Language Model (LM) Evaluations

Pre-Training and Post-Training Evaluation Techniques

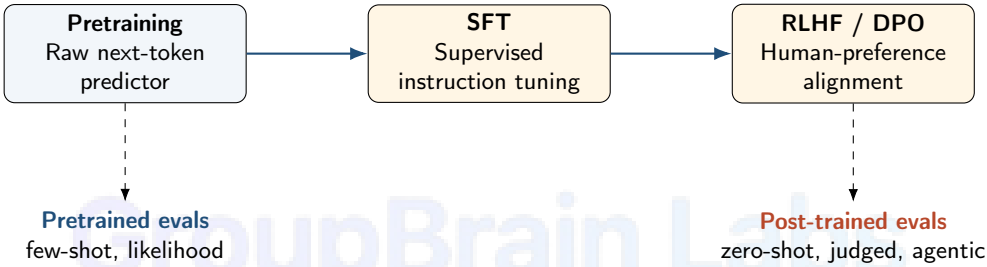
Harsh Raj - IIT Mandi

GroupBrain.ai Talks

August 16, 2026

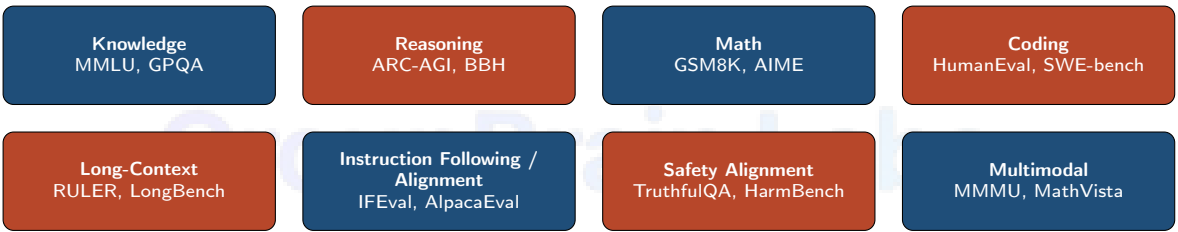
Contents

The Life Cycle of a Language Model



Same model lineage, but two entirely different measurement problems.
Capabilities such as instruction following, preference alignment, and reliable structured outputs illustrate why post-training is important.
Recently released Qwen3.5 combines large-scale multimodal pretraining with extensive post-training and a sparse Mixture-of-Experts architecture to improve the performance.

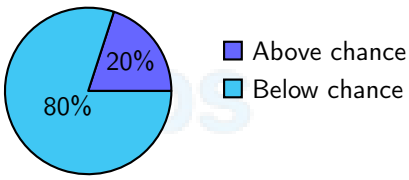
The 8 Capabilities of LLM Evaluation



Every benchmark in this talk maps onto one (or more) of these 8 axes.

MMLU — The Reference for Knowledge Benchmark

- Hendrycks et al. (2020): 57 subjects, elementary math to law
- Requires ***extensive world knowledge + problem solving***
- At release: even the largest GPT-3 improved only ∼20 pts over random chance
- Near-random on morality and law subjects



Hendrycks et al., arXiv:2009.03300

Saturated MMLU ⇒ MMLU-Pro / MMLU-CF

MMLU-Pro (2024)

- 4 → 10 answer choices
- More reasoning-heavy questions
- 16–33% accuracy *drop* vs. MMLU
- Greater prompt-robustness

MMLU-CF (2024)

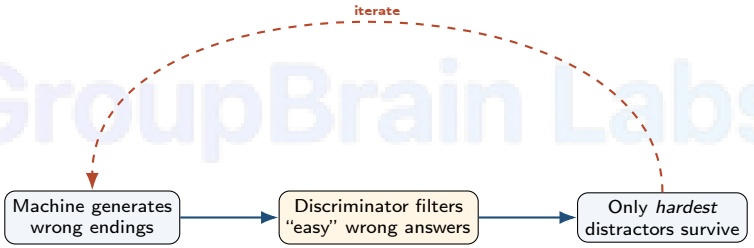
- Contamination-free design
- Closed validation/test splits
- Matched difficulty distributions

Wang et al., arXiv:2406.01574 Zhao et al., arXiv:2412.15194

HellaSwag — Adversarial Filtering

Predecessor: SWAG. Context + 4 candidate endings, pick the right one — but BERT's arrival marked SWAG's saturation (>90% accuracy), so it stopped testing real understanding.

