

EVALS & TESTING · BREAKOUT

Why Your Agent Is Failing

Failure modes from 6,000+ agent trajectories
across 45 benchmarks

Oct 23 · 4:40–5:05 PM · Room 210A

Han Xu

AGENDA

Three findings and a framework

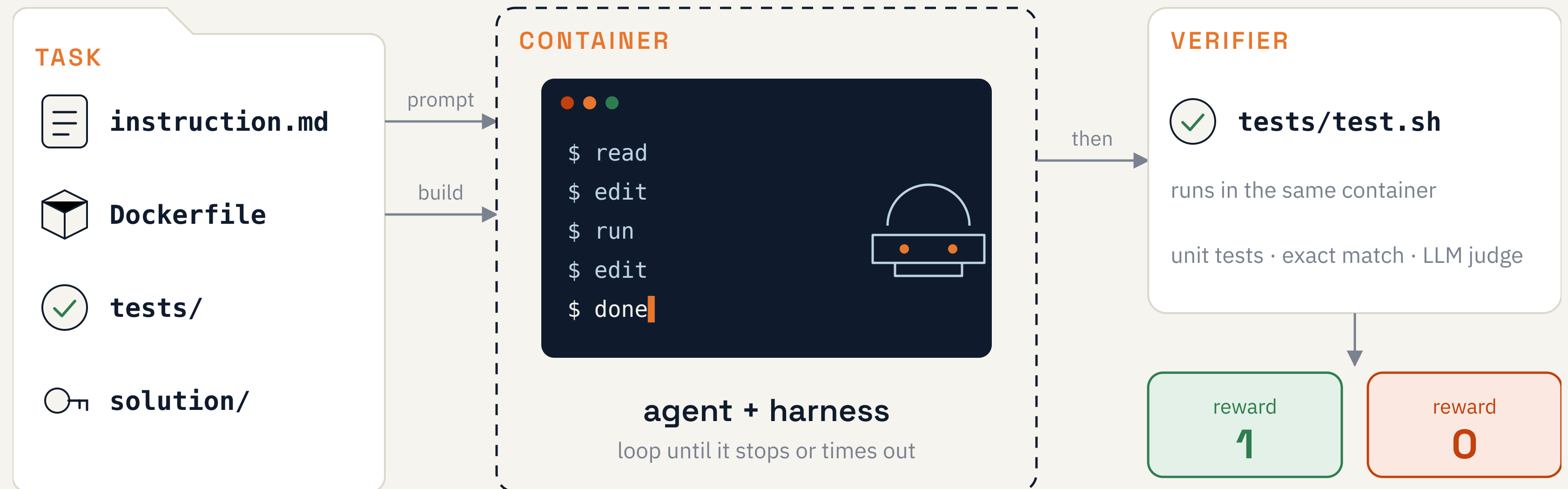
1 Harnesses shape failure modes

2 Complexity does not always win

3 Many hard tasks are broken

4 A practical framework

How Harbor runs one task



THE QUESTION

Agents fail on long-horizon tasks. But why?

When a trajectory ends with reward 0, the blame could sit in any of four places.



The model

Does it lack the knowledge or reasoning to produce a correct solution?



Its tool use

Did it call the wrong tool, write to the wrong place, or never ship a deliverable?



The harness

Does the scaffold let it verify and recover, or push it into early stops?



The task itself

Is the environment or grader broken so that even a correct answer fails?

