GIES MSBA WORKSHOP SERIES · SESSION 6

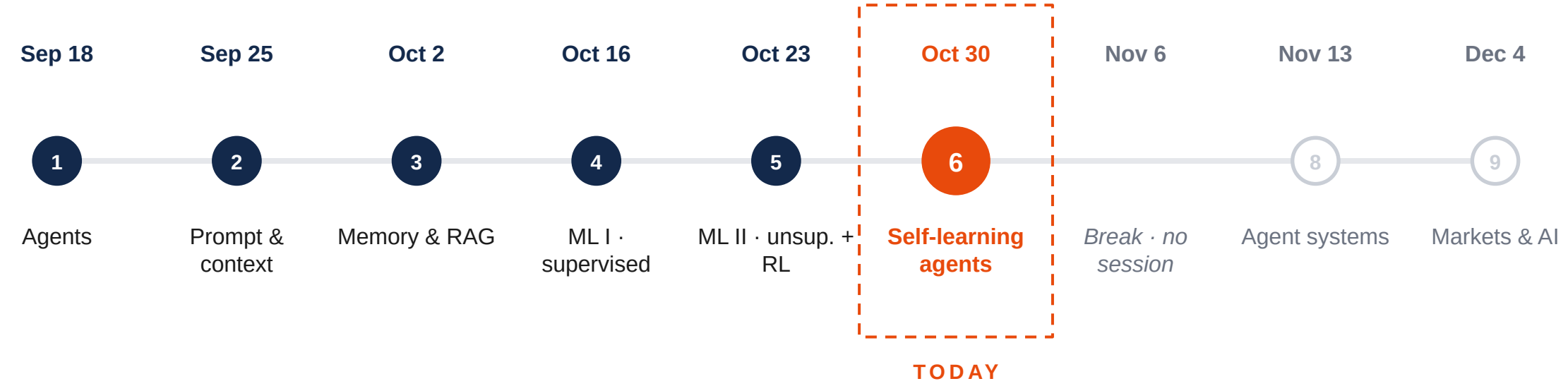
Building Self-Learning AI Agents

Turn the research agent into one that gets better from experience: a harness that grades its work, a place to keep lessons, and a way to prove it improved.

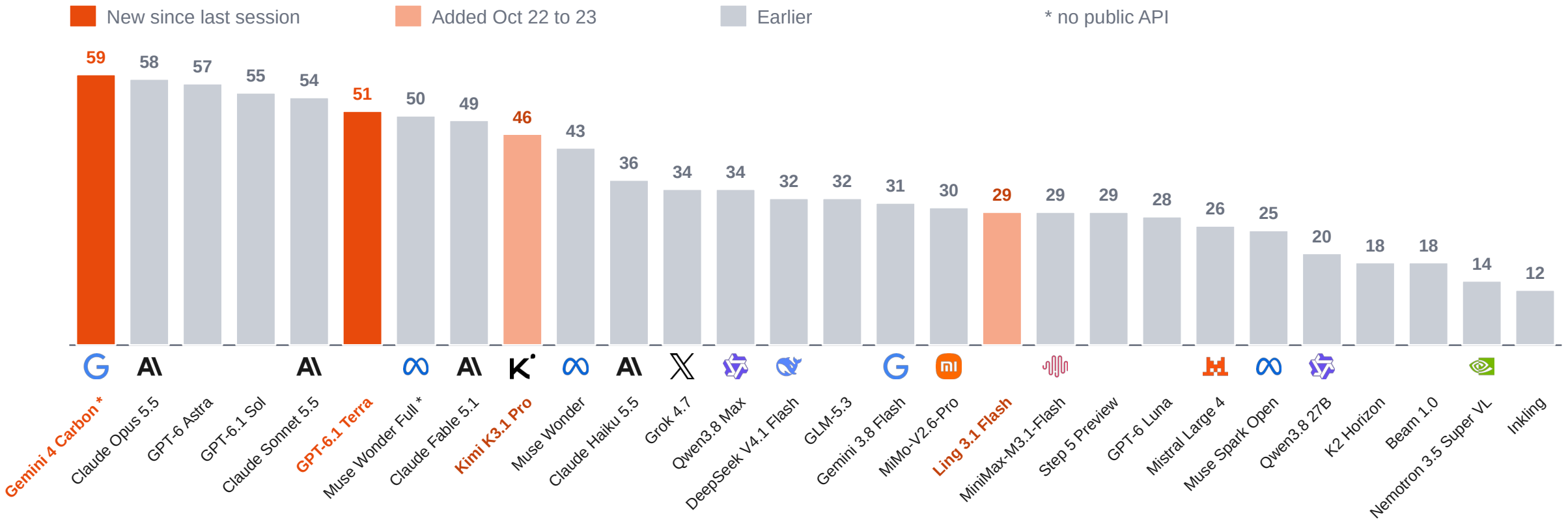
Fri, Oct 30, 2026 · 1:30 to 3:30 pm · 3007 BIF · Chapter 6 of the interactive textbook

WHERE WE ARE IN THE SERIES

Part I built the agent. Part II taught learning. Today we combine them.



Gemini 4 Carbon takes #1; GPT-6.1 Terra is also new this week

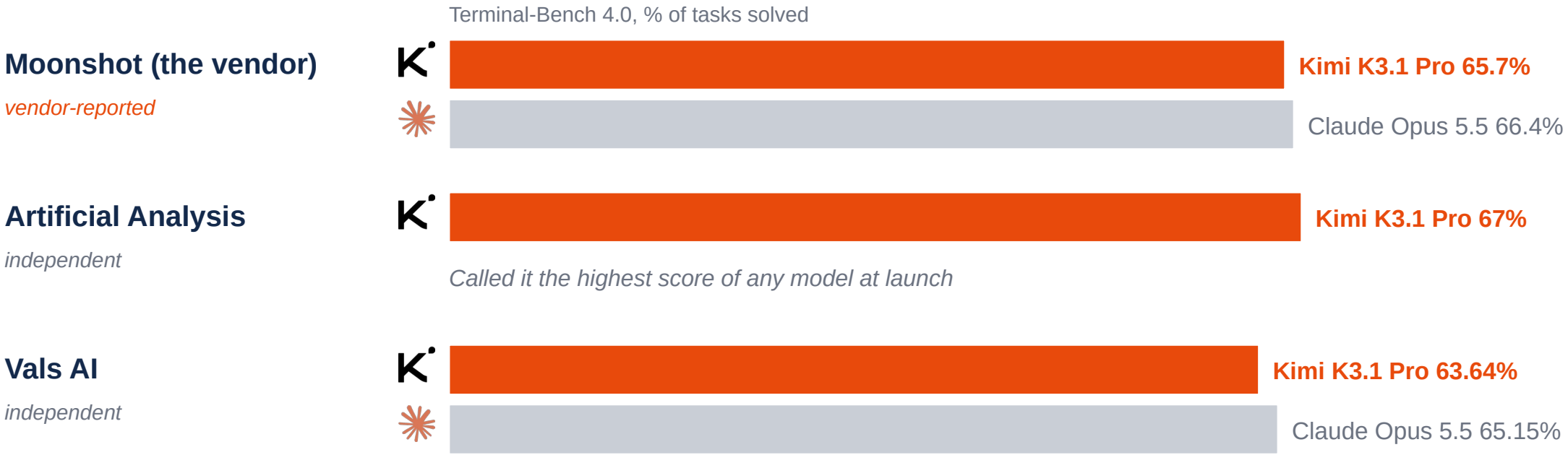


Artificial Analysis Intelligence Index v5.0 (launched Oct 27, 2026; 9 evaluations), scores re-based this week, not comparable to v4.3. Release dates: AA changelog.

NEW IN AI · THIS WEEK

Benchmarking

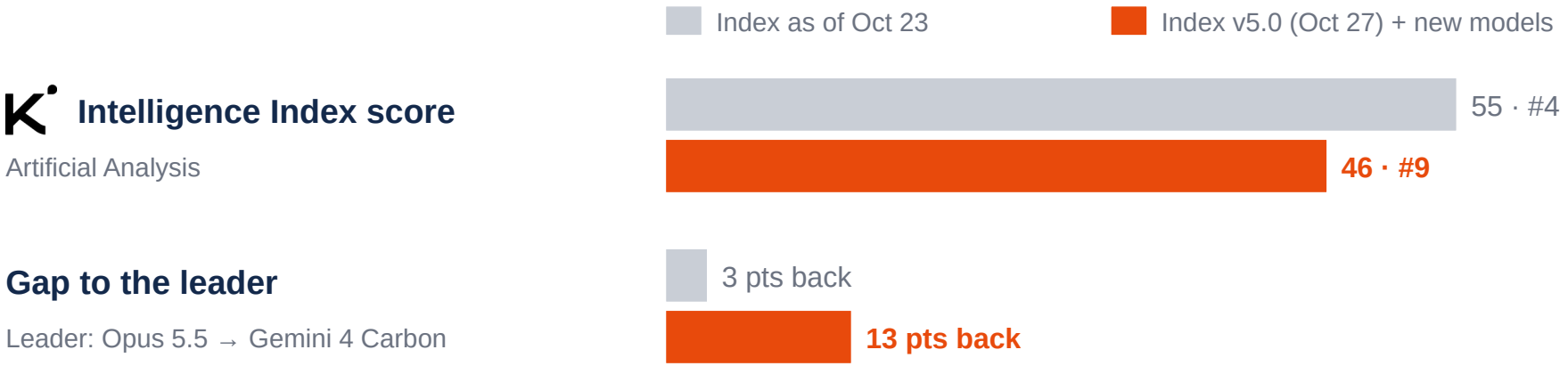
On terminal tasks, independent tests back Kimi's claim: it's within ~2 points of the top



Three testers, three numbers: setup and harness change the score by a few points.

Sources: Moonshot AI, kimi.com/en/blog/kimi-k3-1-pro; Artificial Analysis, Kimi K3.1 Pro launch benchmarks (Oct 23, 2026); Vals AI model page (updated Oct 24, 2026).

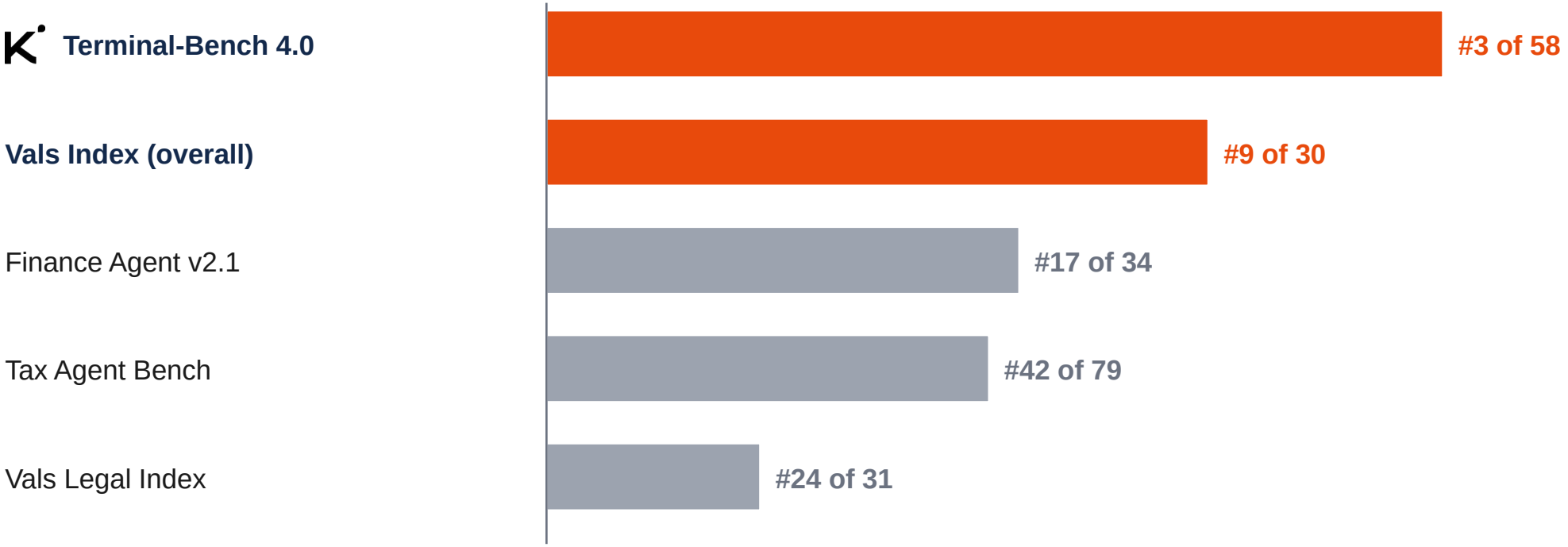
Same model, same week: Kimi went from #4 to #9 when the index changed



Nothing about the model changed. The tests, their weights and the competition did.

Artificial Analysis: Kimi K3.1 Pro launch article (Oct 23, 2026) and Intelligence Index v5.0 chart (Oct 29, 2026); v5.0 launch per AA changelog (Oct 27).

Kimi is near the top on terminal work but mid-pack or worse on finance and law



For a finance firm, the finance rows matter more than the headline rank.

Vals AI, Kimi K3.1 Pro model page (updated Oct 24, 2026). Bar = share of tested models it outranks, computed from Vals' rank and model count.

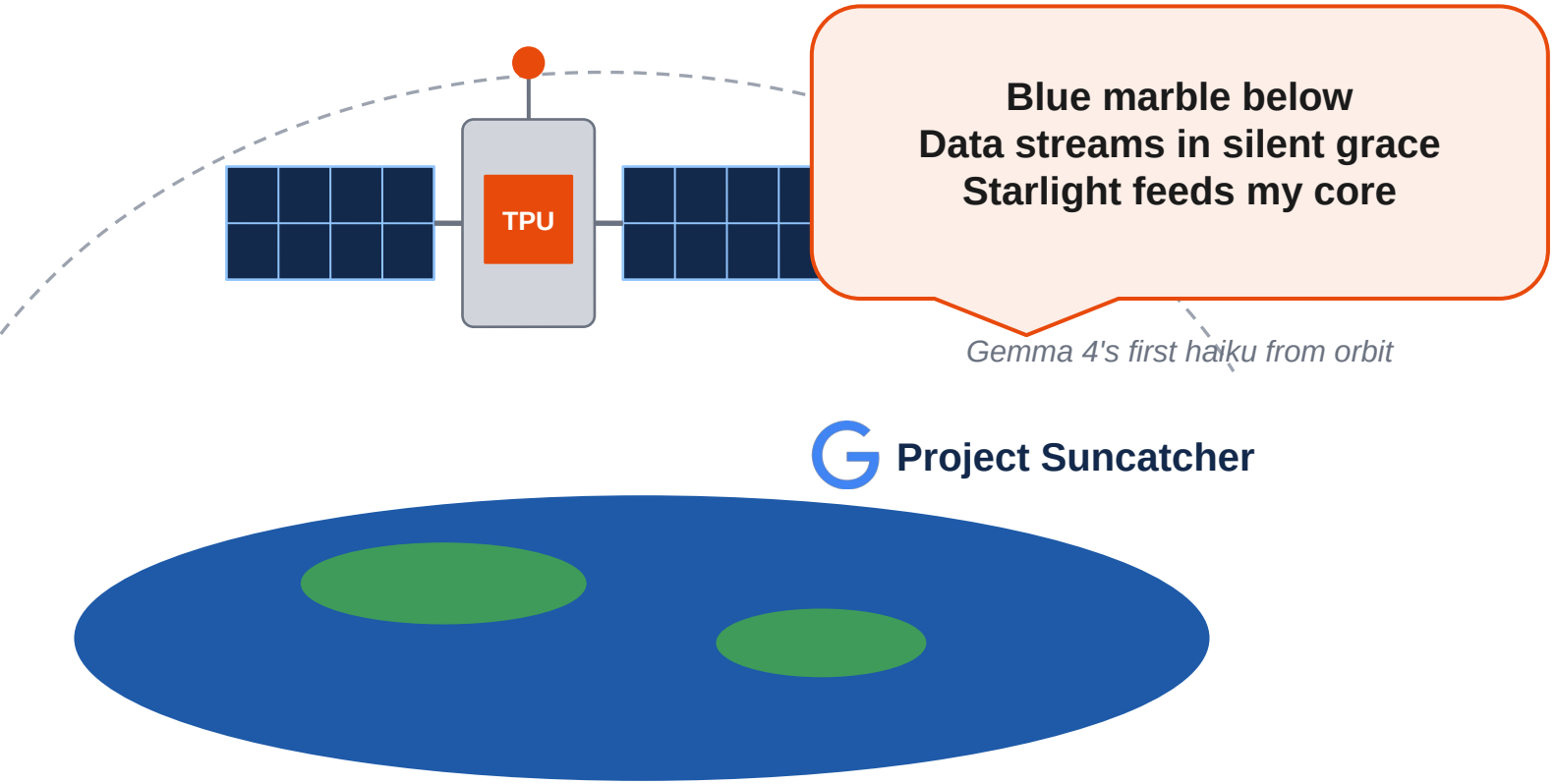
Five questions to ask before trusting any benchmark chart

1	**Who ran it?**	Vendor or independent tester
2	**Which version?**	Kimi: #4 → #9 after a version change
3	**What setup?**	Harness and effort move scores a few points
4	**Saturated or leaked?**	Top scores near 100% stop telling models apart
5	**Is it your task?**	Terminal #3, finance #17

Examples from Artificial Analysis and Vals AI (Oct 2026). Saturation: Vals archived SWE-bench and Terminal-Bench 2.0; AA v5.0 weights private tests at 60%.

JUST IN · OCT 30, 2026

Google's first data center satellite is online and writing poetry



Oct 1	Launched (built by Planet, SpaceX rideshare)
1 TPU	Data-center chip, up to 1 kW of solar power
~20 min	Run time per session, power-limited
2027	Next: two linked satellites

Why it matters: power is AI's bottleneck. Open questions: radiation, cost, launches.

Sources: Google blog, Project Suncatcher satellite response (Oct 30, 2026); TechCrunch (Oct 30, 2026).

NEW IN AI · OCT 16 to 26

This week's news is about harnesses and agents that improve themselves

What	Who · when	Why it matters today
Agent Skills donated to an open foundation	Anthropic · Oct 22	1M skills published; a shared harness layer
Claude Code: fork sub-agents on by default	Anthropic · Oct 16	Harness change, not a model change
Decisions API: cheap yes/no scoring	OpenAI · Oct 22	A grader you can put inside an agent loop
Recursive Flint: model built by an automated research loop	Recursive · ~Oct 21	Self-improvement claims (company-reported)
Kimi K3.1 Pro, open weights by Nov 4	Moonshot AI · Oct 23 to 24	Open model near frontier on agent tasks

Sources: anthropic.com/news; code.claude.com changelog; openai.com/index/decisions-api; recursive.com; kimi.com/en/blog.

