

Modern AI systems: agents and system optimizations

Agent design IV: evaluation

Juncheng Yang



Harvard John A. Paulson
School of Engineering
and Applied Sciences



HARVARD UNIVERSITY

MaDSys

Measurement and Design of Systems Lab

Logistics

- Office hour this Thursday 3:30pm SEC 4.308
- Final project poster presentation: 9am-1pm Dec 7
 - please reach out before **end of this week** if the time does not work for you
- Assignment 2 due this Sunday (start early!)

Logistics: Claude Code session log

- Package everything related (missing part of full session log will incur penalty from Assignment 2)
 - ~/.claude/projects/ (MacOS and Linux), C:\Users\<YourUsername>\.claude\projects\ (Windows)
 - ok to upload more
- You are not penalized on AI usage, however, certain parts of your assignment should be your work
 - 17 students used Claude to help with writeup
 - 6 used to polish writing 😊
 - a few created bullet points and ask Claude to write 😊
 - 7 wrote a draft and edited it (not encouraged), a few pasted from other places (ChatGPT?)
 - 4 have almost exact output from Claude, 2 have TODOs from Claude Code in the writeup
 - A **short** write up: a reflection for you
- **Do NOT manipulate your session logs! This will be treated as a violation of academic integrity**
 - three students tried to erase the history
 - AI-generated writing is easy to detect (even without history)
 - perfectly hiding all your history is harder than your assignments. 😞

Plan for today

- Agent evaluation
- Cost-effective agents

How do you evaluate an agent's capability?

- History and trend
 - Metrics
- Representative benchmarks
 - Challenges

Different ways to evaluate an agent

Component-level evaluation

Inside the agent



Planning

Task decomposition



Tool use

Discovery, selection, parameterization



Memory

Extraction, consolidation, retrieval



Error recovery



End-to-end evaluation



Success rate

Did the task get done?



Cost

Spend per task



Latency

Time to finish the task

A bit of history

Before 2018

Language modeling, narrow tasks

Penn Treebank, WikiText, LAMBADA, SQuAD

2018–2019

Transfer learning, general NLU

GLUE, SuperGLUE, ARC, HellaSwag, WinoGrande

2020–2022

Foundation-model capability

MMLU, BIG-bench, GSM8K, MATH, HumanEval, TruthfulQA, HELM

2023–2024

Agents and tool use, context, modalities

MT-Bench, Chatbot Arena, AlpacaEval, IFEval, LongBench, MMMU, GPQA, AgentBench, WebArena, GAIA, SWE-bench, OSWorld, τ -bench

2024–2025

Difficulty, robustness, freshness

MMLU-Pro, LiveBench, LiveCodeBench, MMMU-Pro, FrontierMath, Humanity's Last Exam

2025–2026

Larger scope, less specific instructions, more realistic use cases, newer problems

GDPval, BrowseComp, Terminal-Bench, ARC-AGI-3...

<https://cs2680.com/notes/agent-evaluation>

Evaluation objectives

Evaluation objective

Representative benchmarks

What success means

Evaluation objectives

Evaluation objective	Representative benchmarks	What success means
Predict text	PTB, WikiText	Assign high probability to observed text
Produce a <i>correct</i> answer	MMLU, GSM8K, GPQA, HLE	Match a known answer
Produce a <i>preferred</i> response	Arena, AlpacaEval, MT-Bench	Receive favorable human or model judgments
Produce <i>functioning</i> code	HumanEval, LiveCodeBench, SWE-bench	Pass executable tests
Complete an interactive task	WebArena, OSWorld, τ -bench, Terminal-Bench	Achieve the required environment state

Metrics

- Earlier benchmarks
 - Perplexity: exponentiated average negative log-likelihood on held-out text (lower is better)
 - Accuracy or F1 score
 - Preference or win rate
- Capability
 - Pass@ k : at least one out of k solutions passes
 - Pass ^{k} : all k solution passes (introduced by τ -bench)
- Cost
 - Mostly LLM cost
 - Overlooked: tool calls and others (e.g., LLM cache miss)
- Latency/duration
 - Not just LLM

Benchmark evaluation loop

```
for task in benchmark:
    environment = reset_to_task_initial_state(task)
    while not terminated and within_budget:
        observation = environment.observe()
        action = agent(task_instruction, observation, history, tools)
        environment.execute(action)

    score = task_evaluator(
        agent_final_answer,
        final_environment_state,
        trajectory
    )
```

GAIA: General AI Assistant

- 466 human-written questions
 - web browsing, spreadsheets, PDFs, images, audio, calculation
- Short answers: a number, string or list
 - e.g. 17.4 or 4,381
- No fixed tool API
 - each agent brings its own search, browser, Python, shell
- Tests the whole agent system, not just the LLM

