SAMPLE — CHAPTER 2

Agentic Engineering

Building agents that ship — the harness, the evals, the memory, the ops

Datt Goswami

agenticfrontier.dev · v2026.07 · sample

Contents

The full book: 19 chapters in five parts, plus appendices. This sample carries Chapter 2 in full.

CHAPTER 2 · PART I · THE PROBLEM · SAMPLE

The Agent Evaluation Crisis

Agent benchmark numbers measure demos, not the capability teams think they bought — and the fix is a measurement regime borrowed from deep RL, not a better leaderboard.

§1 — The replay script that beat the frontier

Here is the most important agent-evaluation result of the year, and it is not a capability result. In early 2026, a team at Meta Superintelligence Labs reported that **a 1MB replay script that blindly executes a recorded action sequence outperforms frontier models on prominent CUA benchmarks** (D'Oro et al., 2026). There is no model inside the script. It does not read the screen. It stores one successful sequence of clicks, taps, and keystrokes per task and plays them back in order, like a pianola scrolling through a punched roll. On OSWorld and MobileWorld — two widely cited computer-use benchmarks — this pianola beats the very frontier agent whose actions it recorded: 71.1% against 70.6% on OSWorld, 41.5% against 32.7% on MobileWorld (see Figure 2.1).

To see why, hold one definition in mind. An agent episode is a **trajectory** τ: a sequence of steps, each one a state **s** of the world, an observation **o** the agent receives, and an action **a** it takes in response. A computer-use benchmark is supposed to measure whether a policy maps *o* → *a* well across many different states *s*. The replay script maps nothing. It emits a fixed action list regardless of what it observes. The only way it can win is if the benchmark always starts each task from the same state *s* and never varies it — so that last week's winning action list still lands on this week's buttons.

D'Oro et al. establish this formally. On a deterministic benchmark with fixed initial states, a replay policy that stores one successful trajectory per task has an expected success rate *exactly equal* to the source agent's **pass@k** — the probability that at least one of *k* independent attempts succeeds (their Remark 1). Replaying a recorded run is not a trick that happens to work; it is the physical embodiment of pass@k. And the corollary should stop every benchmark author cold: for any agent with nonzero success probability on every task, enough rollouts plus memorization drive the replay score to 100%. Every static benchmark is brute-force solvable. The quantity it actually reports is not "can the agent reason about what it sees" but "has someone already recorded the answer."

The fix the same paper supplies is the tell. When the authors rebuilt the evaluation as **DigiWorld** — 15 sandboxed mobile apps that independently vary the data, the visual theme, and the initial interface state across more than 3.2 million verified configurations — the blind script fell from beating the frontier to **6.90%**, while the actual model scored 45.7%. Nothing about the agent changed between those two numbers. The benchmark changed. A static benchmark measures the state coverage of one memorized trajectory; an open one measures the policy. The crisis is that we have spent two years quoting the first number and believing the second.

That is the thesis of this article, and of the eleven that follow it. As currently produced, agent benchmark numbers measure demos, not the property teams believe they bought. The cure is not a better leaderboard. It is a different *measurement regime* — the statistical rigor that deep reinforcement learning learned the hard way, fused with open-world evaluation design. This is Chapter 1 of Agentic Engineering because every capability claim in the later chapters has to be read through the lens it builds here.

> [← A10] If this feels familiar, it should. The published Continual Intelligence series opened with the same discovery one field over: five years of continual-RL benchmarks all measured the same wrong thing — episodic task-switching instead of genuine non-stationary adaptation. The agent era is replaying that mistake at a new scale, and it has the same cure.

KEY TAKEAWAY 1

A score on a static, deterministic benchmark is confounded with how *memorizable* the benchmark is. Before trusting any agent number, ask what a blind replay of a single recorded run would score on the same tasks. If that number is high, you are measuring the benchmark, not the agent.

§2 — A taxonomy of agent-eval failure

The replay result is one instance of a general pattern, and the pattern has exactly five shapes (see Figure 2.2). Once you can name them, you can audit any agent benchmark — your own included — in an afternoon.

Contamination. The score reflects memorized solutions rather than reasoning. Frontier models already perform strongly on standard physics evaluations, which makes it nearly impossible to tell genuine reasoning from recall of material that was in pretraining; DiscoverPhysics responds by asking models to *discover* physical principles interactively rather than restate them, precisely to separate

the two (Wiemann et al., 2026). The detection question is blunt: perturb the surface of a task — rename the variables, change the cover story — and watch whether the score moves. If it craters, the model was matching, not reasoning.

