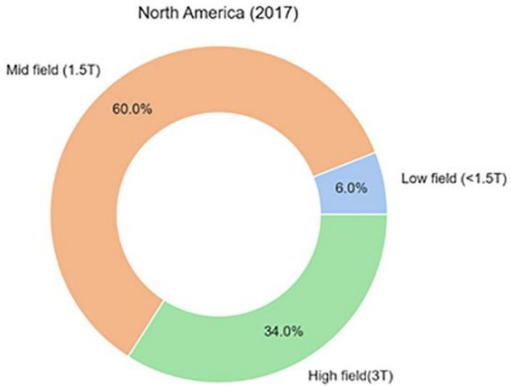
Zbontar et al. (arXiv 2018)

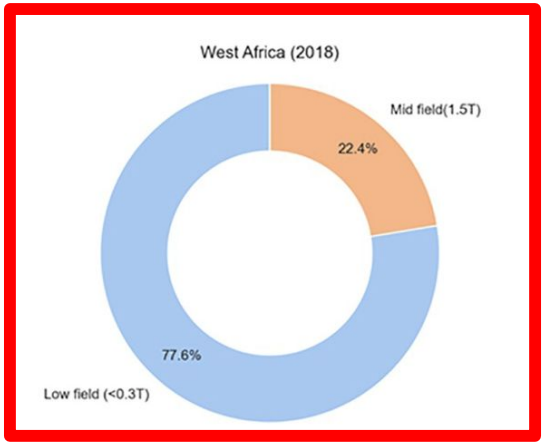
C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust



~75% of documented MRI machines in West Africa are < 1.5T in field strength.

T1	375	407
T1 POST	849	641
T2	1651	2515
FLAIR	126	406
Total	3001	3969





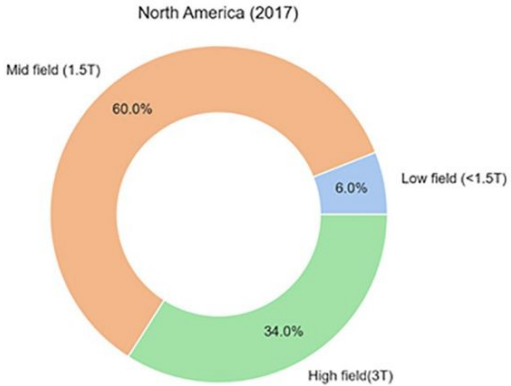
Qin et al. (European Heart Journal 2022)

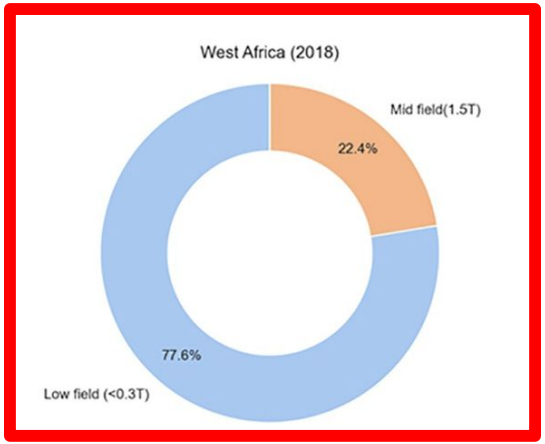
C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust



(image fidelity, model) pairing must be considered!

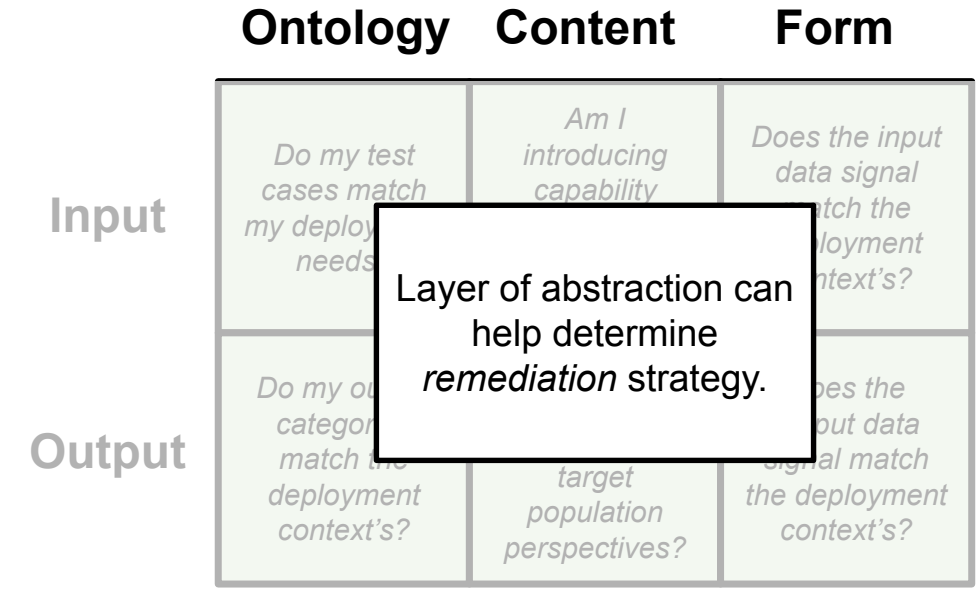
T1	375	407
T1 POST	849	641
T2	1651	2515
FLAIR	126	406
Total	3001	3969