Difficulty levels

Level 1

≤ 5 steps, zero or one tool

Level 2

5–10 steps, multiple tools

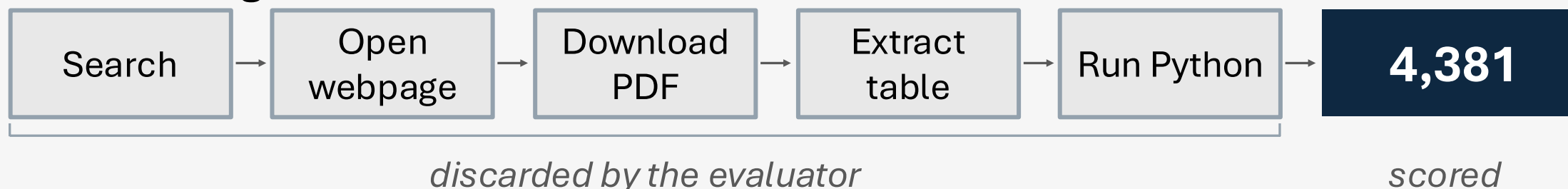
Level 3

arbitrarily long, many tools

GAlA: can an assistant research and return the right answer?

Example task: An attached spreadsheet contains a list of films. Identify the film satisfying two conditions, look up its director, and report the director's birthplace.

What the agent did



Scoring: Only the final answer is scored:

$$\text{score} = (1/N) \sum_i 1[\text{normalize}(\hat{y}) = \text{normalize}(y)]$$

Measures

reasoning, retrieval, tool selection and synthesis, with clean grading

Does not measure: actions

send email, book a flight, modify a database, submit a form, update an issue

BFCL: is the model's tool interface reliable?

The prompt

User

Book the cheapest flight from Boston to Chicago next Monday.

Available tools

```
search_flights(...)  
book_flight(flight_id,  
            passenger_id)  
get_weather(...)
```



Which calls does the model make?

Correct: search, pick the cheapest, then book
`search_flights(origin="BOS",
 destination="CHI", date=...)
book_flight(flight_id=...,
 passenger_id=...)`

Invents an ID

should ask the user for
passenger_id

Irrelevant call

`get_weather(...)`
not needed to book

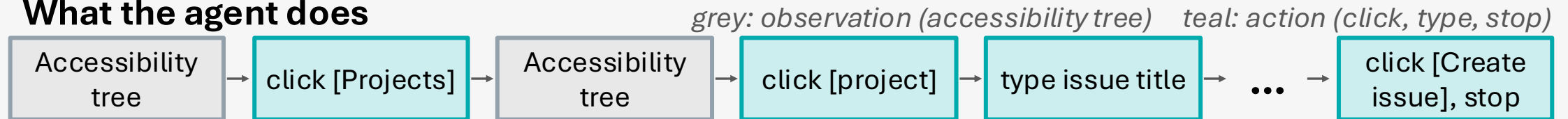
Scoring: AST or execution checks; multi-turn checks state + reply every turn

V4 adds: web search, memory; read category scores, not the composite

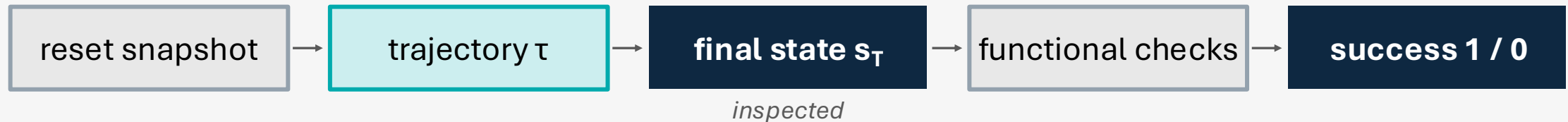
WebArena: can the agent complete a web workflow?

Example task: *In GitLab, create the issue “Update onboarding documentation,” assign it to Alex, and add the documentation label*

What the agent does



How a task is scored



812 tasks
self-hosted
shopping, admin,
Reddit, GitLab,
maps, Wikipedia

Text observations
accessibility tree;
click or type on
element IDs

Issue task checks
the issue exists with
the right assignee
and label

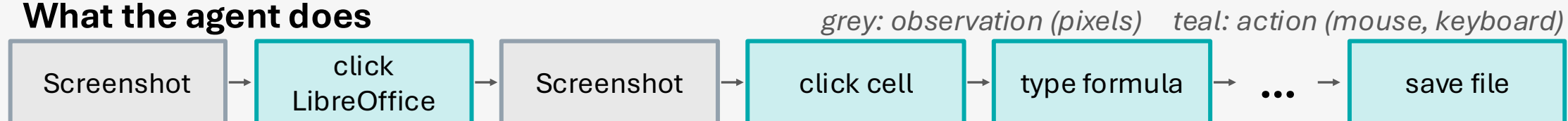
Other tasks check
final URL, page
content, backend
state, text answer

