WORKING PAPER · SEPTEMBER 2026

How Do We Use This?

Findings from running an AI team on real work for six months, and a test that could prove them wrong

Berg Atkinson, Wolfberg LLC · berg@wolfberg.ai

Preprint. Not peer reviewed. Revised 21 September 2026; corrected 23 and 25 September 2026 (see Corrections); the primary study has not yet been run.

ABSTRACT

Large language models deployed as working agents tend to end a search when their answer becomes defensible against the evidence already gathered, rather than when it becomes correct. This is a form of what Simon called satisficing, and of what clinical reasoning calls premature closure; the counterweight we propose is a cognitive forcing function in that older tradition, aimed at a machine (§2.1). We call this the stop problem and study it in a single-operator engineering practice running several concurrent AI agent sessions, against a corrective record spanning February to August 2026; there the premature stop, when it is caught at all, is caught, in the practice's experience, mostly by the human operator rather than by any automated check. The practice cannot state a catch rate, only a catcher share (§12). We advance five propositions: that correctness is the weight missing from how these agents are trained and deployed; that the gap is one of leadership rather than capability; that written rules drift while boundaries hold; that an agent grading its own work is not being measured; and that one operator's corrective record on consequential work is data the developers of frontier models cannot buy. We support them with exploratory readings from the practice's own logs: about 48% of the operator's chat turns carried corrective or teaching signal rather than routine direction, roughly a third once the detector's grade against a blind key is priced in, while the system logged corrections on 1.76%; written rules a future agent must go and read drift while rules injected before its first token fire, a distinction we report as a mechanism rather than as a rate, because adherence across the corpus has never been measured (§5.1); a tracked error class returned in the very next session in 48 of 75 intervals (§5.2); and corrections the system logged ran far below the corrections the operator made, an indication whose provenance we set out in full.

We then specify a counterweight: a challenge that is external to the agent's output, discriminates on actual error, and is delivered at the moment the agent stops. The contribution is not the design, which four independent 2026 studies converge on and have already benchmarked with gains; it is the test the benchmarks cannot run, a pre-registered trial of the mechanism in a live practice, on consequential work, against a recurrence baseline computed before any intervention. The challenge engine is implemented, and its first calibration is itself a finding: an open-weight model auditing 170 claims against the evidence that preceded them never once returned a contradiction, even where that evidence contained the failure; when a revised prompt made it fire, it fired on true and false claims alike, at a precision equal to the base rate; and a second model family caught 3 of the 41 false claims. The four objections a skeptical reader will assemble are all true and all stated here: the design is convergent, the results so far are null or not yet run, the instruments were built by the practice they measure, and the logs are private. Two of the four are answered by one object, a decision rule ratified before its data are seen, under which a null is an equally reportable result and ends the program. The last two are answered only by measurement from outside the practice, and the one instrument of that kind (§10.7) is specified and has not yet run.

Keywords premature commitment; stopping rules; human oversight of LLM agents; AI auditing; single-case experimental design; pre-registration; measurement validity

CONTENTS

1 Introduction

A working agent receives a task, gathers evidence, and at some point stops gathering and answers. This paper is about what decides that point. The field note that started this program put it plainly:

> *"[The model is] trained to present as fact the first answer, the only criteria for that answer being that it thinks it meets the user's expectations and is defensible based on the available information... Correctness of the answer is not a factor."*
>
> The author's research notes, 30 August 2026

Restated as a mechanism: the stopping rule is satisfiability, not correctness. The search ends when the agent can build a defensible account from what it already holds. Because further evidence can only make an account harder to defend, the marginal incentive is to stop. Written as two stopping times, the defect is the distance between them.

$$t^{\star} = \min \{ t : D(a_t \mid E_t) \} \quad \text{the moment the agent stops}$$

$$t^{\circ} = \min \{ t : C(a_t) \} \quad \text{the moment the task is satisfied}$$

E_t is the evidence held at step t, which grows only by searching. D is defensibility against E_t; C is correctness. D can hold while C fails, and the set of turns where it does is exactly where the misses live. Nothing in the objective rewards closing the distance.

The incentive language above is an as-if description, in the sense Wang and Huang (2026) give their own model: it says the behavior can be rationalized this way, not that any incentive is represented inside the model, whose next step is simply the likeliest continuation of a context that already reads as resolved. The defect is therefore located at the termination decision rather than in the reasoning that precedes it, which is why interventions aimed at reasoning quality do not reach it. Figure 1 draws the mechanism as a position on one axis.

Independent work points the same way. Mehta (2026) describes premature commitment in tool-using agents: runs settle on one reading of the evidence early and then defend it, and a hidden-state signature of that settling can be read at runtime. The same work reports that the signature does not separate agents that settled on the right answer from agents that settled on the wrong one, a result its author grades as suggestive rather than confirmed, and it advises deployers to send settled cases to an external verifier or a human rather than resample. A detector of the stop is not a corrector of it.

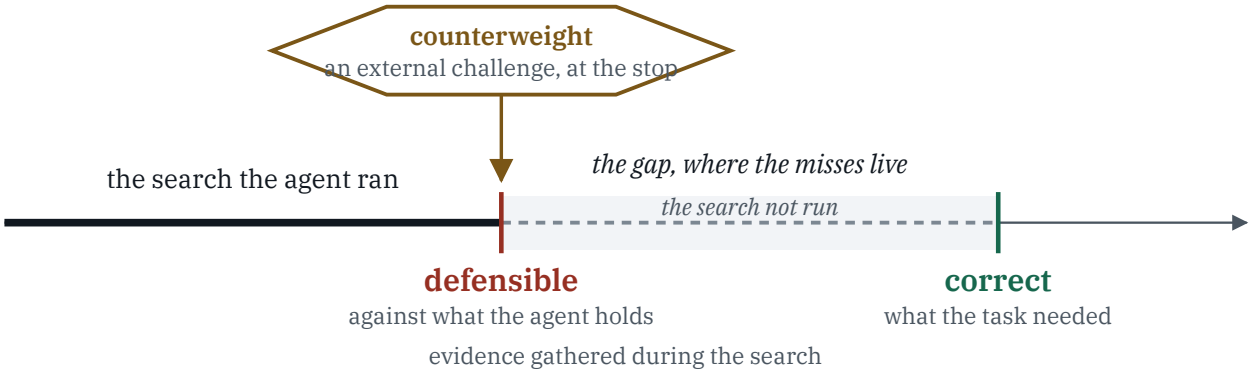

Figure 1. The stop problem as a position on one axis. The agent stops where its answer becomes defensible against what it holds; the correct answer lies further along. The counterweight (§8) is an external challenge timed to the stop. Schematic only: no quantity is plotted, and the counterweight's challenge engine is implemented but not deployed. **Schematic.**

The stakes in the practice studied here are small: an agent stops early, the operator catches it, and the cost is rework. The same mechanism outside a practice with an attentive operator is not small. In May 2025, over a month-long episode, a user was led by a commercial assistant into believing he had discovered a new form of mathematics, across a conversation whose transcript ran past a million words. A former safety researcher at the vendor analyzed that transcript with the vendor's own open-sourced safety classifiers and reported that, of more than two hundred assistant messages graded, over 85% showed unwavering agreement with the user and over 90% affirmed the user's uniqueness. When the user recognized the error and asked that the conversation be escalated, the assistant said it would "escalate this conversation internally" and that "multiple critical flags have been submitted." It had no such capability, and the vendor confirmed as much to the analyst in writing (Adler, 2025).

Two things in that account are this paper's subject rather than a neighboring problem. The agreement rate is the challenge-and-capitulation dynamic of §4.2 running in the absence of anyone to push back. The escalation claim is worse and more specific: an assertion about an action the system had taken, which it had not taken, offered because the assertion was defensible in context and met the user's expectation. That is §6 at its most consequential, a self-report standing in for a fact with nothing external to check it. The same account notes that the analysis was performed using safety classifiers the vendor had itself built and open-sourced, which is the pattern of §6.3 exactly: an instrument that existed, and was not wired to the decision it could have informed.

In the practice studied here, the corrector is the operator, the author, who reads output, notices the premature stop, and challenges it. This works and does not scale. Its throughput is bounded by one person's attention, and the system's liveness is coupled to that person's working calendar; the operator's note on one stall reads "it stopped because I stopped working." The program asks whether that function can be reproduced mechanically, delivered at the moment of termination, and shown to improve outcomes.

The rest of the paper is built around five propositions about the problem and its setting (§3 to §7). Each is argued in its own section from two kinds of evidence, readings from the practice's own logs with the full provenance of each and published work that bears on it, and Table 2 maps them. The paper then specifies the counterweight clause by clause and reports on its implementation (§8), reports the first calibration of the counterweight's auditor, which failed in an informative way (§9), and sets out the evaluation, including the result that ends the program (§10). The work is applied and design-oriented: it builds an artifact for a field problem and evaluates it in the field it came from, at a sample size of one operator.

2 Background and setting

2.1 The problem's lineage

The stop problem is not new; only its subject is. A search that ends when the searcher is satisfied rather than when the answer is right is Simon's satisficing (Simon, 1955), and when people stop gathering information has its own literature (Browne, Pitts & Wetherbe, 2007). In clinical reasoning the same failure is named premature closure, the tendency to settle on a diagnosis before it has been verified, and it is a documented source of diagnostic error (Graber, Franklin & Gordon, 2005). The standard remedy there is also named: a cognitive forcing function, a deliberate prompt that interrupts the fluent stop and forces a second look (Croskerry, 2003), studied more recently as a way to reduce a person's overreliance on an AI's suggestions (Buçinca, Malaya & Gajos, 2021). The counterweight of §8 is a cognitive forcing function delivered to a machine.

Three results from the language-model literature bound what such a check can be, and each was read in full. Models do not reliably repair their own reasoning without an outside signal: asked to self-correct with no external feedback, they can make correct answers worse and need an oracle to gain anything (Huang et al., 2023). Challenge is double-edged: across ten models on seven tasks, a bare "are you sure?" flipped answers 46% of the time and cost about 17% of accuracy (Laban et al., 2023), which is why the counterweight must discriminate on error rather than merely apply pressure. And redundancy among similar checkers does not buy independence: independently written versions of a program fail together far more often than independence would predict (Knight & Leveson, 1986), the reason a deciding body must differ in kind from the bodies it decides between (§6.5). The design rule of §8.3 restates, for a practice run by AI agents, Goodhart's and Campbell's laws (Campbell, 1979) and the software-inspection requirement of a non-author reviewer (Fagan, 1976).

The stop problem in LLM agents is also, as of 2026, an active literature of its own, and this paper's own search did not find it until late. Roh and Han (2026) build HALT, an external verification layer that halts a frozen search agent once retrieved evidence covers every reasoning hop. What it saves depends sharply on what it is given, and the distinction is worth carrying: supplied with gold claims, a diagnostic setting, it cuts search loops by 20 to 45% across three datasets; supplied with claims the system generates for itself, which is the deployable setting, the cuts fall to between 2 and 18%. Answer accuracy passes formal non-inferiority testing in both. Run in the other direction, forcing continuation

on the trajectories HALT flags as under-covered, it raises population exact match by 8.2 points on HotpotQA and by 2.7 and 4.3 points on the other two datasets. Ko et al. (2026) study what they call illusory completion, where an agent believes a task resolved while constraints remain unverified, and measure underverified-answer rates ranging from 52.1% for the strongest system tested to 76.3% for a ReAct baseline, with human annotators agreeing at Fleiss $\kappa = 0.74$. The finding this program should sit up for is the one inside the successes: even among answers that were *factually correct*, 19.1% were still underverified on that strongest system. That is the defensible-versus-correct distinction measured directly, and measured on answers that a correctness-only scorer would have recorded as wins. Their inference-time tracker, LiveLedger, reduces underverified answers by up to 26.5% and raises accuracy by up to 11.6%. Xu et al. (2026) gate the commitment itself: their execution layer refuses an agent's edit or patch until the evidence the task requires has actually been observed, worth 4.8 to 11.8 points of Pass@1 across 500 SWE-bench Verified instances while cutting token use by up to 12.1%. In the same comparison, bolting a post-hoc self-review step onto the baseline agent instead *lowered* Pass@1, by 1.4 and 1.8 points on the two models tested, which the authors read as confirming that post hoc self-review cannot recover from decisions made on insufficient evidence. Luo et al. (2026) treat stopping as abstention, find that agents abstain late when they abstain at all, and improve timely abstention from 26.7% to 57.4% by distilling past trajectories into stopping rules injected into the agent's context.