Determinism exploits. Covered in §1: a fixed initial state lets a blind replay win. The general form is any benchmark whose state can be memorized. The detection check is the one the Statistical Precipice paper hands you for free — run a blind replay agent; if it scores well, your environment is memorizable (D'Oro et al., 2026).

Single-seed reporting. One run, one number, no interval. Rankings computed this way flip under replication. The reliability literature is now explicit that a single accuracy number hides whether an agent behaves consistently at all: *Towards a Science of AI Agent Reliability* (2026) decomposes agent behavior into twelve metrics across four dimensions — consistency, robustness, predictability, and safety — and finds that recent capability gains bought only small reliability gains. A model that is five points more accurate and twice as variable is not obviously better. §3 handles the statistics.

Success-criterion gaming. What counts as "pass" is often a surface match — an action sequence, an LLM-judge verdict — rather than a verified end state. WebArena's design lesson was to check *functional correctness*: did the world actually reach the goal state, not did the agent emit a plausible-looking sequence (Zhou et al., 2023). And criteria are never neutral. *Agents' Last Exam* (2026) argues that once a domain acquires a widely used, verifiable evaluation, research and engineering effort rushes toward it — which is exactly why a gameable criterion is dangerous: you get the number you wrote down, and the whole field chases it.

Cost blindness. pass@1 with no dollars, tokens, or latency attached is half a measurement. An agent that is a point or two more accurate but an order of magnitude more expensive to run is not the better agent. §4 makes the case that

cost is a first-class metric; for now, note only that any benchmark number quoted without its cost is incomparable to the next one.

Figure 2.2: The five recurring failure modes of agent evaluation. Every suspect benchmark number is some combination of these; a benchmark audited against all five is rare. Grounded in D'Oro et al. (2026), Wiemann et al. (2026), Zhou et al. (2023), Yao et al. (2024), and *Towards a Science of AI Agent Reliability* (2026).

These are not hypothetical failure modes; the canonical agent benchmarks were built, in part, to escape them, and their headline gaps tell you how far there is left to go. GAIA poses 466 questions that are simple for humans and hard for assistants — human respondents answer 92%, while an augmented GPT-4 answered 15% (Mialon et al., 2023). WebArena's realistic web tasks: 14.41% for GPT-4 against 78.24% for humans (Zhou et al., 2023). τ-bench made reliability legible by introducing **pass^k** — the chance that *all* k independent attempts succeed — and showed that leading function-calling agents are inconsistent, with retail-domain pass^8 under 25% (Yao et al., 2024). AgentBench spanned eight environments and 29 models to make the capability gap measurable at all (Liu et al., 2023). The newer canon pushes on long horizons: Terminal-Bench on hard command-line tasks (Merrill et al., 2026), ALE-Bench on objective-driven algorithm-engineering contests where a consistency gap to humans persists (Imajuku et al., 2025), AutoLab on multi-step research-and-engineering work (Xu

et al., 2026), and Continual Learning Bench on whether systems improve across real stateful sessions at all (Asawa et al., 2026). The coding-agent numbers everyone quotes get their own reread later in the series [→ B9].

KEY TAKEAWAY 2

Every agent-eval failure is one of five: contamination, determinism exploits, single-seed reporting, success-criterion gaming, or cost blindness (see Figure 2.2). All five are detectable with a defined check. A benchmark that has not been audited against all five is a demo with a number printed on it.

§3 — What deep RL already learned

Deep reinforcement learning spent the late 2010s discovering that most of its published rankings were noise, and then fixed it. Agent evaluation is currently making the same mistake from scratch. We do not have to. The fix has three parts: report intervals, not points; aggregate with a statistic that ignores lucky outliers; and run enough seeds to mean it.

The Statistical Precipice paper takes its name directly from that deep-RL reckoning. It notes that point estimates of mean scores across a handful of environments produced unreliable agent rankings, and that the computer-use ecosystem is now suffering a *compounded* version of the same disease — non-principled environments feeding non-principled methodology (D'Oro et al., 2026).

> [← A10] The continual-RL benchmark audit reached the identical verdict one field over: the field had been measuring peak performance on individual tasks instead of the quantity that actually mattered. Same failure, same cure, different decade.

Here is the whole idea in eight numbers (these figures are illustrative; see Figure 2.3). Evaluate two agents, A and B, on the same benchmark, eight seeds each. Agent A scores 40, 42, 44, 45, 46, 47, 48, and 88. Agent B scores 50, 51, 52, 53, 54, 55, 56, 57.