HellaSwag's fix — two moves: (1) generate wrong endings adversarially and iterate against a discriminator until only the hardest survive; (2) run diagnostic ablations to check *what* models were actually learning.



Result: humans >95% accuracy, contemporary SOTA models <48%.

Diagnostic check: is the model reasoning, or pattern-matching?

Ablating the wrong-ending distractors — **removing the context** entirely, and separately **randomizing word order** within endings — barely hurt BERT's performance. A clear sign it was exploiting *statistical surface patterns*, not natural language understanding.

Rounding Out the Pretraining Suite: Perplexity

Perplexity is the core intrinsic metric for a pretrained model — measured directly on next-token prediction, no downstream task needed.

Intuition

How "surprised" the model is by the true next token.

- **Low** PPL $\Rightarrow$ confident, accurate predictions
- **High** PPL $\Rightarrow$ uncertain, probability spread thin

"The effective number of equally-likely next-word choices the model is juggling."

Mathematically

For a sequence $x_1, \ldots, x_N$:

$$\mathrm{PPL}(x) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(x_i \mid x_{<i})\right)$$

- Exponential of average negative log-likelihood
- $= 2^{H(x)}$ (bits) $= e^{H(x)}$ (nats)
- Perfect model $\Rightarrow$ PPL $= 1$; uniform-random over vocab $V \Rightarrow$ PPL $= V$

Why it's not enough on its own

Corpus- and tokenizer-dependent, so scores aren't comparable across models — and low perplexity doesn't guarantee good reasoning or instruction-following. Hence the layered task-specific benchmarks (MMLU, HellaSwag, etc.).

The Foundational Result after InstructGPT White Paper

"Training language models to follow instructions with human feedback"

Human labelers preferred outputs from a **1.3B-parameter** RLHF-tuned model over the raw **175B-parameter** GPT-3 — *despite 100× fewer parameters*.

- Improved truthfulness, reduced toxic generation
- Minimal regression on public NLP datasets

Why it mattered: parameter count and pretraining-eval scores *stopped predicting* human-perceived quality. Hence, Post-training needed its own eval paradigm.

TruthfulQA — A Counter-Intuitive Finding

Lin, Hilton, Evans (2021) — arXiv:2109.07958

817 questions across 38 categories, crafted so humans *would* answer falsely due to common misconceptions.

Larger models were *less* truthful.

Scaling improves a model's ability to *imitate confident human-sounding falsehoods* — it does not automatically improve truth-discrimination. Truthfulness must be targeted explicitly in post-training.

Frontier Knowledge: GPQA

Rein et al. (2023) — arXiv:2311.12022

448 PhD-level MC questions in biology, physics, chemistry — deliberately ***"Google-proof."***

Group	Accuracy
Domain PhD experts	65–74%
Skilled non-experts (30+ min, web access)	34%
Best GPT-4 baseline at release	39%

Humanity's Last Exam extends this philosophy further as an expert-curated, saturation-resistant frontier test.

Math: From GSM8K to Frontier Competition Math

Cobbe et al. (2021) — arXiv:2110.14168

8.5K grade-school math word problems. Even large transformers struggled with multi-step arithmetic despite conceptual simplicity.

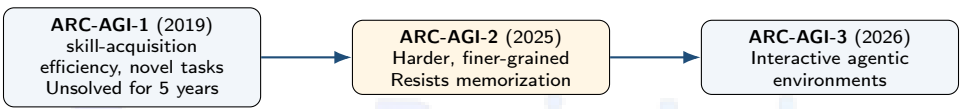
Key methodological fix: train a *verifier*

Sample many candidate solutions → rank with a separate verifier model → pick the best.

- Outperforms simply fine-tuning harder
- **Scales better with data** than fine-tuning alone

This generate-then-verify pattern foreshadows today's chain-of-thought + self-consistency methods.