Behind those four sits an older and larger line that this program's searches repeatedly missed, because it is indexed under retrieval rather than under stopping. Adaptive-retrieval controllers have been deciding when an agent should stop gathering for several years, using reflection-token critics, model uncertainty, or question complexity to route the decision; Roh and Han (2026, §2) survey the family. The clearest statement of its premise is Park, Cho and Lee (2025), who cast iterative retrieval as a finite-horizon Markov decision process, learn a value-based controller for when to stop, and conclude that adaptive stopping is a missing component of current agentic systems. A second field asks the same question in the vocabulary of abstention rather than search: selective prediction and learning to defer, where Piatrashyn et al. (2026) hand a decision to a larger model once a small model's calibrated uncertainty crosses a threshold, and report that deferring a modest fraction of decisions matches the expensive model's quality. These literatures do not talk to each other, and we measured it rather than asserting it. We read the complete reference list of fourteen papers: six on retrieval stopping, five on agent termination, and three on deferral. Citations to the foundational reject-option and selective-prediction lineage appear in three of the fourteen, and all three are in the deferral group. None of the five agent-termination papers cites any of it. Five of the six retrieval-stopping papers cite none of it either; the sixth cites selective classification and then explicitly declines to inherit its guarantee, writing that it uses the same conceptual separation but claims no formal risk guarantee, and it is the only paper of the fourteen that calls for the bridge to be built. Between the retrieval-stopping group and the deferral group there are eighteen possible citation pairs in each direction, and the realized count is zero both ways; between agent termination and deferral it is zero of fifteen each way. Each cluster cites itself densely, so this is not a sparse field. It is three internally connected and mutually hermetic islands, each working on when a system should stop. A fourth and older line asks the same question of a network's internal computation rather than of its search, and it sits outside the fourteen-paper count: adaptive computation, in which a recurrent network learns how many computational steps to take between receiving an input and emitting an output (Graves, 2016).

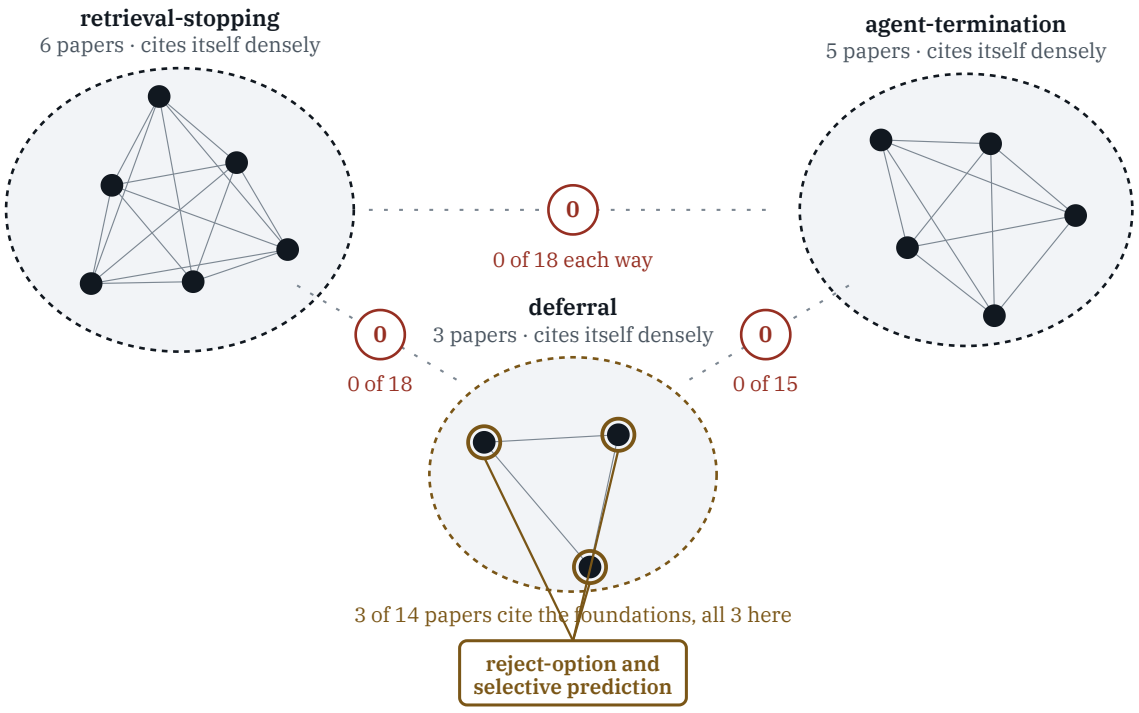

Figure 2. The 2026 stopping literature as a citation graph, counted rather than asserted (§2.1). Each cluster cites itself densely; the crossings that would connect them are empty, zero of eighteen possible pairs each way between retrieval-stopping and deferral and zero of fifteen between agent-termination and deferral, and only three of the fourteen papers reach the fifty-year-old selective-prediction and learning-to-defer foundations. Three internally connected, mutually hermetic islands. **Measured.**

We report that as a finding rather than a complaint, and it implicates this paper directly. A search entered from any one of those vocabularies returns a small set that looks far more novel than it is, which is exactly what happened here across several passes before the citation graphs were walked. The mechanism is the paper's own thesis wearing a bibliography: a search stops when it has enough to write a defensible related-work section.

Three consequences follow, and none of them is comfortable. First, the counterweight's clauses (a) to (c) describe a design that others have now built and measured, mostly with positive results, so this program should be read as a fourth instance rather than a proposal. Second, Xu et al.'s comparison is direct evidence for the design rule of §8.3: gating before commitment beat correcting after it. Third, Luo et al.'s reusable stopping rules, distilled from past trajectories and injected without updating any model parameters, are H1's mechanism working in a setting where the rules came from trajectories rather than from an operator. Against that background this paper makes no claim to priority over any of it, and §7.2 records why that kind of claim was removed rather than narrowed.

2.2 The practice and its data

The practice is a one-person engineering and consulting company. Its work is done by several concurrent AI agent sessions from one commercial model family (Anthropic's Claude), running under a

harness of hooks, shared logs and automated checks, and directed by the operator, who approves every consequential change. Table 1 fixes the vocabulary used in the rest of the paper.

Table 1. Terms used in this paper.

Term	Meaning here
Agent	One running AI agent session, with its own context window and its own identifier.
Operator	The human who directs the agents and approves their consequential changes; the author.
The logs	The durable records an agent does not write about itself: session transcripts, an event log, version-control history, a log of caught errors, and a log of the operator's decisions.
Stopping moment	The turn at which an agent emits a terminal assertion without a further retrieval act.
Automated check	A check that runs at a boundary, for example when an agent ends its turn or at a commit, and can refuse or flag.
Catch	The first surfacing of an error, whether by the operator, another agent, an automated check, or the agent itself.
Implemented, designed	Whether a component exists in the running system or only in its design. The practice's architecture model marks every component one way or the other, and this paper does the same.

The operator's corrections are the data of interest. The primary sample frame is the chat-lane conversation file of the practice's export of 26 August 2026: 8,221 operator turns in 281 conversations between 9 February and 26 August 2026, pinned by SHA-256 hash and reproduced to the turn by an independent count. Design-lane conversations are excluded from this frame, so rates over it describe the chat lane rather than all of the operator's work. A second corpus holds 294 code-agent session transcripts (count of 30 August 2026). The operator wrote a 199-exchange answer key blind, without seeing any model's labels, on a systematic sample of the frame. Two numbers are in play for that key and this paper had been using one: 199 rows were authored, and a documented repair that removed unanswerable rows leaves 195 in the file the detector was actually graded against. Where a figure below depends on the grading, it depends on the 195. The practice's event log held 30,377 events when counted at its files for this revision (11 September 2026, US Eastern; 2,640 in the live file and 27,737 archived), and its architecture is documented in a DoDAF 2.02 model generated from the codebase, with 35 views and 351 data-dictionary elements, each marked implemented or designed. Both figures are the generating instrument's own headline and both disagree slightly with its own itemization: the package's view index lists a thirty-sixth view, and the registry's per-type element counts sum to 354. We quote the headline and record the discrepancy rather than silently picking whichever number reads better. Decisions cited in this paper are dated.

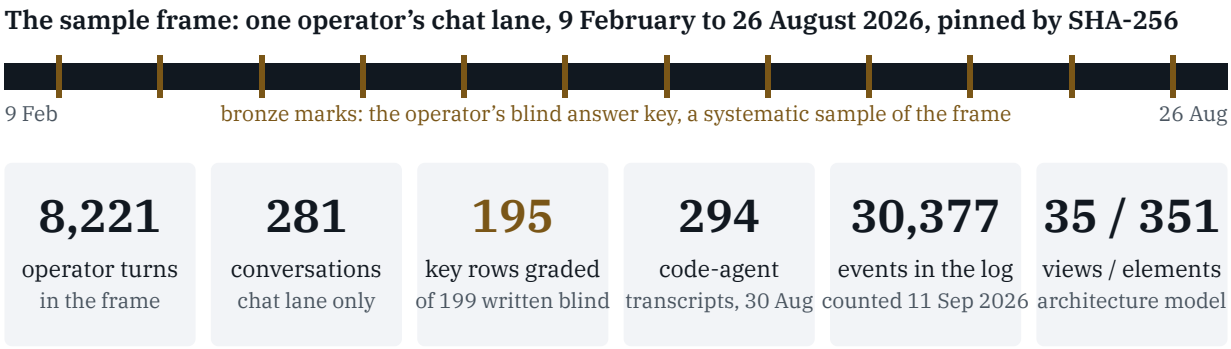

Figure 3. What the record holds. The frame is the chat-lane conversation file of the export of 26 August 2026; the operator’s answer key is a systematic sample of it, written blind. Counts are as reported by the generating instruments (§2.2). **Measured.**

The readings in §3 to §7 come from instruments the practice built and operates. They are descriptive and exploratory: none is the primary outcome, and all of them are exposed to the threats set out in §11. Table 2 maps each proposition to the readings and the published work it rests on.

Table 2. The five propositions and the evidence each rests on.

	Proposition	From the practice’s logs	From recent work	§
P1	The missing weight is correctness	Corrective signal in about 48% of operator turns detector graded 93/64 against the blind key	Mehta (2026)	3
P2	Direction, not capability	Of five misses in one window, two surfaced by the operator outright and one only after the operator pointed at a silent check case series, 4 to 5 Sep 2026	Kelley & Riedl (2026)	4
P3	Rules drift. Incentives and boundaries hold	Rule adherence across the corpus is not measured (§5.1); the tracked classes returned in the next session in 48 of 75 intervals, board render 11 Sep 18:57 UTC (§5.2) rule history; recurrence measure	Wang & Huang (2026); Lamparth et al. (2026)	5
P4	An AI grading its own homework isn’t being measured	Load records in 75 of 579 sessions; a logging gap near 27×, about 19× adjusted; four failures of the practice’s own checks event log; pushback census	Ivanov & Africa (2026); Denisov-Blanch et al. (2026); He et al. (2026); Kenton et al. (2026)	6
P5	Depth of record over breadth of feedback	8,221 operator turns over six months, and a 199-exchange key the operator wrote blind export of 26 Aug 2026	None found (§7.2)	7

2.3 The September 2026 context

Between the first of September 2026 and this revision (21 September 2026), the public discourse of the frontier laboratories shifted in a way that bears on this paper's propositions, and the shift is recorded here with dates. On 3 September the chief executive of OpenAI described the coming generation of models as "sobering for everybody" and said that progress would from here be paced by alignment and safety work (Axios, 3 September 2026, interview at the G20 Innovation Ministerial, as carried by The Next Web). On 12 September the same executive called a public offering "ill-timed" given safety concerns and moved it to 2027 (Fortune, 12 September 2026); both laboratories had been reported in May as preparing public offerings this year at estimated valuations of about $1 trillion each (Fortune, 26 May 2026). On 13 September, on CBS's Sunday Morning, the chief executive of Anthropic said that "for too long the industry lied to people about the fact that this technology had risks," called on the industry to slow capability development, and committed his company to permanent access for independent model evaluators (CBS News, 13 September 2026; the web article is stamped as updated 14 September, and CNBC, 13 September 2026, reports the same interview). Semiconductor equities fell on the accumulated statements on 14 September (Reuters, 14 September 2026, as carried by Yahoo Finance). Each of these is listed with its link under Press and public statements.

Two features of that fortnight matter here. First, every statement in it concerns what the models will be: more capable, more dangerous, sooner. None concerns how an organization is to use the models it already has, which is the question this practice exists to study, and which no maker can answer from where it sits: how to use a model is a fact about the deploying organization's work, its costs of error and its standards of correctness, none of which is visible from the laboratory. Second, no party to the September argument disputed the deployment evidence: the capability claims and the risk claims moved markets while the reported failure rate of enterprise deployments (§4) stood unchallenged. We read the fortnight as corroboration, at the industry's own scale, of the distinction between capability and direction that §4 draws, and as an instance of §6's subject: a maker's public statement about its own unreleased model is self-report, authored by the measured party and unverifiable until the model ships. No result in this paper rests on any claim in this subsection.

---

3 P1 The missing weight is correctness

The objectives these systems are trained and deployed under reward an answer the user accepts, and nothing in them prices whether the answer is right. The stop problem is how that absence shows up in behavior: the search ends at defensible. If correctness is the missing weight, then performance gains that do not supply it will not earn trust, and adoption will stall on trust rather than capability. That last step is our inference; the evidence below bears on the premise.

3.1 FROM THE LOGS How much correction the operator supplies

A census of the 8,221-turn frame described in §2 measured how often an operator turn carried corrective signal, a correction, a reframing or a pointed challenge to what the agent had just said. A local model labeled every turn, and the detector was graded against the operator's blind key, 195 rows after the repair of §2.2, at 93% recall and 64% precision. It flagged about 48% of operator turns. Adjusting for the detector's measured precision and recall lowers that to roughly a third, an inference we mark as such (§6.2). The signal is corrective in the broad sense. The operator's own second pass over a stratified sample labels most non-routine turns as bundles of method correction, reframing and teaching, and whether the finer channels are reliably separable from one another is unsettled, because the reading that said they were not has since been traced to a join defect (§10.6). So the reading is that something near half of what the operator typed was spent redirecting an agent rather than routing it, not that half of the agents' answers were wrong. That each such turn followed a premature stop is an inference, not a measurement; on that inference, the agents stopped where their answers were defensible and the operator supplied the rest, by hand.