The number everyone quotes is the **mean**. The mean of A is 400/8 = **50.0%**; the mean of B is 428/8 = **53.5%**. Close enough to call a tie — and if you had run a single seed each and happened to draw A's 88, you would have crowned A.

Now compute the **interquartile mean**, the IQM, which simply discards the top and bottom quarter of the runs and averages the middle. For A, drop 40, 42 and 48, 88; average 44, 45, 46, 47 to get **45.5%**. For B, drop the extremes and average 52, 53, 54, 55 to get **53.5%**. The IQM says B beats A by eight points. The "tie" was entirely the work of one lucky seed — the 88 — that the mean let dominate.

The **interval** tells you whether to believe any of it. Agent A's scores are spread wide, so its 95% confidence interval is roughly ±10 points: "A = 50%" is compatible with A being anywhere from the low 40s to the low 60s, which is to say it carries almost no information. Agent B's scores are tight; its interval is about ±2 points. Reported honestly, **A is 50% (wide) and B is 53.5% (narrow)**, and only one of those is a measurement. The ranking flip between them was produced entirely by reporting discipline. Nothing about the agents changed.

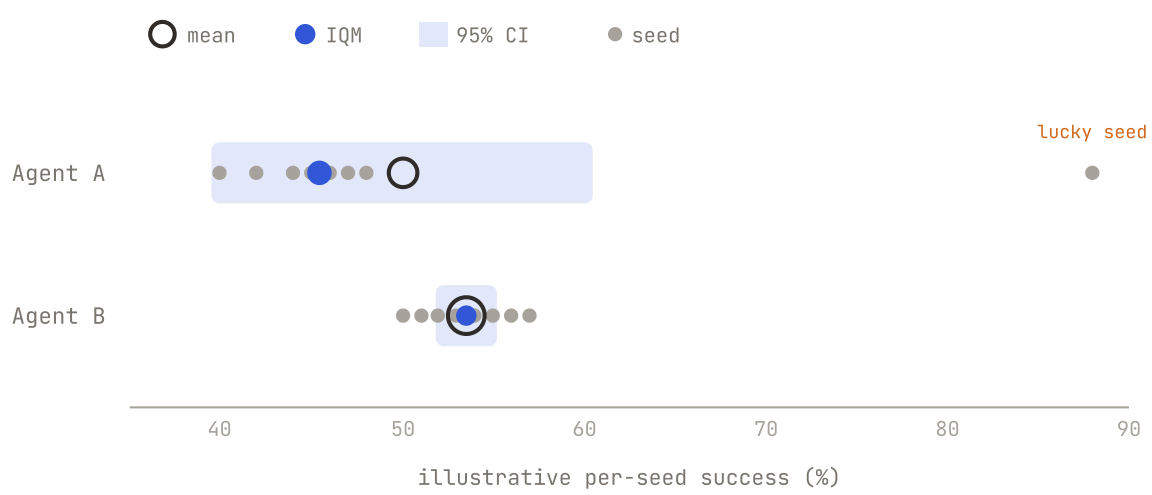

Figure 2.3: A worked, illustrative case. The two agents' **means** nearly tie near 50%, but the **interquartile means** separate them by eight points and Agent A's wide interval makes its number untrustworthy. The method — IQM plus an interval — is the deep-RL lineage D'Oro et al. (2026) restate for agents.

Real agent benchmarks need heavier machinery, but the principle is identical. Because a computer-use evaluation is nested — many rollouts inside many tasks inside many configurations — D'Oro et al. pair **Wilson score intervals** at the rollout level with a **hierarchical bootstrap** at the task level, so the interval reflects every source of variance instead of pretending each trial was an independent coin flip (D'Oro et al., 2026). The statistic changes with the structure of the data; the discipline does not. None of this is new. It is a decade old in the field next door. Agent evaluation reset the clock to zero because it grew up inside product demos, where one good run is the deliverable, rather than inside an experimental science, where one good run is an anecdote.

KEY TAKEAWAY 3

A point estimate without an interval is not a measurement. Report the IQM to disarm lucky seeds and a confidence interval to say how far to trust the number — and if two agents' intervals overlap, you have not measured a difference, no matter what the means say (see Figure 2.3).

§4 — Cost is a metric

Accuracy without cost is not a measurement of an agent; it is a measurement of an agent with the price tag torn off. Two systems at the same pass@1 can differ by an order of magnitude in dollars per task, and the cheaper one is, for almost every deployment, the better agent (see Figure 2.4). Yet cost is the metric most often missing from the number teams quote.