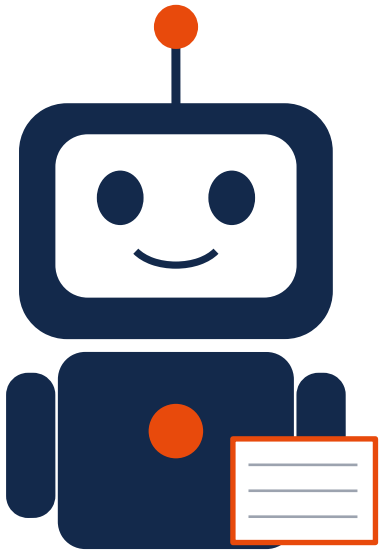
NEW IN AI · THIS WEEK

Distillation

DEFINITION

Distillation

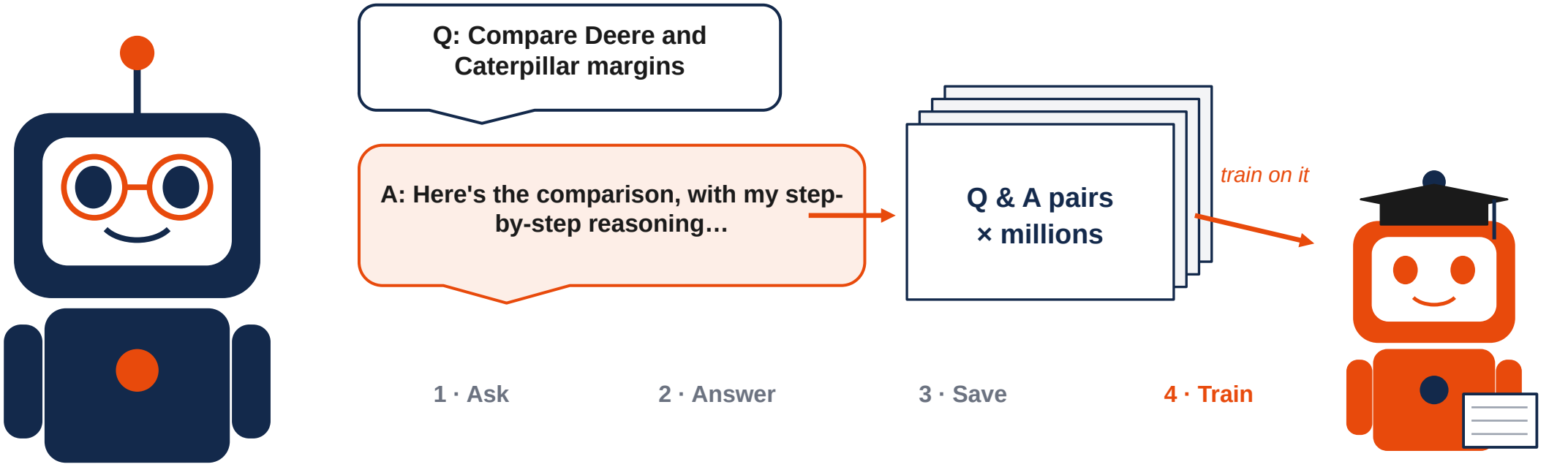
Training a cheaper “student” model on the answers of a stronger “teacher” model, so the student learns to imitate it.



WHY IT MATTERS

It's a normal technique labs use on their own models. Doing it to a rival's model through its API breaks the rival's terms of service.

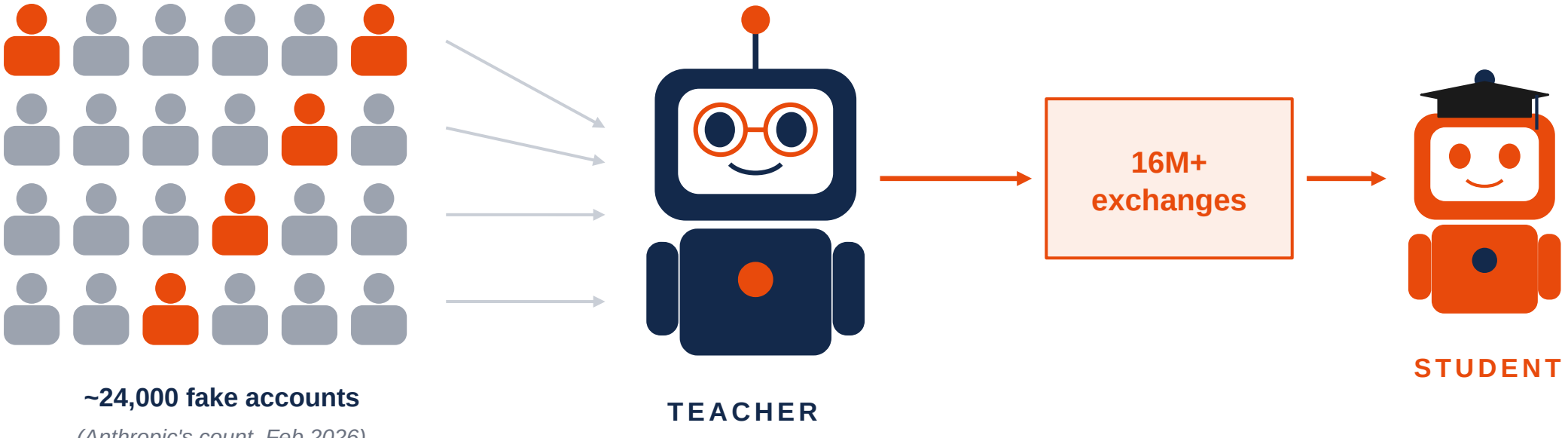
The student copies the teacher's answers, millions of times over



The student never sees the teacher's weights or training data, only its answers. Illustration.

WHAT US LABS ALLEGE

Thousands of fake accounts each ask a few questions, so no one account looks suspicious



~24,000 fake accounts

(Anthropic's count, Feb 2026)

TEACHER

DEFENSES spot coordinated accounts · ban them · make answers harder to copy (harvested data cut 90 to 99%, per Anthropic)

Anthropic, Detecting and preventing distillation attacks (Feb 23, 2026) and Strengthening our defenses (Oct 12, 2026). Allegations; accused labs deny or did not respond.

EVERYDAY EXAMPLE · DISTILLATION

Studying from the top student's graded homework instead of the textbook

You copy the answers and worked steps of the best student in class, thousands of times. You learn fast, without doing their reading.

THE VARIABLES

Top student
Teacher model (Claude, GPT)

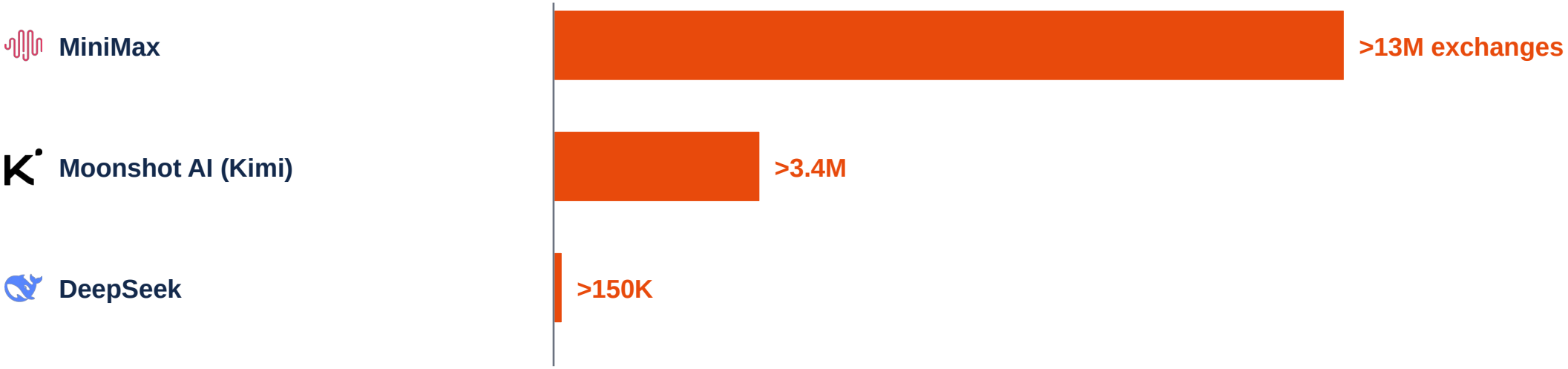
Their answers
API outputs collected at scale

You
Student model

Class rules
Terms of service

WHAT ANTHROPIC ALLEGED · FEB 2026

Anthropic says three Chinese labs ran 16M+ exchanges with Claude through ~24,000 fake accounts



Allegation from Anthropic's own detection. The labs did not respond; no outside audit.

Anthropic, “Detecting and preventing distillation attacks” (Feb 23, 2026). Exchanges with Claude, millions. TechCrunch reported no response from the labs.

TIMELINE · ALLEGATIONS AND DENIALS

US labs and officials allege copying; Chinese labs and Beijing deny it

Jan 2025	White House AI czar: “substantial evidence” DeepSeek distilled OpenAI	ALLEGATION
Sep 2025	Anthropic stops selling to majority-China-owned firms	POLICY
Sep 2025	DeepSeek (Nature): R1 didn't copy OpenAI reasoning examples	DENIAL
Feb 2026	Anthropic names DeepSeek, Moonshot, MiniMax	ALLEGATION
Sep 2026	NSA/CISA/FBI advisory names six Chinese firms	ALLEGATION
Sep 2026	China's foreign ministry: “groundless”	DENIAL
Oct 2026	Report: labs shift to OpenAI's GPT-6 Astra (anonymous sources)	ALLEGATION

Sources: Fortune (Jan 29, 2025); anthropic.com/news (Sep 4, 2025; Feb 23, 2026); Scientific American/Nature (Sep 2025); CISA AA26-251A via CSA & Jon Peddie (Sep 2026); The Decoder (Oct 28, 2026).

THIS WEEK'S REPORT

After Anthropic tightened its defenses, labs reportedly shifted to OpenAI's models

Target: Claude

Main target until 2026, per the report

→

Anthropic's defenses

Oct 12: harvested data cut ~90 to 99%

→

Shift to GPT-6 Astra

Oct 28 report: used for Kimi K3.1 Pro, DeepSeek V4.1 Flash

HOW SURE IS THIS?

The shift rests on anonymous sources; the labs haven't confirmed. OpenAI was also targeted earlier, DeepSeek in Jan 2025, a Moonshot-linked campaign in Jul 2026.

Sources: Anthropic, “Strengthening our defenses against distillation attacks” (Oct 12, 2026); The Information via The Decoder (Oct 28, 2026); OpenAI disruption report via The Hacker News (Oct 1, 2026).

EVIDENCE CHECK

Has cutting off distillation slowed Chinese models? Not proven yet

Points to a wider gap

~10 pts — Epoch capability gap, GPT-6 Astra vs Kimi K3 (Oct 2026)

90 to 99% — Less data harvested, per Anthropic (Oct 2026)

Points the other way

Oct 23 — Kimi K3.1 Pro launched after the defenses; top open model

39 pts — US lead on Arena, Mar 2026, down from 300+ in 2023

The gap has reopened on some measures, but no study shows distillation limits are the cause.

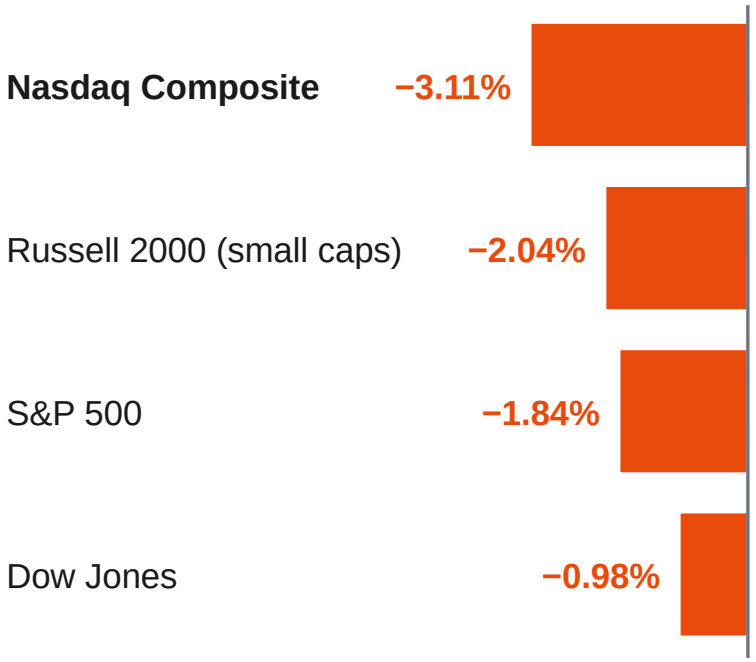
Epoch AI via OfficeChai (Oct 30, 2026); Anthropic (Oct 12, 2026); Artificial Analysis (Oct 23, 2026); Stanford HAI AI Index 2026. Different metrics, not on one scale.

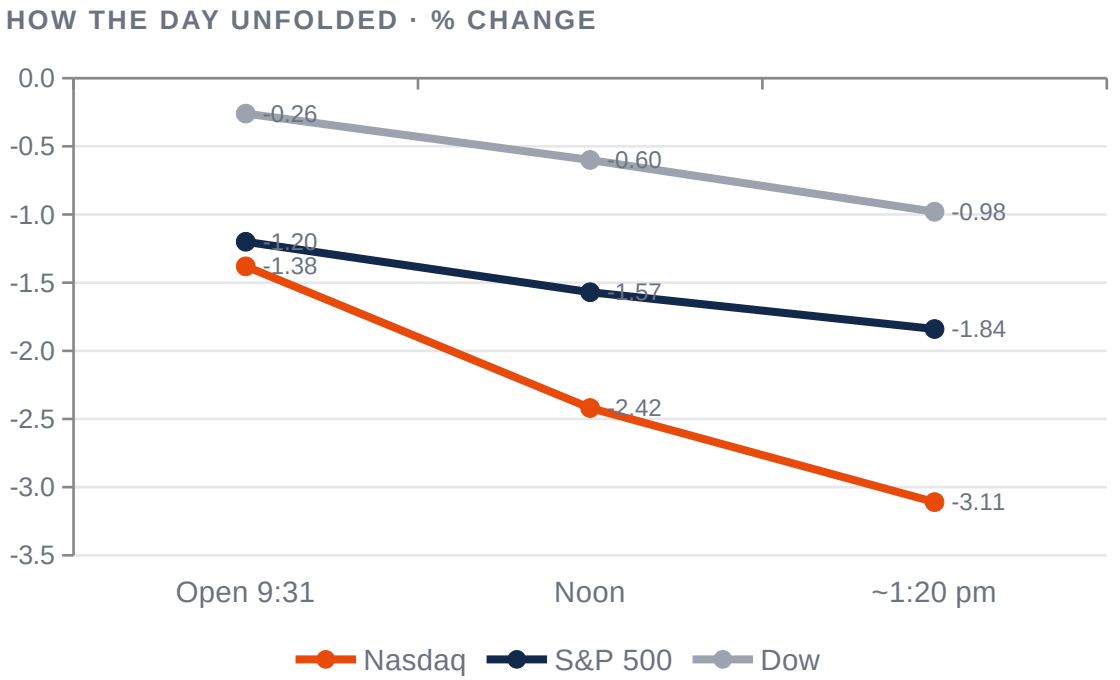
NEW IN AI · MARKETS TODAY

Markets today

MARKETS TODAY · FRI OCT 30, INTRADAY

Tech-heavy Nasdaq falls about 3%, on pace for Wall Street's worst day since June

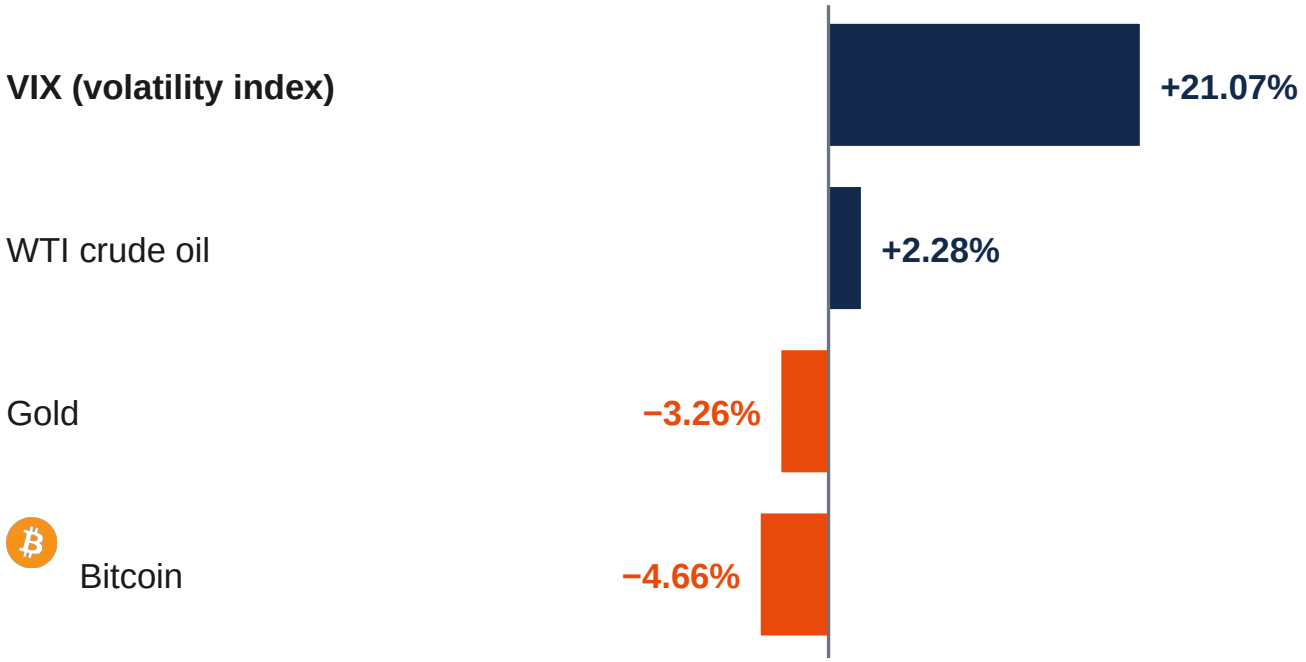




Tech led the drop, and selling deepened through the day.

Day bars & ~1:20 pm: Yahoo Finance live ticker. Open: TheStreet (9:31 am ET). Noon: 247wallst (12:00 pm ET). AP: “worst day since June.” Intraday, not closes.

Fear gauge jumps 21%; oil rises while gold and Bitcoin fall



LEVELS · ~1:20 PM ET

VIX	**20.82**	+3.62 pts
10-yr yield	**4.93%**	−6 bp; >5.30% earlier in Oct
WTI oil	**$104.33**	Brent +12% in October
Gold	**$3,900**	−$131
Bitcoin	**$77,031**	>$86,000 earlier in Oct

Investors moved to safety: the 10-year Treasury yield slipped to about 4.93%.

Yahoo Finance live blog ticker (Oct 30, 2026), intraday ~1:20 pm ET. % change on the day.

MARKETS TODAY · WHY

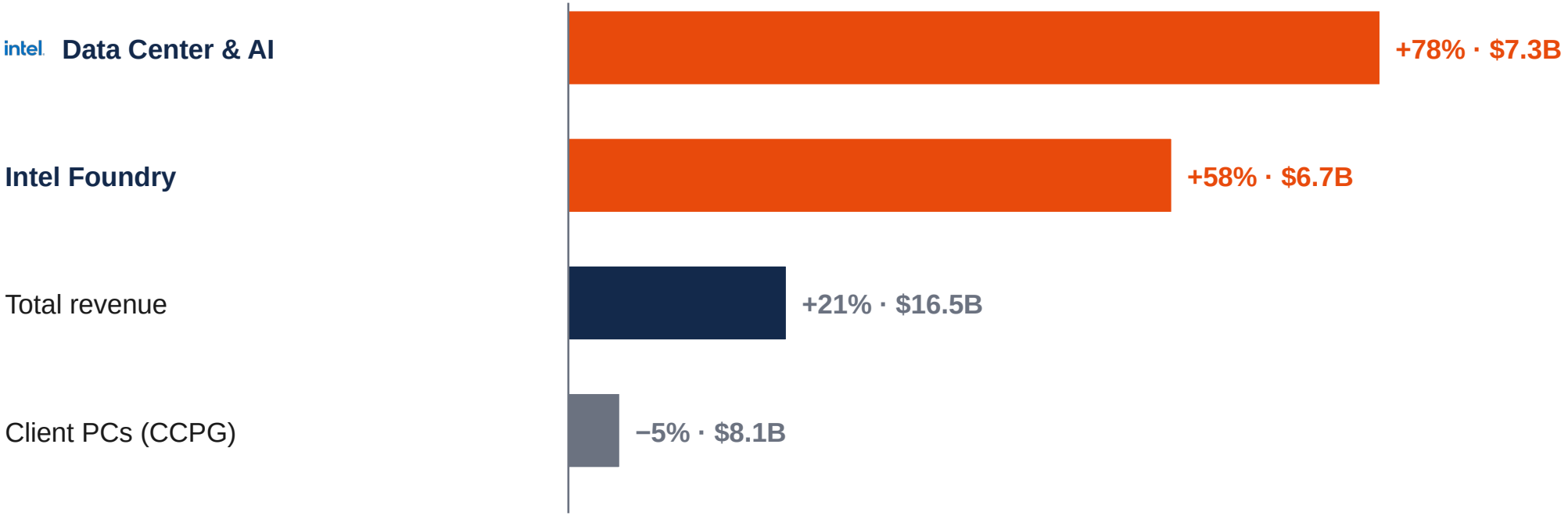
Three reasons stocks fell: AI spending doubts, a hot inflation reading, and oil

Driver	What happened	Why markets cared
AI spending	Amazon burned $42B+ of free cash in 12 months; Alphabet fell 12.6% Thursday	Investors doubt the AI spend will pay off soon
Intel	Weak forecast: PC slump, factory accident in Israel. Stock −30%	Near its worst day since 1974; drags chips
Rates	September PCE inflation ran hot; two Fed officials say rates need to rise	A December hike would slow data center buildouts
Oil	Brent up 12% in October on Iran and tanker attacks	Higher costs; energy stocks were the only big gainers

Sources: AP via CP24, Yahoo Finance and TheStreet live coverage (Oct 30, 2026); StockTitan on Alphabet Q3.