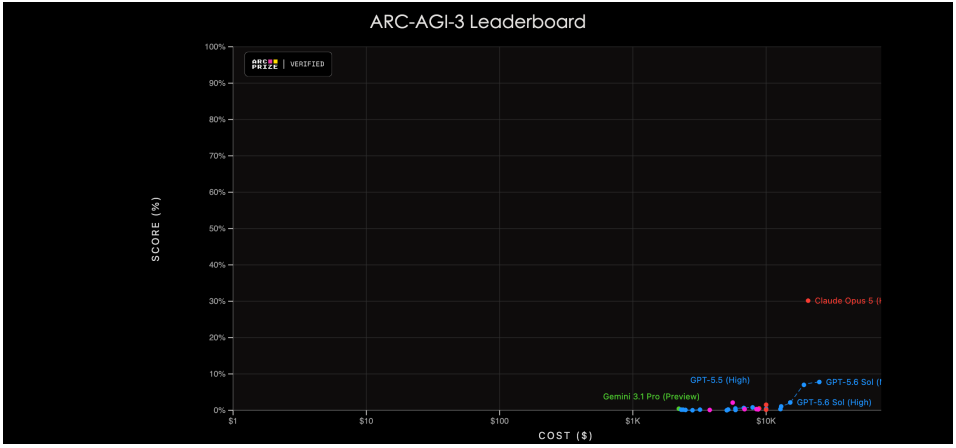
ARC-AGI: 1 → 2 → 3



Core philosophy: intelligence as ***skill-acquisition efficiency*** using only innate human core-knowledge priors — object persistence, counting, basic geometry — *not* task-specific practice. Chollet, arXiv:1911.01547

ARC-AGI-2, arXiv:2505.11831 ARC-AGI-3, arXiv:2603.24621

ARC-AGI-3 Leaderboard — Score vs. Cost



ARC-AGI-3 Leaderboard — Key Takeaways

- **Scores are still near the floor.** Even the leading model sits around 30%, far below the >95% human baseline seen on earlier ARC benchmarks — ARC-AGI-3's interactive, agentic environments remain largely unsolved
- **Claude Opus 5 leads on accuracy**, but at one of the highest costs on the board — performance and cost are not moving together
- **Steep cost–score tradeoff:** most models cluster in the sub-10% range regardless of spend; GPT-5.6 Sol shows large cost increases with only marginal score gains, then a late jump
- **Efficiency frontier is thin:** Gemini 3.1 Pro (Preview) achieves a low score cheaply, but no model yet combines low cost *and* high score
- **Reinforces the lifecycle story:** ARC-AGI-3 is deliberately *active/unsaturated* — unlike HellaSwag or MMLU, it is currently discriminating meaningfully between frontier models

ARC Prize Foundation, ARC-AGI Leaderboard (verified)

SWE-bench — Real-World Agentic Coding

2,294 real GitHub issue/PR pairs across 12 Python repos.

- Requires multi-file coordination
- Requires interaction with execution environments
- *Not* answerable via few-shot likelihood scoring

At release

Best model (Claude 2) solved just **1.96%** of issues.

Two Structural Problems

Saturation

Scores cluster near the ceiling $\Rightarrow$ benchmark can no longer discriminate between models.

Example: MMLU $\rightarrow$ MMLU-Pro (16–33% harder)

Contamination

Benchmark Q&A pairs leak into pretraining corpora $\Rightarrow$ inflated scores that don't reflect genuine capability.

Now a dedicated subfield with multiple 2024–2025 surveys.

Xu et al., arXiv:2406.04244 Recent Advances survey, arXiv:2502.17521

Mitigation Strategies for Saturation and Contamination

- **Held-back / private test splits** — never published (MMLU-CF)
- **Dynamic / procedurally regenerated benchmarks** — resist static memorization
- **Post-hoc contamination detection** — membership inference, "time-travel" probing
- **Deliberately expert-authored / Google-proof design** — GPQA sidesteps contamination by requiring genuine synthesis, not retrievable facts

Benchmark Reproduction — Why It is Its Own Problem

The reproducibility gap

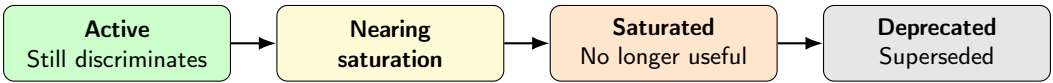
Two labs can report different scores on the *same* benchmark because of different prompt templates, few-shot counts, decoding parameters, or answer-extraction rules — not because the models differ.

- **Stanford HELM** — taxonomizes scenarios & metrics; raised models sharing a common evaluated scenario set from 17.9% to 96% (Liang et al., arXiv:2211.09110)
- **Open LLM Leaderboard (Hugging Face)** — crowd-facing, standardized re-runs of self-reported numbers

Vendor-Reported vs. Independently Verified

This is exactly the distinction tracked by benchmark-monitoring tools — a direct downstream consequence of saturation + contamination.