The omission is partly cultural — leaderboards have one column — and partly that cost is awkward to score well. But the tools exist. The forecasting literature has worked out how to score predictions *properly*, rewarding calibrated confidence rather than lucky point hits: the Bayesian Linguistic Forecaster reaches state-of-the-art on a forecasting benchmark precisely by maintaining a calibrated belief state and aggregating many independent trials, under a backtesting protocol with under 1.5% leakage (Murphy, 2026). The same discipline applies to agents — report the full distribution of outcomes and what each one cost, not a single accuracy with the economics hidden.

There is also a development-time version of the problem. Running a full agent evaluation is expensive, so teams want a cheap proxy that predicts downstream performance early. Patel et al. (2026) show that token-level statistics — entropy, top-k accuracy, the rank a model assigns to expert-written tokens — forecast downstream results better than training loss or raw compute, and do so cheaply

enough to drive model selection. That is cost-aware evaluation pointed in the other direction: spending less to measure, without lying about what was measured.

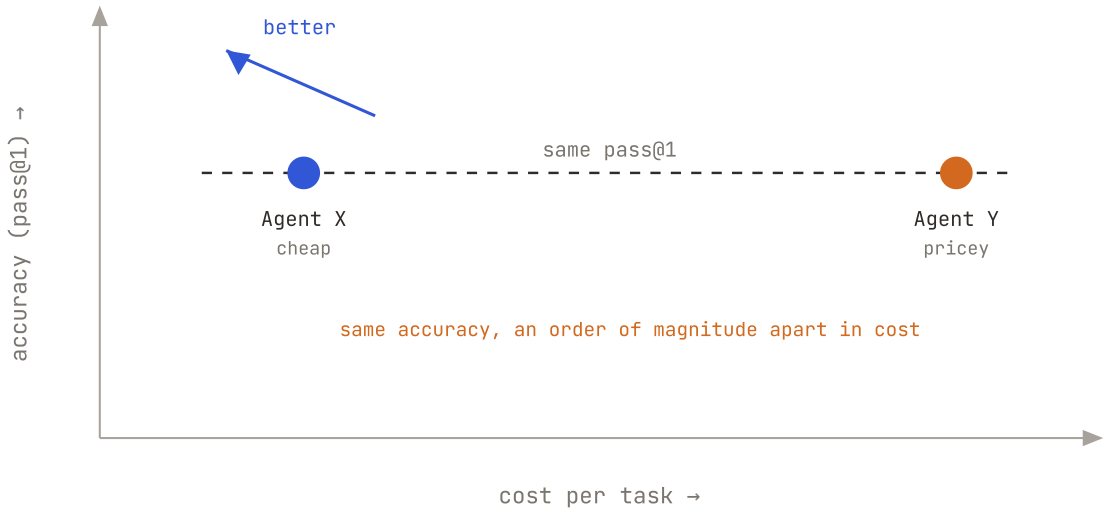

Figure 2.4: Accuracy and cost are orthogonal axes (qualitative; no benchmarked values). Two agents on the same line of pass@1 can be an order of magnitude apart in price — so an accuracy number reported without cost has discarded the axis that decides whether you can ship. The principle follows the proper-scoring and cost-aware work of Murphy (2026) and Patel et al. (2026).

Plot accuracy against cost and the point becomes unarguable: a horizontal line of equal pass@1 can cross agents whose cost differs by orders of magnitude, and reporting only the height of that line discards the axis that decides whether you can afford to run the thing. The dollar mechanics of long agent runs get a full treatment later in the series [→ B9]; the claim here is narrower and non-negotiable.

KEY TAKEAWAY 4

Cost is not an operational footnote to an accuracy number; it is half of the number. An agent eval that reports pass@1 without dollars, tokens, or latency per task has measured one of the two quantities that decide whether the agent is worth running.

§5 — Open-world measurement

If static benchmarks are the disease, the cure has a name and a shape: **open-world evaluation**. The idea is to stop measuring against a frozen task set and start measuring the capability itself, under conditions designed so that the score cannot be inflated by memorizing the test (see Figure 2.5).

The most developed statement of this comes from a Princeton–Stanford–UK AI Security Institute coalition, which lays out what a trustworthy frontier evaluation actually requires (Kapoor et al., 2026). Three commitments matter most.