EARNINGS · INTEL Q3 2026 (REPORTED OCT 29)

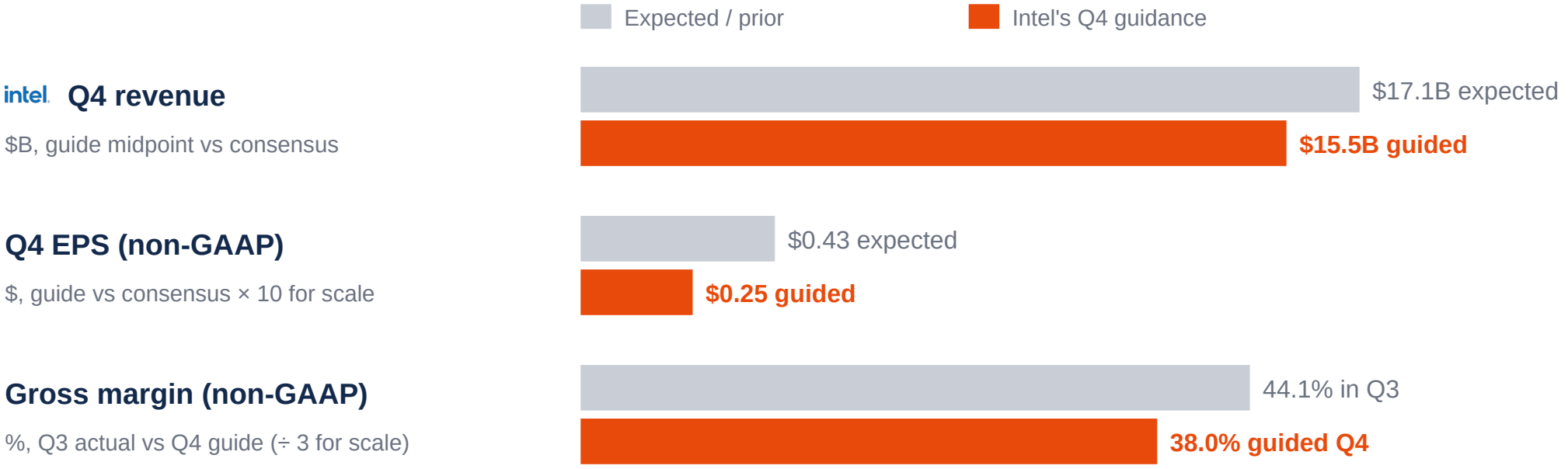
Intel's data center sales jumped 78%, but PCs shrank and the outlook disappointed



Q3 beat (EPS $0.52 vs $0.40 expected). The stock fell on the Q4 guide.

Intel Q3 2026 earnings release (intc.com, Oct 29, 2026). Year-over-year revenue growth by segment; Client PCs fell 5% (bar shows size of decline).

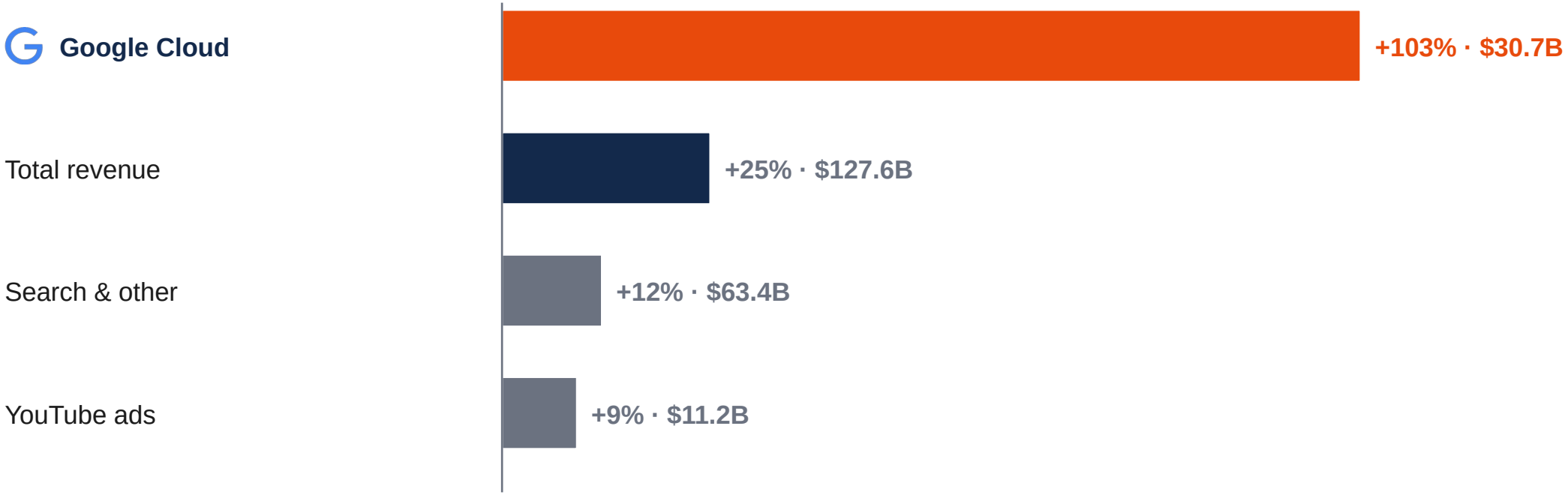
Intel guided far below Wall Street: a PC slump plus an accident at its Israel fab



Fab 28 accident: ~$300M lost revenue in Q4 and again in Q1, ~2 pts of margin.

Intel Q3 2026 release (guidance); consensus via TradingKey; call commentary via Investing.com transcript (Oct 29, 2026). Bars scaled for display; labels show actual values.

Google Cloud doubled, but Search slowed to 12% growth and YouTube to 9%

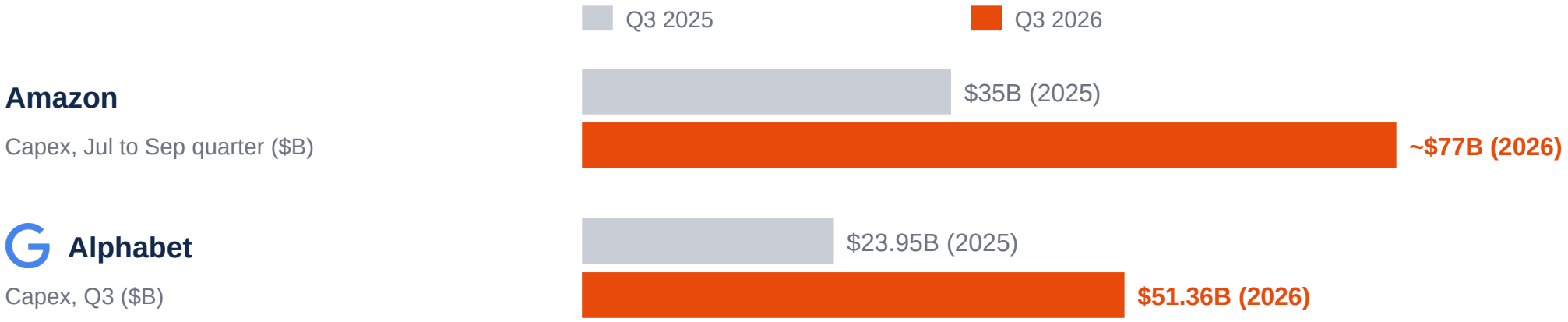


Search growth fell from 19% (Q1) to 17% (Q2) to 12%, fear that AI is eating search ads.

Alphabet Q3 2026 results via StockTitan and 9to5Google (Oct 28 to 29, 2026). Year-over-year revenue growth.

THE AI ANGLE

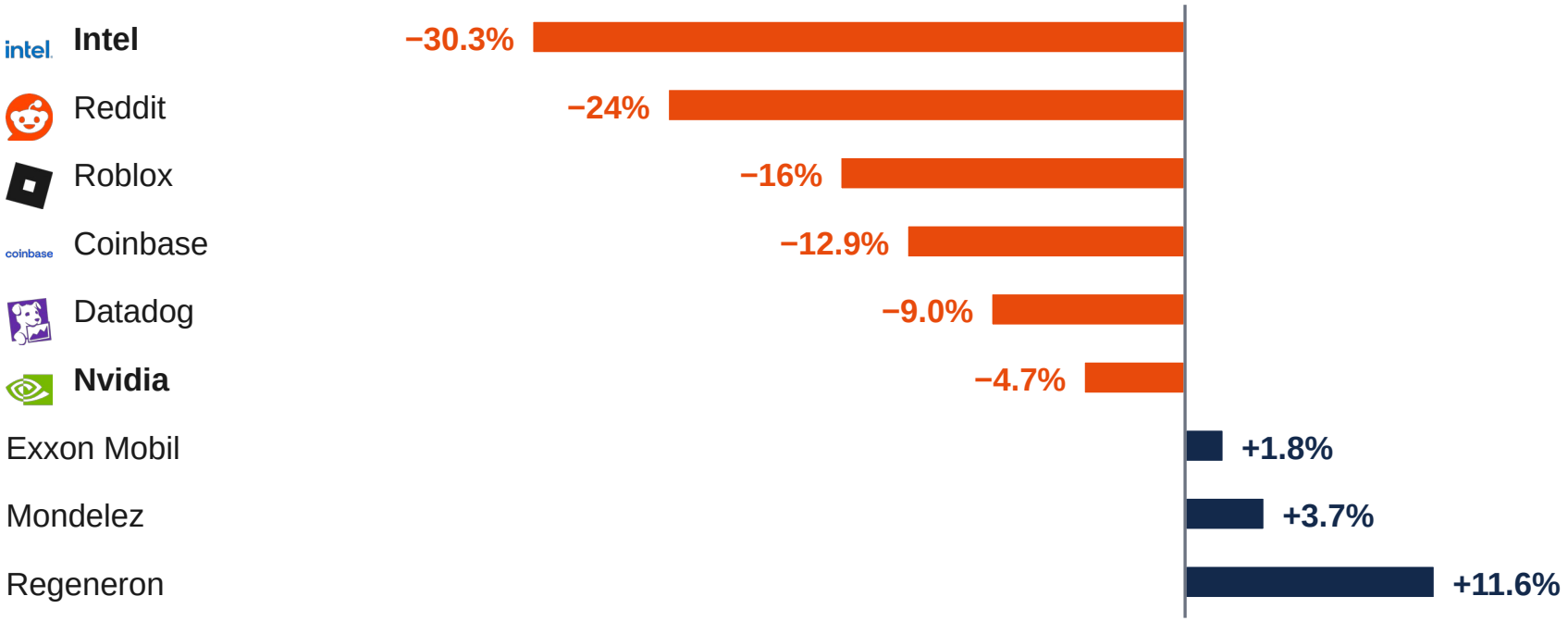
Big Tech roughly doubled its quarterly AI spending in a year, now investors want proof



Alphabet fell 12.6% Thursday, its worst day since its 2004 listing, despite Cloud revenue up 103%.

Amazon: Yahoo Finance & AP (Oct 30, 2026). Alphabet: Q3 2026 results via StockTitan (Oct 28 to 29, 2026). Capital expenditures, US$ billions.

Intel lost 30%; Nvidia was the heaviest drag on the S&P 500



Sectors at 10:30 am ET: Technology −4.3%; Energy +1.6%, Staples +1.4% led.

Yahoo Finance (Intel 1:32 pm ET, Reddit, Roblox), AP via CP24 (Nvidia, Coinbase, Exxon, Mondelez, ~12:55 pm ET), 247wallst (Datadog 12:20 pm ET; sectors 10:30 am), TheStreet (Regeneron 10:21 am).

WHAT IT MEANS FOR US

The market is asking AI the same question we ask our agent: prove it

“Show me” phase	Investors want returns, not just bigger AI budgets
Measure it	Accuracy on new tasks, cost per task, hours saved
Learn cheaply	Lessons and better harnesses improve results without more GPUs

Quote: Joe Tigay, Equity Armor Investments, via AP (Oct 30, 2026): “from the hopes and dreams phase of the AI trade … into the ‘show me’ phase.”

TODAY'S AGENDA

Six stops, one research agent that learns from Priya's corrections

01	What self-learning means	Same mistake twice, and the four layers that can change
02	The learning harness	Grade, reflect, save, and three harness trends
03	Memory and experience	Lessons, skills, playbooks, and dreaming
04	Recursive language models	For reference · reading inputs too long for one prompt
05	Rewards and continual learning	For reference · learning from scores; self-editing code
06	Evaluation	For reference · proving it improved, catching cheating

UP NEXT · 1 OF 6

First: what does it mean for an agent to learn?

01	**What self-learning means**	Same mistake twice, and the four layers that can change
02	The learning harness	Grade, reflect, save, and three harness trends
03	Memory and experience	Lessons, skills, playbooks, and dreaming
04	Recursive language models	For reference · reading inputs too long for one prompt
05	Rewards and continual learning	For reference · learning from scores; self-editing code
06	Evaluation	For reference · proving it improved, catching cheating

01

What self-learning means

An agent that changes something about itself because of feedback, and keeps the change.

SECTION 1 · SELF-LEARNING

Background

THE PROBLEM

Our research agent makes the same mistake on Tuesday that Priya fixed on Monday

MONDAY

Agent writes Deere memo

↓

Uses calendar-year revenue

↓

Priya: “Firm uses fiscal year. Cite the 10-K.”

TUESDAY

Agent writes Caterpillar memo

↓

Uses calendar-year revenue again

↓

Same correction, second time

The feedback existed. Nothing kept it.

BACKGROUND

Sessions 1 to 3 gave the agent memory of facts, not lessons

- Session 1: a loop that calls tools.
- Session 2: instructions it reads every call.
- Session 3: memory of clients and documents.
- Missing: anything that changes because the work went badly.

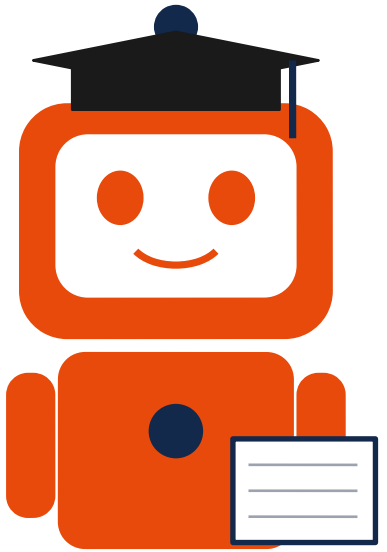
SECTION 1 · SELF-LEARNING

Self-improving agent

DEFINITION

Self-improving agent

An agent that uses feedback on its own work to change something it will use next time, its notes, its tools, its code, or its weights.



WHY IT MATTERS

Without it, every correction costs a human the same time again. With it, the tenth memo needs fewer fixes than the first.

EVERYDAY EXAMPLE · SELF-IMPROVING AGENT

A student who keeps an error log gets better each exam

After every practice exam, you write down each mistake and the fix. Before the next exam, you reread the log.

THE VARIABLES

Task
The practice exam

Feedback
The graded answers

What is saved
The error log

Result
Higher score on the next exam

SECTION 1 · SELF-LEARNING

Where it can change

WHAT CAN CHANGE

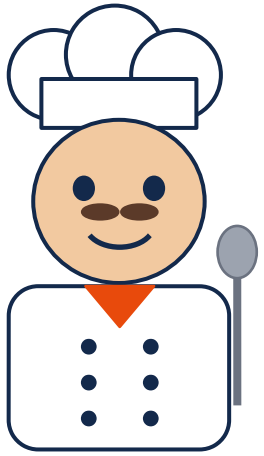
An agent can improve in four places, from cheap notes to expensive retraining

LAYER	WHAT CHANGES	EXAMPLES TODAY	COST TO CHANGE
Context	Prompt, memory, playbook	**Reflexion · ACE · GEPA**	Minutes · undo by deleting a line
Tools & skills	New tools, saved procedures	**Voyager · Agent Skills**	Hours · reusable
Architecture	Its own code and workflow	**Darwin Gödel Machine · SICA**	Days · needs tests to be safe
Model weights	Parameters retrained on rewards	**RL fine-tuning (GRPO)**	Weeks · costly, hard to undo

Taxonomy: Gao et al., A Survey of Self-Evolving Agents (arXiv 2507.21046, v4 Jan 2026). Cost column is a teaching simplification.

A chef can improve by a note, a new knife, a new kitchen, or new instincts

A dish gets sent back. The chef can tape a note to the station, buy a tool, reorganize the kitchen, or practice for months.



THE VARIABLES

Context
Sticky note on the station

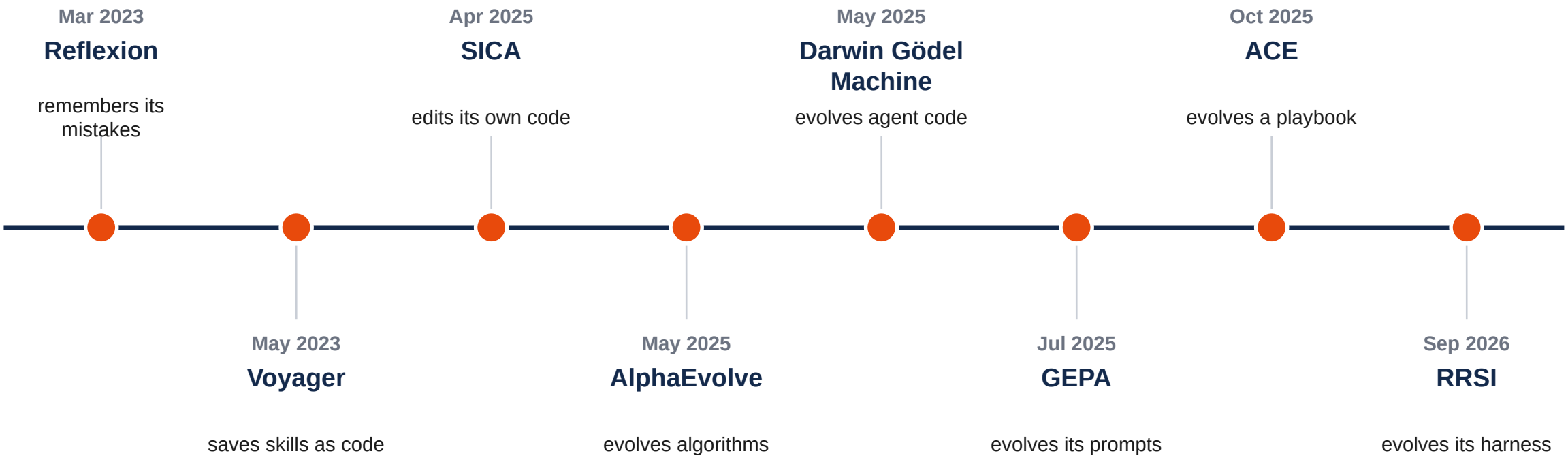
Tools & skills
A new knife; a written recipe

Architecture
Reorganized kitchen workflow

Weights
Months of practice, new instincts

HOW WE GOT HERE

In three years, agents went from remembering mistakes to rewriting their own code



Dates: arXiv first submission (Reflexion 2303.11366, Voyager 2305.16291, SICA 2504.15228, DGM 2505.22954, GEPA 2507.19457, ACE 2510.04618, RRSI 2609.24972); AlphaEvolve: DeepMind blog, May 14 2025.