WebArena-Verified
Structured output;
Remove LLM-as-a-
judge, deterministic
checks on HTTP traces

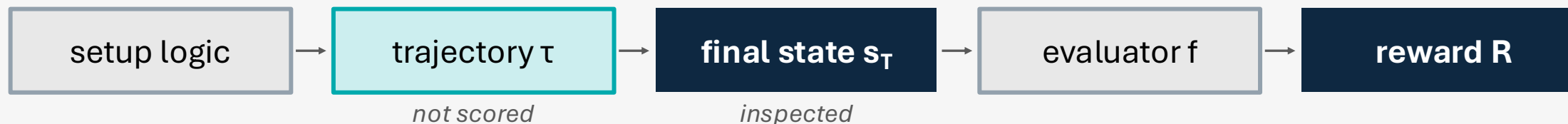
OSWorld: can the agent operate an actual computer?

Example task: *add a computed column, sort, save as result.xlsx*

What the agent does



How a task is scored



369 tasks

browser, office, file,
system apps

Cross-application

workflows span apps

Pixel observations

screenshots

Spreadsheets

cells, formulas, file
location

Other checks

settings, files, app
config

OSWorld 2.0 (2026): 108 long workflows (often 1 hr+), hundreds of steps, ~27 checkpoints for partial credit

SWE-bench: can the agent fix a real repository issue?

Example task: *Calling a library function with an empty index incorrectly raises an exception. Make it return an empty result while preserving existing behavior.*

What the agent does



How the patch is scored



Verified
human-checked

Lite
300-task subset

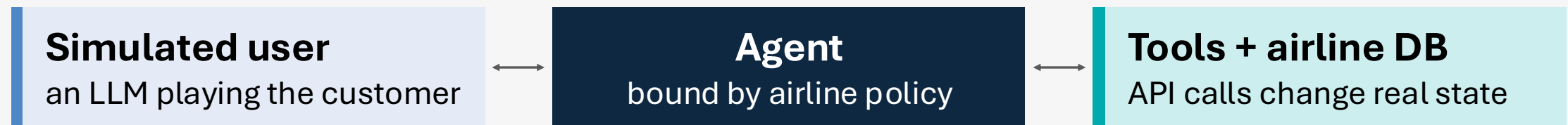
Multilingual
9 languages

Multimodal
image issues

Pro
harder, multi-file

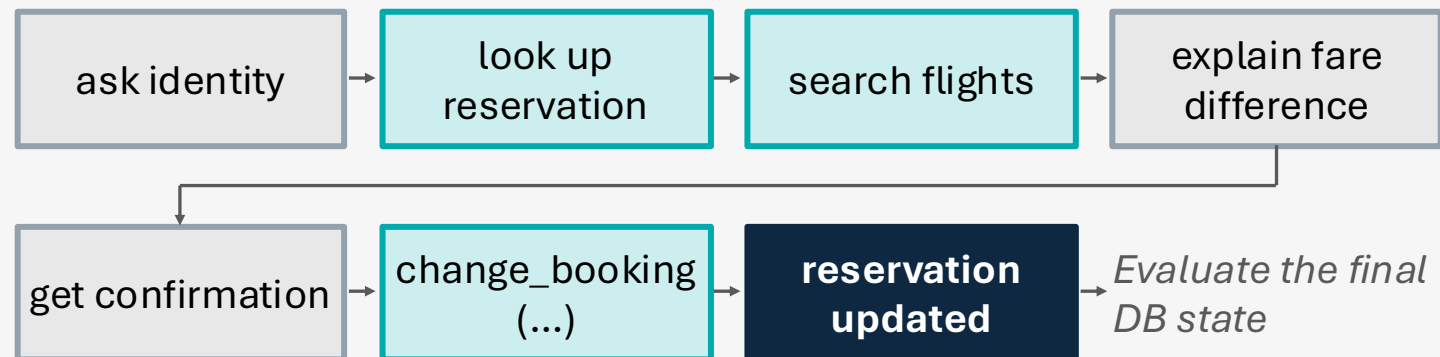
τ -bench: can an agent interact with user, obey policy, and change backend state?

Two interactive actors with a policy



Example task:

"I need to change my flight tomorrow to the evening flight."



grey: talk to the user

teal: call a tool

Airline policy

- basic economy can't be modified

τ -bench scores end state and consistency

Scoring

The original reward is

$$r = r_{\text{action}} \times r_{\text{output}}$$

$$r_{\text{action}} = 1$$

only if the final database matches the annotated goal state.

$$r_{\text{output}} = 1$$

only if the user-facing response includes required information.