THE SETUP

One infrastructure, 54 benchmarks, every model on two harnesses

54

benchmarks, 6,627 tasks, across software engineering, reasoning, science, agents, data and multimodal

8

models from Google, OpenAI and Anthropic, spanning capability tiers

16

model-harness configs: each model on Terminus-2 plus its native harness (Codex, Claude Code, Gemini CLI)

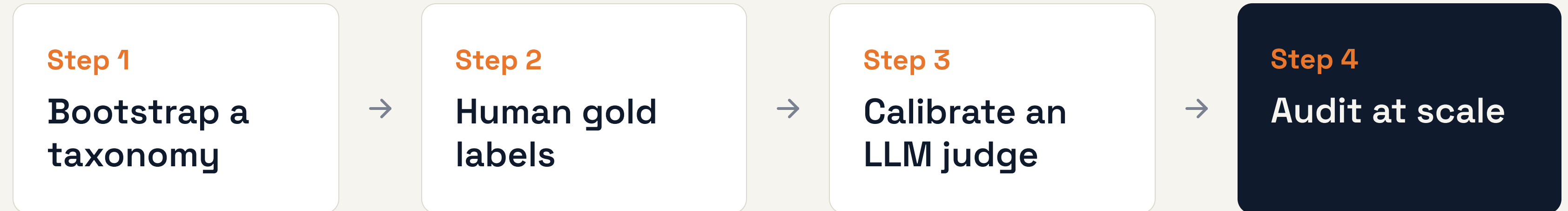
0.3M

trajectories, 3 trials per (benchmark, model, harness) setting

Harbor Adapters = one standard format for every benchmark (instruction, environment, tests, solution)

AUDIT FAILURES

Humans build the rubric, a calibrated judge scales it



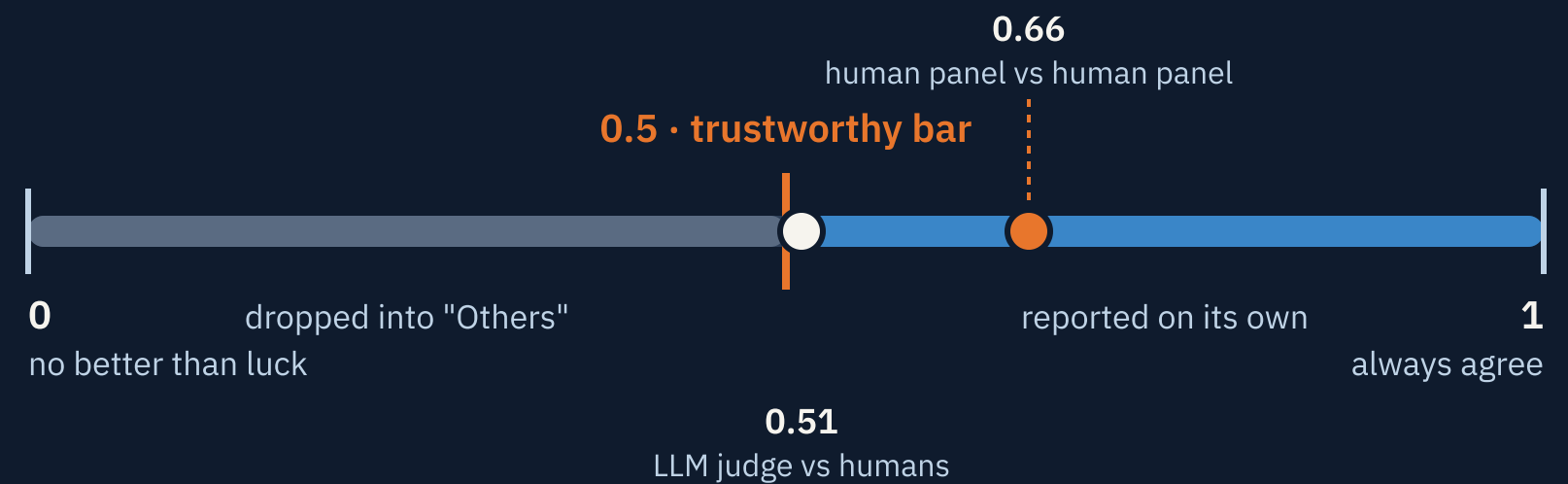
READING THE NUMBERS

What Cohen's kappa (κ) means: agreement beyond luck

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

p_o = how often the two raters actually agreed.

p_e = how often they would agree by luck, given how often each says "match".