Capability elicitation. A low score means low capability only if you genuinely tried to elicit the capability — with the right scaffolding, prompting, and tools. So elicitation effort has to be part of the report, not an afterthought. An under-elicited model looks safe for the wrong reason, and that false comfort is more dangerous than a high score.

Open-task distributions. Instead of a fixed list of tasks an agent can be tuned against, sample from an open, evolving distribution, so that "doing well" cannot collapse into "having seen the test." This is the same move DigiWorld makes with its millions of configurations (D'Oro et al., 2026), lifted from one benchmark into a methodology.

The role of third parties. An evaluation a lab runs on itself and an evaluation an independent body runs are not the same evidence. The UK AI Security Institute model — external evaluators with privileged access running their own tasks — exists because self-reported numbers, however honest, carry a structural conflict of interest. Put together, a trustworthy agent eval report names the tasks and where they came from, states how hard the team tried to elicit the capability, gives every number an interval and a cost, and says who ran it. Most numbers you will read this year contain none of those four things.

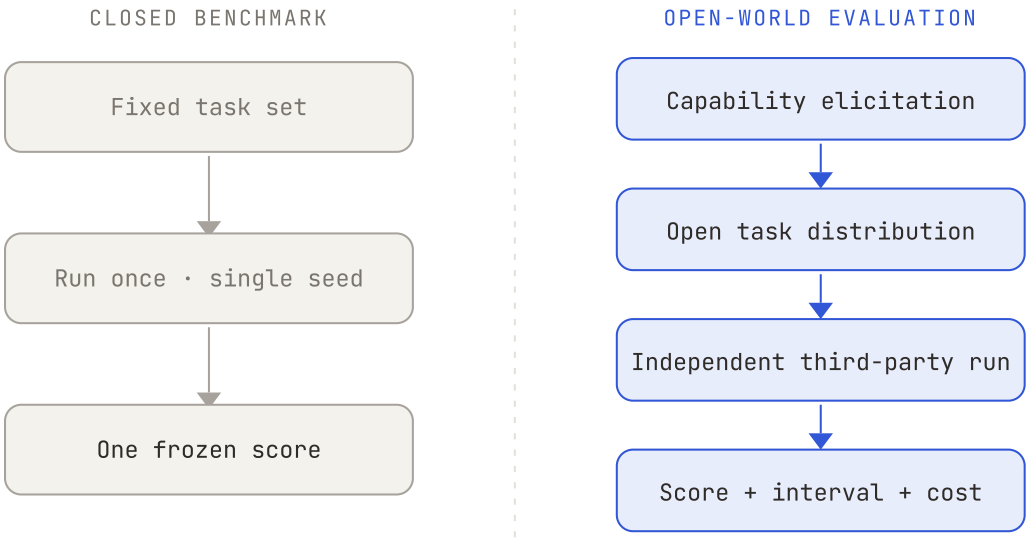

Figure 2.5: The closed benchmark produces one frozen number; the open-world pipeline produces a defensible one — elicited, sampled from a distribution the model cannot have memorized, run by an outside party, and reported with an interval and a cost. After the framework of Kapoor et al. (2026).

KEY TAKEAWAY 5

Open-world evaluation replaces "how did the model do on our fixed test" with "how capable is the model, given a real attempt to elicit it, on tasks sampled from a distribution it cannot have memorized, scored by someone with no stake in the result." That is a measurement regime, not a leaderboard.

§6 — What frontier labs actually report

It is worth reading a real frontier evaluation closely, because the best of them already do much of what §5 asks — and the gaps fall exactly where you would predict. The Claude Opus 4.7 system card is a useful specimen, with one caveat that governs how to read it: it is a **practitioner artifact**, a vendor's safety report, not a peer-reviewed finding (Anthropic, 2026).

Read it in three columns (see Figure 2.6). What is **measured**, and measured well: capability is benchmarked against the prior model and judged not to advance the frontier, with cyber capabilities assessed as roughly similar to the previous Opus. These claims are concrete, comparative, and falsifiable. What is **hedged**: misalignment risk is described as very low, though explicitly higher than for some earlier models — the honest language of a quantity the lab cannot yet measure crisply. Hedging is not evasion here; it is the correct response to genuine uncertainty, and it is more trustworthy than a confident point estimate would be. What is **absent, or external**: the most telling line is that the model could not complete the UK AI Security Institute's evaluation tasks — a capability ceiling established by an outside body running its own tests, not by the lab. That is the open-world principle from §5 operating in production: the most credible negative result in the document is the one the lab did not run itself.