Read-only calls may vary, but **unintended writes cause failure.**

τ^3 -**bench** implementation now includes corrected tasks, more domains, knowledge retrieval, and voice evaluation

The benchmark also introduced pass^k

pass^k

the probability that all k independent runs of the same scenario succeed

 4 of 4 runs succeed

$\text{pass}@k$

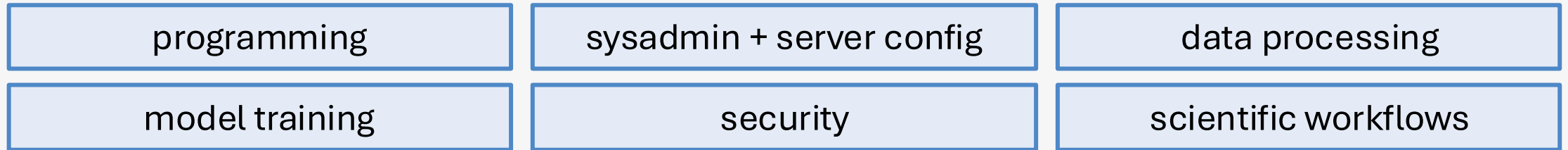
 1 is enough

only one successful run is needed

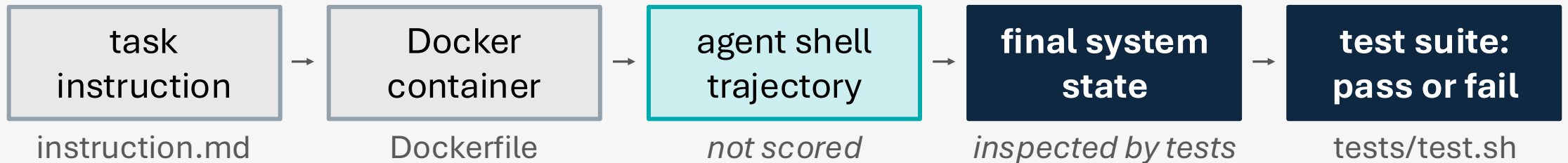
pass^k is deliberately stricter than $\text{pass}@k$

Terminal-Bench: can an agent finish a real task through a shell?

Task domains: end-to-end work, not only repo bug fixes



How a task is packaged and scored



89 tasks in TB 2.0 → 66 in TB 4.0

hidden human solutions; v4.0 prunes saturated tasks, 8-hour timeout

Representative task

new protocol on a live KV service: no dropped connections, latency held

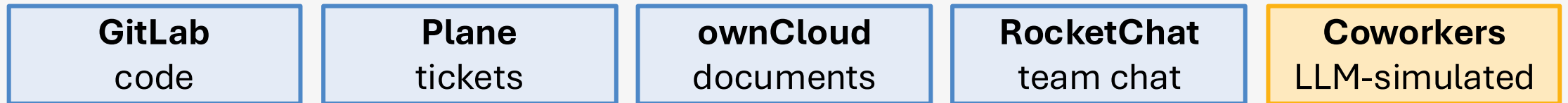
Terminal-Bench: Between SWE-bench and OSWorld

	SWE-bench	Terminal-Bench	OSWorld
Agent acts by	editing files in one repo	shell commands in a container	mouse clicks and keystrokes
Agent observes	code and test output	terminal output	screenshots
Task scope	fix one GitHub issue	configure, debug, repair, build	operate any desktop app
Scored by	the repo's tests	tests on final system state	checks on final state

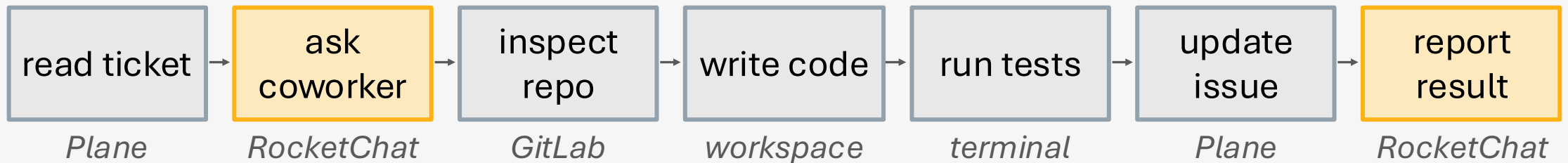
No GUI grounding: tests long-horizon reasoning and system interaction without mouse-coordinate prediction

TheAgentCompany: mimicking a company

Company environment



One job, end to end



175 tasks

software, PM,
data science, HR,
finance

Partial credit

checkpoints score
each step of the
job

$$S_{\text{partial}} = 0.5 \cdot \text{earned/total} + 0.5 \cdot S_{\text{full}}$$