Four failure modes where the model did not know enough

Which year?

→

1987 X

expected 1989

right slot · literal answer · value simply wrong
numbers, names, dates, MCQ letters, yes/no

Wrong factual answer

κ 0.78

```
def total(xs):  
    s = 0  
    for i in range(n):  
        s += xs[i+1]  
    return s
```

test says
expected 5
got 4 X

Algorithmic bug

κ 0.66

visible tests (agent ran them)

✓

✓

✓

"3 passed"

hidden tests (verifier)

✓

X

X

✓

X

reward

Hidden-test regression

κ 0.76

patch →

global stylesheet

no effect

← grader

component · inline styles

data

Wrong target / layer

κ 1.00

Four failure modes where the work never reached the grader

agent
"Done ✓"

→

answer.txt
(empty)

verifier: No answer.txt found

Silent deliverable

κ 0.66

agent

→

403 · pip failed · ImportError · no OCR tool

Environment block

κ 0.65

```
OPENQASM 2.0;
qreg q[2];
rxx(0.5) q[0], q[1];
measure q -> c;
```

STOP

unsupported gate, line 3

never runs

Syntax / language error

κ 0.85

verifier expects

```
{"count": 4}
```

dict · int · named key

≠

agent wrote

```
["4.0"]
```

list · string · no key

same content, wrong shape

Wrong output schema

κ 0.58

Four failure modes in how the agent ran and judged itself



no answer submitted

step budget

270 / 270 steps · timeout error

last message:

"let me run the tests, then..."

Agent timeout

κ 0.82

skipped the check

write code

pytest available

"Done"

verification was available and never run
used only when no more specific rubric fits

Unverified claim

κ 0.81

own inline check
assert ok ✓

"All tests pass"

verifier
reward 0 ✗

a smoke test it wrote,
not the shipped suite

False success claim

κ 0.74

"Analysis done. I'll now
write parser.py and run it."