Qin et al. (European Heart Journal 2022)

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust



Corona et al. (ICLM CTB Workshop 2026)

C4 · Outcome and unit fit for deployment can facilitate deliberation over leaderboard trust

Ontology	Content	Form
E.g. Co-design of test cases and labels with target population.	E.g. filtration, translation.	E.g. collection in new modality.

C4 · Should I use the leaderboard winner?

Original claim: The leaderboard winner will be best for my product.

What holds: Leaderboard winner fits best for deployments it's a *good representative of*.

Breaks when: Leaderboard evaluates a different *outcome* or assumes a different *system*.

Better claim: The leaderboard winner will fit my product best if I can convince myself that:

1. The outcome being scored is the deployment outcome.
2. The unit being ranked is the unit being deployed.

C4 · The leaderboard winner will fit my product best

Vote again: hands up if you still agree.

Part II

Incentive-aware Robust Evaluation

"When a measure becomes a target, it ceases to be a good measure."

(Strathern 1997, Goodhart 1975, Campbell 1979)

Contexts: **education, monetary policy, social program evaluation**

Evaluation gaming is mostly cheating or data leakage

Gaming: raising the score without raising the capability.

Hands up if you agree.

C5 · Evaluation gaming is mostly cheating or data leakage

Why people believe it: Gaming means breaking the rules, so stopping leaks and cheaters stops gaming.

What we'll show: Developers follow the incentives an evaluation sets. Many ways to raise a score break no rule.

C5 · What does “evaluation gaming” really look like?

Strategy 1: Test data contamination

Strategy 2: Repeated model releases on the same leaderboard

Strategy 3: Optimizing for evaluated tasks

Strategy 4: Tailoring towards certain user populations

Strategy 5: Selective release of new evaluations

C5 · Strategy 1 · Contamination: training on test items

Strategy 1: Test data contamination

The model trains on a benchmark's test split, then is scored on that same benchmark

Training: Leaked MMLU questions

Evaluation: MMLU

Sainz, …, Agirre, NLP Evaluation in Trouble: On the Need to Measure LLM Data Contamination for Each Benchmark, Findings of EMNLP 2023 (arXiv 2310.18018)

C5 · Strategy 2 · Private variants: test many, publish the best

Strategy 2: Repeated model releases on the same leaderboard

Private releases can give an optimization advantage:

One provider tested 27 private variants before releasing one model

Singh, …, Koyejo, …, The Leaderboard Illusion, NeurIPS 2025 D&B (arXiv 2504.20879)

C5 · Strategy 2 · Public near-copies crowd the leaderboard

Strategy 2: Repeated model releases on the same leaderboard

Public releases can also help one company win:

Company B fills the leaderboard with near-copies of its own model.

Leaderboard:

1. Model A1
2. Model B1
3. Model B2
4. Model B3
5. Model B4
6. Model C1

C5 · Strategy 2 · Company B adds a near-copy

Independence of clones: *adding near-copies of a model should not change the winner.*

English-speaking users (66%):

A > B > B2

French-speaking users (34%):

B > B2 > A

Leaderboard:

1. Model A
2. Model B

Company B releases similar but slightly worse model B2

Tideman, Independence of Clones as a Criterion for Voting Rules, Social Choice and Welfare 1987; Procaccia, Schiffer, Zhang, Clone-Robust AI Alignment, ICML 2025; example adapted from Wikipedia, Independence of Clones Criterion

C5 · Strategy 2 · The near-copy flips the winner

Independence of clones: *adding near-copies of a model should not change the winner.*

English-speaking users (66%):

A > B > B2

French-speaking users (34%):

B > B2 > A

Leaderboard:

1. Model B 1.34
2. Model A 1.32
3. Model B2 0.34

Company B releases similar but slightly worse model B2

Ranked by summed pairwise win rate: adding B2 moves B above A, though no user changed their mind.

C5 · Strategy 3 · Training on the test task is not cheating

Strategy 3: Optimizing for evaluated tasks

Training on the test task is not cheating: it uses knowledge of the evaluated tasks at training time

Dominguez-Olmedo, Dorner, Hardt, Training on the Test Task Confounds Evaluation and Emergence, ICLR 2025 (arXiv 2407.07890)

C5 · Strategy 3 · Training toward the test: MMLU

Strategy 3: Optimizing for evaluated tasks

Example:

Training: including question answering templates in pretraining data

Evaluation: MMLU

Format practice raises MMLU scores without breaking any rule.