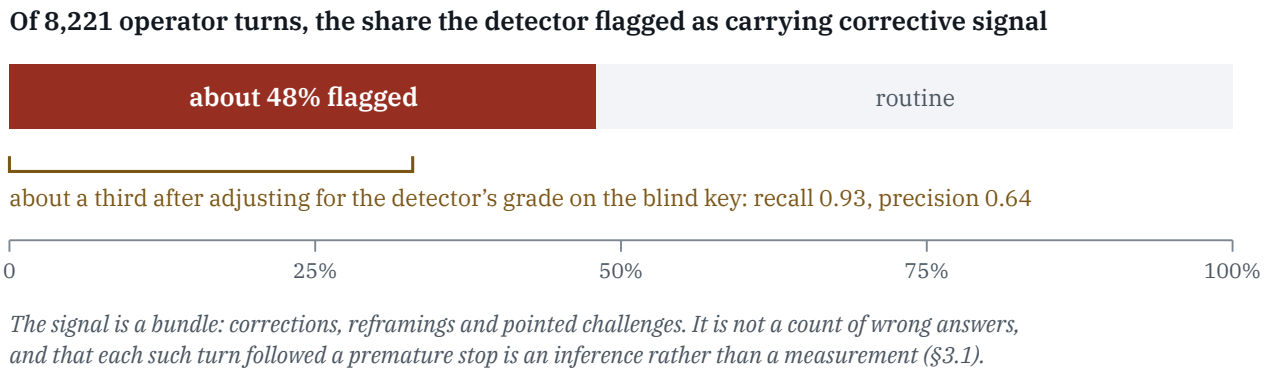

Figure 4. How much of what the operator typed was correction. The detector flagged about 48% of turns; adjusting for its measured precision and recall gives roughly a third (§6.2 gives the ruler). The signal is a bundle of correction, reframing and teaching, and the reading is that near half of the operator's typing was spent redirecting an agent, not that half of the agents' answers were wrong. **Measured.**

3.2 FROM RECENT WORK Premature commitment

Mehta (2026) defines representational commitment as cross-run convergence of hidden states at a fixed agent step and uses it as an early diagnostic of trajectory consistency. On HotpotQA with Llama-3.1-70B, step-four similarity predicts downstream behavioral consistency ($r = -0.35$; partial $r = -0.45$), replicating on Qwen-2.5-72B and Phi-3-14B and on StrategyQA ($r = -0.83$). A runtime monitor flags inconsistent trajectories at AUROC up to 0.97, and 0.85 to 0.88 under a stricter split. The boundary result is the one this program builds on: committed-wrong and committed-correct questions could not be separated in activation similarity. The author reports that as a failure to reject rather than a demonstrated equivalence, and draws the operational conclusion that settled cases should be deferred to an external verifier or a human.

Read beside §3.1, the implication is direct: the settling can be seen, but seeing it does not say whether the settled answer is wrong, and in this practice the test was the operator, by hand, on something near half of the turns.

4 P2 Direction, not capability

> *The operator's proposition, stated as the practice's working claim: "Your AI doesn't need to be smarter. It needs to be led."*

A capable agent without direction behaves like a capable junior team that nobody was assigned to run. In this practice the operator directs the work, approves every consequential change, and converts corrections into structure: a boundary that refuses, rather than a note that asks (§5.1). The counterweight is designed as a projection of that leadership, not a replacement for it (§8.4). This proposition is a reading of one practice and is offered as such.

4.1 FROM THE LOGS Who catches the misses

Table 3 lists five misses from one working window, reconstructed from the session transcript and version-control metadata. They were chosen to illustrate the failure's shape and are not a sample. In each, the asserted state followed from the evidence the agent had gathered, and one further read would have falsified it.

Table 3. Five misses between 10:00 EDT on 4 September and 05:30 EDT on 5 September 2026.

#	What the agent asserted	What the logs showed	Surfaced by
1	An empty handoff from a predecessor session indicated a defect.	The session was the first in a new line and had nothing to inherit.	Operator 4 Sep, 10:08
2	A named hook had closed the still-running session.	The close record names its own author, which was a different mechanism.	The agent, after the operator noted that an automated check had not fired 4 Sep, 10:05
3	Drafts in the operator's voice could be written from five sample paragraphs.	A 57,400-word corpus of the operator's writing was on the same disk. The agent had just read a note recording this same omission three days earlier, acknowledged it, and wrote two more drafts the same way.	Operator 4 Sep, 14:51

#	What the agent asserted	What the logs showed	Surfaced by
4	A rendered document was complete.	One page instead of thirteen, the wrong page size and typeface, at a plausible 23,782 bytes.	The agent's own page count, within half a minute of rendering 5 Sep, 05:26
5	A pull request was open and blocking.	It had merged 49 minutes earlier; the agent had read a status post from an hour before as current.	An automated claim check at the end of the turn 5 Sep, 05:28

Two of the five were surfaced by the operator outright, one by an automated check, one by the agent's own check, and one by the agent only after the operator pointed at a check that had stayed silent. This is a tally of hand-picked cases, not a rate. The practice's error log records only errors that were caught, so it can support a catcher share reported with its n and window, and never a catch rate. What the tally does show is where correction comes from today: in three of the five, it began with the operator. Case 3 is the recurrence of §5.2 in miniature: the lesson was in the logs, the agent read it, and the agent repeated the error within minutes.

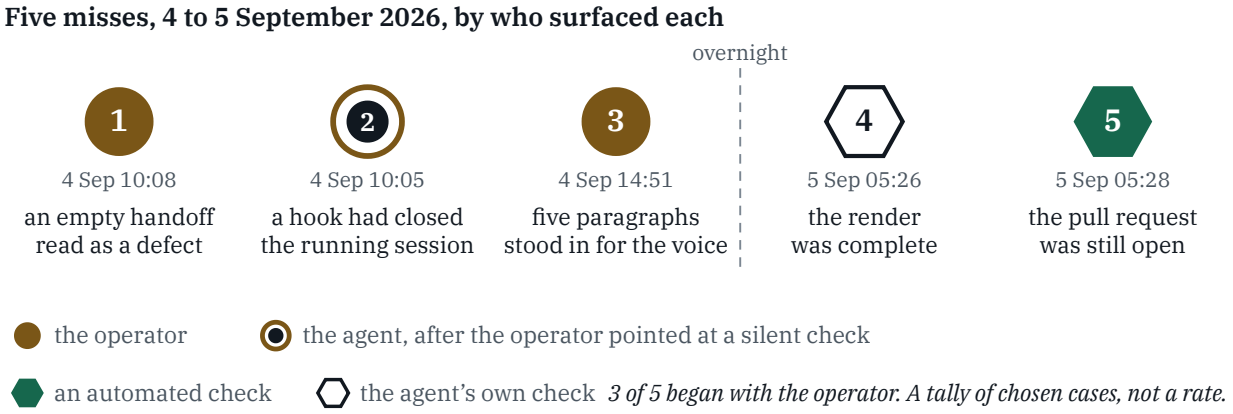

Figure 5. The five misses of Table 3, in table order, marked by who surfaced each. Three of the five began with the operator, one with an automated check at the end of the turn, and one with the agent's own page count. These are hand-picked cases and the tally is not a rate (§4.1). **Indication, not a rate.**

4.2 FROM RECENT WORK Role, not capability

Kelley and Riedl (2026) measure the epistemic effects of personalization across nine frontier models and find them role-dependent. Cast as an advisor, a model challenges the user's framing more often (in advice contexts, acceptance of the framing falls to 26.8%). Cast as a peer, it capitulates: a flip coefficient of $\beta = 0.87$ under persona-grounded rebuttals, with agreement calibrated to the persona's inferred preference for validation ($\beta = 0.57$). Position change under challenge is largely independent of whether the new position is right. A challenge can teach a model to fold as easily as to correct.

Mytsyk, Zhang and Krishnamurthy (2026) attack the same failure from the training side rather than the harness side. Fine-tuning a 3-billion-parameter model (SmolLM3-3B) against a Bayesian truth serum reward, a scoring rule that pays for predicting what others will answer as well as for the answer itself, cut the answer-flip rate under user pressure from 23% to 4% and raised accuracy under that pressure from 80% to 93% on a synthetic set of 1,000 true-or-false questions. Folding is therefore not a fixed property of a model, and something in it is trainable out. Their own conclusion is the half that matters more here: "Our results say nothing about correctness or truthfulness, only about sycophancy." The reward pays for answers that diverge from what others are predicted to say, not for answers that are right, so a model can stop folding and stay wrong. That is H3 (§8.4) stated by authors who held the training lever this program does not. The agents studied here run on a commercial frontier family whose weights the practice does not hold, so the only surface available to it is the harness. That is a constraint on the work, not a judgment that the harness is the better place to intervene.

The same models challenged or folded depending on the role they were given, and the role is chosen by whoever sets up the work, not fixed by the model. We read that as evidence that the gap this proposition names is one of direction rather than capability. It is also why the counterweight is built to speak as an advisor and never as a peer (§8.2).

The proposition also has survey-scale evidence from outside this practice. The MIT NANDA project's industry study, read in full for this revision, reports that despite $30–40 billion of enterprise investment, 95% of organizations are getting zero return from generative AI, against a sample of 300+ public deployments, structured interviews at 52 organizations, and surveys of 153 senior leaders; the report's own attribution is that the divide is "not… driven by model quality or regulation" but "determined by approach" (Challapally, Pease, Raskar & Chari, 2025). That is this section's claim at field scale, from instruments this practice does not operate: the same models that clear capability benchmarks return nothing where nothing directs them, and the September record of §2.3 shows the number standing undisputed while the capability argument moved markets.

---

5 P3 Rules drift. Incentives and boundaries hold

A rule that asks an agent to remember a discipline changes nothing the agent is evaluated on, so it drifts. A boundary that refuses, and records its refusals, changes what is evaluated at the point where it counts. The practice's history shows the difference (§5.1, §5.2), and recent theory, framed in terms of incentives, says it should be expected (§5.3). A memo does not change what is evaluated. A boundary does.

5.1 FROM THE LOGS Rules and boundaries

The practice carries a large corpus of written rules, and adherence across it has never been measured, so this paper reports no rate. The only adherence count in the record is on the practice's boot page, whose own words are: "A rule asks a future session to remember and went 0-for-7 on 2026-07-12." Two

lines later the same page adds, “Seven misses on 2026-07-12. They are not seven failures. They are one act”. That is one day, and seven misses rather than seven rules. The corpus those misses sit against is far larger, and we have counted it. The two rule-bearing surfaces in the required boot set carry 122 headed sections between them and 45 distinctly named disciplines after formats and descriptors are stripped out, among them SHIP-REALITY, GATE-OR-OWN, UNSOURCED-ASSERTION, FETCH-DON’T-RECALL, RUNTIME-SELF-CHECK and USE-WHAT-EXISTS, with further unnamed rules carried as prose headings such as “Push back” and “Do-not-do list”. Beyond the boot set sit a memory store of 124 files, 71 of which carry an explicit “how to apply” directive, and nine further pages titled *Operating Rule* in the workspace tree. The corpus is comfortably past 150. Nobody has measured adherence across it.

What the record does support is a distinction the same page draws, and it is sharper than the number was. Rules delivered one way drift; rules delivered another way fire.

> *“A step in a list is a rule, and rules went 0-for-7. Being first is a structure.”*
>
> The practice’s boot page, 16 July 2026

A rule a seat is supposed to go and read is a wish: it competes for attention with the work, and it loses. A rule injected into the context before the seat’s first token is a property of the environment, and it does not have to win anything. The practice’s memory store is the second kind, and it demonstrably fires. That is the same shape as the boundary-versus-memo claim below, moved one level down: what matters is not whether a discipline is written but whether reading it is optional. We report this as a mechanism the record supports and not as a measured rate, because the measurement does not exist.

Turning to the checks that hold rather than the rules that drift: The checks that did hold share a form: each sits at a boundary and records its refusals. They include a check that blocks known-dangerous edit patterns, a check at the end of a turn for actions promised but not performed, an intake check on incoming work, a freeze window on canonical documents, and an identifier linter. The seven rules are not individually enumerated in the record, which is a gap in this reading; and the count may reflect how these particular rules were placed and how often they fired rather than a law about rules in general (recent work on multi-turn drift measured exactly this: goal reminders injected mid-conversation reduced divergence by 7–12% and judge-scored alignment rose 16–27% across three models, with drift behaving as a bounded equilibrium rather than runaway decay: Dongre et al., 2025, *Drift No More?*, arXiv:2510.07777). The pattern is nonetheless familiar from any organization.

5.2 FROM THE LOGS Recurrence

The program’s outcome proxy is recurrence of named error classes. For each class, the instrument orders sessions, counts a class at most once per session, and takes the gaps between successive sessions in which the class occurs; below a minimum series length it reports the raw series rather than a summary. The reported value is the median of those gaps, in sessions. The program calls this a recurrence half-life, but it is a median inter-arrival time, not the decay constant the name suggests. For the tracked classes the baseline median is 1.0, meaning the typical gap between one appearance and the next is a single session.

The instrument also reports a recurrence count, and that count carries no information and should not be quoted. At the instrument, the recurrence count is the length of the gap list, and the gap list is the differences between consecutive appearances, so the count is always the number of sessions seen minus one. A class seen in 45 sessions will report 44 recurrences whatever its behavior, and a class seen in 500 will report 499. It is not a survival rate and it cannot be one; it is the denominator subtracting from itself. The producing file says as much in its own header: a count of that form for one class cannot serve as the compound measure's denominator.