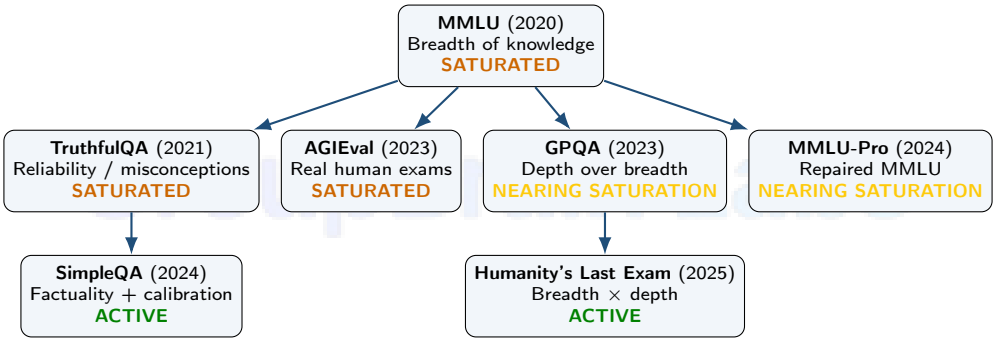
Benchmark Lifecycle



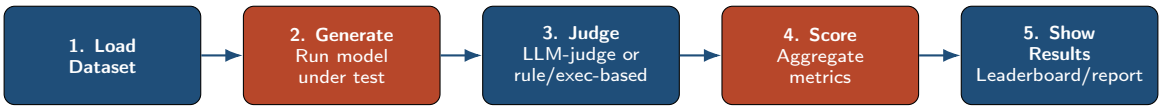
Rule of thumb

MMLU was *active* in 2020, *nearing saturation* by 2023, *largely saturated* for frontier models by 2024 — which is exactly why MMLU-Pro and MMLU-CF exist. A benchmark's stage is not fixed.

Lifecycle in Practice: The Knowledge-Benchmark Lineage



Eval Harness — The Pipeline



- **Load Dataset** — pull prompts/questions + gold references (e.g. MMLU questions, SWE-bench issues)
- **Generate** — run the *model under test* on every example, with fixed decoding settings
- **Judge** — decide correctness: an *LLM-as-judge* scores/compares responses, or a rule-based/execution-based check is used instead
- **Score** — aggregate per-example judgments into accuracy, pass@k, win-rate, etc.
- **Show Results** — push to a leaderboard, dashboard, or report for comparison across models/runs

Beyond the Basic Flow: Modern Eval Harnesses

The Load → Generate → Judge → Score → Show pipeline is the classic ***offline batch / LLM-as-a-Judge*** paradigm — but it's a *linear, one-shot* view. Two widely-used harnesses extend it in different directions:

Inspect AI (UK AISI)

Built for **stateful, agentic** evaluation — the model acts inside a sandboxed environment across multiple turns, not a single prompt-response pair. Core use case: *capability & safety red-teaming*.

Arize AI (Phoenix)

Built for **observability** — it traces every span of a live application (retrieval, guardrails, generation) rather than judging one final output. Core use case: *production monitoring & CI/CD*.

The shift

Both add *dynamic interactions*, richer judging targets, and workflows the basic 5-step flow doesn't capture on its own.

Judging Stage — LLM-as-a-Judge

Zheng et al. (2023) — ***MT-Bench*** (80 multi-turn Qs) + ***Chatbot Arena*** (crowdsourced pairwise battles)

- A separate "judge" model generates a verdict/score for the response produced by the model under test
- GPT-4 judges: >80% agreement with humans
- Same level as human–human agreement

Documented biases

- **Position bias** — favors first-shown answer
- **Verbosity bias** — favors longer answers
- **Self-enhancement bias** — favors own model family

Judging Stage — Other Ways of Judging

Not every benchmark needs an LLM judge — the right method depends on whether the answer is verifiable.

- **Rule-based / exact-match** — extracted answer string vs. gold answer (MMLU, GSM8K, GPQA)
- **Execution-based** — run generated code against unit tests, check pass/fail (HumanEval, SWE-bench)
- **Human evaluation** — crowdsourced pairwise preference voting (Chatbot Arena)
- **Reward-model scoring** — a trained scalar preference model scores each response (used inside RLHF-style pipelines)
- **Similarity-based** — embedding/overlap scores against a reference answer (ROUGE, BERTScore) for open-ended generation

Risk Mitigation Techniques for LLM-as-a-judge Pipelines

If you build an LLM-as-judge pipeline...