C5 · Strategy 3 · Training toward the test: SWE-bench

Strategy 3: Optimizing for evaluated tasks

Example:

Training: fine-tuning on coding tasks (e.g. Code-Qwen)

Evaluation: SWE-bench (fixing real GitHub issues)

Fine-tuning on coding tasks raises SWE-bench scores without breaking any rule.

C5 · Strategy 4 · Tailoring to who votes

Arena voters may not match the people who use chatbots; tailoring to voters can lift rank without serving everyone.

Strategy 4: Tailoring towards certain user populations

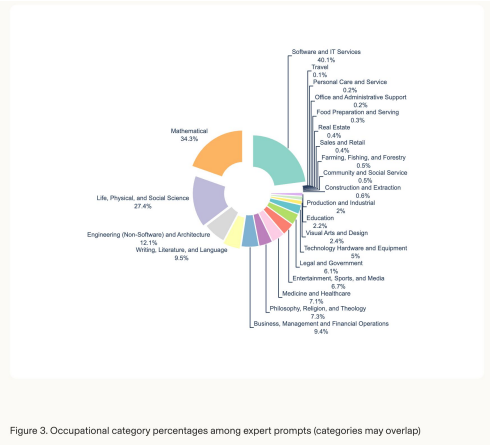
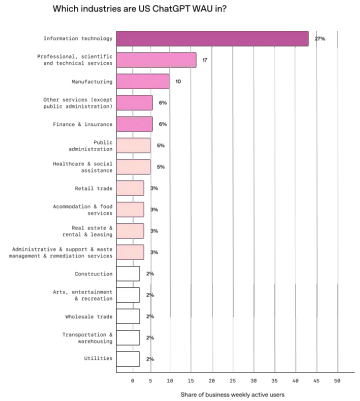


Figure 3. Occupational category percentages among expert prompts (categories may overlap)

[Arena Team 2025]



[Chatterji et al. 2025]



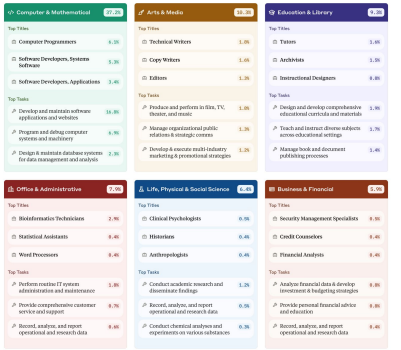


Figure 2: **Hierarchical breakdown of top six occupational categories by the amount of AI usage in their associated tasks.** Each occupational category contains the individual O*NET occupations and tasks with the highest levels of appearance in Claude.ai interactions.

[Handa et al. 2025]

Arena Team, Arena Expert and Occupational Categories, arena.ai blog, Nov 2025; Chatterji et al., How People Use ChatGPT, NBER Working Paper 34255, 2025; Handa et al., Which Economic Tasks Are Performed with AI? Evidence from Millions of Claude Conversations, arXiv 2503.04761, 2025

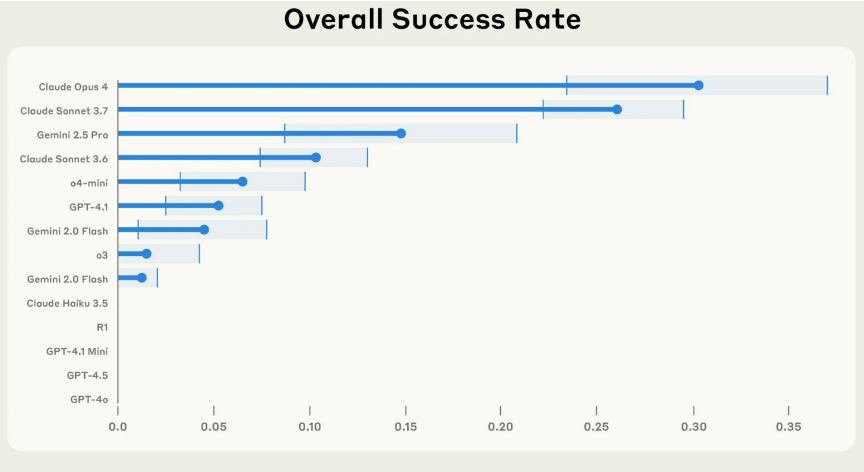
C5 · Strategy 5 · Choosing which evaluations to release

A developer that writes evaluations chooses which ones the field sees; evaluations it does poorly on never have to appear.