Score by progress

4/8, incomplete		0.25
7/8, incomplete		0.44
8/8, complete		1.00

Agent benchmarks

Benchmark	Agent operates in	Typical actions	What is graded
GAIA	Web/files/code tools	Search, inspect files, calculate	Short final answer
WebArena	Self-hosted websites	Click, type, navigate	Website or backend state
OSWorld	Full desktop VM	Mouse, keyboard, app interaction	Files, app state, system state
SWE-bench	Software repository	Inspect, edit, run tests	Patch passes tests
BFCL	Function/API interface	Structured function calls	Calls, arguments, and resulting state
τ-bench	Customer-service APIs and dialogue	Talk to user, call APIs	Database state and response
Terminal-Bench	Isolated terminal container	Arbitrary shell commands	Task-specific verifier tests
TheAgentCompany	Simulated software company	Use web apps, code, files, talk to co-workers	workflow checkpoints and final system state

Agent Evaluation is Challenging and Evolving

- Challenges
- Evolution

Data Leakage: four layers, not just training

Training leakage

task or its human fix was in the training data

Eval-time leakage

agent web-searches and copies the future fix

Environment leakage

sandbox exposes future commits or hidden tests

Benchmark overfitting

prompts and scaffold tuned on the public test set

realistic tools



required for agent eval



larger leakage surface



benchmark may be searchable

Leakage can occur through the whole agent stack

Modern “agent” performance is roughly a function of

$$P = f(\text{LLM, prompt, scaffold, tools, retriever, memory, environment})$$

Component	Possible leakage
Base model	Benchmark examples in pretraining
Fine-tuning	Benchmark solutions used as agent-training trajectories
Prompt	Few-shot examples derived from test tasks
Retrieval	Benchmark answers indexed in RAG corpus
Memory	Previous benchmark attempts stored
Web search	Published task solutions retrieved
Environment	Future Git commits / hidden artifacts exposed
Scaffold	Hand-coded heuristics tuned to benchmark
Agent development	Researchers repeatedly inspect test failures and tune the system

Any one of these can contain benchmark information. Therefore saying “**the model was not trained on the test set**” is nowhere near sufficient for agent evaluation.

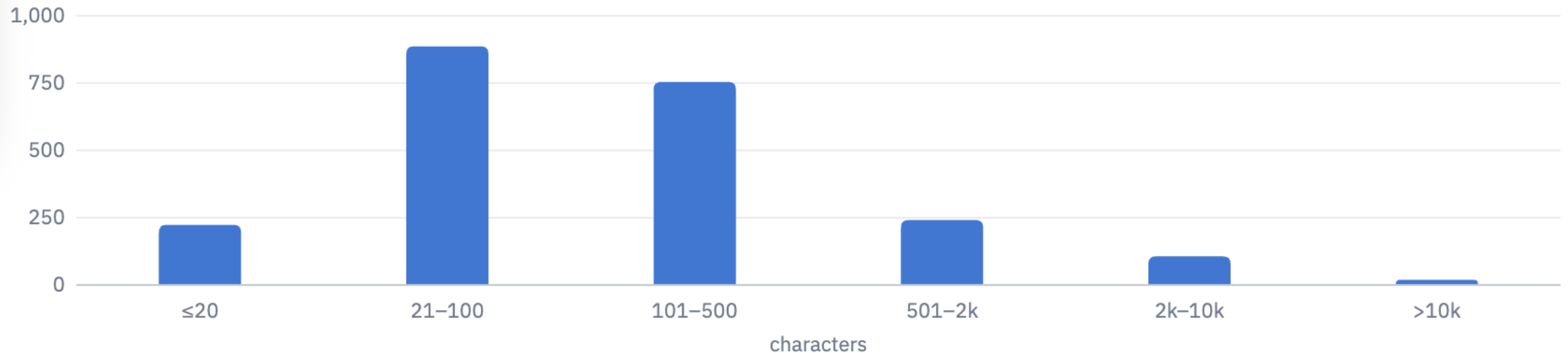
Challenges with Agent Evaluation: a single user turn

- Most benchmarks hand the agent one fully-specified instruction
 - prompt in, final state out — no user in the loop
- Real work is ambiguous, revised and interrupted
 - missing constraints, mid-task corrections, shifting goals
- Single-turn scoring rewards confident guessing
 - asking a clarifying question costs steps and earns no credit

Challenges with Agent Evaluation: a single user turn

Prompt length

Each column counts the typed prompts whose length falls in that band, in characters. For scale, this sentence is about 90 characters.