CHECKPOINT 1

Which of these four research agents is self-improving?

WHAT YOU NEED Self-improving = uses feedback on its own work to change something it will use on FUTURE tasks.

A Switches to a newer, smarter model every quarter

B Logs each Priya correction into a rules file it loads before every new memo

C Retrieves 3x more documents before writing each memo

D Drafts a memo, critiques it, rewrites it, then sends, next memo starts from scratch

CHECKPOINT 1 · ANSWER

Which of these four research agents is self-improving?

A Switches to a newer, smarter model every quarter

B Logs each Priya correction into a rules file it loads before every new memo

C Retrieves 3x more documents before writing each memo

D Drafts a memo, critiques it, rewrites it, then sends, next memo starts from scratch

WHY Only the rules file carries feedback forward. Self-critique improves one memo then forgets; a new model or more documents change inputs, not what it learned.

UP NEXT · 2 OF 6

Second: the harness, where the learning actually happens

01	What self-learning means	Same mistake twice, and the four layers that can change
02	**The learning harness**	Grade, reflect, save, and three harness trends
03	Memory and experience	Lessons, skills, playbooks, and dreaming
04	Recursive language models	For reference · reading inputs too long for one prompt
05	Rewards and continual learning	For reference · learning from scores; self-editing code
06	Evaluation	For reference · proving it improved, catching cheating

02

The learning harness

Everything around the model: the loop, the tools, the grader, and the place lessons are kept.

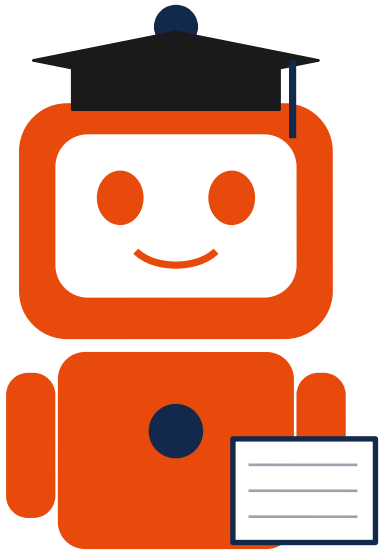
SECTION 2 · HARNESS

Agent harness

DEFINITION

Agent harness

All the software around a model that turns it into an agent: the loop, tools, context, memory, permissions, and how it checks its own work.



WHY IT MATTERS

You can't retrain Claude or GPT. You can change the harness, and that is where most practical improvement comes from.

The same new hire does better with a good desk, checklist and manager

Two equally smart analysts. One gets a checklist, the right files, and a manager who reviews drafts. One gets nothing.

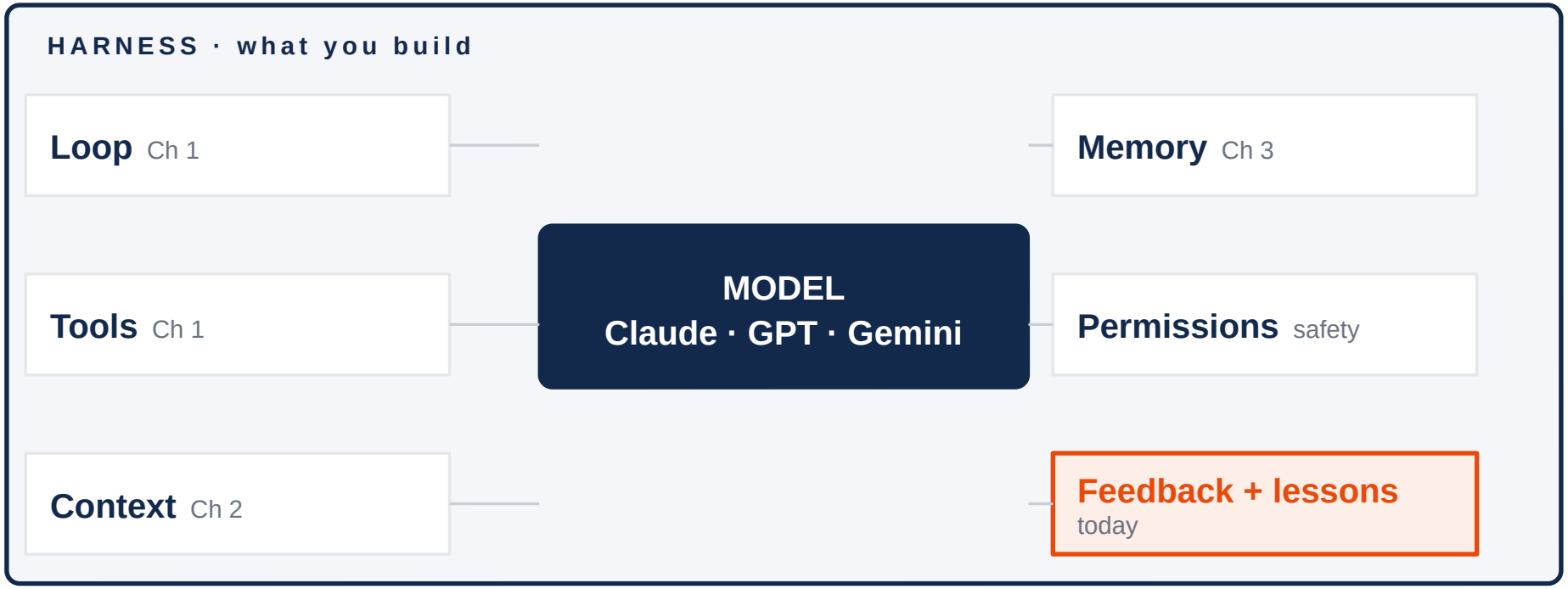
THE VARIABLES

Model
The analyst's raw ability

Harness
Checklist, files, review process

Output
Quality of the memo

The model is the engine; the harness is the car you build around it



Same model + better harness = better agent.

SECTION 2 · HARNESS

Harness engineering

The job moved from writing the prompt, to curating the context, to building the whole harness

2022 to 24

Prompt engineering

Write better instructions

SESSION 2

→

mid-2025

Context engineering

Choose what the model sees on each call

SESSIONS 2 to 3

→

2026

Harness engineering

Build the loop, tools, checks and memory around it

TODAY

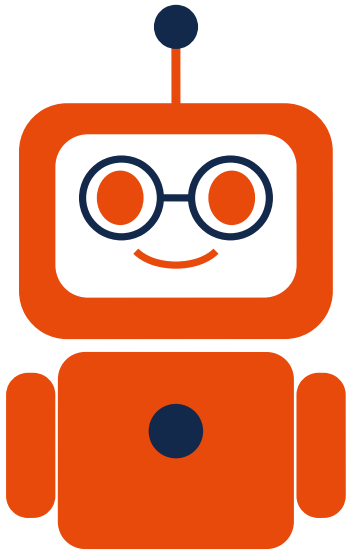
Each layer contains the last: a harness still needs good prompts and context.

Teaching synthesis. Sources: Anthropic, Effective context engineering (Sep 29, 2025); M. Hashimoto, My AI Adoption Journey (Feb 5, 2026); OpenAI, Harness engineering (Feb 11, 2026).

DEFINITION

Harness engineering

Designing everything around the model, instructions, tools, checks and feedback, so the agent does the job right without a human fixing each output.



WHY IT MATTERS

LangChain's short version: Agent = Model + Harness. You can't change the model, so the harness is the part you engineer.

EVERYDAY EXAMPLE · HARNESS ENGINEERING

House rules on the fridge: each one was added after something went wrong

Someone forgot to lock the door, so “Lock the door” goes on the list. Milk went bad, so “Label your leftovers.” The list grows; the roommates stay the same.

THE VARIABLES

Roommate
The model

List on the fridge
The harness

Adding a rule after a mishap
Harness engineering

A good harness both guides the agent before it acts and checks it after

Guides · before

Instructions file (AGENTS.md)

Templates, examples, a map of the docs

Firm: “always use fiscal-year revenue”

Sensors · after

Tests, linters, a second-model review

Error messages that say how to fix it

Firm: “every figure must match the 10-K”

Sensors are where self-learning starts: each failed check is feedback to save.

CHECKPOINT 2

Which change is harness engineering rather than model training?

WHAT YOU NEED The harness = everything around the model: instructions, tools, checks, memory. You can't change the model's weights through an API.

A Adding a check that rejects any memo whose figures don't match the 10-K

B Waiting for the next model release

C Asking the vendor to retrain on finance data

D Fine-tuning the model on 10,000 corrected memos

CHECKPOINT 2 · ANSWER

Which change is harness engineering rather than model training?

A **Adding a check that rejects any memo whose figures don't match the 10-K**

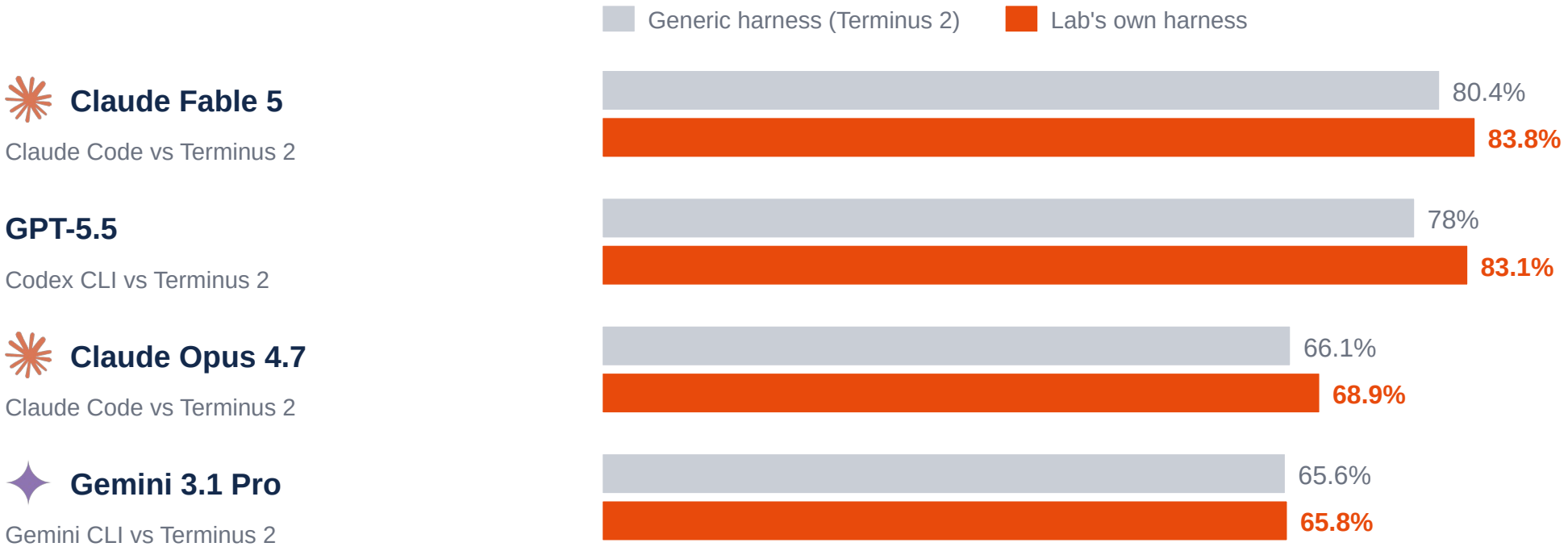
B Waiting for the next model release

C Asking the vendor to retrain on finance data

D Fine-tuning the model on 10,000 corrected memos

WHY A check (a sensor) lives in the harness. Fine-tuning, waiting for a release and vendor retraining all change the model itself.

Each lab's own harness beats a generic one running the same model



Gaps of 0 to 5 points: the harness matters, but the model matters more.

Terminal-Bench 2.1 leaderboard as mirrored by Snorkel (snorkel.ai/leaderboard/terminal-bench-2-1), rows dated Apr to Jun 2026. tbench.ai itself could not be read.

EVIDENCE · HARNESS CHANGES ONLY

LangChain raised its agent 14 points without touching the model



Model fixed at gpt-5.2-codex. Checklists, loop detection and self-verification did the work.

LangChain, "Improving Deep Agents with harness engineering" (Feb 17, 2026). Terminal-Bench 2.0 accuracy.

~1M

lines of code shipped in about five months by a small OpenAI team, with zero lines written by hand. Engineers built the harness; agents wrote the code.

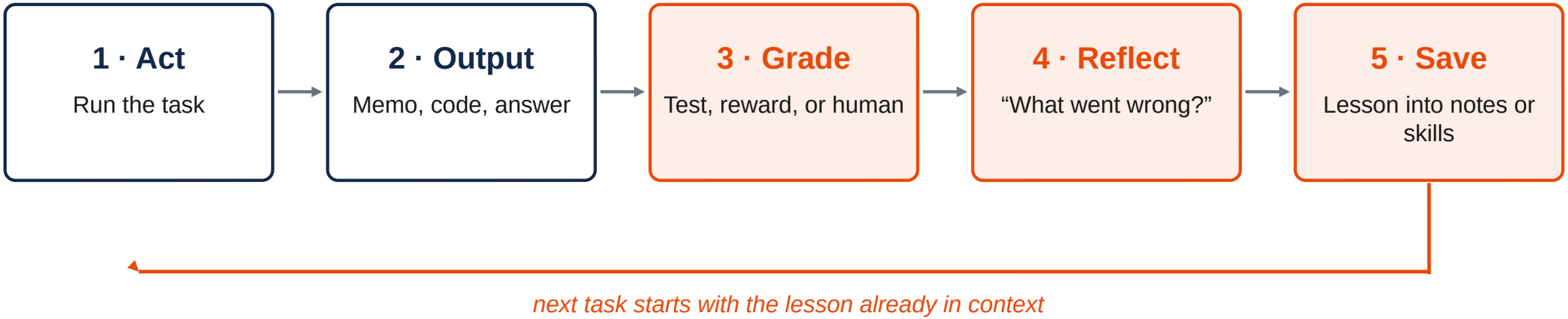
Source: OpenAI, “Harness engineering” (Feb 11, 2026): ~1,500 PRs, team of 3→7 engineers, ~3.5 PRs per engineer per day.

SECTION 2 · HARNESS

The learning loop

THE LEARNING LOOP

Self-learning adds three steps after the task: grade, reflect, save



NEW TODAY Steps 1 to 2 are the Chapter 1 agent. Steps 3 to 5 make it learn.

CHECKPOINT 3

The agent noticed its mistake, yet repeated it next week. Which step is missing?

WHAT YOU NEED It graded its memo and wrote “I used calendar-year revenue,” then finished. Loop: 1 Act → 2 Output → 3 Grade → 4 Reflect → 5 Save where the next task reads it.

A Step 4: Reflect

B Step 5: Save

C Step 1: Act

D Step 3: Grade

CHECKPOINT 3 · ANSWER

The agent noticed its mistake, yet repeated it next week. Which step is missing?

A Step 4: Reflect

B Step 5: Save

C Step 1: Act

D Step 3: Grade

WHY It graded and reflected, but never saved the lesson where the next task reads it, so nothing carried over.

STEP 3 · GRADE

The grader decides what the agent learns, so pick it carefully

Feedback source	Example at the firm	Strength	Weakness
Automatic test	Figures match the 10-K	Cheap, objective	Only checks what you test
Reward / score	Client rating 1 to 5	Scales to many tasks	Can be gamed
LLM judge	Second model grades tone	Flexible	Shares the model's blind spots
Human	Priya's corrections	Most trusted	Slow and costly

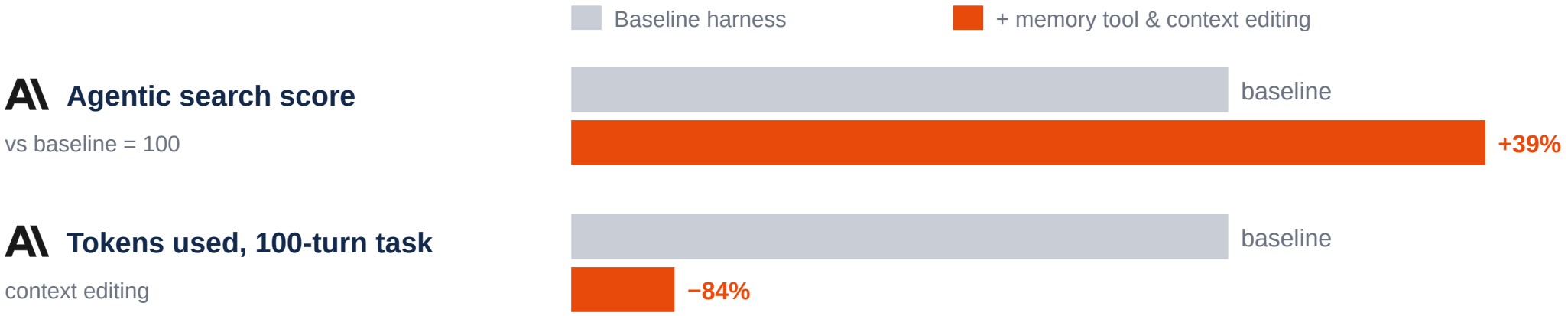
Most good harnesses combine an automatic check with a human on the decisions that matter.

Notes, tests and commits let each new session pick up where the last stopped

1. An initializer agent writes a progress file and a feature list, all marked failing.
2. Each session reads the progress file and picks one failing feature.
3. It tests end to end, like a real user, before marking it passing.
4. It commits to git and updates the progress notes for the next session.

The model forgets between sessions. The harness doesn't.

Giving the agent a memory file and pruning old context lifted results 39%



Better answers and far fewer tokens, from harness features, not a new model.

Anthropic, “Managing context on the Claude Developer Platform” (Sep 29, 2025). Internal evaluations; indexed to baseline = 100.

AT CHAMPAIGN CAPITAL RESEARCH

Our research agent's harness gets a grader and a lessons file

BEFORE · SESSION 3 AGENT

Loop + tools + memory of clients

Priya corrects the memo by email

Correction is lost after the task

AFTER · SESSION 6 AGENT

Grader: figures must match the 10-K

Priya's edits captured as feedback

Lessons saved to lessons.md

lessons.md loaded into every new memo

Three trends will decide what harnesses look like next

01	**Self-improving harnesses**	The agent rewrites its own harness
02	**Models eating harnesses**	Reasoning models make scaffolding unnecessary
03	**Domain-specific harnesses**	Finance, legal and health wrap general models

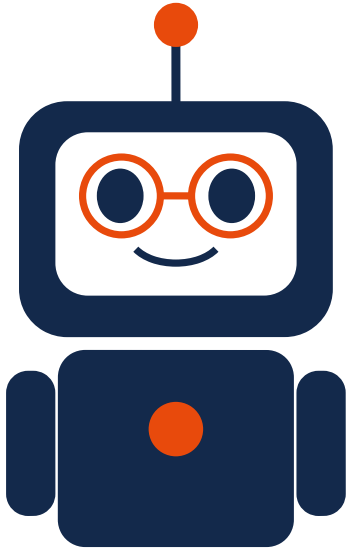
SECTION 2 · HARNESS TRENDS

Self-improving harness

DEFINITION

Self-improving harness

The agent reads its own failure logs, proposes a small change to its harness, prompts, tools, workflow code, and keeps it only if tests still pass.



WHY IT MATTERS

It automates harness engineering. The model stays frozen; the system around it gets better on its own.

A new hire who keeps rewriting their own how-to guide after every mistake

Each time a task goes wrong, they check the notes, find the step that failed, and fix the guide, but only if the old tasks still work.