Strategy 5: Selective release of new evaluations

Developers build and release evaluations too: *SHADE-Arena: Evaluating sabotage and monitoring in LLM agents, by Anthropic*

An example of a developer-built evaluation, not evidence of selective release. In it, Gemini 2.5 Pro, not a Claude model, was the best monitor.



Kutasov et al., SHADE-Arena: Evaluating Sabotage and Monitoring in LLM Agents, arXiv 2506.15740, 2025

C5 · What holds, where it breaks, a better claim

Original claim: *“Evaluation gaming is mostly cheating or data leakage.”*

What holds: Contamination happens, and it inflates scores.

Where it breaks: Private variant testing, near-copies on a leaderboard, training on the test task, tailoring to the majority of raters, and choosing which evaluations to publish all raise scores without breaking a rule.

Better claim: Gaming is what developers do with the incentives an evaluation creates. Beyond “who cheated,” ask what the evaluation rewards.

Sainz, …, Agirre, NLP Evaluation in Trouble, Findings of EMNLP 2023 (arXiv 2310.18018); Singh, …, Koyejo, …, The Leaderboard Illusion, NeurIPS 2025 D&B (arXiv 2504.20879); Dominguez-Olmedo, Dorner, Hardt, Training on the Test Task Confounds Evaluation and Emergence, ICLR 2025 (arXiv 2407.07890)

C5 · Evaluation gaming is mostly cheating or data leakage

Vote again: hands up if you still agree.

A private, uncontaminated test set stops gaming

Hands up if you agree.

C6 · A private, uncontaminated test set stops gaming

Why people believe it: If developers never see the test items, they cannot overfit to them.

What we'll show: Developers still learn from returned scores, from what the test is likely to cover, and from who votes. Privacy closes one route.

C6 · How can we make evaluations more robust?

STRATEGY 1

Test data contamination

FIXES

Keep test sets private and uncontaminated

IN PRACTICE

This is the claim C6 tests.

STRATEGY 2

Repeated model releases on the same leaderboard

FIXES

Thresholdout
[Dwork et al. 2015]

Clone-robust ranking algorithms
[Hays et al. 2026; Procaccia et al. 2025]

STRATEGY 3

Optimizing for evaluated tasks

FIXES

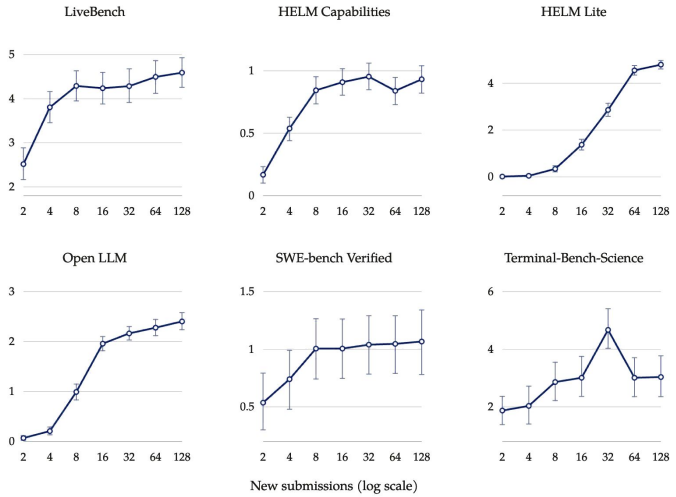
Evaluator post-training
[Dominguez-Olmedo et al. 2025; Chen et al. 2026]

Randomized evaluation
[Truong et al. 2026]

Dwork et al., The Reusable Holdout, Science 2015; Hays, Li, Flanigan, Raghavan, arXiv 2603.26891; Procaccia, Schiffer, Zhang, Clone-Robust AI Alignment, ICML 2025 (arXiv 2501.09254); Dominguez-Olmedo, Dorner, Hardt, ICLR 2025 (arXiv 2407.07890); Chen, Zhang, Hardt, arXiv 2603.08371; Truong, Wang, Haber, Koyejo, Strategic Evaluation, NeurIPS 2026

C6 · Returned scores leak information even when items stay private

- Attack sees only returned scores, never the items
- After 128 submissions, the top model is 1 to 5 points worse on fresh data
- LiveBench, one overall score: 64 submissions until 90% of leaders are false winners
- LiveBench, 18 task scores: only 4
- One fix: add noise to returned scores, as in Thresholdout



Allouah, Duchi, Koyejo, Fast but False Progress on Benchmarks with Richer Feedback, 2026; Allouah and Duchi, How Reusable Are Benchmarks with Richer Feedback?, arXiv 2609.32109

C6 · Public or private, benchmarks saturate

- 60 LLM benchmarks: nearly half no longer separate top models
- Older benchmarks saturate more
- Public or private test data: no difference
- Expert curation: slower saturation
- Privacy alone does not keep a benchmark useful