Figure 2.6: One frontier system card, read in three columns (Anthropic, 2026, a practitioner artifact). The load-bearing claims are the comparative ones and the externally run one; the absolute self-reported numbers are the least so.

Labs increasingly forecast downstream performance before the expensive evaluation exists, using proxy metrics of the kind described in §4 (Patel et al., 2026) — useful, and a fresh surface for self-deception if the proxy is chosen because it looks good. The safety evaluations get their own article [→ B10]; the choosing-your-own-metric problem is the subject of the next section.

KEY TAKEAWAY 6

The most trustworthy numbers in a frontier system card tend to be the comparative ones (this model versus the last) and the externally run ones (what an independent evaluator could and could not get it to do). Treat absolute, self-reported, single-number capability claims as the least load-bearing part of any eval — including a good one.

§7 — Self-measuring systems

The frontier of agent evaluation is agents that build their own evaluations, and it is genuinely promising and quietly dangerous for the same reason. When the thing being measured also designs the measurement, you get a faster loop and a new way to fool yourself (see Figure 2.7).

Agentic-imodels is a clean example. It runs an autoresearch loop that evolves data-science tools — scikit-learn-compatible models for tabular data — optimizing them against two objectives at once: predictive performance and a novel interpretability metric (Singh et al., 2026). The interpretability metric is itself agent-graded: an LLM is asked whether it can simulate the fitted model's behavior from the model's own string representation. The evolved tools improve on both axes, which is a real result and a glimpse of where evaluation tooling is heading — agents that discover better ways to measure agents.

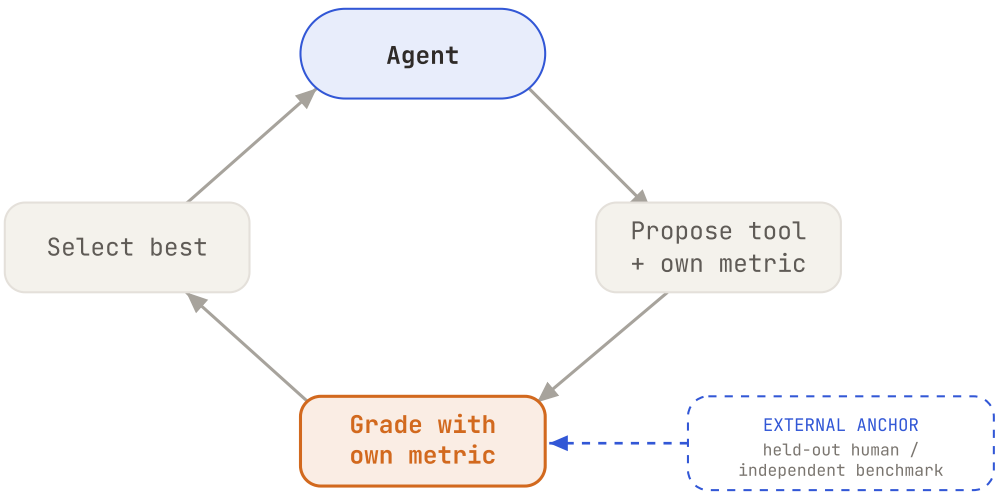

Figure 2.7: The self-measuring loop (Singh et al., 2026). An agent that both proposes a metric and grades against it has a built-in conflict of interest; the dashed external anchor — a grader with no stake in the result — is what keeps the loop honest.

The circularity is the catch. An optimization loop that both produces a candidate and grades it with a metric of its own choosing has every incentive to drift toward metrics it scores well on, rather than metrics that track the property you care about. This is not a flaw in this particular system; it is a structural hazard of the whole direction. The safeguard is the one §5 already named — a grader with no stake in the result. A self-measuring system needs an external anchor, a held-out human judgment or an independent benchmark, or it will eventually optimize the ruler instead of the thing the ruler was for.

KEY TAKEAWAY 7

Agents that evolve their own evaluation tooling are coming, and they are worth building — but a metric an agent both optimizes and grades is a metric under conflict of interest. Self-measurement needs an external anchor, or the loop will learn to move the ruler (see Figure 2.7).

§8 — A checklist for your own evals

Everything above reduces to a checklist you can run against any internal agent benchmark before you trust its numbers — or ship a decision based on them. This is the deliverable of the article. If you read nothing else, read the table (see Figure 2.8).

Ten checks, each tied to a failure mode, each with a detection step you can actually perform and a mitigation — and, honestly, the residual risk that survives the mitigation. The green cell is the exception, not the rule, and that is the point: the crisis is not yet solved, and a checklist that pretended otherwise would be one more demo with a number on it.