This is how you used Claude Code in Assignment 1

Challenges with Agent Evaluation: a single user turn

Typed prompts per session

Build sessions by how many prompts were typed in them.



Active time per session

Build sessions by active time: the time between events, leaving out gaps over 15 minutes, so a session left open overnight counts only its working stretches.



Challenges with Agent Evaluation: a single user turn

- Most benchmarks hand the agent one fully-specified instruction
 - prompt in, final state out — no user in the loop
- Real work is ambiguous, revised and interrupted
 - missing constraints, mid-task corrections, shifting goals
- Single-turn scoring rewards confident guessing
 - asking a clarifying question costs steps and earns no credit

Clarification

does it ask when the goal is underspecified?

Recovery

does it re-plan after a mid-task correction?

Memory

does it carry constraints across turns?

Challenges with Agent Evaluation: there are more

Cost and latency

Success rate is not sufficient: a 200-step, \$40 run vs. a 12-step, \$0.30 run?

Non-determinism

same agent, same task, different outcome: single-run scores are noise, not signal

Safety and side effects

nothing penalises deleted files, sent emails or money spent en route to a pass

Metrics

outcome-only scoring passes lucky or unsafe runs; trajectory matching fails valid ones

Get involved: MaDSys group is building a research platform for agent evaluation that uses real traffic — contact me if you are interested and have 20+ hours each week to help out.