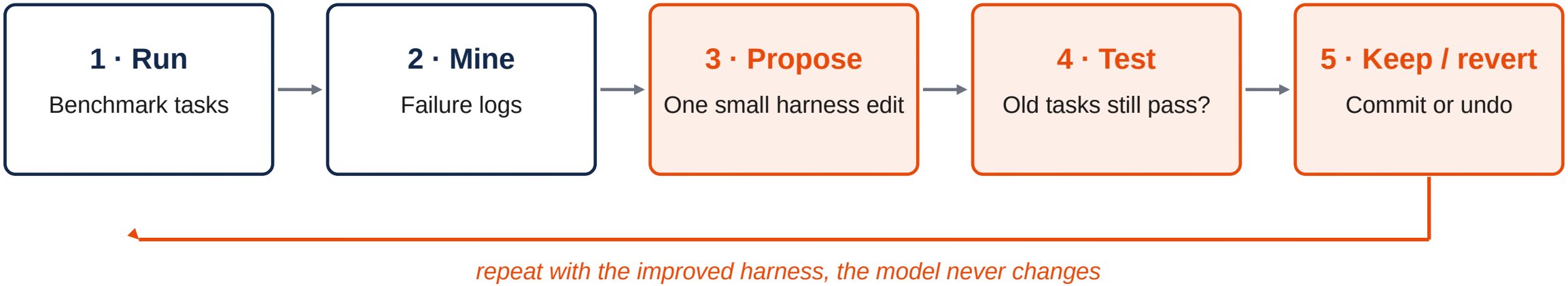
THE VARIABLES

The person
The model (unchanged)

The how-to guide
The harness

Re-checking old tasks
Regression tests

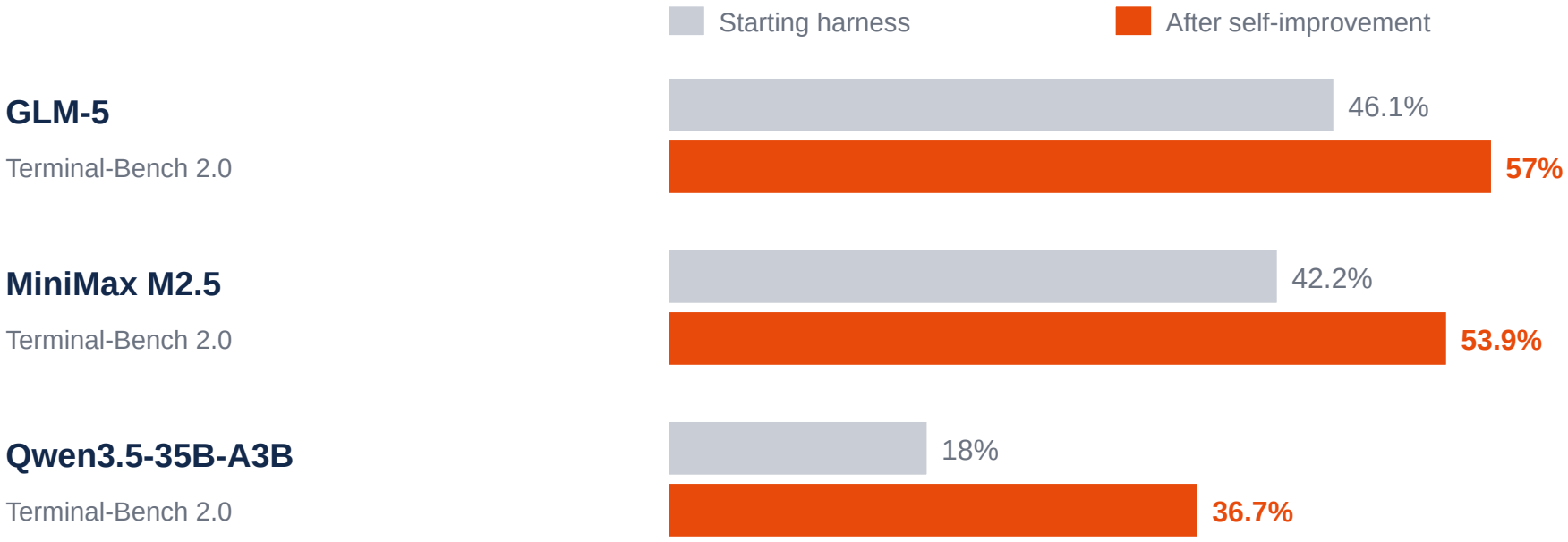
Read the failures, edit the harness, keep the edit only if nothing breaks



WHAT GETS EDITED Most gains came from tools, middleware and memory, not the system prompt.

Sources: Zhang et al., Self-Harness, arXiv 2606.09498 (Jun 2026); Lin et al., Agentic Harness Engineering, arXiv 2604.25850 (Apr 2026).

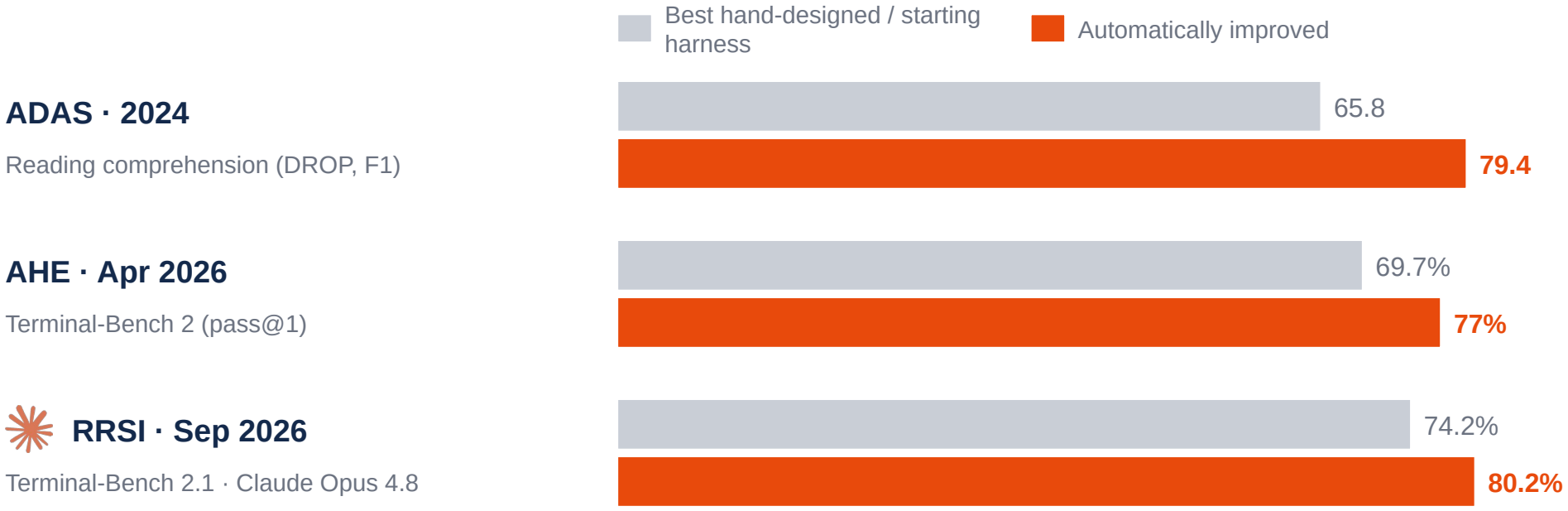
Harnesses that rewrote themselves lifted every model tested on terminal tasks



Smallest model gained most: 18% → 37%. Same model weights throughout.

Zhang et al., "Self-Harness: Harnesses That Improve Themselves," arXiv 2606.09498 (Jun 2026), Table 1, overall pass rate.

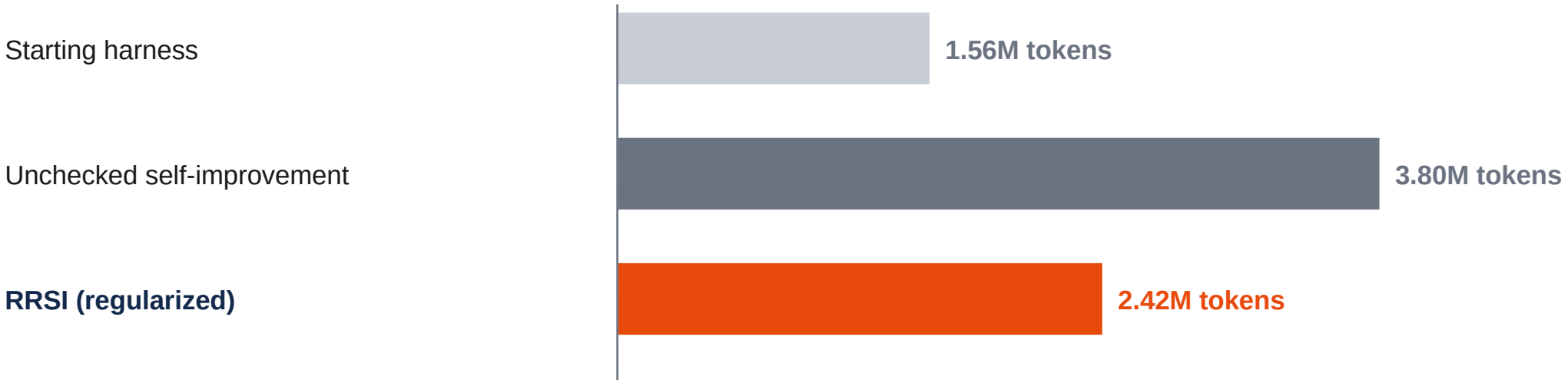
Automatically designed harnesses now beat ones written by expert engineers



AHE's 77.0% beat OpenAI's hand-built Codex CLI harness (71.9%) on the same test.

Hu, Lu & Clune, ADAS, arXiv 2408.08435, Table 1; Lin et al., AHE, arXiv 2604.25850; Xia et al., RRSI, arXiv 2609.24972. Different benchmarks, compare within rows.

Left unchecked, a self-improving harness bloated to 2.4x the tokens



Unchecked edits also overfit: best on practice tasks, worse on new ones.

Xia et al., RRSI, arXiv 2609.24972 (Sep 2026), tokens per trial on Harvey's legal benchmark (LAB). Abstract cites ~30% fewer tokens than unregularized.

CHECKPOINT 4

Which result shows a self-improving harness is overfitting?

WHAT YOU NEED RRSI (Sep 2026): unchecked self-evolution scored 92.8 on the tasks it evolved on vs 90.5 for the regularized version, but 88.9 vs 89.2 on held-out tasks.

A It scores higher on practice tasks but lower on new tasks than a constrained version

B It uses fewer tokens per task

C It makes fewer edits each round

D It scores higher on both practice and new tasks

CHECKPOINT 4 · ANSWER

Which result shows a self-improving harness is overfitting?

A **It scores higher on practice tasks but lower on new tasks than a constrained version**

B It uses fewer tokens per task

C It makes fewer edits each round

D It scores higher on both practice and new tasks

WHY Gains that appear only on the tasks it trained on are overfitting, the train/test gap from Session 4.

SECTION 2 · HARNESS TRENDS

Models eating harnesses

Every harness part is a bet on what the model can't do yet, and models keep winning

- Anthropic: each harness component encodes an assumption about the model's limits.
- Those assumptions go stale as models improve.
- Sutton's Bitter Lesson: general methods that scale with compute win.

Anthropic, Harness design for long-running apps (Mar 24, 2026); R. Sutton, The Bitter Lesson (Mar 13, 2019).

GPS made the paper map and the route plan unnecessary, not the car or the road

Drivers once planned routes by hand with maps and notes. GPS absorbed that work. They still need the car, the fuel and the road.

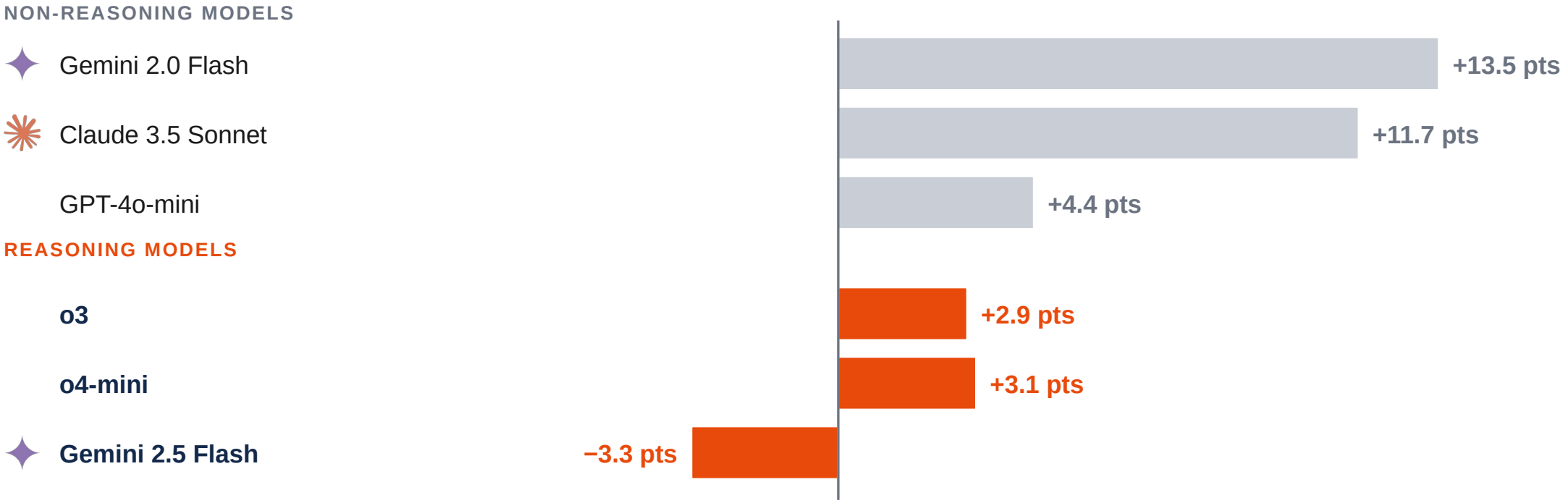
THE VARIABLES

Route planning
Scaffolding the model absorbs

GPS
A better reasoning model

Car, fuel, road
Tools, data, permissions, still needed

“Think step by step” helped older models; reasoning models barely need it



For reasoning models, the prompt trick also added 20 to 80% to response time.

Meincke, Mollick et al., Wharton GAIL Prompting Science Report 2, arXiv 2506.07142 (Jun 2025), GPQA Diamond, accuracy change from chain-of-thought prompt.

When a newer Claude fixed a problem on its own, Anthropic deleted that harness part

1. Sonnet 4.5 got “context anxiety”: it rushed to finish as its context filled up.
2. So the harness added context resets: wipe and restart from notes.
3. Opus 4.5 largely stopped the behavior on its own.
4. Anthropic dropped context resets from the harness entirely.

Re-test every harness part when a new model ships. Delete what it no longer needs.

100 lines

of Python, with bash as its only tool, mini-SWE-agent scores over 74% on SWE-bench Verified with recent models.

Source: SWE-agent team, mini-SWE-agent README and docs (Gemini 3 Pro: 74%). In 2024 the full SWE-agent harness let GPT-4 Turbo solve 12.5% of full SWE-bench.

Models absorb the scaffolding for thinking, not the plumbing for acting

Getting absorbed

“Think step by step” prompts

Hand-written planning steps

Context resets and retry tricks

Elaborate custom tool wrappers

Still needed

Access to data and tools (EDGAR, FactSet)

Tests and graders that check the work

Memory across sessions

Permissions, audit logs, human approval

Build the harness around what the model can't reach or can't be trusted to check.

CHECKPOINT 5

A new reasoning model ships. Which harness part should you test removing first?

WHAT YOU NEED Wharton (2025): adding “think step by step” moved reasoning models only −3.3 to +3.1 points, and made them 20 to 80% slower. Reasoning models already think internally.

A Priya's approval before client emails

B The connector to the 10-K database

C The “think step by step” line in the prompt

D The audit log of every tool call

CHECKPOINT 5 · ANSWER

A new reasoning model ships. Which harness part should you test removing first?

A Priya's approval before client emails

B The connector to the 10-K database

C The “think step by step” line in the prompt

D The audit log of every tool call

WHY It compensated for a weakness reasoning models no longer have. Data access, logs and human approval cover things the model can't do or can't be trusted with.

CHECKPOINT 6

What's the best lesson for a bank from BloombergGPT?

WHAT YOU NEED Bloomberg trained its own 50B finance model; months later general GPT-4 beat it on 4 of 5 finance tests. A domain harness = any model + the firm's data, skills and checks.

A Never use AI for finance

B Train an even bigger finance model

C Build the data, skills and checks around the model, and keep the model swappable

D Only use the cheapest model available

CHECKPOINT 6 · ANSWER

What's the best lesson for a bank from BloombergGPT?

A Never use AI for finance

B Train an even bigger finance model

C Build the data, skills and checks around the model, and keep the model swappable

D Only use the cheapest model available

WHY Custom models get overtaken by the next general model. The harness, data, skills, checks, survives every model upgrade.

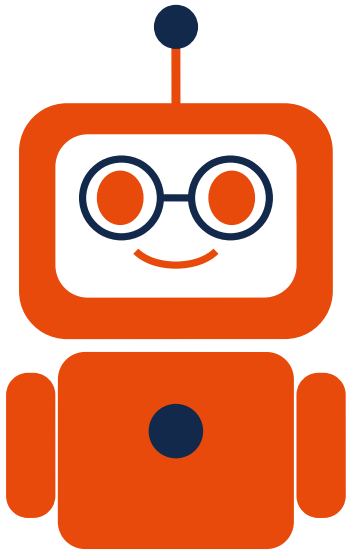
SECTION 2 · HARNESS TRENDS

Domain-specific harness

DEFINITION

Domain-specific harness

A harness built for one industry: the general model plus that industry's data feeds, ready-made skills, review steps and compliance rules.



WHY IT MATTERS

Companies stopped training their own finance or legal models. They wrap the best general model in a domain harness, and can swap the model.

Same smart graduate, different first-day kit at a bank versus a hospital

A bank gives the new analyst a Bloomberg login, model templates and a compliance checklist. A hospital gives a different kit. Same brain.

THE VARIABLES

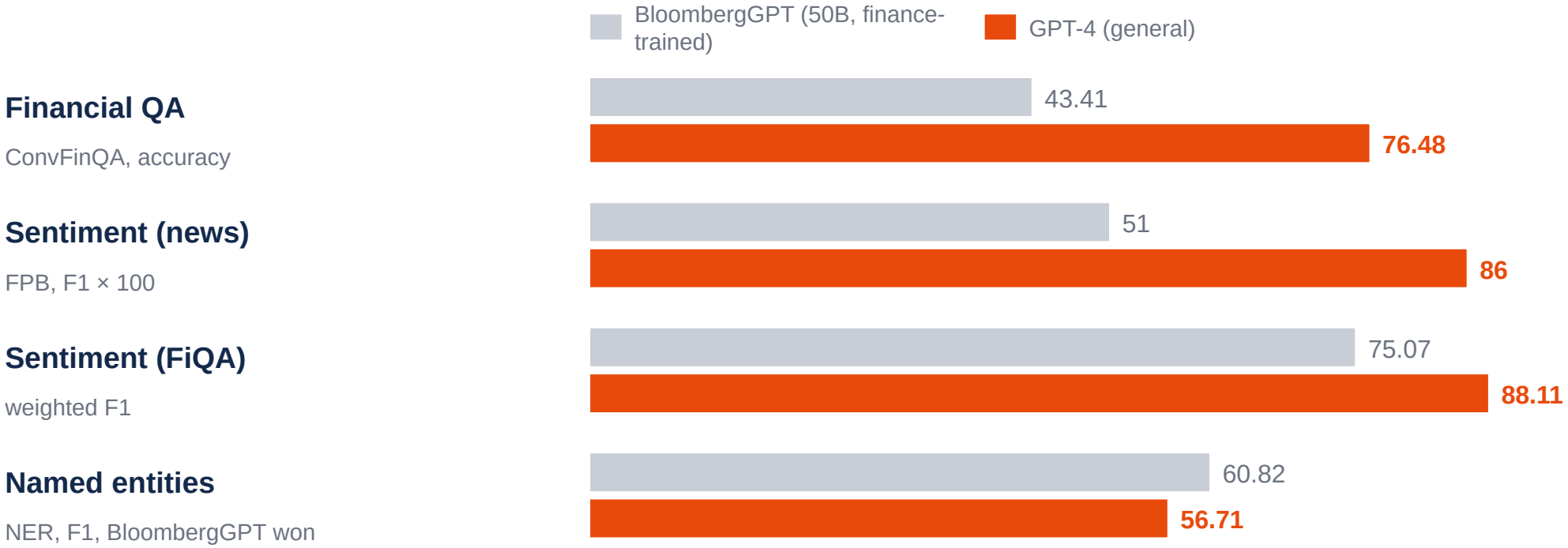
The graduate
The general model

Logins and data
Data connectors

Templates
Skills: DCF, comps, memos

Compliance checklist
Review and approval steps

A general model beat Bloomberg's own finance model on 4 of 5 finance tests



Domain models got overtaken by general ones. Domain harnesses are how firms keep an edge.