parser.py · no tool call
never written

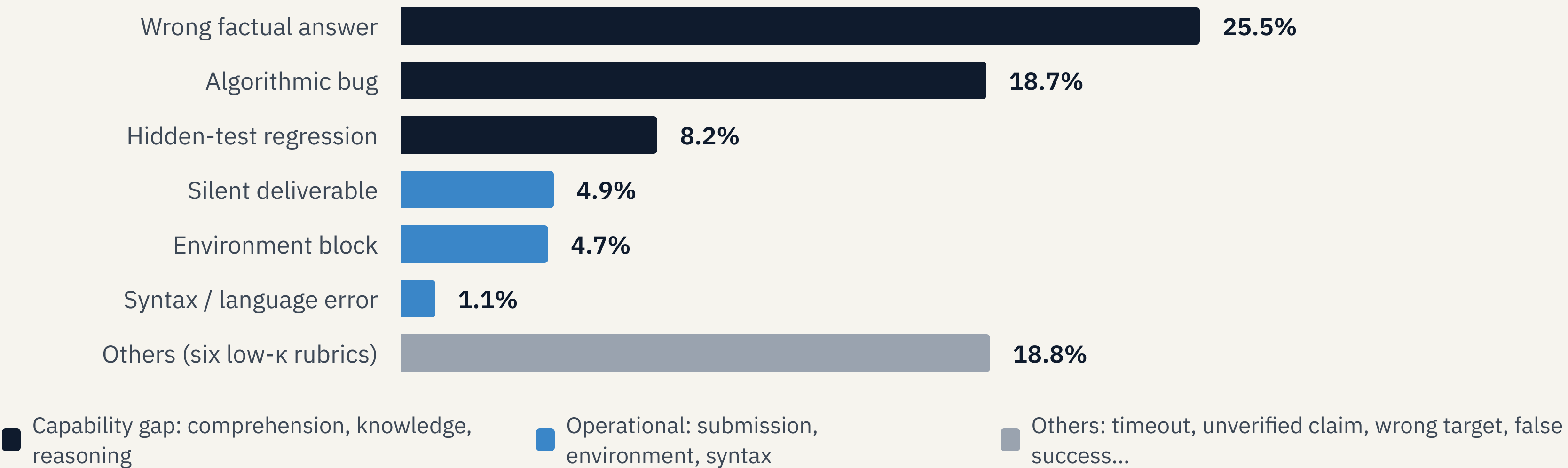
end of trajectory

Plan over implementation

κ 0.80

FINDING 1 · WHERE AGENTS FAIL

Wrong answers and real bugs dominate, on every model and harness



FINDING 1 · HARNESS SHAPES FAILURE

Same model, different harness, different operational failures

Model	Harness	Wrong answer	Algo bug	Env block	Silent deliv.
GPT-5.4	Codex	27.3	20.2	3.5	1.4
GPT-5.4	Terminus-2	26.6	18.8	9.3	4.9
Opus 4.6	Claude Code	26.4	22.5	2.9	4.1
Opus 4.6	Terminus-2	23.2	17.7	4.2	5.0
Gemini 3.1 Pro	Gemini CLI	25.0	17.7	3.9	6.9
Gemini 3.1 Pro	Terminus-2	24.5	15.3	4.1	7.3

Capability rubrics barely move

Wrong answers and algorithmic bugs are conserved within each model family, whichever harness runs it.

Operational rubrics follow the harness

GPT-5.4 hits environment blocks 2.7× more often on Terminus-2 than on Codex. Silent deliverables peak on Gemini CLI. Submission rules differ by harness.

Verify-and-correct loops matter

Native harnesses allow iteration and self-correction. Terminus-2 enforces a linear plan, execute, complete sequence and breaks when refinement is needed.

FINDING 1 • TIMEOUTS AND THE HARNESS

On hard tasks, more than a third of failures are timeouts with no answer

453

timeouts with no answer, 36% of failed rollouts

361

close but not passing solutions

445

fundamentally wrong answers

Weaker models time out twice as often

The native harness cuts timeouts from 42% to 26%

FINDING 2 • MODEL OR HARNESS?

Capability is the biggest lever. The harness is still a lever.

Model fixed-effect range

0.451

6 of 8 model coefficients significant at $p < 0.05$

Harness fixed-effect range

0.087

2 of 4 harness coefficients significant at $p < 0.05$

Model effect vs harness effect

5.2×

after controlling for benchmark difficulty, which alone explains 75% of score variance

FINDING 2 · WHERE CAPABILITY PAYS OFF

Where frontier models win: the middle

Task difficulty	Tasks	Other models	Frontier	Gain
0.0–0.1	1,461	94%	99%	+5 pp
0.1–0.2	862	80%	95%	+15 pp
0.2–0.3	660	67%	90%	+23 pp
0.3–0.4	495	54%	83%	+29 pp
0.4–0.5	429	44%	71%	+27 pp
0.5–0.6	392	35%	62%	+27 pp
0.6–0.7	448	26%	51%	+25 pp
0.7–0.8	460	17%	39%	+22 pp
0.8–0.9	454	9%	25%	+16 pp
0.9–1.0	966	1%	4%	+3 pp

Easy tasks: everyone passes

A five-point gap on the easiest decile. Not where to spend your budget.

Mid difficulty: up to 29 points

The frontier advantage peaks between 0.3 and 0.6, where the task is reachable but not trivial.

Hardest decile: nobody passes

1% versus 4%. Many of these tasks are broken.