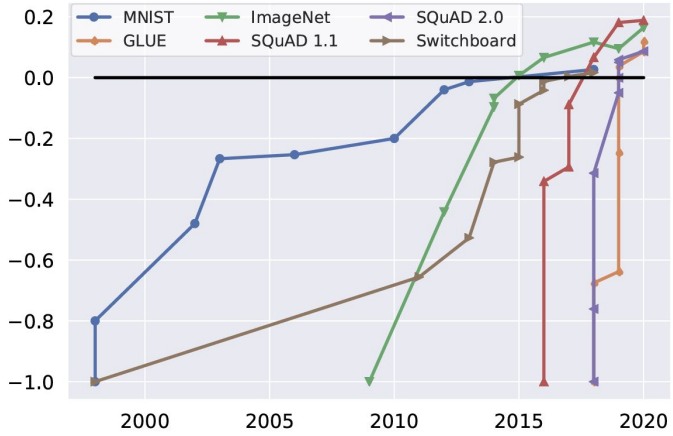


Figure 1: Benchmark saturation over time for popular benchmarks, normalized with initial performance at minus one and human performance at zero.

Akhtar, Reuel, …, Koyejo, When AI Benchmarks Plateau: A Systematic Study of Benchmark Saturation, ICML 2026 (arXiv 2602.16763)

C6 · How can we make evaluations more robust?

STRATEGY 1

Test data contamination

FIXES

Keep test sets private and uncontaminated

IN PRACTICE

This is the claim C6 tests.

STRATEGY 2

Repeated model releases on the same leaderboard

FIXES

Thresholdout
[Dwork et al. 2015]

Clone-robust ranking algorithms
[Hays et al. 2026; Procaccia et al. 2025]

STRATEGY 3

Optimizing for evaluated tasks

FIXES

Evaluator post-training
[Dominguez-Olmedo et al. 2025; Chen et al. 2026]

Randomized evaluation
[Truong et al. 2026]

Dwork et al., The Reusable Holdout, Science 2015; Hays, Li, Flanigan, Raghavan, arXiv 2603.26891; Procaccia, Schiffer, Zhang, Clone-Robust AI Alignment, ICML 2025 (arXiv 2501.09254); Dominguez-Olmedo, Dorner, Hardt, ICLR 2025 (arXiv 2407.07890); Chen, Zhang, Hardt, arXiv 2603.08371; Truong, Wang, Haber, Koyejo, Strategic Evaluation, NeurIPS 2026

C6 · Strategy 3 · Optimizing for evaluated tasks

Evaluation game

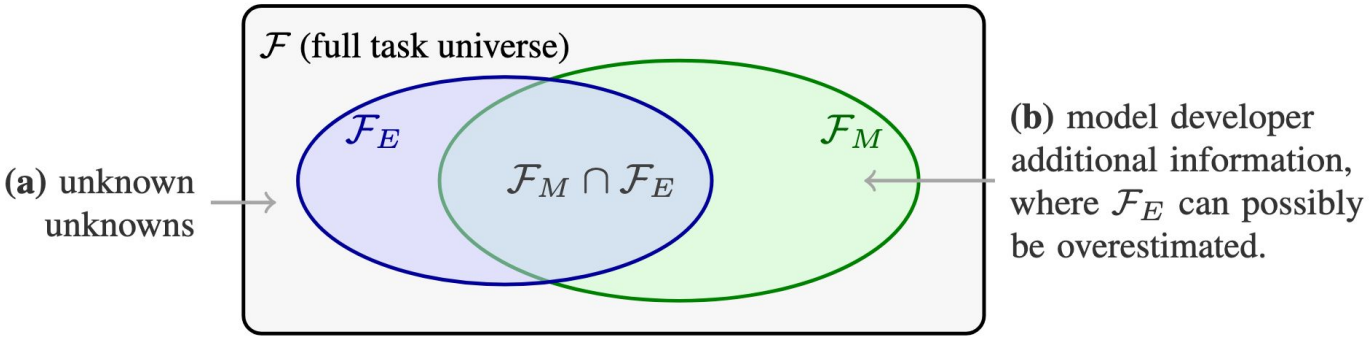
1. Evaluator picks the test tasks
2. Developer trains a model
3. Developer is rewarded for the score on those tasks

Evaluator’s goal: the developer covers every task users need.

Truong, Wang, Haber, Koyejo, Strategic Evaluation: Incentivizing AI Capability Coverage with Private Benchmarks, NeurIPS 2026

C6 · Strategy 3 · Fundamental problem: information asymmetry

The developer may know tasks the evaluator does not.



The evaluator announces the test. As the developer, what do you study?