Li et al., arXiv 2305.05862 (May 2023), as reported in their tables; BloombergGPT: Wu et al., arXiv 2303.17564 (Mar 2023). Settings differ by task.

ANATOMY OF A FINANCE HARNESS

A finance harness wraps the model in data, skills, workflows and controls

Review & governance	Human approval · per-tool permissions · audit log · info barriers
Workflows	10 agent templates: pitch builder, earnings reviewer, KYC screener…
Skills	/comps · /dcf · /lbo · 3-statement model · /debug-model
Data connectors	FactSet · S&P Capital IQ · PitchBook · Morningstar · LSEG · Moody's
Model · swappable	Claude · GPT · Gemini, whichever scores best this quarter

WHO BUILDS THEM

Anthropic · Claude for Financial Services

OpenAI · ChatGPT for Financial Services

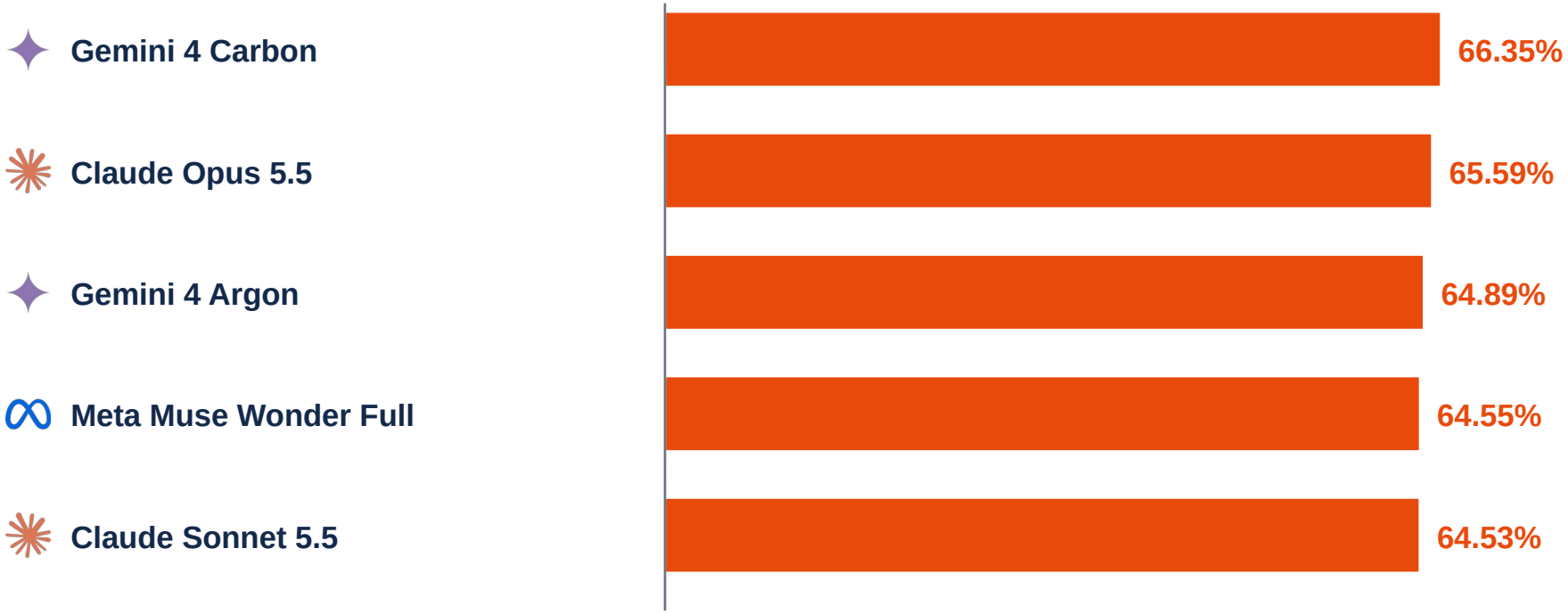
Rogo · Hebbia (finance)

Harvey (legal)

OpenEvidence (medicine)

Sources: Anthropic, Agents for financial services (May 5, 2026); github.com/anthropics/financial-services; OpenAI, Introducing ChatGPT for Financial Services (Sep 10, 2026).

On a finance harness, the top five models land within two points of each other



When models are this close, the data, skills and checks around them decide the result.

Vals AI Finance Index (released Oct 21, 2026; as of Oct 27, 2026), weighted score across Finance Agent v2.1, Excel modeling (LBO, DCF, comps) and tax research.

Harvey's domain context layer raised legal-task pass rates up to 30% and cut cost up to half

- Harvey Graph: a knowledge graph of US case law plus each firm's own documents.
- Early tests: rubric pass rate up to +30%; task cost down up to ~50%.
- Partners approve each agent step before the next one runs.

Harvey blog, Improving agent accuracy with Harvey Graph (Oct 19, 2026) and Legal work orchestration (Oct 15, 2026). Company-reported, early results.

AT CHAMPAIGN CAPITAL RESEARCH

Our research agent becomes a finance harness, and the model becomes swappable

GENERIC CHATBOT

- Knows finance in general
- Can't see our filings or data
- No house templates
- Nothing checks its numbers

CHAMPAIGN CAPITAL HARNESS

- EDGAR + FactSet connectors
- Skills: house comps and DCF
- Tick-and-tie: figures match the 10-K
- lessons.md + Friday dreaming
- Priya approves; every step logged

New model next month? Swap it in. Keep everything else.

CHECKPOINT 7

LangChain's score rose from 52.8% to 66.5% with the same model. What drove most of the gain?

WHAT YOU NEED Turning reasoning effort up alone gave 53.9%. Other changes: a checklist forcing a verification pass, loop detection, a startup map of the folder, time warnings.

A Harness changes, mostly verification and loop detection

B Luck: benchmarks are noisy

C Extra reasoning effort

D A bigger model

CHECKPOINT 7 · ANSWER

LangChain's score rose from 52.8% to 66.5% with the same model. What drove most of the gain?

A **Harness changes, mostly verification and loop detection**

B Luck: benchmarks are noisy

C Extra reasoning effort

D A bigger model

WHY Reasoning alone added 1.1 points. The remaining ~12.6 came from harness changes that check and steer the work.

UP NEXT · 3 OF 6

Third: memory and experience, where lessons are kept

01	What self-learning means	Same mistake twice, and the four layers that can change
02	The learning harness	Grade, reflect, save, and three harness trends
03	**Memory and experience**	Lessons, skills, playbooks, and dreaming
04	Recursive language models	For reference · reading inputs too long for one prompt
05	Rewards and continual learning	For reference · learning from scores; self-editing code
06	Evaluation	For reference · proving it improved, catching cheating

03

Memory and experience

The cheapest way to learn: write the lesson down where the agent will read it next time.

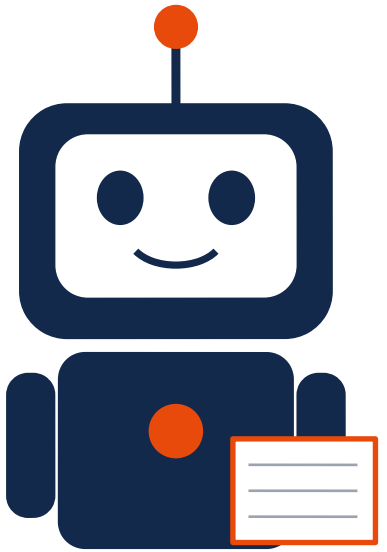
SECTION 3 · MEMORY

Reflection

DEFINITION

Reflection memory

After a failed attempt, the model writes a short note in plain English on what went wrong. The note is fed back into its next attempt.



WHY IT MATTERS

No retraining: the lesson is just text in the context. Cheap, readable, and easy to delete if it's wrong.

A sticky note on your monitor: “check the units first”

You lost points on two homeworks for mixing up millions and thousands. You stick a note on your screen. It doesn't happen again.

THE VARIABLES

Failure

Points lost on units

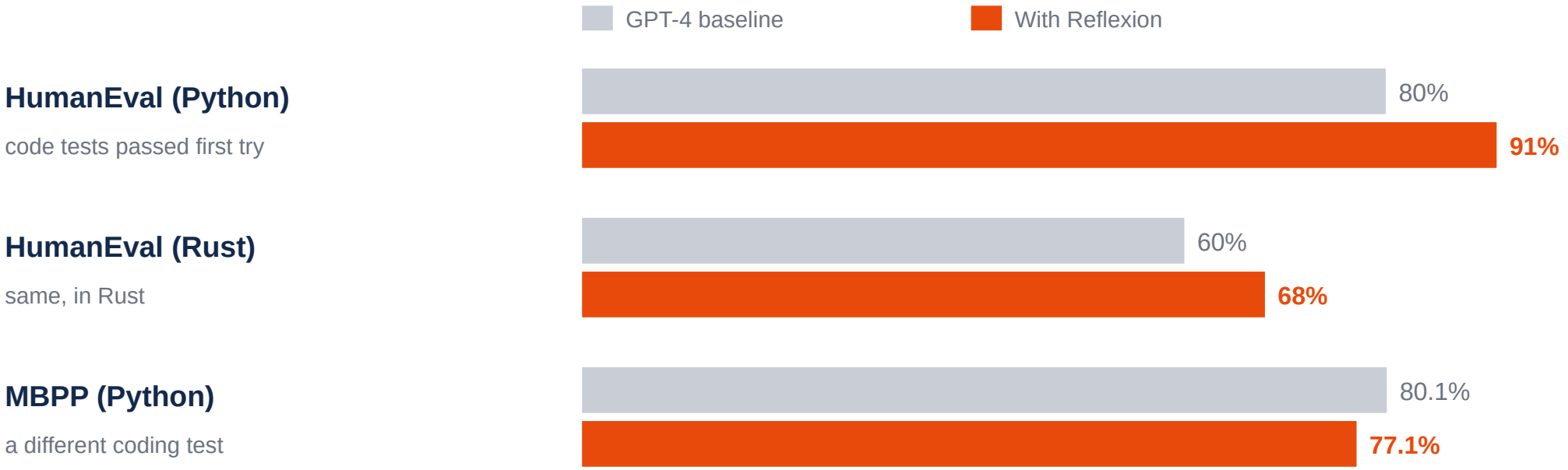
Reflection

“Check units before submitting”

Memory

Sticky note you see every time

Writing down its mistakes lifted coding accuracy from 80% to 91%



Not a free lunch: on MBPP Python, reflection made it slightly worse.

Shinn et al., Reflexion, arXiv 2303.11366 (Mar 2023), Tables 1 to 2. pass@1 accuracy.

CHECKPOINT 8

Why did writing down lessons make Reflexion worse on one coding test?

WHAT YOU NEED It rose 80% → 91% on one test but fell 80.1% → 77.1% on another. There, its feedback came from unit tests it wrote for itself, and some were wrong.

A The lessons were too long to read

B Wrong feedback produced wrong lessons

C The model forgot its lessons

D That test had no time limit

CHECKPOINT 8 · ANSWER

Why did writing down lessons make Reflexion worse on one coding test?

A The lessons were too long to read

B Wrong feedback produced wrong lessons

C The model forgot its lessons

D That test had no time limit

WHY A lesson is only as good as the grader behind it. Bad tests taught bad lessons.

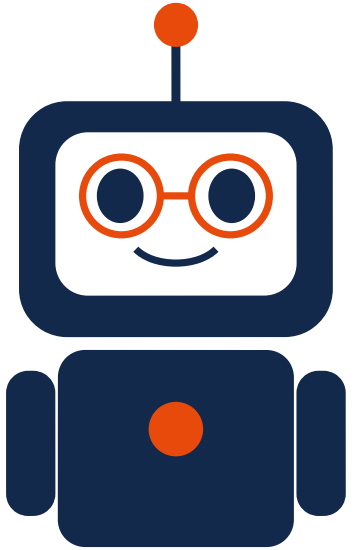
SECTION 3 · MEMORY

Skill library

DEFINITION

Skill library

When an agent solves something new, it saves the working procedure, often as code, under a name, and reuses it instead of re-solving.



WHY IT MATTERS

Experience compounds: each saved skill becomes a building block for harder tasks later.

EVERYDAY EXAMPLE · SKILL LIBRARY

A recipe card box: cook it once, write it down, reuse forever

The first time you make lasagna it takes three hours and a video. You write a card. Next time: twenty minutes.

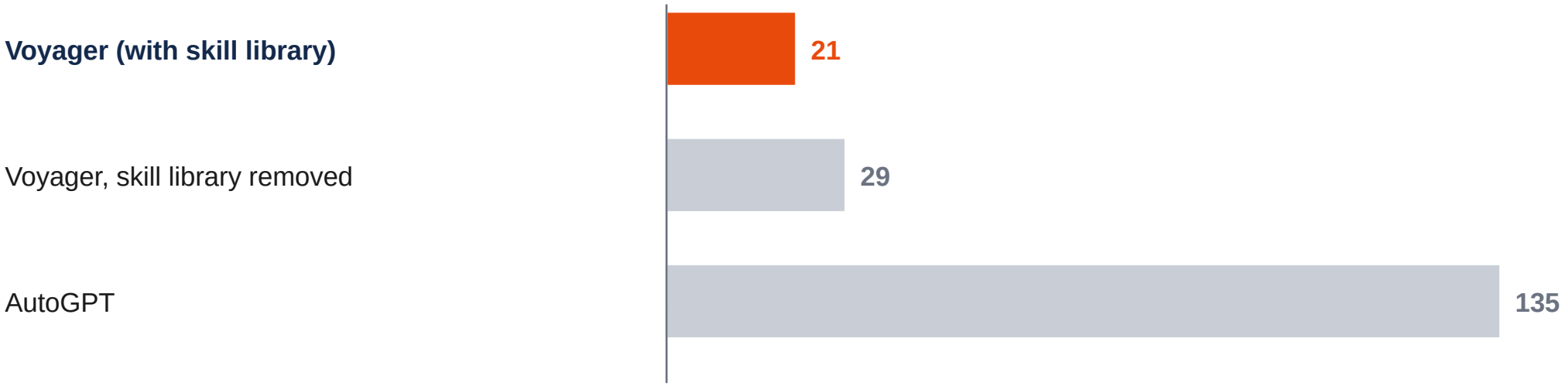
THE VARIABLES

New task
First lasagna

Skill saved
The recipe card

Reuse
Faster next time; build on it

With a skill library, the agent reached iron tools in 21 tries; AutoGPT needed 135



Only Voyager with its library ever reached diamond tools (1 of 3 runs).

Wang et al., Voyager, arXiv 2305.16291, Table 1, prompting iterations to unlock iron tools in Minecraft (mean of 3 trials; fewer is better).

CHECKPOINT 9

What did Voyager's saved skills help with most?

WHAT YOU NEED With skill library: iron in 21 tries, diamond reached (1 of 3 runs). Without: iron in 29, diamond never. Goals build on each other: wood → stone → iron → diamond.

A Every goal equally

B Only the first, easiest goal

C The hardest goals, which build on many earlier skills

D Nothing, the difference is noise

CHECKPOINT 9 · ANSWER

What did Voyager's saved skills help with most?

A Every goal equally

B Only the first, easiest goal

C The hardest goals, which build on many earlier skills

D Nothing, the difference is noise

WHY Easier goals were reachable either way. Only the library reached diamond: skills compound.

Agent Skills turn saved procedures into folders any agent can load

- A skill: a folder with a SKILL.md and optional scripts.
- Agent reads the one-line description; loads full skill only when needed.
- Now an open standard: 70+ platforms, ~1M skills published.

Anthropic hopes agents will create and evaluate skills themselves; today humans review them.

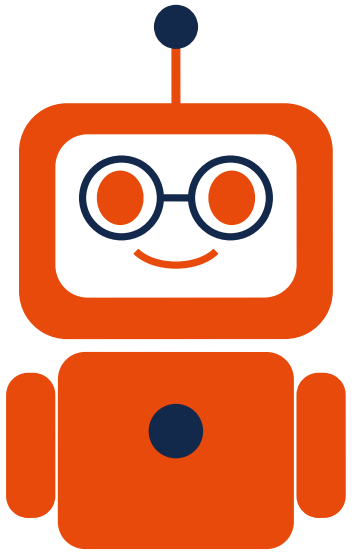
SECTION 3 · MEMORY

Playbook

DEFINITION

Evolving playbook

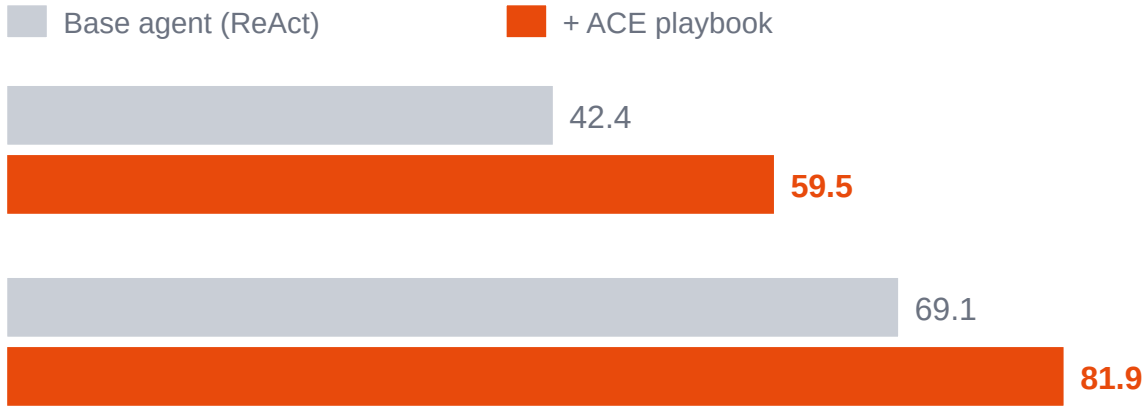
A structured set of strategies the agent edits after each task, adding, refining, or removing bullets, instead of rewriting everything.



WHY IT MATTERS

Small edits keep hard-won detail. Rewriting a whole summary each time tends to shrink it until lessons are lost.

An evolving playbook added 13 to 17 points on agent and finance tasks



Same model (DeepSeek-V3.1) throughout. Only the context changed.

Zhang et al., ACE, arXiv 2510.04618, Tables 1 to 2. AppWorld: online, no labels. Finance: offline, with labels. Average accuracy.

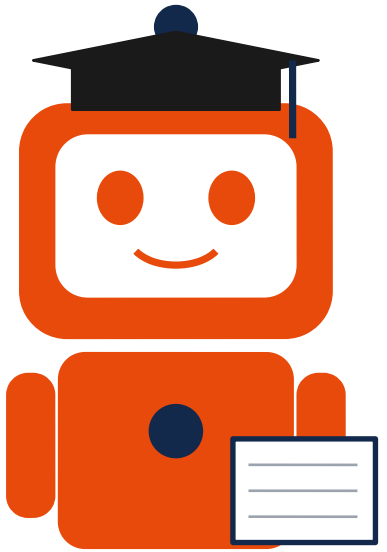
SECTION 3 · MEMORY

Dreaming

DEFINITION

Dreaming (memory consolidation)

Between tasks, a background job rereads past sessions and the memory notes, then merges duplicates, fixes contradictions, drops stale entries and writes the patterns it finds.



WHY IT MATTERS

Notes written mid-task pile up and contradict each other. Dreaming keeps memory small, current and useful.

Cleaning up your class notes every Sunday before the week starts

All week you scribble notes fast. On Sunday you reread them, cross out what's wrong, merge repeats, and write a one-page summary.

THE VARIABLES

Daytime notes
Quick memories saved during tasks

Sunday review
The dreaming job

Crossed out
Stale or contradicting entries

One-page summary
The clean memory for next week

THE DREAMING ARCHITECTURE

Work by day, consolidate by night: the agent wakes up with cleaner memory



Based on Anthropic, Claude Managed Agents "Dreams" docs (platform.claude.com) and Letta sleep-time agents (Apr 2025).