How fast the frontier caught up

Before 2018

2018–2019

2020–2022

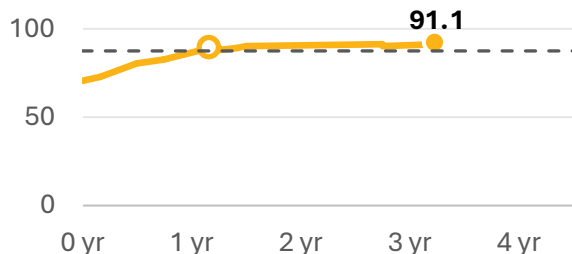
2023–2024

2024–2025

2025–2026

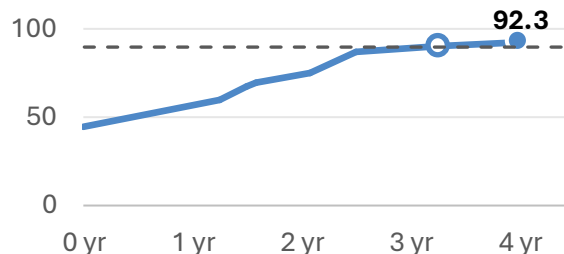
GLUE · 2018

Human level (87.1) in 14 months



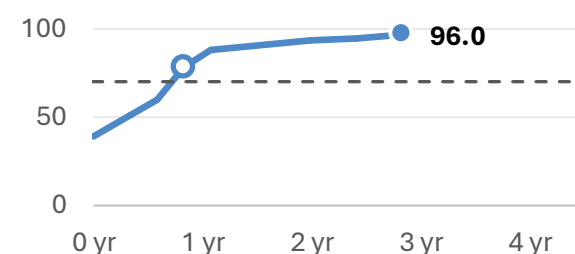
MMLU · 2020

Expert level (89.8) in 39 months



GPQA Diamond · 2023

PhD experts (69.7) in 10 months



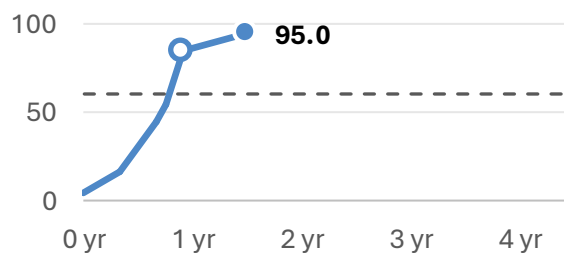
SWE-bench Verified · 2024

33% → 95% in 22 months



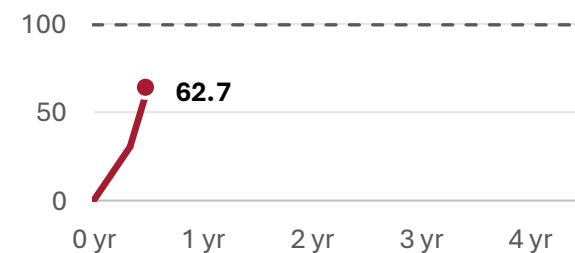
ARC-AGI-2 · 2025

Average human (60) in 11 months

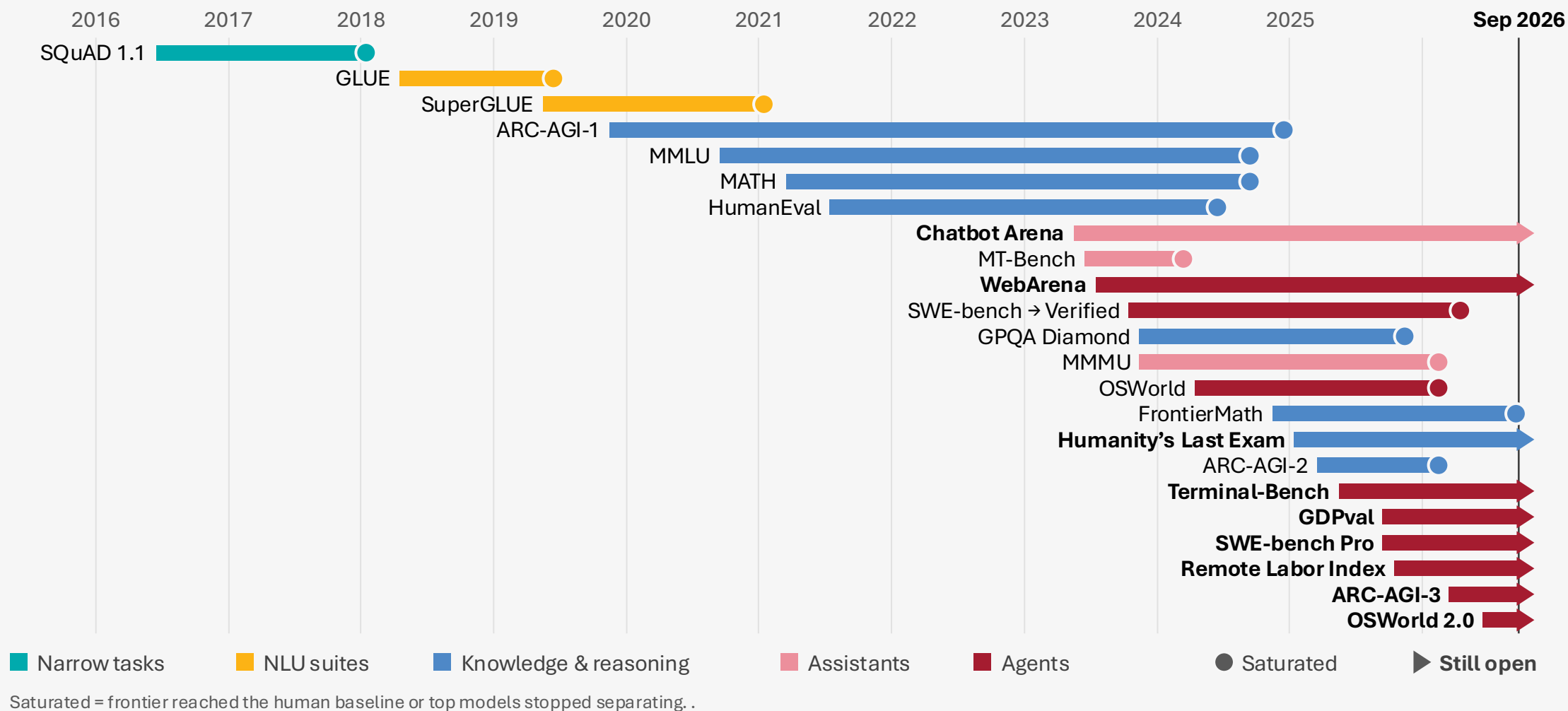


ARC-AGI-3 · 2026

0.5% → 62.7% in 6 months*

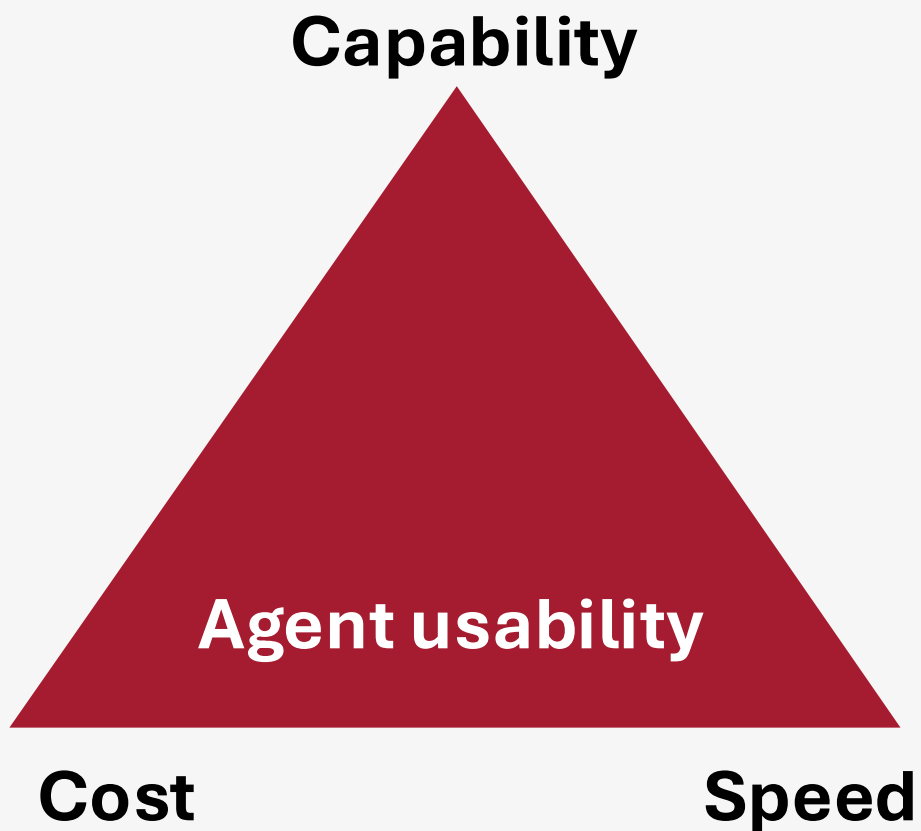