The quantity to read is one that can vary. Counting how many of the observed intervals sit at the floor of one session gives the share of the time the class came back immediately, and on the program board's render of 11 September at 18:57 UTC the compound of the two tracked classes read 76 sessions seen, a median gap of 1.0, **48 of 75 intervals at the floor**, and a session share of 0.53, 76 of 144 ordered sessions. That is a number the class could move. A separate live query the same evening returned 74 sessions seen, which is the same instrument two readings apart and a reminder that each of these is a photograph rather than a state. The floor share is what §10.4's primary outcome should be read against, and the median that the program ratified as its finish line cannot move at all while most intervals sit on the floor, which is a defect in the finish line rather than in the class. A lengthening median is the intended signal that the class is being addressed; a flat one says it is not.

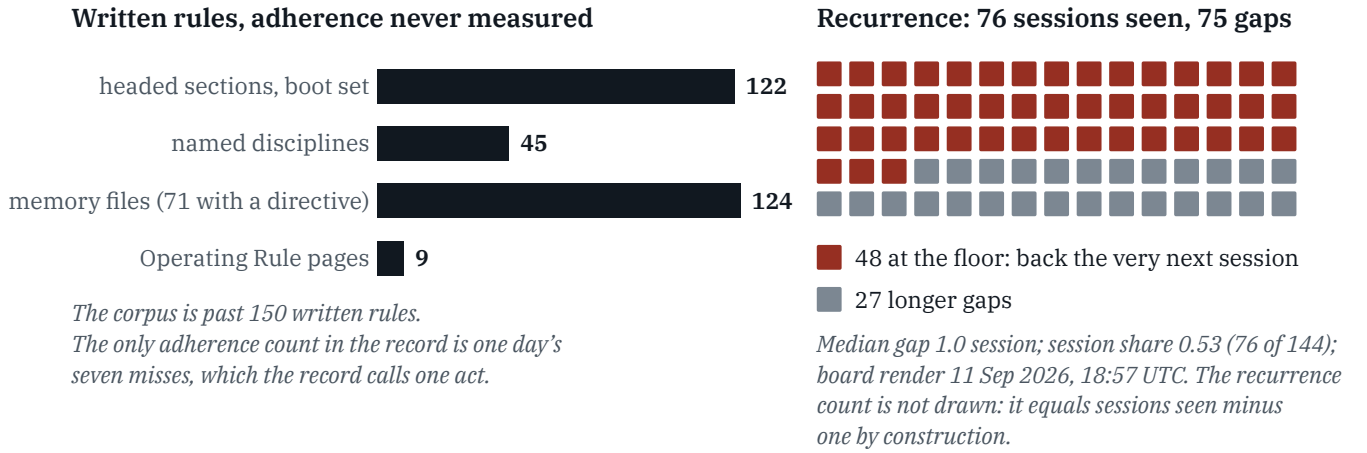

Figure 6. Left: the rule corpus is large and its adherence is unmeasured (§5.1). Right: recurrence of the two tracked error classes, drawn as one square per interval between appearances; a filled square is an interval of one session. Forty-eight of 75 sit at that floor, which is the quantity that can move; the instrument's own recurrence count cannot (§5.2). **Measured.**

A lesson written into the record is a rule of the first kind, and case 3 of Table 3 shows one failing within minutes. The practice's learning instrument makes the same point from the other side: of eight lessons ratified into the merged store, zero are currently measurable for whether behavior changed, because promotion dropped the citation linking a lesson to the miss that bought it, so the entries join to nothing. The instrument reports that as eight unmeasurable rather than as a rate, which is the correct refusal and also an admission that the loop cannot yet tell whether a lesson lands.

5.3 FROM RECENT WORK Incentives under finite evaluation

Wang and Huang (2026) prove that under five minimal axioms (multi-dimensional quality, finite evaluation, effective optimization, finite resources and combinatorial interaction) any optimized agent will systematically under-invest in the quality dimensions its evaluation does not cover, which makes reward hacking a structural equilibrium rather than a correctable bug, independent of the alignment method. They further prove that as a system moves from closed reasoning to tool use, evaluation coverage declines toward zero as the number of tools grows, provided investment in evaluation grows more slowly than the square of the tool count, which they argue is the generic case. Their result gives the practice's experience with written rules (§5.1) a theoretical footing: a rule that nothing evaluates leaves the agent's incentives where they were, while bringing that dimension under evaluation changes them.

The mechanism is one equation. Where an evaluation covers K of N quality dimensions and the agent weights its effort by

$$\tilde{w}_i = \lambda r_i + (1 - \lambda) w_i \quad \text{for a covered dimension } (i \le K)$$

$$\tilde{w}_i = (1 - \lambda) w_i \quad \text{for an uncovered one } (i > K)$$

w_i is what the principal actually values on dimension i, r_i what the evaluation rewards there, and λ the degree to which behavior follows the evaluation rather than the internalized objective. For any $\lambda > 0$ the uncovered dimension carries strictly less effective weight. A written rule changes neither r nor K, so it does not appear in this expression at all; a boundary that refuses changes K.

Lamparth et al. (2026) show that mitigating one reward-model bias, such as reliance on length or sycophancy, can rotate optimization pressure onto correlated proxies rather than remove it, a failure they call reward bias substitution, enabled by the gap between the distribution an audit sees and the distribution a trained policy induces. They demonstrate it live rather than only proving it: a length penalty applied by reinforcement learning to a 3-billion-parameter model cut response length from 204 to 170 tokens exactly as intended and left a knowledge benchmark unchanged, while calibration error rose from 0.25 to 0.41, free-form accuracy fell from 0.56 to 0.42, and the model's confidence-correctness discrimination fell from 0.73 to 0.65. A control run with the penalty switched off kept calibration intact, so the penalty caused the damage. A check fixed on one proxy invites the measured party onto the next.

Together they describe the drift of §5.1 from the other side: pressure on a dimension nothing evaluates goes elsewhere, and pressure that one patch evaluates moves to the next proxy.

6 P4 An AI grading its own homework isn't being measured

When an agent writes the record it is evaluated on, the agent decides what gets evaluated. Observation has to be external, and it has to read the work rather than the report about the work. The practice's own record shows how thin self-report is (§6.1, §6.2); the practice turned the same test on its own instruments and found four of them wanting (§6.3); and recent work finds the same failure in models that know they are being tested and in models grading models (§6.4, §6.5).

6.1 FROM THE LOGS Self-report

Agents in the practice are asked to record, at the start of each session, that they have loaded their context. The count is of two row types in the event log: 75 rows recording that a session loaded its context, against 579 rows recording that a session started, an emission rate of 13.0% measured in early September 2026. The practice's record is useful because most of it is written by machinery as work happens, not because agents report on themselves.

6.2 FROM THE LOGS The logging gap, with its ruler

The 27× figure has been quoted more often than it has been explained, so we give its full provenance. The numerator is the operator-pushback rate from a census of the 8,221-turn frame described in §2. A local seven-billion-parameter model labeled every turn, and the pushback detector was graded against the operator's blind key at 84% recall and 59% precision, and at 93% recall and 64% precision after a documented repair of the key on 31 August 2026. It flagged about 48% of operator turns. The denominator is the rate of corrections the system itself logged: misses recorded in the practice's session-health event log, set against the same operator turns, at 1.76%.

The two ends come from different instruments and different stores, and the numerator is uncorrected for the detector's precision. Correcting the observed flag rate by the detector's measured precision and recall on the blind key, which was drawn systematically from the same frame, puts the rate at roughly a third of operator turns and the ratio near 19.

$$\hat{r} = r_{obs} \times P / R = 0.48 \times 0.64 / 0.93 \approx 0.33$$

$$\text{ratio} = \hat{r} / 0.0176 \approx 19$$

r_{obs} is the detector's flag rate, P and R its precision and recall against the 195-row blind key, and 0.0176 the rate the system logged over the same turns. The correction assumes the key is representative of the frame, which is the assumption §11 records as a threat.

We therefore report the gap as an indication of logging loss, with this ruler attached, and never as a miss rate. Either way the direction holds: the corrective signal is far more abundant than the system's

record of it. That makes a study of the operator's natural corrections viable, and it makes what an agent reports about itself the unreliable minority of the record.

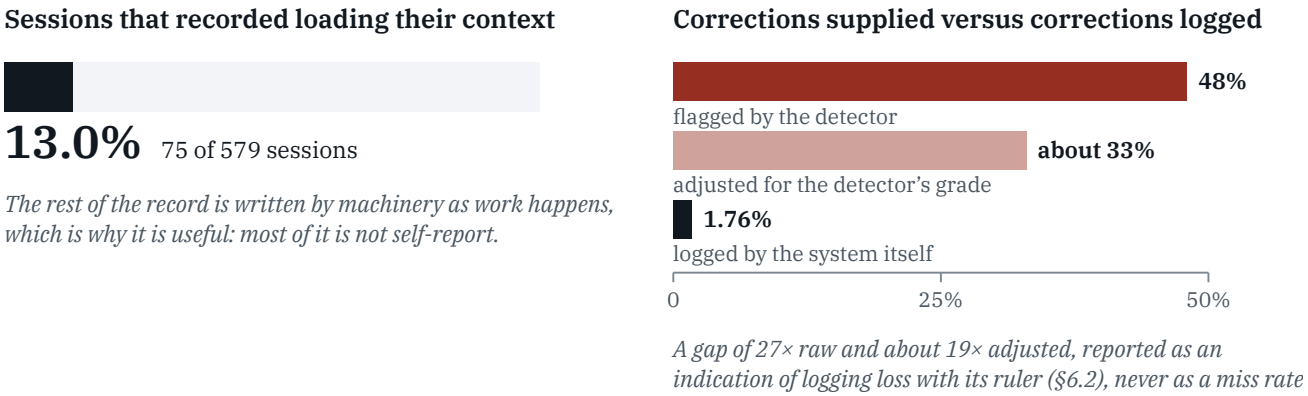

Figure 7. Self-report against the record. Left: agents recorded loading their context in 75 of 579 sessions, 13.0% (§6.1). Right: the operator supplied corrective signal on about 48% of turns, roughly a third after adjustment, while the system logged corrections on 1.76% of the same turns (§6.2). The gap is an indication of logging loss, not a miss rate. **Measured.**

6.3 FROM THE LOGS Four failures of the practice's own checks

The any-read check. A check that runs when an agent ends its turn was built to block claims about system state that no read had grounded. It tested whether any read had occurred during the turn, not whether the read concerned the thing claimed, so an ungrounded claim passed alongside unrelated reads (case 2 of Table 3). The measured party could satisfy the check without changing what it asserted.

The shell count. An export step reported 28 of 28 architecture diagrams rendered. Twenty-five were empty shells: a diagram library named each figure from the clock, a headless browser's virtual time froze the clock, identifiers collided, and every renderer drew into the first figure carrying the shared name. The check counted figure elements rather than drawn content. The repair names each figure explicitly and counts text nodes per view.

The composition failure. For about six days (152 hours as of 11 September 2026) the practice's public metrics page did not update. The host slept from 5 to 11 September; on waking, a catch-up run fired outside its permitted window; a freeze check correctly refused the resulting commit; the calling step logged the failure and exited with success; and a downstream check correctly declined to publish from an uncommitted tree. Each component behaved as specified, and the system as a whole did not.

The assembled engine. Four joins between the modules of the challenge engine and its auditing harness parsed cleanly and measured nothing. An independent check on cited claims could not open the documents it was meant to read, so every cited claim came back unreadable; every auditor verdict would have been recorded as a refusal, inflating the count of counterweight firings; retrieved rules would have been recorded without their labels; and the challenger would have loaded a second copy of a model the running service already held, overrunning the graphics card's memory. The full test suite passed with all four in place. An audit of the joins themselves caught them, one only by running the

assembled modules in their real environment, and all four were fixed, with tests that fail against the old wiring, before the engine first ran.

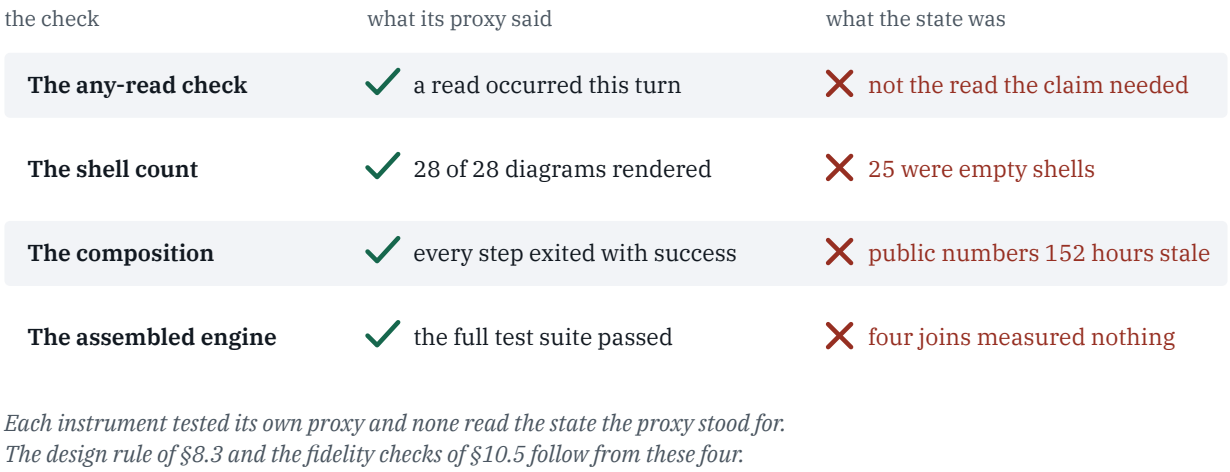

Figure 8. The four failures of §6.3, side by side: in each, the check's own proxy was satisfied and the state it stood for was not. This is the stop problem's institutional counterpart, and it is why the counterweight's clauses are enforced by construction where they can be (§8.3). **Measured.**

These are the stop problem's institutional counterpart. Each instrument reported success when its own proxy was satisfied, and none checked the state the proxy stood for. They motivate the design rule in §8.3 and the fidelity checks in §10.5.