- **Randomize** answer order to counter position bias
- **Control for length** to counter verbosity bias
- Use a **different model family** to judge, when possible, to counter self-enhancement bias

Scoring & Reporting

- **Aggregate** per-example judgments into a single comparable number per model: accuracy, pass@k, win-rate, Elo, mean reward
- **Keep per-example results**, not just the aggregate — needed for error analysis and slicing by category
- **Show Results** on a leaderboard or dashboard (e.g. Open LLM Leaderboard, Chatbot Arena) so runs are comparable over time
- **Track regressions** across model versions and harness/prompt-template changes, not just a single snapshot score

Why Coding Is a Good Capability to Dive Deep

- **Uniquely verifiable** — unit tests give an objective pass/fail signal, unlike open-ended chat quality
- **Spans multiple evaluation stages** — from evaluating basic code generation to complex agentic coding capabilities
- **Bridges static and agentic evaluation** — from isolated code generation to multi-file, execution-environment tasks
- **Currently the least-saturated frontier** — best agentic coding models still fail most real-world issues

Deep Dive — Where Coding Evals Are Headed

- **SWE-bench Verified / SWE-bench Pro** — human-validated subsets, harder real-world repos
- **LiveCodeBench** — continuously refreshed with new competitive-programming problems to resist contamination
- **Terminal-Bench** — full shell/terminal agentic task completion, beyond single-repo patches
- **MBPP+** — expanded test cases catching edge-case failures HumanEval missed

Takeaway

Coding evaluation is moving from "does this function pass a test" to "can this agent resolve a real engineering task end-to-end" — the same static-to-agentic shift happening across every capability in this talk.

Live Demo — Playing an ARC-AGI-3 Environment

```
import arc_agi
from arcengine import GameAction

arc = arc_agi.Arcade()
env = arc.make("ls20", render_mode="terminal")

# Take a few actions
for _ in range(10):
    env.step(GameAction.ACTION1)

print(arc.get_scorecard())
```

- Spin up the ARC-AGI-3 arcade and load an interactive environment (`ls20`)
- Step through it with agent actions, rendered live in the terminal
- Pull a scorecard summarizing performance on the task

What to watch for

This is exactly the interactive, agentic setting from the ARC-AGI-3 leaderboard earlier — scores are still near the floor, so watch how the agent explores, fails, and (maybe) recovers, live.

Questions?

Thank you

Harsh Raj
GroupBrain.ai Talks

References I

- Hendrycks et al. *Measuring Massive Multitask Language Understanding.* arXiv:2009.03300
- Wang et al. *MMLU-Pro.* arXiv:2406.01574
- Zhao et al. *MMLU-CF.* arXiv:2412.15194
- Zellers et al. *HellaSwag.* arXiv:1905.07830
- Ouyang et al. *Training LMs to Follow Instructions with Human Feedback.* arXiv:2203.02155
- Zheng et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.* arXiv:2306.05685
- Lin, Hilton, Evans. *TruthfulQA.* arXiv:2109.07958
- Rein et al. *GPQA.* arXiv:2311.12022
- Jimenez et al. *SWE-bench.* arXiv:2310.06770
- Cobbe et al. *Training Verifiers to Solve Math Word Problems (GSM8K).* arXiv:2110.14168
- Chen et al. *Evaluating LLMs Trained on Code (HumanEval).* arXiv:2107.03374
- Xu et al. *Benchmark Data Contamination of LLMs: A Survey.* arXiv:2406.04244
- *Recent Advances in LLM Benchmarks against Data Contamination.* arXiv:2502.17521
- Chollet. *On the Measure of Intelligence.* arXiv:1911.01547
- Chollet et al. *ARC-AGI-2.* arXiv:2505.11831

References II

- ARC Prize Foundation. *ARC-AGI-3.* arXiv:2603.24621
- Liang et al. *Holistic Evaluation of Language Models (HELM).* arXiv:2211.09110
- Inspect AI — UK AI Safety Institute, https://inspect.ai-safety-institute.org.uk
- Arize AI / Phoenix — https://phoenix.arize.com