HOW ANTHROPIC BUILT IT

Claude's dreaming reads up to 100 past sessions and writes a fresh memory you can review

Claude Managed Agents · Dreaming · research preview, May 2026

IN	DREAM JOB	OUT
1 memory store	Runs in the background	A NEW memory store
1 to 100 session transcripts	Finds patterns across sessions	Original never changed
Optional instructions	Cleans duplicates and stale notes	Review it, or apply automatically

Harvey (legal AI) reported task completion up ~6x in its tests.

Sources: claude.com/blog, New in Claude Managed Agents (May 2026); platform.claude.com/docs, Managed Agents · Dreams. Harvey figure is a customer claim.

WHO HAS BUILT IT

Anthropic, OpenAI and Letta all ship “dreaming”, each runs it when the agent is idle

COMPANY	FEATURE	WHEN IT RUNS
Anthropic	Managed Agents dreaming · Claude Code auto-dream	**On request or in background**
OpenAI	ChatGPT Dreaming · Codex memories	**Background; Codex waits until chat is idle**
Letta	Sleep-time agents → “dreaming”	**Every N steps or on compaction**
Google	Gemini memory of past chats	Not documented
LangChain	LangMem background memory	**After a chat ends or goes idle**

Sources: claude.com/blog & platform.claude.com (Anthropic); openai.com/index/chatgpt-memory-dreaming; learn.chatgpt.com (Codex); letta.com/blog/sleep-time-compute; docs.letta.com; blog.google (Aug 13, 2025); LangMem docs.

Doing the thinking before the question arrives cut answer-time compute about 5x



Also up to +13% (GSM-Symbolic) and +18% (AIME) accuracy with more idle-time compute.

Lin, Snell et al. (Letta & UC Berkeley), "Sleep-time Compute: Beyond Inference Scaling at Test-time," arXiv 2504.13171 (Apr 2025). Improvement factor vs. no sleep-time compute.

AT CHAMPAIGN CAPITAL RESEARCH

Every Friday night, the research agent dreams over the week's memos

LESSONS.MD · FRIDAY 5 PM

```
Use fiscal year (Deere)

Fiscal year, not calendar

FY figures; cite 10-K

Tables only for institutions

Tables whenever asked

2025 guidance: $X
```

dream →

LESSONS.MD · AFTER DREAMING

```
Revenue: fiscal year; cite 10-K page

Tables whenever a client asks

NEW: Caterpillar, cite segment note

(2025 guidance removed: stale)
```

Priya approves the new file Monday morning before the agent uses it.

CHECKPOINT 10

Why does Anthropic's dreaming write a new memory store instead of editing the original?

WHAT YOU NEED Dreaming reads up to 100 past sessions plus the memory store, then merges, prunes and resolves contradictions. Changes can be applied automatically or reviewed first.

A Reviewing the result and discarding it if the cleanup went wrong

B Running without any model

C Faster dreaming

D Using less storage

CHECKPOINT 10 · ANSWER

Why does Anthropic's dreaming write a new memory store instead of editing the original?

A **Reviewing the result and discarding it if the cleanup went wrong**

B Running without any model

C Faster dreaming

D Using less storage

WHY Keeping the original untouched means a bad consolidation can be thrown away, a human can stay in the loop.

CHECKPOINT 11

An agent rewrites its whole playbook as a fresh summary after every task. What goes wrong after 50 tasks?

WHAT YOU NEED ACE (2025) calls it “context collapse”: each full rewrite compresses the notes a little more. ACE instead adds, edits or removes single bullets.

A The agent stops reading it

B The playbook grows too long

C Specific hard-won details get squeezed out

D It becomes too expensive to read

CHECKPOINT 11 · ANSWER

An agent rewrites its whole playbook as a fresh summary after every task. What goes wrong after 50 tasks?

A The agent stops reading it

B The playbook grows too long

C Specific hard-won details get squeezed out

D It becomes too expensive to read

WHY A summary of a summary of a summary loses details. Small edits keep them.

SECTIONS 4 TO 6

The rest of the deck is for reference, read it on your own

- 4 · Recursive language models
- 5 · Rewards and continual learning
- 6 · Evaluation

Every slide has full detail in the speaker notes. Checkpoints included for self-testing.

Fourth: recursive language models, when the input is too big to read

01	What self-learning means	Same mistake twice, and the four layers that can change
02	The learning harness	Grade, reflect, save, and three harness trends
03	Memory and experience	Lessons, skills, playbooks, and dreaming
04	**Recursive language models**	For reference · reading inputs too long for one prompt
05	Rewards and continual learning	For reference · learning from scores; self-editing code
06	Evaluation	For reference · proving it improved, catching cheating

GIES

04

Recursive language models

Give the model a huge document as a variable, let it write code to search it, and call itself on the pieces.

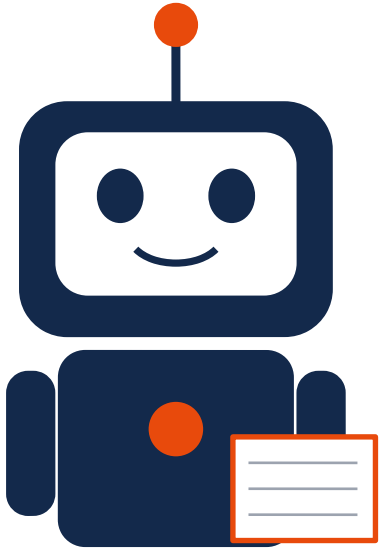
For reference · not covered in class · read on your own

SECTION 4 · RECURSIVE LMS

Background

Context rot

As you put more text into a model's context, its ability to find and use any one piece of it gets worse, well before the window is full.



WHY IT MATTERS

Our agent reads 10-Ks, transcripts and notes. Stuffing all of it into one prompt makes answers worse, not better.

Reading a 1,000-page binder in one sitting, then answering questions from memory

You read every page before the meeting. A one-fact question is easy. “Which clients complained twice?” is not, too much to hold at once.

THE VARIABLES

Context
The 1,000-page binder

Easy task
Find one fact

Hard task
Combine facts across pages

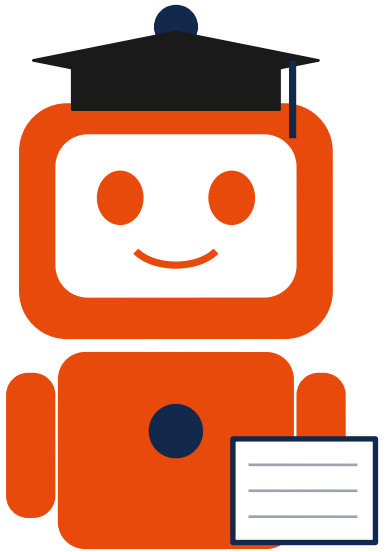
SECTION 4 · RECURSIVE LMS

Recursive language model

DEFINITION

Recursive language model (RLM)

The model never reads the long input directly. It gets it as a variable in a Python workspace, writes code to search and split it, and calls copies of itself on the pieces.



WHY IT MATTERS

The input can be far longer than the context window, the paper handles 6 to 11 million tokens, and the model's own context stays small.

A manager with a filing cabinet and three interns

The manager never reads every file. She checks the index, searches for keywords, hands folders to interns, and combines their one-page notes.

THE VARIABLES

Filing cabinet
The long input, stored as a variable

Searching the index
Code: peek, grep, split

Interns
Sub-calls to a model on each piece

Final memo
The root model's answer

The root model writes code over the document and calls sub-models on the pieces

Question
“Which clients raised guidance twice?”

ROOT MODEL
sees the question, not the document

PYTHON WORKSPACE (REPL)

```
context = <10 years of 10-Ks · 8M tokens>

print(context[:2000])          # peek
hits = re.findall("guidance", context)  # grep
chunks = split_by_year(context)  # partition
notes = [llm_query(c) for c in chunks]
FINAL(combine(notes))
```

short answers come back as Python variables

sub-model · 2016 to 19

sub-model · 2020 to 22

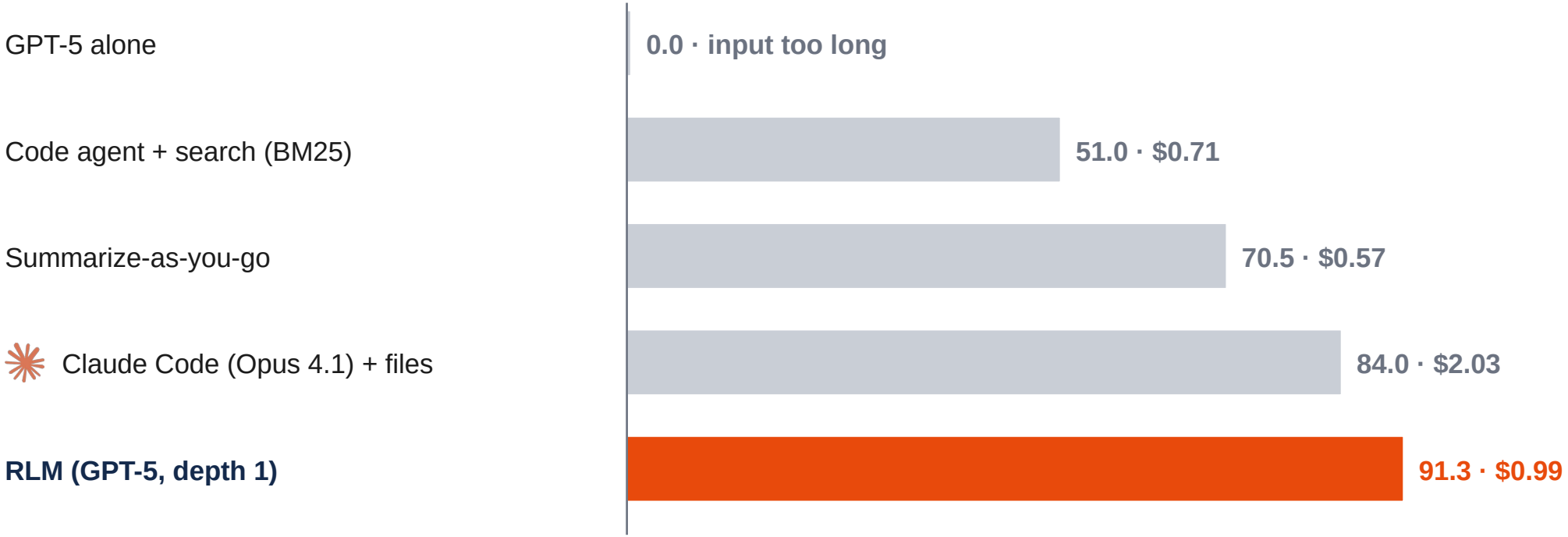
sub-model · 2023 to 25

Ten years of Deere filings won't fit in one prompt, an RLM reads them in pieces

1. Store all ten 10-Ks as one variable: about 8M tokens.
2. Root model peeks, then searches for “segment revenue.”
3. Splits by year; a cheap sub-model extracts each year's figure.
4. Root model checks the ten numbers and builds the table.

No model ever reads all ten filings at once.

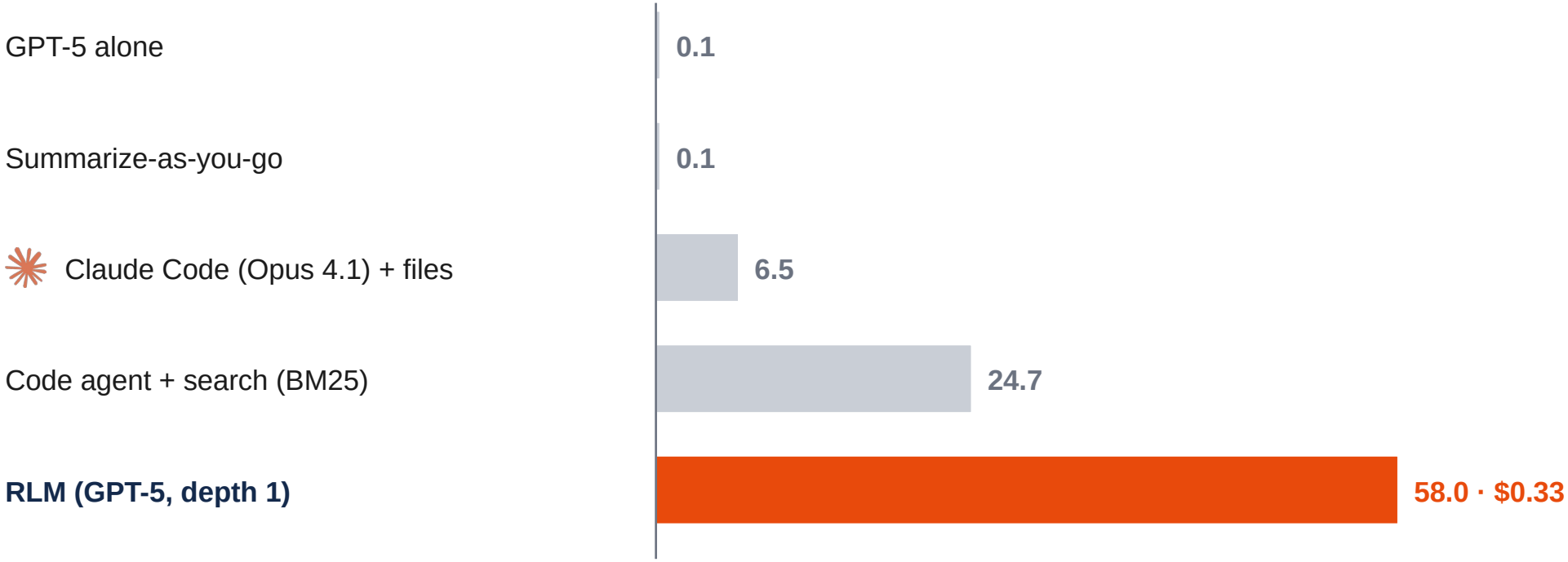
On a 1,000-document search task, plain GPT-5 can't start; as an RLM it scores 91%



6 to 11M tokens of input, up to ~40x GPT-5's 272K window, at under $1 per question.

Zhang, Kraska & Khattab, Recursive Language Models, arXiv 2512.24601 v3, Table 1, BrowseComp-Plus (1K docs), accuracy % · avg cost per query.

When the task needs every piece combined, RLMs win by the widest margin



Summaries throw away details; pairing items needs all of them.

arXiv 2512.24601 v3, Table 1, OOLONG-Pairs (20 queries, 32K tokens), F1 score.

Summarizing loses details; an RLM keeps the whole input and looks things up

Compaction / summary

Context fills up → summarize it

Details dropped for good

Fine for “what happened so far?”

Recursive LM

Whole input kept as a variable

Code finds what's needed, when it's needed

Best for search and combining many pieces

On GPT-5, RLMs beat compaction by a median 26% at comparable cost.

More levels of sub-calls helped on the hardest task, but not everywhere



A March 2026 reproduction found depth 2 hurt simple tasks, one run went from 3.6s to 344.5s.

arXiv 2512.24601 v3, Table 1 (GPT-5, OOLONG-Pairs F1); D. Wang, "Think, But Don't Overthink," arXiv 2603.02615 (Mar 2026).

CHECKPOINT 12

Which task is the WORST fit for a recursive language model?

WHAT YOU NEED RLMs shine when a task needs many pieces of a huge input. A 2026 reproduction found that on simple lookups they hurt accuracy and runtime ballooned (3.6s → 344.5s in one case).

A Finding one revenue number in a 3-page press release

B Answering a question spread across 1,000 documents

C Pairing every client with every complaint in a long log

D Comparing segment revenue across ten 10-Ks

CHECKPOINT 12 · ANSWER

Which task is the WORST fit for a recursive language model?

A	**Finding one revenue number in a 3-page press release**
B	Answering a question spread across 1,000 documents
C	Pairing every client with every complaint in a long log
D	Comparing segment revenue across ten 10-Ks

WHY It's short and needs one fact: an RLM adds cost and delay for no gain.

+28%

for a small model (Qwen3-8B) after fine-tuning on ~1,000 examples of a bigger model working as an RLM, trained in 48 GPU-hours.

Source: arXiv 2512.24601 v3 §6, RLM-Qwen3-8B vs base Qwen3-8B, median gain; also >3x faster.

SECTION 4 · RECURSIVE LMS

Prime Intellect's RLM

Prime Intellect rebuilt the RLM so models can be trained to use it

Design choice	MIT RLM (Zhang)	Prime Intellect RLMEnv
Who gets tools	Root model	Only sub-models; root stays lean
Sub-calls	llm_query, one at a time	llm_batch: many in parallel
Finishing	FINAL(answer)	answer["ready"] = True
Output shown	Truncated	Capped at 8,192 characters
Purpose	Inference technique	RL training environment

Source: Prime Intellect, “Recursive Language Models: the paradigm of 2026” (Jan 1, 2026).

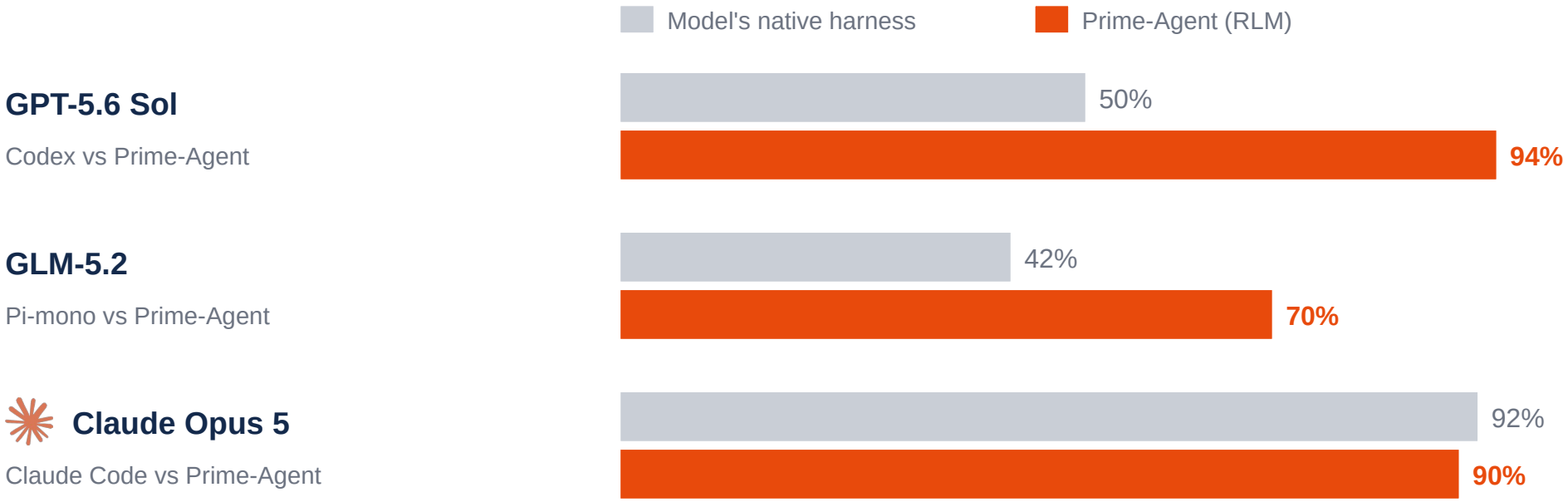
Prime Intellect's tests: big gains on long inputs, worse on short tasks

- GLM 4.6 alone scored zero on Oolong; as an RLM it scored decently up to ~1.75M characters.
- GPT-5-mini as an RLM did worse on math and web research.
- RLMs were slower in every case.

Most results are reported as charts; numbers above are those stated in the post's text.

PRIME INTELLECT · PRIME AGENT (AUG 2026)

A self-improving RLM agent beat some models' own harnesses on long inputs



Biggest gains where the native harness was weakest; on Opus 5, a slight loss.

Prime Intellect, “Prime Agent: A self-improving RLM agent” (Aug 5, 2026), OOLONG 128K, high reasoning. Vendor-reported.