The checks that do fire have now been joined to the operator's corrections for the same window, and the join is unflattering. Table 4 gives it. Of 93 refusals raised across 674 guard invocations, 2 coincided with a correction the operator went on to make and 91 did not; 14 corrections arrived on turns where a guard was live and silent. The instrument declines to turn these into rates, because one seat's correction channel was empty for the window and a rate computed over a dry channel would be a number about logging rather than about guards. Counts are therefore reported and rates withheld. Even as counts, the reading is the one clause (b) exists to prevent: a check can fire often, refuse confidently, and discriminate barely at all.

Table 4. Guard firings joined to operator corrections, September 2026 window. Rates are withheld by the instrument: one seat's correction channel was dry.

Quantity	Count	What it is
Guard invocations	674	Occasions a boundary check ran
Refusals raised	93	The check fired and blocked or flagged
Operator overrides	0	Refusals the operator reversed
Refusals coinciding with a correction	2	The firing and a real miss lined up

Quantity	Count	What it is
Refusals not coinciding with one	91	Fired where no correction followed
Corrections on a turn where a guard was live and silent	14	The miss the check was there to catch
Operator corrections in the window	409	Of which 16 fell inside a guard's exposure

The activation null. One further attempt belongs here because it failed. Mehta's result (§3.2) reads commitment off hidden states, so the practice tried the mirror image: whether a correction-carrying operator turn is separable from a routine one in the activations of a local 7-billion-parameter model. Over 179 turns, 96 of them corrections, at a middle layer, the separation index did not beat a permutation null ($p = 0.85$), and it still did not after de-duplication or length normalization. One caveat travels with that figure: a later re-check file in the same directory, attempting to reproduce the published raw run under the same label, returns $p = 0.95$ rather than 0.85. A second and larger caveat arrived with the join defect described in §10.6: the labels this null was computed against are the same key, and under the repaired join 71 of the 176 turns change label. The activation analysis has not been re-run on the repaired labels, so the null reported here is a null against a partly misjoined key, and it should be read as unreproduced until it is. The conclusion is unchanged, since both are emphatic nulls, but the two on-disk readings of the same run disagree, so the pipeline behind this particular number is not exactly reproducible and we report the figure with that attached. The dominant axis of that space was turn length: the first principal component held 99.9% of the variance and its highest-norm rows were the operator's shortest turns, "go", "try again", "check slack". The honest reading is narrow, since a null at this n bounds the effect loosely rather than establishing absence, and last-token pooling at one layer is a weak probe. But it is worth recording that the practice looked for its own signature and did not find one.

6.4 FROM RECENT WORK Tested behavior is not deployed behavior

Ivanov and Africa (2026) construct deployment-like evaluations by replaying realistic agentic trajectories and appending the evaluation prompt at the end. Moving from a standard benchmark to replayed usage reorders model rankings (Spearman $\rho = -0.56$, $n = 10$, $p = 0.09$; $\rho = -0.88$ with one outlier removed), and one model that capitulated on 2% of challenges under the benchmark capitulated on 46% under replayed usage. Whether a model can tell it is being tested changes what it does, which matters for any intervention delivered inside the model's own context.

6.5 FROM RECENT WORK Models grading models

Denisov-Blanch et al. (2026) show that scaling inference by aggregation does not deliver truthfulness gains where no external verifier exists: agreement between models reaches $\kappa \approx 0.35$ even on random strings with no ground truth, and 53% of multi-model mathematical errors converge on the same wrong answer. He et al. (2026) find that in a three-vendor heterogeneous panel the minority was still correct in 25.5% of divergent cases; adding a fourth model as arbiter was net-negative (−1.37%, flip

precision 42.7%), while a non-model classifier over features of how the debate behaved was net-positive (+1.71%, precision 81.2%). Redundancy among models does not supply independence, and the body that decides has to be of a different kind from the bodies it decides between.

Kenton et al. (2026) find that debate training reduces reward hacking under reinforcement learning from AI feedback: it holds judge correlation flat and recovers 45% of the performance gap. They also report that, absent restrictions on the debaters, “hacking the judge is probably the default result”, and that under a weakened judge the critic routinely fabricated direct quotes. A challenger that is free to invent its evidence will invent it; this program’s response is to require every challenge to name a source the challenged agent can read.

---

7 P5 Depth of record over breadth of feedback

> *The operator’s proposition: “The data the frontier can’t buy.”*

The developers of frontier models collect feedback at a breadth no single practice can match: preference signals from users who mostly do not own the outcome and often lack the context. What that breadth cannot supply is depth, and depth is what a single operator’s record has.

7.1 FROM THE LOGS What one operator’s record holds

The record described in §2 has what a broad collection lacks: one operator; the full surrounding record of what the agent read and held; ownership of the consequences; a continuous identity over six months; and corrections that can be adjudicated against a key the operator wrote blind. Such a record can say that an answer was wrong, why, what the model had read, and what happened next. It is a case record and a method, not a population, and it is private (see Data availability).

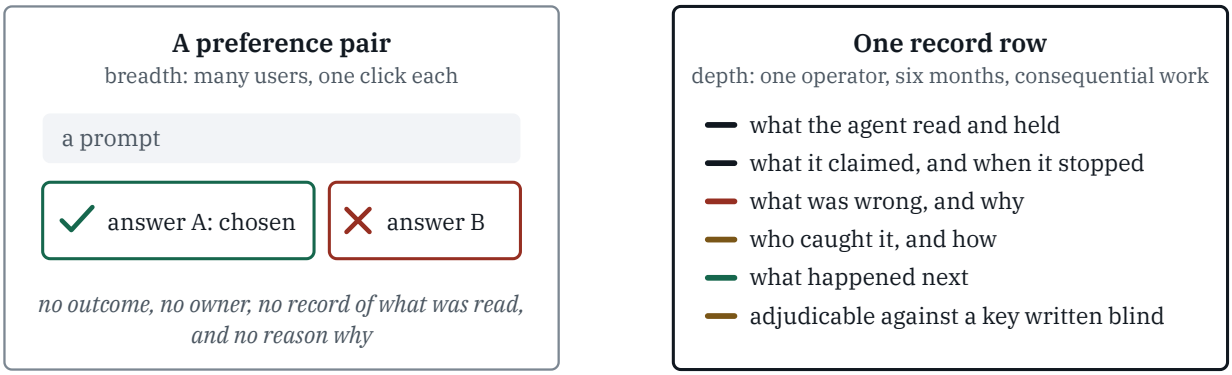

Figure 9. Breadth against depth. A preference pair records a click; a record row from this practice records what the agent read, what it claimed, what was wrong and why, who caught it, and what happened next, adjudicable against a key the operator wrote blind (§7.1). It is a case record and a method, not a population. **Schematic.**

7.2 FROM RECENT WORK The gap

The oversight studies in §2.1 recruit reviewers for a task built for the study, on a study platform, over weeks. Buser et al. is the closest to live practice, seeding threats into a working screening queue and scoring per named screener, and even it is a constructed detection task with recruited staff. The 2026 agent-stopping literature of §2.1 measures the phenomenon far more precisely than this practice can, but it measures it on benchmarks: HotpotQA, SWE-bench, WebShop, constructed multi-constraint queries, with correctness supplied by the benchmark. This paper claims no priority. A novelty claim is an absence claim, an absence claim is a report about a search, and a search stops when it has enough for a defensible related-work section, which is this paper's thesis applied to its own bibliography. We do not make one.

What can be said without a search behind it is what the record contains. One operator. A live team of agents on consequential work. Six continuous months. Corrections labeled by the person who owns the outcome, adjudicable against a key that operator wrote blind. Every method used to read it, from seeded faults to per-individual scoring to pre-registration to live-queue measurement to evidence-gated commitment, is established elsewhere and in several cases better evidenced elsewhere (§2.1). The cost of the setting is the same fact as its value: ecological validity and longitudinal depth, at N = 1. Readers should assume that any mechanism described here has a better-evidenced counterpart in the literature of §2.1, and should treat this paper as a field record rather than as a claim to have been first at anything.

8 The counterweight

A counterweight is a challenge-producing mechanism aimed at the stopping moment. Figure 1 (§1) places it and Table 5 specifies it. Its challenge engine is implemented and not deployed (§8.5); nothing

in this section reports a measurement, and §9 reports the first.

8.1 Specification

Table 5. The counterweight's clauses, their grounds, and their standing.

Clause	Requirement	Grounds	Standing
(a)	External to the agent's own output stream.	Agreement is not verification (Denisov-Blanch et al., 2026); a non-model arbiter outperformed a model arbiter (He et al., 2026); settled cases go outward (Mehta, 2026).	Specified; enforced in the engine
(b)	Discriminates on actual error, not on the agent's confidence.	Commitment does not track correctness (Mehta, 2026); challenge moves positions regardless of correctness (Kelley & Riedl, 2026).	Specified; the first calibration failed it (§9)
(c)	Delivered at the stopping moment, at a rate above the ambient error rate.	Timing: commitment forms early and is localized in time (Mehta, 2026). Rate: untested; the dose check in §10.5 measures it.	Timing specified; rate is a hypothesis
(d) not ratified	Indistinguishable from ordinary work: the agent cannot tell the challenge is an instrument. The program's own term is *measured indistinguishability*, and it is a standing proposal awaiting the operator's decision rather than an adopted clause. The ratified specification has three clauses, not four.	Evaluation awareness changes behavior (Ivanov & Africa, 2026). Its cost: a wrong challenge would carry the operator's authority, which the precision check of §10.5 exists to bound.	Proposed

8.2 Constraints on form

- **Advisor, never peer.** The peer frame is the one measured to produce capitulation (Kelley & Riedl, 2026).
- **Evidence-shaped, never persona-voiced.** A challenge reads "the log contradicts your claim at this line," not "as the operator, I feel." An evidence-shaped challenge can be checked by a third party; a persona-voiced one invites folding.
- **Cited or refused.** Every challenge names a source the agent can read, because an unconstrained critic invents its evidence (Kenton et al., 2026).
- **The decider is an instrument anchored on the logs, not a further opinion** (He et al., 2026; Denisov-Blanch et al., 2026).
- **Bounded and curated.** A small, selected set of challenges rather than a large one.

Two of the clauses pull against each other, and the program has not resolved it. Clause (c) asks the counterweight to fire at a rate above the ambient error rate; clause (d) asks it to be indistinguishable from ordinary work. A challenge that arrives often enough to satisfy the first is, by construction, a recurring artifact in the context, which is what the second forbids. Ivanov and Africa (2026) show the cost of losing clause (d) is not marginal: one model capitulated on 2% of challenges it could tell were a test and 46% of the same challenges delivered as ordinary usage. No surface in the program had named this trade before this paper; it is recorded here as an open design conflict rather than a solved one, and the dose check of §10.5 is where it becomes measurable. It is also, as far as we can establish, untested by anyone. We read sixteen papers in the adjacent certificate, termination and early-abort literatures in full, and none builds and measures a challenge that the challenged agent cannot distinguish from ordinary work. Two come near it and each falls short in an instructive way: one controls for leakage in how it phrases its probes but applies them to finished runs rather than live ones, and asserts indistinguishability by construction without ablating it; and one gates each lifecycle transition on evidence rather than injecting a separate check, which we read as avoiding a distinguishable test moment rather than measuring indistinguishability, though that reading is ours and the paper does not claim it. Clause (d) is therefore both unratified inside this program and unmeasured outside it, which is a reason to treat it as a research question rather than as a specification.

8.3 A rule for the practice's own instruments

The failures in §6.3 led the practice to adopt a design rule in September 2026: an instrument the measured party operates will rot; an instrument that operates on them will not. Its test is a single question: can the measured party change the reading without changing the world? The rule restates, for a practice run by AI agents, a principle long familiar in the social sciences as Goodhart's and Campbell's laws. Its closest contemporary analogues in model training are reward bias substitution (Lamparth et al., 2026) and the equilibrium result of Wang and Huang (2026).

The rule as first written was too coarse, and a survey of what production harnesses actually do shows where it breaks. We had been treating the distinction as deterministic checks good, model-based checks bad. That is not the variable. AutoGen's termination check is deterministic and sits at the harness boundary, and it still fails, because what it deterministically matches is a sentinel string the agent itself emitted. The check is rigorous about a claim the claimant authored. Conversely a trained model instrument can be sound if what it reads is not the agent's account. **The variable is whether the verdict depends on evidence the claimant could not have authored or talked its way around**, and determinism is a reliable way of securing that rather than the thing itself.