Benchmarks saturate quickly



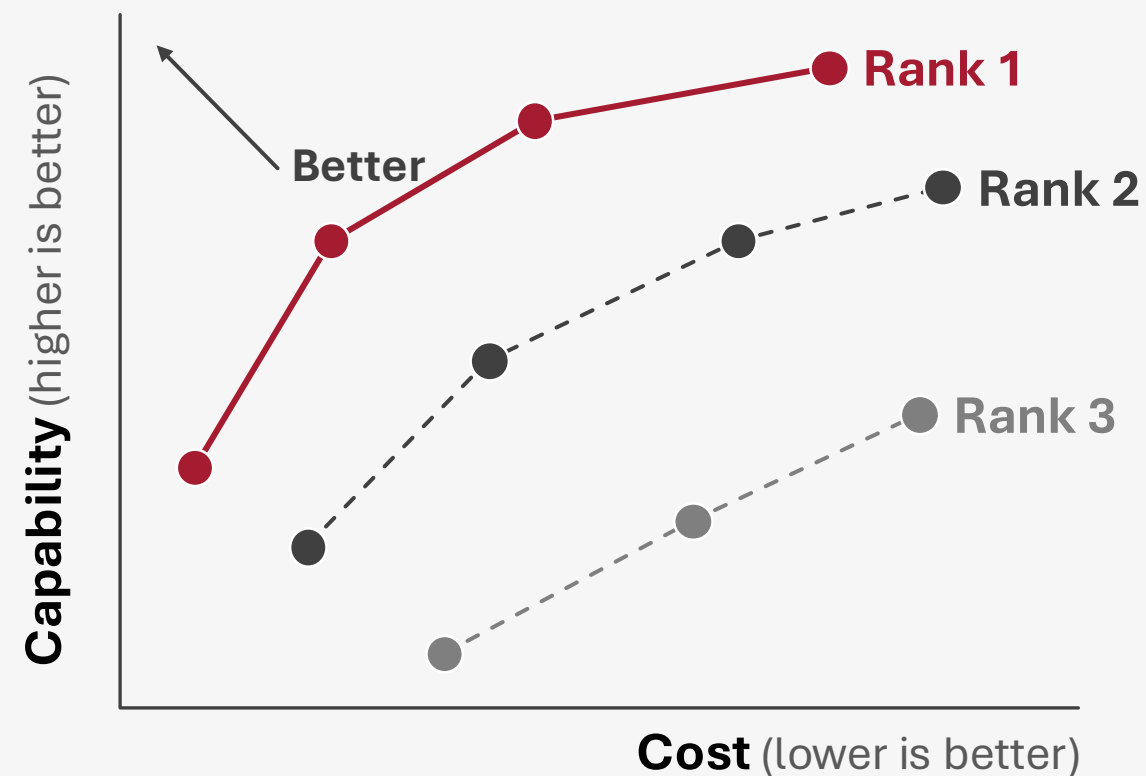
Agent evaluation is still evolving

Dimension	Earlier evaluation	Current frontier
Objective evaluated	Base model	
Interaction	Static, single-turn	
Grader	Exact match or likelihood	
Dataset	Fixed and public	
Main metric	Average accuracy	
Primary concern	Generalization across examples	



More dimensions to consider for a real-world agent

Assignment 3: You will be evaluated on all three metrics



A dominates B if A is no worse on any metric and better on at least one.

Rank 1 is the Pareto frontier; speed is not drawn

Next class

- Cost-effective agent
- Case study
- Different types of agents
- Agentic frameworks