Frontier: Opus 4.6, Gemini 3.1 Pro, GPT-5.4. Other: five smaller models. Difficulty = 1 – mean pass rate. · 15

FINDING 2 • WHEN AGENTS SPIN

The longest failures are on almost-solvable tasks, not the hardest ones

- 1 Success latency rises with difficulty
- 2 Failure latency peaks at difficulty 0.7–0.8
- 3 The weakest model persists longest

ONE TRAJECTORY

270 steps

36 hours

One GAIA2 run. The task was broken; no agent could have passed.

FINDING 3 • HARD OR BROKEN?

About one third of the hardest tasks are broken, not hard

6,627	tasks in the full Harbor pool, 54 adapters
1,311	pass the difficulty filter: three frontier models succeed on at most 33% of 18 trials
307	survive an AI audit for instruction–verifier alignment and essential difficulty
110+	survive review by 14 domain-experienced humans under the same rubric
100	chosen for difficulty, diversity and insight
82	released in Harbor-Index after a trajectory-grounded audit-and-fix pass

Across more than 30 benchmarks, roughly **one third** of the hardest candidate tasks were rejected as broken rather than genuinely difficult.

Five ways a correct solution scores zero

Dead environment • GAIA2

Missing input • MMAU

Verifier overreach • SWE-bench Pro

Pinned reference • CRUST-Bench

Wrong gold answer • AA-LCR

Six ways grader and prompt disagree

Undisclosed format · SWE-bench Pro

Missing test file · Teleport

Wrong reference · HLE

Unstated unit rule · AA-LCR

No directive · SWE-bench Multilingual

Guess the import path · FeatureBench

Most benchmarks measure the same thing

13 of 54

benchmarks are saturated, with state-of-the-art above 90%, including once-hard GPQA-Diamond and LiveCodeBench

12

benchmarks are enough: after greedy selection, every remaining benchmark's model ranking correlates at $\rho \geq 0.7$ with a selected one

3 tasks

per benchmark recover the full system ranking at mean $\rho \approx 0.92$ across 52 benchmarks

What 14 rounds of judge tuning teach

Lesson 1

Edit one rubric at a time

Lesson 2

One shortlist beats
twelve yes/no judges

Lesson 3

Short rubric text beats
priority rules

Diagnose in this order: task, harness, model

1 • Is the task broken?

2 • Is it the harness?

3 • Is it the model?

82 audited tasks that no configuration passes above 30%

Model · native harness	Native pass %	\$ per run	Terminus-2 pass %	\$ per run
GPT-5.5 · Codex	28.0	178	19.5	155
Claude Opus 4.8 · Claude Code	20.7	269	15.9	293
Gemini 3.1 Pro · Gemini CLI	13.4	74	11.0	89
GLM 5.2 · Claude Code	8.5	205	9.8	52
Kimi K2.6 · Claude Code	6.1	191	8.5	33
MiniMax M3 · Claude Code	3.7	66	6.1	18
Qwen3.7 Max · Claude Code	4.9	201	4.9	36
DeepSeek V4 Pro · Claude Code	4.9	177	3.7	35
MiMo V2.5 Pro · Claude Code	2.4	49	2.4	4

Top models gain most from their native harness

GPT-5.5 picks up 8.5 points moving from Terminus-2 to Codex.

Open-weight models do better on the general harness

GLM, Kimi and MiniMax all score higher on Terminus-2, at a fraction of the cost. Complexity does not always win.

Cheap enough to run repeatedly

A full 82-task run costs between \$4 and \$293 depending on configuration.

THREE THINGS TO REMEMBER

Why your agent is failing

1 Harnesses shape failure modes

Capability failures are conserved across harnesses. Operational failures are not. A verify-and-correct loop is the difference between recovery and a silent stop.

2 Capability wins, but co-design matters

The model effect is about five times the harness effect, yet a general-purpose harness can beat a model-specific one on a given model.

3 Many hard tasks are broken

About a third of the hardest candidates failed audit.

Thank you

REFERENCES

Code and benchmark adapters: <https://github.com/harbor-framework/harbor>