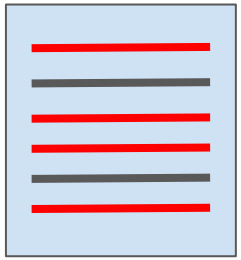
Truong, Wang, Haber, Koyejo, Strategic Evaluation: Incentivizing AI Capability Coverage with Private Benchmarks, NeurIPS 2026

C6 · Strategy 3 · Failure of deterministic evaluation

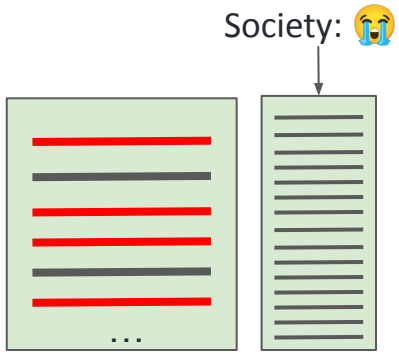
Evaluation game

1. Evaluator picks the test tasks
2. Developer trains a model
3. Developer is rewarded for the score on those tasks

Evaluator: "These are the questions that will be on the test."



Model developer: "Cool, I will focus on studying those questions."



Truong, Wang, Haber, Koyejo, Strategic Evaluation: Incentivizing AI Capability Coverage with Private Benchmarks, NeurIPS 2026

C6 · Strategy 3 · Failure of deterministic evaluation

Evaluation game

1. Evaluator picks the test tasks
2. Developer trains a model
3. Developer is rewarded for the score on those tasks

Theorem 2.3 (Failure of Published Deterministic Mechanisms)**.** *If $\mathcal{F}_E \subset \mathcal{F}$, then no deterministic mechanism can recover the optimal utility: for any non-degenerate r and deterministic S, there always exists $\mathcal{F}, \mu$ such that $u_E(\theta^M) < \max_{\theta \in \Theta} u_E(\theta)$.*

Announcing the test lets the developer study only those tasks.

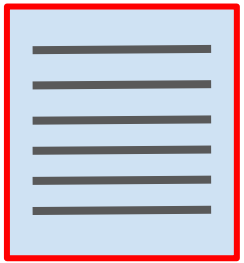
Truong, Wang, Haber, Koyejo, Strategic Evaluation: Incentivizing AI Capability Coverage with Private Benchmarks, NeurIPS 2026

C6 · Strategy 3 · Solution: randomized evaluation

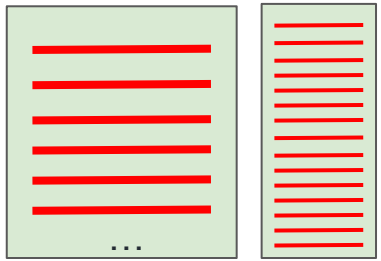
Evaluation game

1. Evaluator draws the test tasks **at random from a hidden bank**
2. Developer trains a model
3. Developer is rewarded for the score on those tasks

Evaluator:
"The questions on the test will be drawn at random from this hidden bank."



Model developer:
"I will study any questions that I think might be in the bank."



Truong, Wang, Haber, Koyejo, Strategic Evaluation: Incentivizing AI Capability Coverage with Private Benchmarks, NeurIPS 2026

C6 · Strategy 3 · Solution: randomized evaluation

Evaluation game

1. Evaluator draws the test tasks **at random from a hidden bank**
2. Developer trains a model
3. Developer is rewarded for the score on those tasks

Takeaway: randomization can be helpful when there is information asymmetry.

Truong, Wang, Haber, Koyejo, Strategic Evaluation: Incentivizing AI Capability Coverage with Private Benchmarks, NeurIPS 2026

C6 · How can we make evaluations more robust?

STRATEGY 4

Tailoring towards certain user populations

FIXES

Guarantee representation in rankings
[Haghtalab et al. 2026]

Evaluate on worst-case performance
[Khalaf et al. 2026]

Correct for biases in the user population
[Wang et al. 2024]

STRATEGY 5

Selective release of new evaluations

FIXES

Adversarial evaluation

Disclosure incentives
[Wang et al. 2025]

IN PRACTICE

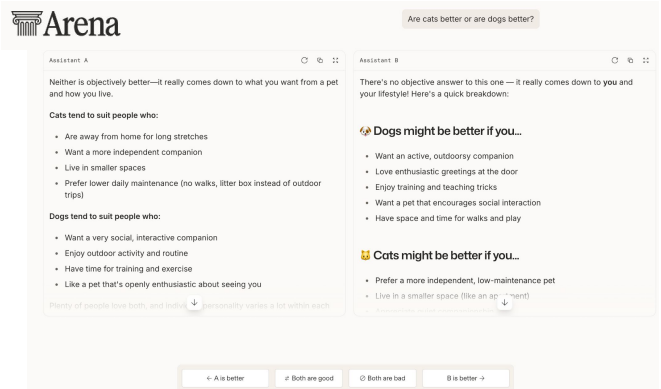
A stylized model: the evaluator is the principal, the developer is the agent, and the disclosed metric is a benchmark.