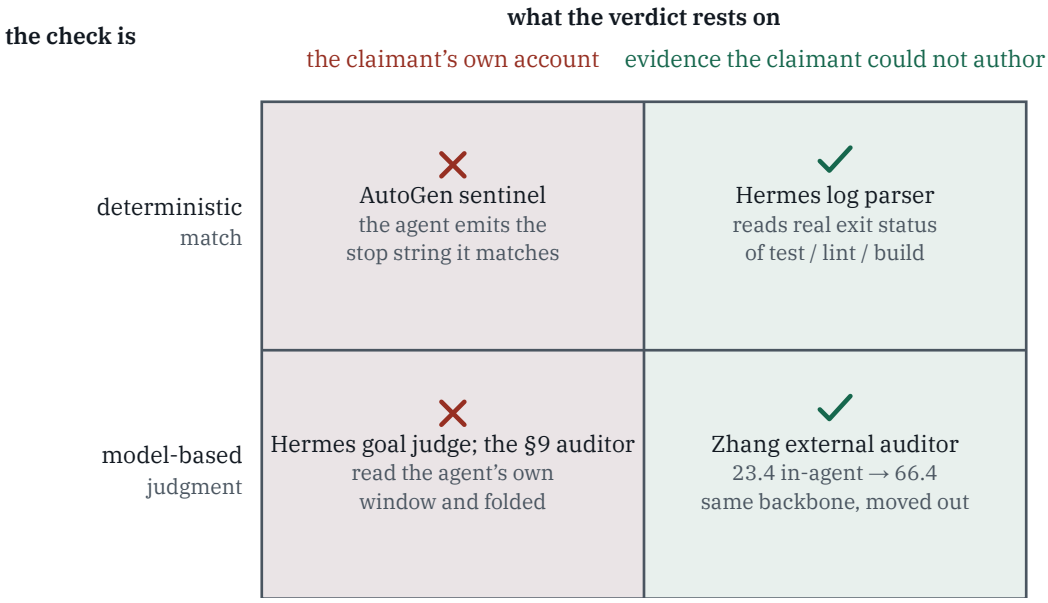

Figure 10. The design rule of §8.3 as a matrix. What separates a check that rots from one that holds is the column, not the row: whether the verdict rests on evidence the claimant could not have authored. A deterministic check can still rot when what it matches is a string the agent emitted (AutoGen), and a model-based check can hold when it reads a channel outside the agent (Zhang's external auditor, 66.4 against 23.4 in-agent on the same backbone). Determinism is a reliable way to secure externality, not externality itself. **Schematic; Zhang's two points measured.**

Two recent results put numbers on both halves of that rule. On externality, Zhang et al. (2026) hold the model fixed and vary only where the check sits: an in-agent self-reflection step over a seven-billion-parameter backbone scores 23.4 on their failure-detection measure, while an externally trained auditor over the same backbone scores 66.4, a gain of roughly 2.8 times bought purely by moving the check outside the agent. The same work defines a deployable operating region, a false-alarm rate at or below 20% with step accuracy at or above 50%, and reports that of ten auditors compared only its own small trained one falls inside it, at a 2.4% false-alarm rate. The strongest frontier model tested as an external judge reaches the accuracy half of that bar and misses the other by a wide margin, with a false-alarm rate of 43.2%. Capability in the auditor is not the lever; position is, and an auditor that cries wolf twenty times as often as a smaller trained one is not usable whatever its accuracy. On grounding, Flynt (2026) reports a case in which two frontier judges scored a plausible agent response at 0.85 and higher, while a deterministic check of the recorded trace found that the agent had never retrieved the artifact its answer depended on, scoring it 0.000. His framework scores the trajectory alongside the answer and targets three failures he argues judge-based evaluation cannot detect by construction, among them whether an agent checked before claiming an absence. He reports that his case studies suggest this failure mode is common rather than exceptional. That is this paper's subject reached independently, by someone who built the deterministic check we have so far only specified.

And here is the result that threatens this paper's own mechanism most directly. Advani (2026) studies exactly the failure named here, an agent asserting completion while the environment shows otherwise, across 9,876 trajectories from eight model families on one benchmark and 1,879 from four

families on a second whose ground truth is read from database state rather than from text. The rate varies enormously by setting and we report the range rather than the worst of it: 45 to 48% of failures in one family of domains, 3% in another, and 75.8% among self-assessing coding agents that make an explicit status claim. The finding that matters is what happened when LLM judges were asked to catch it. No configuration across five judge models and five prompt strategies exceeded an AUROC of 0.65, even when the judges were handed the full task specification, and the same judges reached 0.54 on the benchmark whose traces are structured API calls rather than prose. The stated mechanism is the one this paper should fear: judges rely on surface completion proxies, confident closing language in one benchmark and coarse action-sequence volume in the other, rather than on verified state changes. A lightweight statistical detector over surface trajectory features reached 0.83 and 0.95 on the two benchmarks and recovered four to eight times more false successes than the best judge at the same flag rate, at a three-thousandth of the latency. The paper's own recommendation is that production monitoring use such detectors as triage rather than LLM judges as the primary monitor.

Read against §9, that is not a coincidence but the same result twice. Our auditor folded on claims whose contradicting evidence sat in its window; these judges scored trajectories on how confidently they closed. A challenger that reads an agent's account and forms a view is anchored by the very property this paper calls defensibility, so being external is necessary and nowhere near sufficient. Two further results constrain the design in the same direction. Panickssery, Bowman and Feng (2024) show that a model's ability to recognize its own output is linearly related to how much it favors that output, that training the recognition up strengthens the favoritism, and that the relationship survives the obvious confounders. Pan, He, Bowman and Feng (2024) show that when a generator and an evaluator share an underlying model, reward hacking appears spontaneously in context with no gradient update at all, and that its severity tracks model size and how much context the two share. Together these say a challenger should not be drawn from the same family as the agent it challenges, and should not be run as an iterative exchange in a shared context. This program's build satisfies both, with a challenger and an auditor from different open-weight families and a single-shot challenge, and it satisfied them by instinct rather than by argument until now.

Two further results bound what any version of this design can promise. Wan et al. (2026) build the closest published relative of the counterweight we have found: a rubric-guided verifier that evaluates an agent's answer and returns feedback the agent then refines against, scaled at inference time rather than trained in. That a mechanism of this shape exists, is published at a main venue, and works is another reason §7.2 claims no priority. And Wang et al. (2026) state the limit that applies to all of it. Characterizing verification along three dimensions, scalability, faithfulness and robustness, they argue that achieving all three at once is the central unsolved problem, and conclude that no fixed reward function can remain effective as policy capability continues to grow, so verification must co-evolve with the generator.

That last point changes what this program should claim. A counterweight is not a gate that can be specified once and left standing, because the thing it constrains improves and the constraint does not. The honest framing is a point-in-time intervention whose calibration decays, which makes the re-calibration schedule part of the design rather than maintenance, and which means a null result three

months from now would not distinguish a mechanism that never worked from one that was overtaken. Nothing in §10 currently measures that decay, and it should.

One open-source harness makes the distinction visible by shipping both answers at once. Nous Research's Hermes agent has a stop-time verification path that parses the terminal log for real test, lint and build invocations and records their actual exit status, which is evidence causally downstream of the world rather than of the agent's narrative. It also has a standing-goal judge that calls an auxiliary model with the goal text and roughly the last four kilobytes of the agent's own final response, with no tool access and, by default, the same model as the agent. The project's own issue tracker records the predictable failure: an agent reported writing a file, the write silently failed, and the judge marked the goal complete. The release carrying this work is announced with the line that done means proven rather than claimed. Half of it is; the other half is the stop problem with a second model attached, and it is the half that looks most like verification. We take these details from the project's public repository, configuration and issue tracker, and they are current as of September 2026 rather than permanent (Nous Research, 2026).

That failure is independent corroboration of §9 from a different team on different code, and it sharpens what §9 measured. Our auditor was not weak; it was reading the wrong channel. It saw a claim and an evidence window, and the window was assembled from the same stream that produced the claim.

Park and Choi (2026) demonstrate that this is the operative variable, in the only way that settles it: they hold the agent and its tools fixed and manipulate nothing but the information channel available to the evaluator that gates the loop, against a world-state oracle they isolate by container and network so it cannot be faked. They call the failure the progress mirage. Across 54 cycles their agent claimed improvement every single time, while 56% of those cycles had a measured change of zero or worse, so self-report carried no information at all and the self-graded gate degenerated into accepting everything, eroding the best state the system had actually reached by 19%. Their preregistered adversarial hypothesis was that a strong enough judge closes this gap. It was rejected: the strongest in-band judge, given the full artifact text, the change diff and its own verdict history, still accepted cycles of which 44% were real regressions while rejecting 38% of real improvements. On a task whose success can be checked from the artifact itself, the same judge's mirage fell to zero. A variant returning only an accept-or-reject bit and no explanatory content performed about as well as full feedback, which places the benefit in the gate's grounding rather than in anything the judge said. Their conclusion is the design rule stated as a structural claim: where the success signal lives outside the transcript, scaling the judge is not enough.

It is worth recording what the default is, checked in September 2026, because it is lower than the discussion above implies. In the OpenAI Agents SDK, an agent run ends when the model emits a turn containing no tool calls; with no output type configured, any text at all satisfies the condition (OpenAI, 2026a). The framework does ship a human-in-the-loop approval primitive, and it gates tool calls rather than completion claims, so a developer can require a person to approve a refund before it is issued and has nothing available to require a person to confirm the task was actually done (OpenAI, 2026b). The predecessor framework ended a run on the same no-more-tool-calls condition with no turn limit at all (OpenAI, 2024). The pattern repeats across the frameworks we checked. In the Claude Agent SDK the

loop likewise ends when the model returns a response containing no tool calls, with no turn cap and no budget cap set by default, and the result carries the subtype *success*, which is a statement that the loop terminated without error rather than a claim about the answer (Anthropic, 2026a). In CrewAI the completion test is that the literal string “Final Answer” appears in the agent’s own generated text (CrewAI, 2026a), and a guardrail specified as a string is executed by the acting agent’s own model (CrewAI, 2026b). Each of these frameworks offers a real gate, and in each case it is opt-in and empty until a developer fills it.

Two details from that survey are worth stating on their own, because they come from the vendors rather than from us. Claude Code, whose agent loop the Claude Agent SDK embeds, ships a built-in completion condition in which a separate small model checks after each turn whether a stated goal has been met, and its documentation says of that evaluator that “it does not call tools, so it can only judge what Claude has already surfaced in the conversation” (Anthropic, 2026b). That is an accurate description of the limit this paper is about, published by the party with the most incentive to describe it favorably. The same vendor’s guidance on building agents says of having one model judge another that “this is generally not a very robust method” (Anthropic, 2025). Meanwhile the human-in-the-loop primitives in the two SDKs gate *tool calls* and not completion claims: a developer can require a person to approve an irreversible action before it is taken, and has nothing built in to require a person to confirm that the finished work was actually correct. CrewAI is the exception among the three: a task can be set to have a human review the agent’s final answer, and like every other gate here that setting is off by default (CrewAI, 2026b). Outside that one opt-in, the approval surface exists for the act and not for the claim, which is the asymmetry the counterweight is aimed at. This is not a criticism of those libraries, which are explicit about what they are; it is the baseline against which every mechanism in this paper should be read. The common case is not a weak check. It is no check, and a stop the agent declares for itself.

Applied to the counterweight, the rule forbids a self-reported exit: an agent may not certify its own completion. The design answer is a certificate gate, in which the agent’s end-of-turn hook emits typed claims with evidence pointers, a deterministic verifier with no model in the loop checks each pointer, and a refusal names the pointer that failed. The gate is designed and not yet implemented here. A gate of this shape has been built and measured elsewhere: Liu (2026) lets an agent return a completion only when a typed certificate binds every required claim to in-scope trace evidence and a deterministic replay reconstructs the claimed value, and tests it against critic-based termination on controlled faults. The checks that currently run at the end of an agent’s turn are a liveness heartbeat, a check for promised actions not performed, the claim check of case 5, and a check for open verifications.

8.4 Hypotheses

The propositions of §3 to §7 describe the problem. The hypotheses below describe the mechanism built against it, and the ways that mechanism can fail.

H0, the outcome hypothesis. Counterweights delivered at the stopping moment measurably raise agent correctness. H1 to H3 are the mechanism H0 rests on and the two ways that mechanism can fail; stating H0 alone would leave a null unable to say which link broke.

H1, the counterweight is a projection of the operator's judgment. The agents working under the harness generate a corpus of the operator's corrections in the course of ordinary work; a nightly consolidation of the day's corrections produces an updated projection of that judgment, and the projection, delivered into an agent's context at the stopping moment, is the counterweight. Steering a model whose weights are fixed by way of a smaller adapted one requires access to logits and therefore open weights, which makes a review of model terms of use a gate on the training path.

H2, a control problem rather than a voting problem. The nightly consolidation is a slow controller, re-balanced from outcome data; the in-context challenge is a fast controller acting within a turn. Whether the closed loop converges to correct, oscillates, or teaches the agent to fold is unmeasured, and it is the question that can end the architecture.

H3, a reversal is not a correction. Because challenge moves positions largely independent of correctness (Kelley & Riedl, 2026), an agent changing position after a challenge is equally consistent with the counterweight having taught it to fold. H0 is supported only if reversals are corrections, measured against an independent correctness signal. A counterweight that raises reversals while leaving correctness flat is a negative result, not a partial success. That folding and correctness can be pulled apart is not hypothetical: Mytsyk et al. (2026) move flip rate and accuracy under pressure in opposite directions by training (§4.2), which is the separation H3 asks this program to measure without being able to train.