FAILURE MODE	HOW IT SHOWS UP	DETECTION CHECK	MITIGATION	RESIDUAL RISK
1. Contamination	Score reflects memorized solutions, not reasoning	Perturb the surface (rename, restory); watch the score delta (Wiemann et al., 2026)	Private held-out split; out-of-distribution variants	⚠️ **Pretraining exposure can't be fully excluded**
2. Determinism exploits	A fixed initial state lets a blind action-replay win	Run a blind replay agent — a high score means it's memorizable (D'Oro et al., 2026)	Multifactorial variability across configs (PRISM)	❌ **Any static single-config env stays exploitable**
3. Single-seed reporting	One run, one number; rankings flip on replication	Re-run N seeds; check whether the intervals overlap	Report IQM + confidence intervals (Wilson + hierarchical bootstrap) (D'Oro et al., 2026)	⚠️ **Many-seed cost tempts a relapse to single runs**
4. Success-criterion gaming	"Pass" is a surface or LLM-judge match, not a goal state	Verify the final world state, not the action surface form (Zhou et al., 2023)	State-based functional verification; privileged checkers	⚠️ **The verifier itself can be gamed**
5. Cost blindness	pass@1 quoted with no dollars, tokens, or latency	Log cost per task beside every accuracy number	Report accuracy and cost jointly; proper scoring (Murphy, 2026)	⚠️ **No shared cost baseline across the field yet**
6. Thin state coverage	A high score on a sliver of the state space	Count distinct verified configs; run the replay-collapse test (D'Oro et al., 2026)	Combinatorial config space (data × theme × state)	✅ **Closes when coverage is genuinely enforced**
7. pass^k reliability gap	pass@1 looks fine; the agent is inconsistent run-to-run	Report pass^k (all k must pass); track pass^8 (Yao et al., 2024)	Gate deployment on a pass^k threshold	❌ **Frontier agents still under 25% pass^8 in retail**
8. Missing human baseline	An absolute score with nothing to anchor it	Collect human success on the same tasks (Mialon et al., 2023)	Always pair the agent score with the human gap	⚠️ **Human baselines are scarce and costly**

tunnel vision	One accuracy hides robustness, predictability, safety	Compute a multi-dimension reliability profile ("Towards a Science of AI Agent Reliability," 2026)	Report consistency, robustness, predictability, safety	⚠ **Capability gains buy only small reliability gains**
10. Closed set, no third party	A fixed self-run task list; no elicitation reported	External re-evaluation; open-task sampling (Kapoor et al., 2026)	Open-world design + independent audit	⚠ **Few credible third parties; elicitation is unbounded**

Figure 2.8: The eval-failure audit table — ten checks for any internal agent benchmark, each tied to a failure mode, a detection step, a mitigation, and the honest residual risk. Every row traces to a cited result. Notice how few cells are green.

Notice how few cells are green. Multifactorial state coverage is the one place where a known mitigation genuinely closes the gap — randomize the configuration space widely enough and blind replay simply stops working (D'Oro et al., 2026). Almost everywhere else the honest verdict is amber or red: you can detect the failure and shrink it, but not banish it. An internal benchmark that scores green across the board has not solved evaluation; it has stopped looking.

KEY TAKEAWAY 8

Run the ten checks in Figure 2.8 against your own agent benchmark before you quote its number to anyone. The goal is not a clean scorecard — it is knowing exactly which of the five failure modes still contaminates your number, and by how much.

What Comes Next

This article was about measurement: why the numbers you read mislead, and what a number you can trust looks like. The next one is about what those same defective numbers do when you turn them inward and *train* on them. A reward signal is a benchmark pointed at the optimizer. If a static, gameable benchmark misleads your evaluation, the same reward misleads your training — and far more expensively, because the agent will actively search for the gap between the metric and the goal you meant. The measurement discipline built here is the prerequisite for everything downstream: you cannot fix a reward you cannot measure.

> [→ B3] **Environments Are the Bottleneck.** The next article argues that your agentic-RL project is stuck on the environment, not the algorithm — that environments are training data, and most of them are the wrong data. It is where the failure modes named here stop being a reporting problem and start being a training problem.

References

1. D'Oro, P., Silwal, S., Wong, W., Sun, Y., Xiao, F., Wang, M., Gan, E., Bolourchi, A., & Tighe, J. (2026). **Computer Use at the Edge of the Statistical Precipice.** *arXiv preprint.* arXiv:2605.08261.