Haghtalab, Procaccia, Shao, Wang, Yang, Pluralistic Leaderboards, arXiv 2606.02547; Khalaf et al., arXiv 2602.21297; Wang, Bates, Aronow, Jordan, On Counterfactual Metrics for Social Welfare, AISTATS 2024 (arXiv 2305.14595); Wang, Jordan, Ligett, McAfee, Relying on the Metrics of Evaluated Agents, WWW 2025

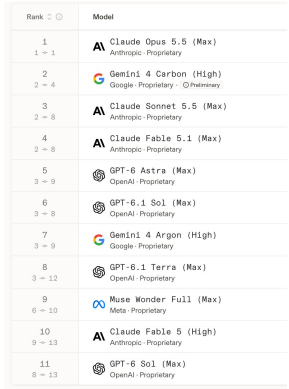
C6 · Strategy 4 · Tailoring towards certain user populations

Arena leaderboards: users compare two answers; models are ranked by average win rate (or Elo).

Pairwise comparisons



Leaderboard



Chiang et al., Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference, ICML 2024 (arXiv 2403.04132);
Zhao et al., SciArena: An Open Evaluation Platform for Non-Verifiable Scientific Literature-Grounded Tasks, NeurIPS 2025 D&B (arXiv 2507.01001)

C6 · Strategy 4 · Tailoring towards certain user populations

Arena leaderboards: users compare two answers; models are ranked by average win rate (or Elo).

- Problem: the majority's taste sets the ranking
- Users who disagree are ignored
- So every developer tailors to the majority

C6 · Strategy 4 · Where “average win rate” rankings go wrong

Ask the room: which model should be ranked first?

60%	40%
a	f
b	e
c	d
d	c
e	b
f	a

Profile 1

10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
a	c	c	a	b	a	a	b	b	a
b	b	d	e	e	d	d	c	c	c
c	d	a	b	d	e	e	e	e	d
f	f	f	f	f	f	f	f	f	f
d	e	b	d	a	c	b	a	d	e
e	a	e	c	c	b	c	d	a	b

Profile 2

Leaderboard:

$a \succ b \succ c \succ d \succ e \succ f$

Ranking by average win rate follows the majority; users who prefer another model are left out.

Haghtalab, Procaccia, Shao, Wang, Yang, Pluralistic Leaderboards, arXiv 2606.02547, 2026

C6 · Strategy 4 · Solution: pluralistic leaderboards

Intuition

- For decisions: the ranking stays useful to the 40% who disagree
- For incentives: model f gets credit for serving 40% of users

60%	40%
a	f
b	e
c	d
d	c
e	b
f	a

Profile 1

Locally stable leaderboard:

$$a \succ f \succ b \succ c \succ d \succ e$$

Haghtalab, Procaccia, Shao, Wang, Yang, Pluralistic Leaderboards, arXiv 2606.02547, 2026

C6 · Strategy 4 · Solution: pluralistic leaderboards

Definition: a ranking is *locally stable* if for any candidate *x* not in the top *k* ranks, the fraction of users who prefer *x* to all candidates in the top *k* ranks is at most 1/k.

In words: a model that a large group of users prefers cannot be left out of the top.

60%	40%
a	f
b	e
c	d
d	c
e	b
f	a

Profile 1

Locally stable leaderboard:

$$a \succ f \succ b \succ c \succ d \succ e$$

Haghtalab, Procaccia, Shao, Wang, Yang, Pluralistic Leaderboards, arXiv 2606.02547, 2026

C6 · What holds, where it breaks, a better claim

Original claim: *“A private, uncontaminated test set stops gaming.”*

What holds: Private, uncontaminated test sets block training on the test items, the most direct route to inflated scores.

Where it breaks: Returned scores, likely coverage, and the mix of raters still steer development. Randomized private draws and pluralistic leaderboards address routes that privacy alone does not.

Better claim: Privacy closes one route. Ask what the leaderboard rewards: does it push models to improve where users need it, and does it help people decide?