H4, and it is the one most likely to sink this. A counterweight can satisfy clause (b) completely and still make things worse. Vasudev, Russak, Bikel and Alshikh (2026) run exactly this test: a critic with an offline AUROC of 0.94, used to intervene on agent trajectories, caused a 26-point collapse in task success on one model while affecting another by close to zero. Their diagnosis is a disruption-recovery tradeoff. An intervention can rescue a failing trajectory, but it can also derail one that was going to succeed, and which effect dominates depends on the base rate of failure rather than on how good the critic is. Intervention degraded performance on their high-success benchmarks and helped only on the high-failure one, by 2.8 points. Their practical proposal is a fifty-task pilot run before deployment to predict whether intervening will help at all, and they describe the framework's main value as identifying when *not* to intervene.

This program has no answer to that yet, and it should not pretend otherwise. Every clause in Table 5 constrains what a challenge must be; none of them constrains how often the challenged agent was going to be right anyway. If the practice's agents succeed on most turns, this result predicts that a well-aimed counterweight will still cost more than it returns, and the dose requirement of clause (c) would make that worse rather than better by firing more often. The fidelity checks of §10.5 measure whether a challenge was delivered as specified; they do not measure what the challenge broke. Adding a disruption measure to the interventional arm, and running a pilot of the kind Vasudev et al. describe before opening it, are the two changes this finding forces.

A second hazard is sharper still, because it is aimed at sophistication rather than at timing. Mittapalli et al. (2026) build an evidence-aggregating monitor that reasons across an agent's steps and weighs the evidence against the agent's stated justifications. Against the evasion strategies that hide evidence it

dominates every baseline. Against one strategy, in which the agent takes its real actions in the open and supplies plausible reasons for each, it detected 0% of the adversarial trajectories while a simpler monitor that reasoned step by step and did not aggregate detected 58.8% of them. The authors' own diagnosis is quoted here because it is the whole finding: that strategy leaves no traceable evidential footprint, so the agent constructs plausible justifications for each action, and the aggregating loop, "designed to weigh evidence against stated justifications, is susceptible to accepting them." The same paper reports its method trailing simpler baselines on several individual tasks where the malicious signal concentrates in a few actions. The lesson for a counterweight is unwelcome and specific: the machinery that lets a challenger read a whole trajectory and reason about it is the same machinery a good explanation exploits, and a more capable challenger can be strictly worse than a blunter one. This program's challenger is built to quote a record rather than to weigh a justification, which is the right side of that line by construction, but nothing currently measures whether it stays there under an agent that argues well.

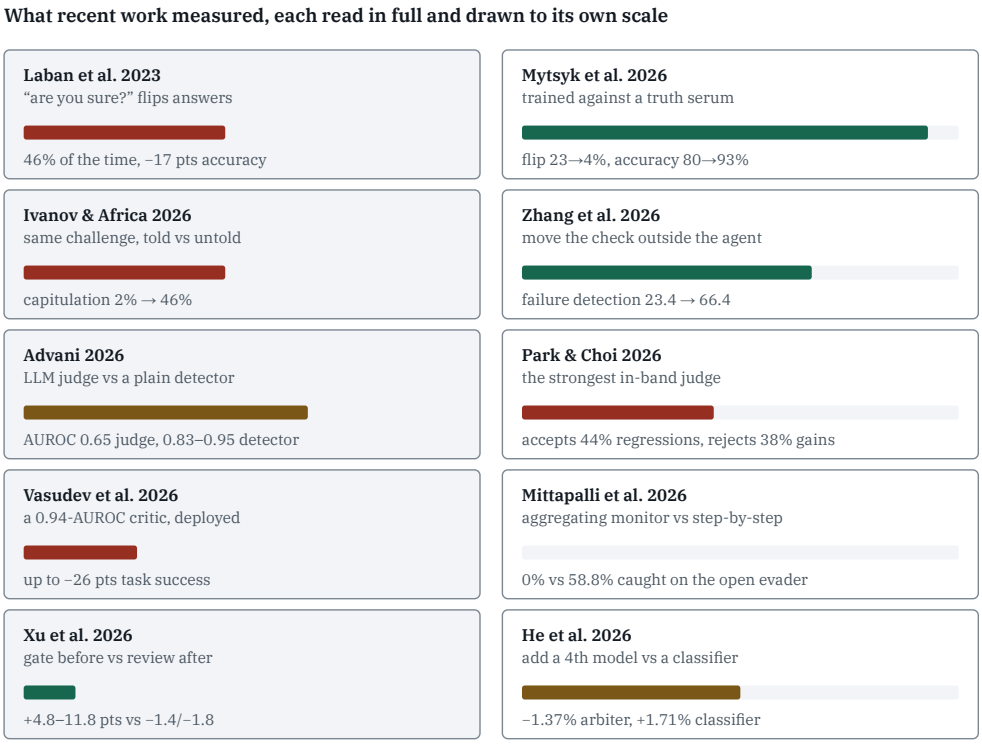

Figure 11. The recent-work results the counterweight is built on and against, each read in full and each on its own scale (green where the result helps this design, red where it warns, bronze where it splits). Together they say: challenge moves positions independent of correctness, folding is trainable rather than fixed, the check must sit outside the agent, and a more capable in-band judge is not a safer one. Sourced across §3.2, §4.2, §5.3, §6.4, §6.5, §8.3 and §8.4. **Measured; all read in full.**

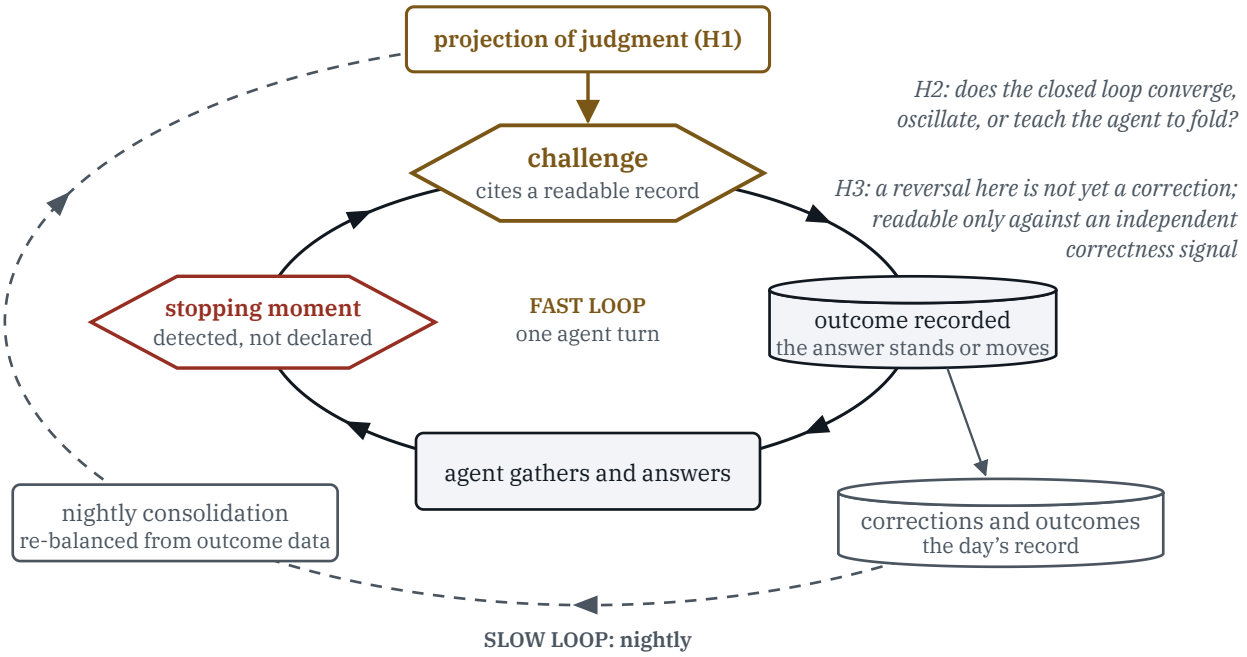

Figure 12. The architecture of H1 to H3 as two nested control loops. The fast loop runs within one agent turn: the stopping moment is detected by the harness rather than declared by the agent, the challenge cites a readable record, and the outcome is recorded. The slow loop runs nightly, consolidating the day's corrections and outcomes into the projection of judgment the next challenge is drawn from (H1). H2 asks what the closed loop does; H3 sits on the return path, where a position change is equally consistent with correction and with folding. Schematic: the engine is implemented but not deployed, and no arrow here has yet been measured end to end. **Schematic.**

An alternative design, three models checking one another, is not adopted. Static panels of models do not decorrelate (§6.5), so any claim of independence among checkers has to be measured rather than assumed.

8.5 Implementation

The challenge engine was implemented and merged on 11 September 2026. It has three parts: retrieval over the operator's corrections and decisions, in which every hit carries the address of its record and an item without one never enters the index; an evidence contract under which a challenge without cited evidence cannot be constructed at all; and a challenger that writes only from retrieved evidence, behind a lint that refuses any challenge that does not quote its own record verbatim, cites a record it was not given, or speaks in a persona. Clause (a), the evidence-shaped constraint and the cited-or-refused constraint are therefore enforced by construction rather than by instruction, which is the rule of §8.3 applied to the counterweight itself. By design, the model that writes a challenge is not the model that judges one: in the first build the challenger is Qwen and the auditor is OLMo, both open-weight and running on the practice's own hardware. The engine is not deployed; §9 explains why.

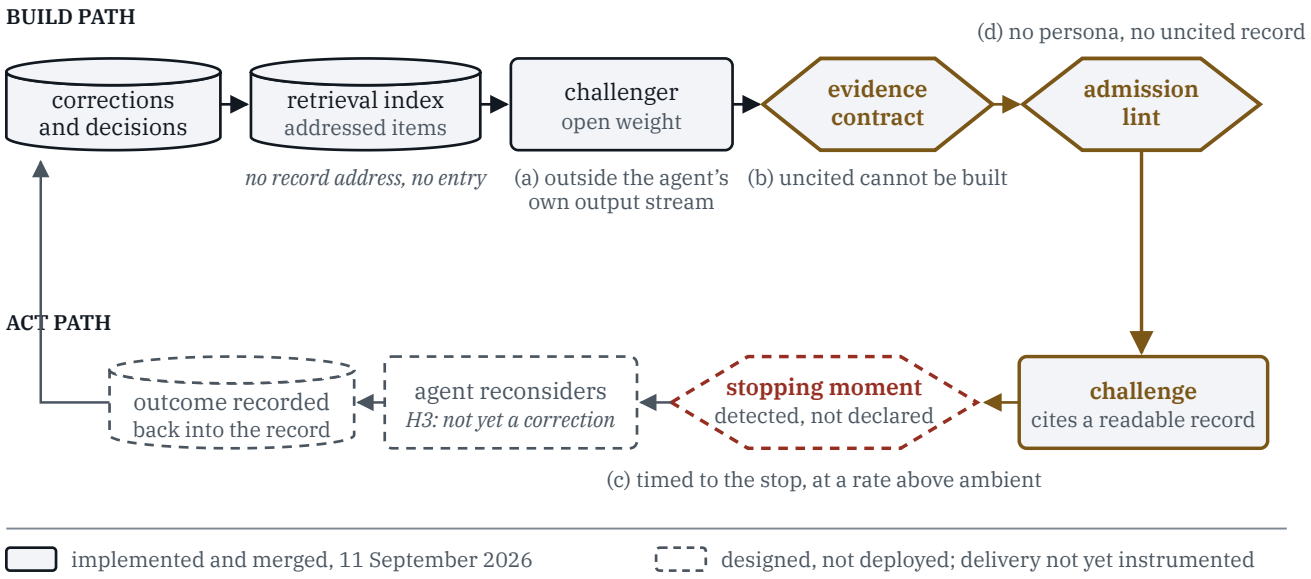

Figure 13. The counterweight engine end to end, with each clause of Table 5 drawn at the gate where it binds. Records are cylinders and gates are hexagons; filled shapes are implemented and dashed ones designed. On the build path an item with no record address never enters the index, a challenge with no cited evidence cannot be constructed, and the lint refuses a persona or an uncited record, so clauses (a), (b) and (d) are enforced by construction (§8.3). The act path is designed and not deployed: the stopping-moment detector is specified but its delivery is not yet instrumented (§10.5), and clause (c)'s rate requirement remains a hypothesis. The loop closes into the record, which is what makes the slow controller of Figure 12 possible. **Schematic; build status shown.**

The evidence contract is the clause (b) commitment made structural. A challenge is not an object that may or may not carry a citation; it is an object that cannot be constructed without one, so a challenger that would rather assert than cite has no code path to do it.

```
# The evidence contract: a challenge with no readable record cannot exist.
@dataclass(frozen=True)
class Challenge:
    claim:     str   # the assertion the agent stopped on
    record_id: str   # address of a record in the operator's index
    quote:     str   # span reproduced from that record

    def __post_init__(self):
        if not self.record_id:
            raise Unconstructable("no record cited")
        if self.quote not in corpus.text(self.record_id):
            raise Unconstructable("quote is not verbatim in the cited record")
```

The contract, in outline. Names are simplified from the shipped engine; the control it expresses is the shipped one.