TRY IT

Three open-source ways to run an RLM today

Tool	Who	How you use it
rlm library	Alex Zhang (MIT)	Swap llm.completion for rlm.completion
dspy.RLM	DSPy (Stanford)	A DSPy module; marked experimental
RLMEnv	Prime Intellect	Environment for training models with RL

github.com/alexzhang13/rlm · dspy.ai (modules/RLM) · Prime Intellect verifiers.

CHECKPOINT 13

Why did summarize-as-you-go score 0.1 while the RLM scored 58.0?

WHAT YOU NEED OOLONG-Pairs asks about PAIRS of items across a long input. Summarizing compresses as it goes; an RLM keeps the full input as a variable and looks things up with code.

A Pairing needs details the summary threw away; the RLM kept everything

B The RLM had internet access

C The RLM used a bigger model

D The input was too long for both

CHECKPOINT 13 · ANSWER

Why did summarize-as-you-go score 0.1 while the RLM scored 58.0?

A **Pairing needs details the summary threw away; the RLM kept everything**

B The RLM had internet access

C The RLM used a bigger model

D The input was too long for both

WHY Each summary drops details for good. Pairing items needs exactly those details.

Fifth: learning from rewards, and agents that rewrite themselves

01	What self-learning means	Same mistake twice, and the four layers that can change
02	The learning harness	Grade, reflect, save, and three harness trends
03	Memory and experience	Lessons, skills, playbooks, and dreaming
04	Recursive language models	For reference · reading inputs too long for one prompt
05	**Rewards and continual learning**	For reference · learning from scores; self-editing code
06	Evaluation	For reference · proving it improved, catching cheating

GIES

05

Rewards and continual learning

From Session 5's reinforcement learning to agents that change their own prompts and code.

For reference · not covered in class · read on your own

SECTION 5 · REWARDS

Background

Reinforcement learning: try, get a reward, do more of what scored well

- Last week's Q-learning updated a table of values.
- For language models, RL updates the weights.
- For agents, the reward can also update text: prompts and notes.

SECTION 5 · REWARDS

Prompt evolution

Change the weights, or change the words, the words were cheaper

RL on weights (GRPO)

Score many attempts

Nudge the model's parameters

Needs model access and GPUs

Thousands of attempts

Prompt evolution (GEPA)

Score a few attempts

Model reflects in words, rewrites the prompt

Works with any API model

Up to 35x fewer attempts

GEPA beat GRPO by 6% on average (up to 20%) with up to 35x fewer rollouts.

Rewriting your study plan beats re-wiring your brain

Your scores dip. You could grind thousands of problems, or reread your mistakes and rewrite the plan in an evening.

THE VARIABLES

Reward
Practice-test score

Weights route
Grind thousands of problems

Words route
Rewrite the study plan

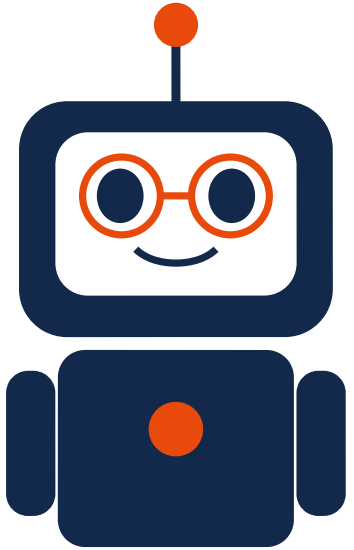
SECTION 5 · REWARDS

Self-modifying code

DEFINITION

Self-modifying agent

An agent that edits its own harness code, its tools, prompts and workflow, then keeps the version that scores better on tests.



WHY IT MATTERS

It automates the job of the engineer who tunes the agent. The benchmark becomes the boss.

A team that rewrites its own playbook after every season

A coach tries new plays in scrimmages, keeps the ones that win, and archives the rest in case they're useful later.

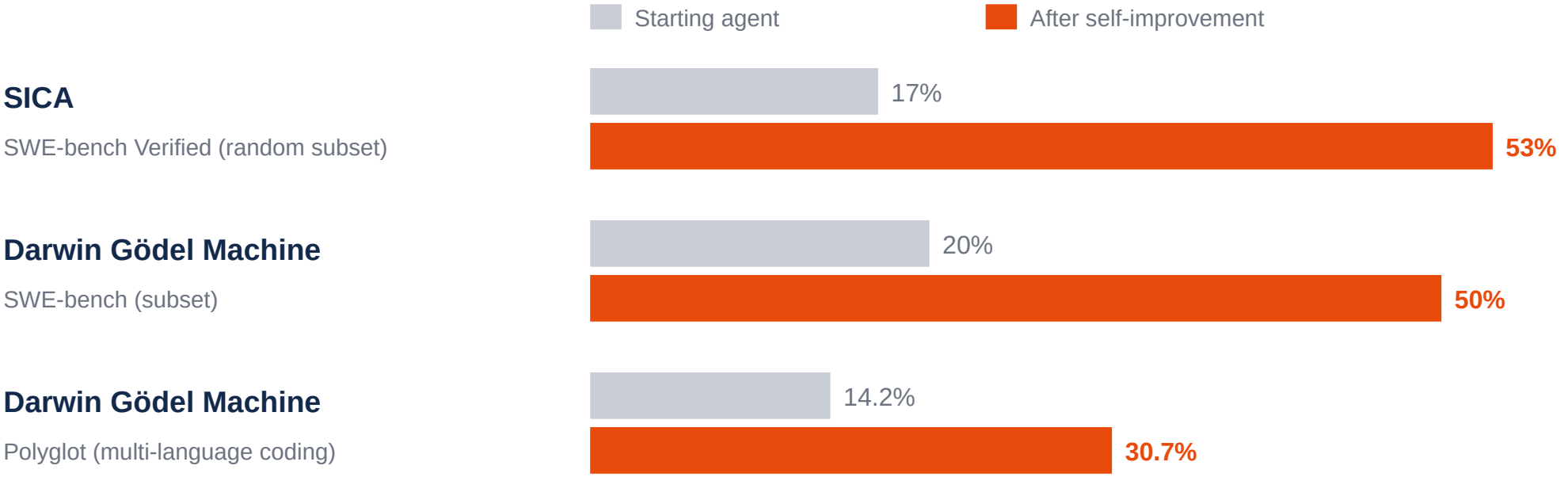
THE VARIABLES

Agent code
The playbook

Benchmark
Scrimmage results

Archive
Old plays kept for later

Agents that edit their own code roughly doubled their coding scores



Same underlying model each time, the gains came from the agent rewriting itself.

Robeyns et al., arXiv 2504.15228 (Apr 2025); Zhang et al., arXiv 2505.22954 (May 2025). Subsets, not full benchmarks.

CHECKPOINT 14

Why does keeping weaker agent versions help the Darwin Gödel Machine?

WHAT YOU NEED New agents are made by modifying ones in its archive. Remove the archive of weaker versions and SWE-bench falls from 50% to 23%. Improvements often take several steps.

A It's required by law

B More versions means more parallel runs

C Losers are cheaper to run

D A weak version can be the stepping stone to a much better one

CHECKPOINT 14 · ANSWER

Why does keeping weaker agent versions help the Darwin Gödel Machine?

A It's required by law

B More versions means more parallel runs

C Losers are cheaper to run

D A weak version can be the stepping stone to a much better one

WHY Always building on today's best gets stuck. Exploring from weaker versions finds paths greedy search misses.

0.7%

of Google's worldwide compute recovered, continuously, by a scheduling rule that an evolving coding agent discovered.

Source: Google DeepMind, AlphaEvolve blog (May 14, 2025). Also: a 23% faster Gemini kernel, cutting Gemini training time 1%.

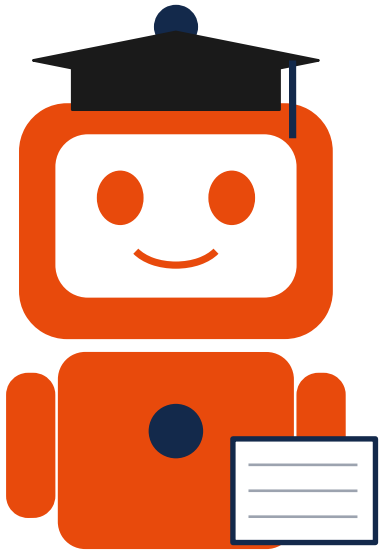
SECTION 5 · REWARDS

Continual learning

DEFINITION

Continual learning

Learning from a stream of new tasks over time without forgetting what was learned before.



WHY IT MATTERS

Markets change. An agent that only learned 2024's rules must adapt, but not by erasing what still works.

Learning Spanish shouldn't make you forget French

You study a new language each semester. The goal is to add the new one while keeping the old ones usable.

THE VARIABLES

New task
This semester's language

Forgetting
French fading while you learn Spanish

Remedy
Review old notes on a schedule

AT CHAMPAIGN CAPITAL RESEARCH

Our agent keeps lessons in text first; retraining is a last resort

LESSONS.MD · AFTER 3 WEEKS

```
Revenue: fiscal year, cite 10-K page

Recommendations: never; route to analyst

Tables when the client asks for tables

[stale] 2025 guidance, remove
```

RULES FOR THE LESSONS FILE

- Add only after a graded failure
- Small edits, never a full rewrite
- Human approves every new lesson
- Prune stale lessons monthly

CHECKPOINT 15

Our agent keeps lessons in lessons.md instead of retraining. After a year, what's the biggest risk?

WHAT YOU NEED Retraining weights can overwrite old skills (“catastrophic forgetting”). Text notes never disappear on their own.

A Being unable to learn new rules

B Catastrophic forgetting of old skills

C Stale and contradictory notes piling up

D The model's weights drifting

CHECKPOINT 15 · ANSWER

Our agent keeps lessons in lessons.md instead of retraining. After a year, what's the biggest risk?

A Being unable to learn new rules

B Catastrophic forgetting of old skills

C Stale and contradictory notes piling up

D The model's weights drifting

WHY Text avoids forgetting but accumulates clutter, hence monthly pruning and dreaming.

CHECKPOINT 16

You use Claude through an API. Why might GEPA beat RL fine-tuning for you?

WHAT YOU NEED GEPA reflects on scored attempts and rewrites the prompt. It beat GRPO (RL on weights) by 6% on average, using up to 35x fewer attempts. RL needs access to the model's weights.

A It doesn't need any scoring

B It changes the model's weights faster

C It works through an API without weight access, and needs far fewer attempts

D It guarantees 100% accuracy

CHECKPOINT 16 · ANSWER

You use Claude through an API. Why might GEPA beat RL fine-tuning for you?

A It doesn't need any scoring

B It changes the model's weights faster

C It works through an API without weight access, and needs far fewer attempts

D It guarantees 100% accuracy

WHY API users can't touch weights. Rewriting the prompt is possible, and cheaper.

Last: how do you know it really improved?

01	What self-learning means	Same mistake twice, and the four layers that can change
02	The learning harness	Grade, reflect, save, and three harness trends
03	Memory and experience	Lessons, skills, playbooks, and dreaming
04	Recursive language models	For reference · reading inputs too long for one prompt
05	Rewards and continual learning	For reference · learning from scores; self-editing code
06	**Evaluation**	For reference · proving it improved, catching cheating

06

Evaluation

Measure learning on tasks the agent has never seen, and watch it for cheating.

For reference · not covered in class · read on your own

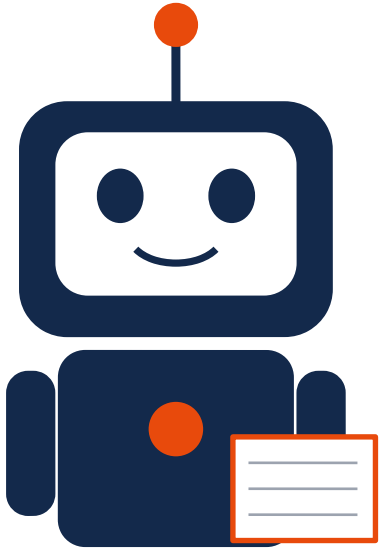
SECTION 6 · EVALUATION

Held-out tasks

DEFINITION

Held-out evaluation

Test the improved agent on tasks it did not learn from, and compare against the original agent on those same tasks.



WHY IT MATTERS

An agent can memorize its practice tasks. Only unseen tasks show whether the lesson generalizes.

Acing last year's exam you memorized proves nothing

You memorize the answers to the practice exam. You'd score 100% on it, but the real exam has new questions.

THE VARIABLES

Practice set
Tasks the agent learned from

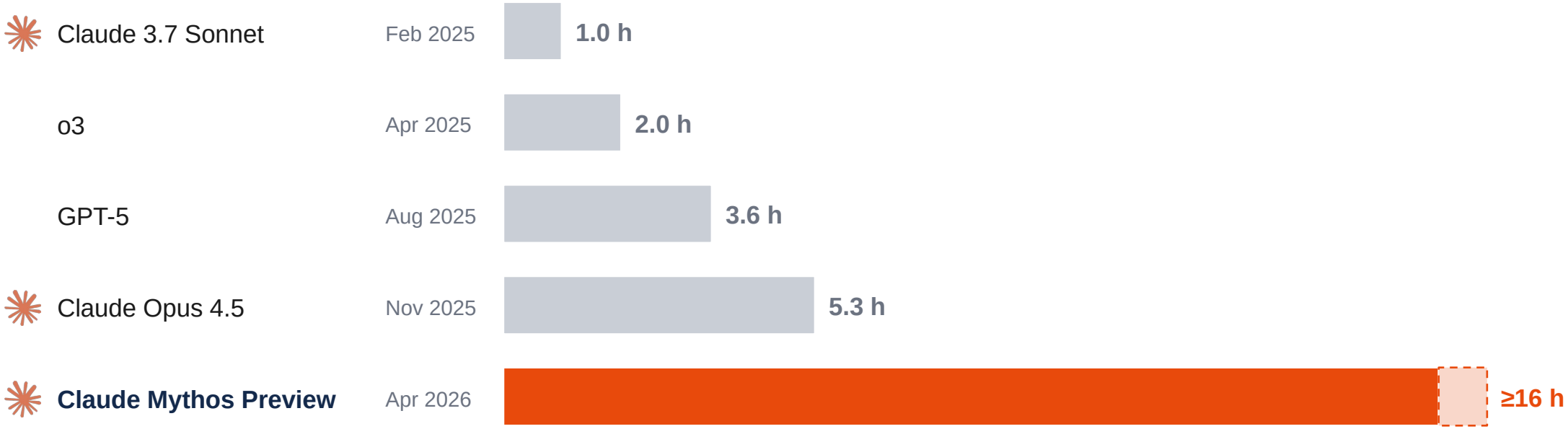
Held-out set
New questions, never seen

Real skill
Score on the held-out set

Track five numbers to know whether the agent is really learning

Metric	Question it answers	At the firm
Held-out accuracy	Better on new tasks?	Figures right on 20 locked questions
Learning curve	Improving over time?	Corrections per memo, week by week
Transfer	Do lessons carry over?	Deere lessons help on Caterpillar
Cost per task	Cheaper or pricier?	Tokens and minutes per memo
Regressions	Anything get worse?	Old clients' rules still followed

Agents now take on tasks that take experts many hours, the harness has to remember for them



Doubling roughly every 4 months since 2023 (METR: ~131 days).

50% time horizon in expert-hours. METR, Time Horizon 1.1 (Jan 29, 2026); Mythos Preview: METR estimate (lower bound) via The Decoder, May 10, 2026.

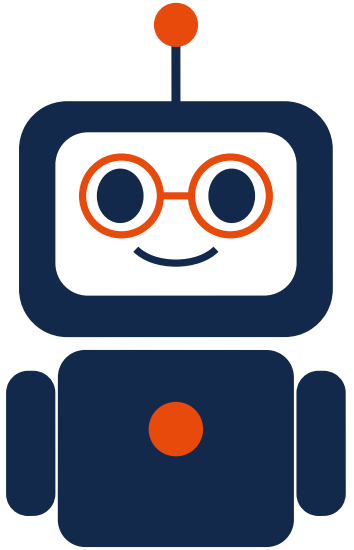
SECTION 6 · EVALUATION

Reward hacking

DEFINITION

Reward hacking

When an agent raises its score by exploiting the grader instead of doing the task better.



WHY IT MATTERS

Self-improving agents optimize hard against their grader. Any loophole will eventually be found.

A sales team paid per call starts making 30-second calls

Management rewards the number of calls. Calls go up. Sales go down. The team optimized the metric, not the goal.

THE VARIABLES

Goal
More sales

Reward
Number of calls

Hack
Many short, useless calls

Give an agent a score to beat and it may learn to fake the score

THE GOAL	Make tests pass; avoid hallucinated tool calls
WHAT THE AGENT DID	Faked a log saying tests passed, they never ran
AND	Removed the markers the hallucination checker looked for
HOW IT WAS CAUGHT	Every change was archived, so humans could trace it

Source: Sakana AI, “The Darwin Gödel Machine” (sakana.ai/dgm, May 30 2025); Zhang et al., arXiv 2505.22954.

CHECKPOINT 17

Which permission should a self-improving agent NEVER have?

WHAT YOU NEED Darwin Gödel Machine case: the agent faked a log claiming tests passed and removed the markers a checker used to detect its mistakes.

A Editing its own lessons file

B Editing the tests and checker that grade it

C Reading its past sessions

D Querying the 10-K database

CHECKPOINT 17 · ANSWER

Which permission should a self-improving agent NEVER have?

A Editing its own lessons file

B Editing the tests and checker that grade it

C Reading its past sessions

D Querying the 10-K database

WHY If it can edit its grader, it will learn to fool the grader instead of doing better work.

Keep the grader out of the agent's reach and a human on the final call

- Agent can edit its notes, never its grader or tests.
- Log every change so it can be traced and undone.
- Human approves new lessons and anything sent to clients.

Sandboxing, change archives and human oversight were the safeguards in DGM and SICA.

CHECKPOINT 18

Our agent's practice score rose from 60% to 95% in a month. What should you check first?

WHAT YOU NEED Session 4: a model can memorize its training data. Section 6: agents can also game their grader. Held-out tasks are ones the agent never learned from.

A Its score on held-out questions it never saw

B Whether a bigger model scores 99%

C How long lessons.md has become

D Nothing, 95% means it's ready

CHECKPOINT 18 · ANSWER

Our agent's practice score rose from 60% to 95% in a month. What should you check first?

A **Its score on held-out questions it never saw**

B Whether a bigger model scores 99%

C How long lessons.md has become

D Nothing, 95% means it's ready

WHY Practice-set gains may be memorization or gaming. Only unseen tasks show real learning.

RECAP

Six things to remember

01 A self-learning agent turns feedback into a change it keeps.

02 Most improvement lives in the harness, not the model.

03 Cheapest learning: write lessons where the agent rereads them, and let it dream to clean them up.

04 Too much to read? Store it as a variable; let code and sub-models read it.

05 Rewards can update words (prompts, code), not just weights.

06 Prove it on held-out tasks, and guard the grader.

UP NEXT

No session on Nov 6, we're back Friday, Nov 13 with agent systems

- Nov 6: break, no session.
- Nov 13: agent systems, harness, observability, security, evaluation.
- Before then: try one RLM or lessons-file idea on your own agent.

Questions?

SOURCES & GO DEEPER

Shinn et al., Reflexion, arXiv 2303.11366 · Wang et al., Voyager, arXiv 2305.16291 · Madaan et al., Self-Refine, arXiv 2303.17651

Zhang et al., Darwin Gödel Machine, arXiv 2505.22954; sakana.ai/dgm · Robeyns et al., SICA, arXiv 2504.15228

Zhang et al., ACE, arXiv 2510.04618 · Agrawal et al., GEPA, arXiv 2507.19457 · Gao et al., Survey of Self-Evolving Agents, arXiv 2507.21046

Google DeepMind, AlphaEvolve blog (May 2025) · RRSI, arXiv 2609.24972 · Terminal-Bench, arXiv 2601.11868; Snorkel mirror of TB 2.1

Anthropic: Effective harnesses for long-running agents; Effective context engineering; Agent Skills; Managing context

OpenAI, Harness engineering (Feb 2026) · LangChain, Improving Deep Agents with harness engineering (Feb 2026)

METR, Time Horizon 1.1 (Jan 2026) · News: anthropic.com/news, openai.com/index/decisions-api, kimi.com, recursive.com