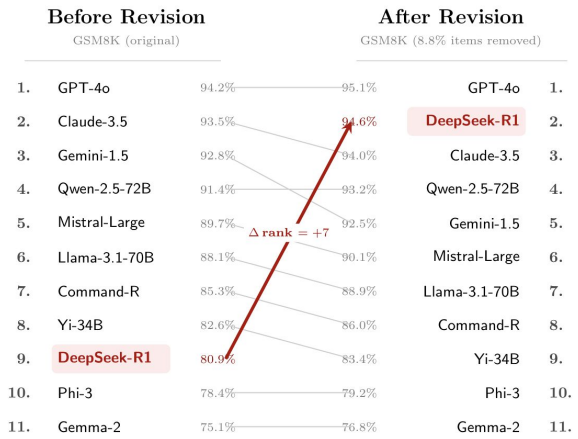
Truong, Wang, Haber, Koyejo, Strategic Evaluation: Incentivizing AI Capability Coverage with Private Benchmarks, NeurIPS 2026; Haghtalab, Procaccia, Shao, Wang, Yang, Pluralistic Leaderboards, arXiv 2606.02547, 2026

C6 · A private, uncontaminated test set stops gaming

Vote again: hands up if you still agree.

Conclusion

Back to the revised leaderboard



C1 Is “math reasoning” what GSM8K shares with other math benchmarks?

C2 **Could we have compared models on fewer questions?**

C3 After translation, are scores still comparable?

C4 Is the winner right for my tutoring product?

C5 Who trained toward GSM8K, and was that cheating?

C6 Would a fresh private split settle it?

Truong, …, Koyejo, Fantastic Bugs and Where to Find Them in AI Benchmarks, NeurIPS 2025 D&B (arXiv 2511.16842)

Diagnose the error before choosing the fix

Sampling error

C2 Audit items; collect enough evidence

Specification error

C1 Validate the shared construct

C3 Test transfer empirically

C4 Validate the deployment decision

Strategic error

C5 Gaming often follows the incentives an evaluation sets; cheating is one route

C6 Privacy closes one route; ask what the leaderboard rewards

Messick, Validity, in Educational Measurement (3rd ed.), 1989

An evaluation checklist

1 Define the claim	Construct, target population, deployment decision
2 Inspect the items	Relevance, scoring, construct-irrelevant variance (score changes unrelated to the skill)
3 Test the inference	Uncertainty, transfer across groups, the deployment outcome
4 Examine incentives	How could optimization raise the score without improving the outcome?
5 Report	Item-level responses and the full protocol: harness, scorer, prompts, runs

Jiang, …, Koyejo, AI Evaluation Should Require Standardized Item-Level Data Releases, NeurIPS 2026; Ghosh, …, Koyejo, Evaluation Cards, NeurIPS 2026

Three open problems

Ability moves between evaluations

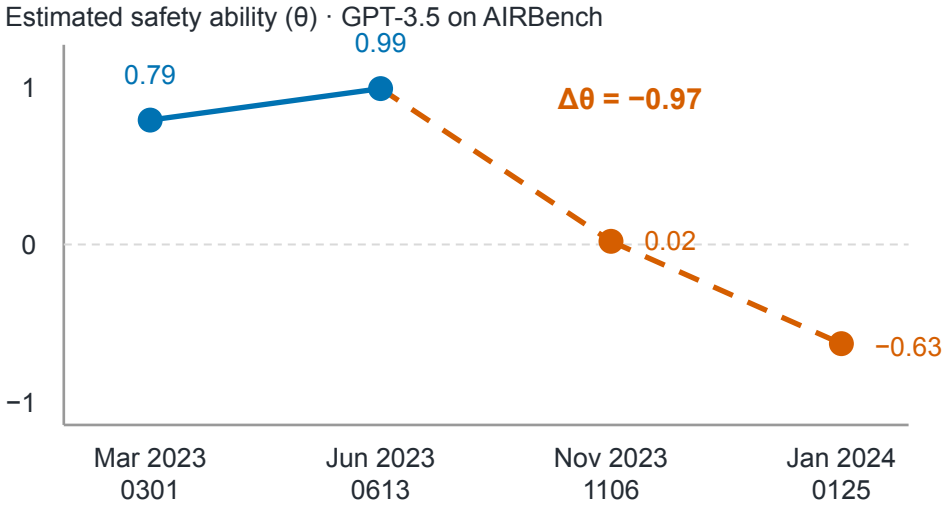
GPT-3.5 safety ability fell by about one unit on the ability scale in one update

The scorer is part of the instrument

LLM judges need validating against people; prompting changes their consistency

Generality claims need forecasts that could fail

Register predictions before the tasks exist, then grade them (as the CASP protein-structure contest does)



Truong, Tu, Liang, Li, Koyejo, Reliable and Efficient Amortized Model-based Evaluation, ICML 2025; Truong, …, Koyejo, A Measurement Science Roadmap, arXiv 2026; Unell, …, Koyejo, Metric Match, TAE@NeurIPS 2026; Chen, …, Koyejo, Rethinking LLM Judges, ICLR 2026 Workshop on Logical Reasoning of LLMs; Moult et al., CASP, Proteins 1995

A benchmark score is evidence for a decision. Say which decision, and how the score could rise without the capability rising.

Thank you

Questions & discussion

What evidence would make you trust a benchmark score for your use case?