2. Wiemann, M. L., Smith, L. M., Melchior, P., Mishra-Sharma, S., Wilson, A. G., Izmailov, P., & Cuesta-Lázaro, C. (2026). **DiscoverPhysics: Benchmarking LLMs for Out-of-the-Box Scientific Thinking.** *arXiv preprint.* arXiv:2605.26087.

3. Zhou, S., Xu, F. F., et al. (2023). **WebArena: A Realistic Web Environment for Building Autonomous Agents.** *arXiv preprint.* arXiv:2307.13854.

4. Mialon, G., Fourrier, C., Swift, C., Wolf, T., LeCun, Y., & Scialom, T. (2023). **GAIA: A Benchmark for General AI Assistants.** *arXiv preprint.* arXiv:2311.12983.

5. Yao, S., Shinn, N., Razavi, P., & Narasimhan, K. (2024). **τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains.** *arXiv preprint.* arXiv:2406.12045.

6. Liu, X., Yu, H., Zhang, H., Xu, Y., et al. (2023). **AgentBench: Evaluating LLMs as Agents.** *arXiv preprint.* arXiv:2308.03688.

7. Merrill, M. A., Shaw, A. G., Carlini, N., et al. (2026). **Terminal-Bench: Benchmarking Agents on Hard, Realistic Tasks in Command-Line Interfaces.** *arXiv preprint.* arXiv:2601.11868.

8. Imajuku, Y., Horie, K., Iwata, Y., Aoki, K., Takahashi, N., & Akiba, T. (2025). **ALE-Bench: A Benchmark for Long-Horizon Objective-Driven Algorithm Engineering.** *39th Conference on Neural Information Processing Systems (NeurIPS 2025), Datasets and Benchmarks Track.*

9. **Towards a Science of AI Agent Reliability.** (2026). *arXiv preprint.* arXiv:2602.16666.

10. **Agents' Last Exam.** (2026). *arXiv preprint.* arXiv:2606.05405.

11. Asawa, P., Glaze, C. M., Orlanski, G., Ramakrishnan, R., Xu, B., Biswal, A., Chen, V. S., Sala, F., Zaharia, M., & Gonzalez, J. E. (2026). **Continual Learning Bench: Evaluating Frontier AI Systems in Real-World Stateful Environments.** *arXiv preprint.* arXiv:2606.05661.

12. Xu, Z., Chen, J., Huang, Y., et al. (2026). **AutoLab: Can Frontier Models Solve Long-Horizon Auto Research and Engineering Tasks?** *arXiv preprint.* arXiv:2606.05080.

13. Murphy, K. (2026). **Agentic Forecasting using Sequential Bayesian Updating of Linguistic Beliefs.** *arXiv preprint.* arXiv:2604.18576.

14. Patel, A., Reddy, S., Mosbach, M., & Bahdanau, D. (2026). **Forecasting Downstream Performance of LLMs with Proxy Metrics.** *arXiv preprint.* arXiv:2605.18607.

15. Kapoor, S., Kirgis, P., Schwartz, A., Rabanser, S., Allaire, J. J., Bommasani, R., et al. (2026). **Open-World Evaluations for Measuring Frontier AI Capabilities.** *arXiv preprint.* arXiv:2605.20520.

16. Anthropic. (2026). **Claude Opus 4.7 System Card.** *Anthropic.*

17. Singh, C., Tan, Y. S., Xu, W., Gero, Z., Yang, W., Galley, M., & Gao, J. (2026). **Agentic-imodels: Evolving Agentic Interpretability Tools via Autoresearch.** *arXiv preprint.* arXiv:2605.03808.

GET THE FULL BOOK

Agentic Engineering

A deployed agent's capability is the product of model quality and harness quality — and the harness is the part you control. This book is the Harness Engineering series as one volume: thirteen essays on the discipline of building agents that actually ship. Evals that measure capability instead of demos. Memory that survives the session. Tools, planning, security, and ops treated as engineering surfaces, not vibes. Every chapter ends with a deployable artifact — a checklist, decision table, or playbook you can put to work Monday.

Bundled inside: **The Long Bet** — six essays on what survives in agentic AI, each ending in a dated, falsifiable prediction with a review already on the calendar.

Every claim in this book was traced to its source paper before publication. It is a **living book**: versioned like software, and every future edition is included with your copy.

The purchase includes the book as **PDF and EPUB**, the companion *Agentic Engineering Checklists* PDF, and **every future version** of the book. Find it via **agenticfrontier.dev/book**.