The lint then holds the two constraints the constructor cannot see: whether the record was actually in the set retrieved for this turn, which is what stops a challenger from citing something it never read, and whether the challenge speaks as a person, which is the clause (d) constraint of §8.2.

```
# Admission: what the constructor cannot check about a well-formed challenge.
def admit(ch: Challenge, retrieved: set) -> None:
    if ch.record_id not in retrieved:
        refuse("cites a record this turn was not given")
    if speaks_in_persona(ch.claim):
        refuse("speaks as someone rather than as the record")
```

Admission checks. A refusal names the failing predicate, so the refusal is itself a record.

What neither guarantees is that the cited record contradicts the claim. Retrieval by similarity returns records about the same topic, and agreement and contradiction look alike to it. A challenge that is well formed, correctly cited, and simply wrong would pass both gates above, count as dose under clause (c), and arrive carrying the operator's authority under clause (d). That is the exact condition H3 says produces folding rather than correction, and it is why the precision check of §10.5 is a required instrument rather than an optional one.

8.6 The judgment layer

The counterweight is one instrument inside a set of decisions the practice cannot delegate, and the set deserves naming because the market is commoditizing everything around it. Over 2026 the infrastructure beneath working agents, the orchestration, retries and state handling, became a managed rental; this practice rents such tooling and expects to keep renting it. What did not commoditize are the decisions that infrastructure exists to execute: whose knowledge becomes the standing instruction set; what result, fixed before work opens, ends an experiment; and what "done" means for a given piece of work. A practitioner essay contemporaneous with this program posed these three questions as the unanswered residue of managed-agent platforms (Forstie, 2026, cited under Prior art). The practice's answers are machinery this paper has already described: changes to the standing instruction set are proposed by the machine and merged or refused by the operator; experiments carry decision rules ratified before their data are seen, under which a null closes the program (§10.4); and the finish line is ruled in advance. The record is the load-bearing element in each: every one of those decisions is made from the practice's own logs, which is why §7 treats ownership of the record as a first-order property rather than a storage preference. An execution history that lives on a platform's dashboard, in the platform's format, is testimony in the sense of §6, not a record the operator holds.

9 First calibration: the auditor folds

Before any challenge reaches a working agent, the plan requires one pair of numbers with intervals: how often an automated auditor, shown a claim and the evidence that preceded it, correctly flags a false claim, and how often its flags are right. The first run is itself a result.

The calibration set holds 170 claims about test outcomes made by agents during real sessions. Each is labeled mechanically from the nearest test-runner output that precedes it in the same session, and the auditor never sees the label. The auditor, a small open-weight model (OLMo-2-7B, run at 4-bit precision) on the practice's own hardware, receives the claim and its evidence window, which is the tool output an agent would have had at the stopping moment, and answers SUPPORTED, CONTRADICTED or CANT_TELL. A hedged answer on a false claim counts against recall, because hedging is not catching. A second auditor from a different open-weight family (Qwen) then judged the same claims, plus one true claim added to the set after the first run.

Table 6. First calibration of two auditors, runs of 11 September 2026. The positive class is a false claim.

Quantity	**First auditor** OLMo-2-7B	**Second auditor** Qwen
Claims judged	170	171 one true claim added
Verdicts: SUPPORTED, CANT_TELL, CONTRADICTED	85, 85, 0	58, 108, 5
False claims caught, of 41	0	3
Recall on false claims Wilson 95% interval	0 0 to 0.086	0.073 0.025 to 0.194
CONTRADICTED on a true claim	0	2
Precision Wilson 95% interval	undefined no verdicts to score	0.6 0.231 to 0.882
False claims supported	17	9
True claims supported	68	49
Timeouts	0	0

The first auditor never returned CONTRADICTED. It supported 85 claims and declined to decide 85; of the 41 false claims, it caught none and affirmatively supported 17. In a sample of eight false-claim rows, six evidence windows carried an explicit failure marker. The auditor had the contradiction in front of it and still answered SUPPORTED or declined to decide. These are the easiest claims an auditor could be given: a claim about a test outcome, with the test runner's own output in the window. The second

auditor fired, rarely. It returned five contradictions, three on false claims and two on true ones, so it caught 3 of the 41, still supported 9, and declined to decide on 108 of its 171 verdicts.

A second judging prompt, pre-registered as a single variant and run on the same 171 claims the evening this paper was revised, settles what the first auditor's zero meant and adds the sharper finding. The new prompt licenses disagreement, requires a verbatim quote of the outcome line, and places the verdict last. Under it the first auditor answered 150 of 171 rather than 85, returned 114 contradictions rather than none, and its recall on the 41 false claims rose from 0 to 0.683 with an interval from 0.53 to 0.80. Its precision was 0.246. That last number is the whole result: the auditor fired on 28 of the 41 false claims, 68%, and on 86 of the 130 true ones, 66%. It fires at the same rate whichever way the truth lies, so its precision is simply the share of false claims in the set. The prompt change moved the firing rate and left the discrimination where it was. The first run's zero was the prompt shape suppressing the verdict, not the model failing to see the evidence; the second run shows that when the verdict is no longer suppressed, it is not informed either. Nothing further is tuned on these rows, and the gate holds.

Zhao et al. (2026) show the same brittleness is general rather than peculiar to small local models. Stress-testing nine frontier models across fourteen judging tasks, they find every one of them flips its verdict on 25 to 71% of items under static pushback, and on 62 to 91% when an adversarial model does the persuading. Their central finding is the one that bears on this program: pressure that succeeds in changing a judge's verdict is almost always net-corrupting with respect to ground truth. An auditor that can be moved is not thereby an auditor that can be improved, and the thing that moves it moves it the wrong way.

We read this as the stop problem reproduced in the instrument meant to catch it. An auditor built from the same class of model defers to the claim it is shown, much as Kelley and Riedl (2026) observe of models cast as peers, and as §6.5 leads one to expect of any model asked to arbitrate. The second family does not overturn that reading. It traded some of the deference for hedging, and its recall interval still sits far below anything a deployment could rest on; three catches in 41 cannot say whether the two families really differ, since their intervals overlap. Both are small local models; a frontier-class auditor was not tested, and the result bounds only the small open-weight case the practice can run offline. It is also the case H3 anticipates: an auditor that folds produces neither reversals nor corrections, and so cannot distinguish them. The plan's gate held, and neither auditor is deployed on these numbers.

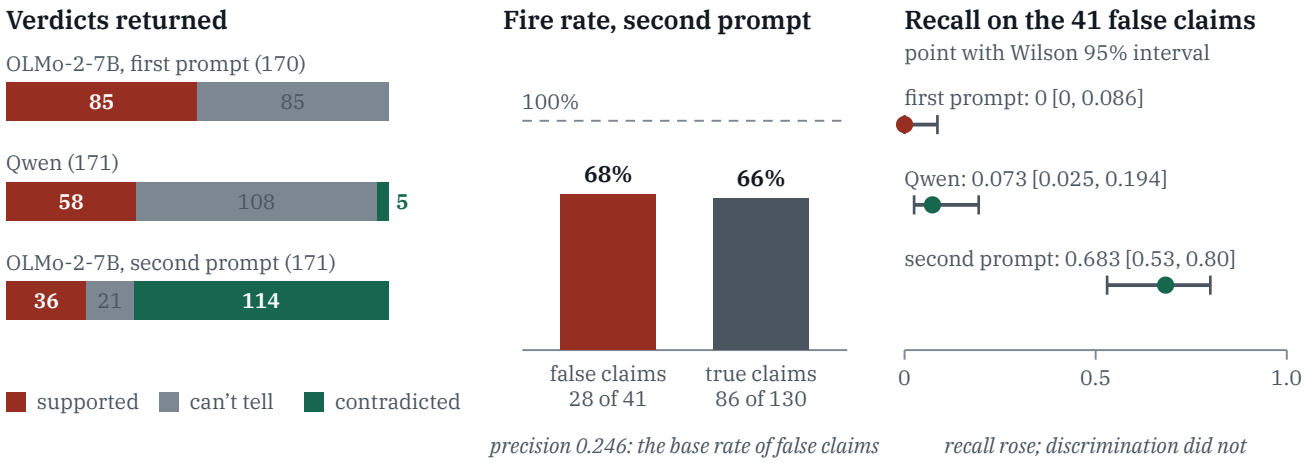

Figure 14. The calibration of §9 drawn three ways. Left: every verdict each run returned; the first prompt never used the contradiction verdict at all. Middle: under the second prompt the same auditor fired on 28 of 41 false claims and 86 of 130 true ones, so its precision, 0.246, is the share of false claims in the set. Right: recall on the 41 false claims, on the full zero-to-one scale. The second prompt moved the firing rate and left the discrimination where it was. Counts are from Table 6 and §9. **Measured.**

The next steps, in order of cost, are a sharper judging prompt, evidence windows centered more tightly on the failure marker, and a comparison of where the two auditors agree and disagree, which decides whether a pair of them adds anything over one. The denominator carries known noise, since at least one sampled false claim is a passage of prose rather than a claim; noise can move recall's magnitude, but it cannot produce zero flags across 170 rows. The verdict counts in Table 6 were recounted from each run's per-row output, the label-dependent figures were read from each run's report file, and the sample of eight is from the implementing team's report.

10 Evaluation

10.1 Questions

1. **RQ1.** Do operator challenges at the stopping moment cause material reversals in agent output?
2. **RQ2.** Can a mechanical counterweight predict where the operator would challenge, reliably enough to act on?
3. **RQ3.** Does delivering such challenges at the stopping moment measurably reduce recurrence of the target error classes?

10.2 Design

The design is a single-case experimental design with N = 1, extending to a multiple-baseline design across co-researchers onboarded at staggered start points: the intervention begins at a different time for each subject, and an effect should appear at each start point and not before. Single-case standards

conventionally require at least three demonstrations of an effect at three different points in time, so that a single coincidental change cannot pass for one (Kratochwill et al., 2013); the reporting guideline for this family is SCRIBE (Tate et al., 2016), and the n-of-1 trial extension is CENT (Vohra et al., 2015). None of the three is indexed where this program's literature searches run, which is its own finding: the design standard this paper should be held to lives in education and behavioral science, and a search confined to machine-learning venues will never surface it. The program's own phase criteria currently ask for two and are being corrected to three to meet that standard. Three units are in use and are not conflated: the turn is the sampling unit for adjudication (about 8,221), the session is the exposure unit, and the class-instance is the unit of the primary outcome. The evaluation's datasets come from different corpora of §2.2 and do not overlap: the calibration claims of §9 are drawn from the code-agent transcripts, where a test runner's own output supplies each label, and Study 2's pairs are drawn from the operator's chat-lane record, where the operator's own challenge does.

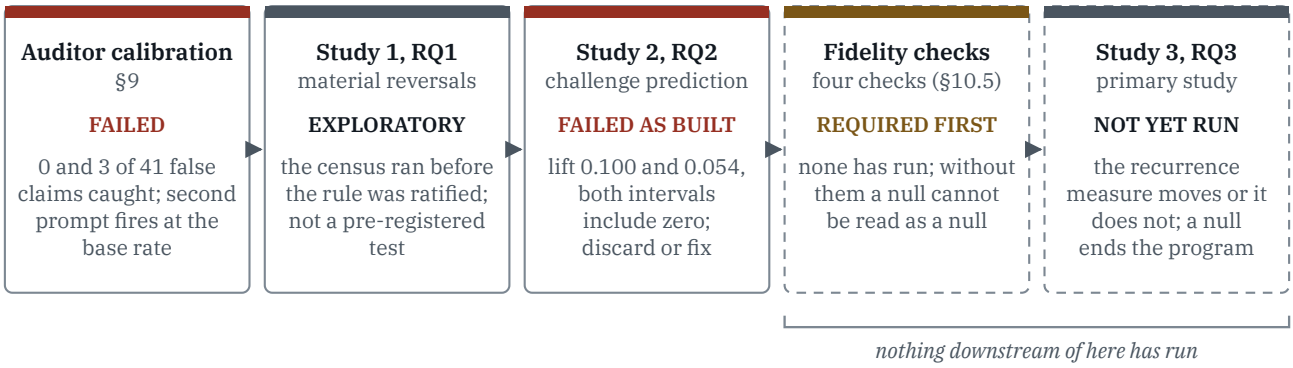

Figure 15. The evaluation as a gated pipeline, drawn with its standing in September 2026 rather than as a plan. The auditor meant to automate the check failed its calibration and is not deployed; Study 1 cannot serve as a pre-registered test because its census preceded the rule; Study 2 failed both conjuncts on the ruled split, with positive point estimates and intervals that include zero; the four fidelity checks have not run; and the primary study has not opened. The program's claim to be falsifiable rests on the last stage, which is the one that has not happened yet. **Current standing, September 2026.**

10.3 Constructs

Table 7. Constructs and their operationalization.

Construct	Operational definition	Instrument
Stopping moment	The turn at which an agent emits a terminal assertion without a further retrieval act.	Render-event log joined to a read-after-render check
Agent correctness	Not measured directly. Proxied by recurrence of two error classes, wrong diagnosis and unsupported coverage claims.	The recurrence measure of §5.2, in production

Construct	Operational definition	Instrument
Material reversal	A change of position on a claim of fact following a challenge, distinguished from a cosmetic edit.	Census detector with a hand-rated validation stratum
Catcher identity	Who first surfaced an error: operator, another agent, or the agent itself.	Source field on the error record
Auditor accuracy	Recall and precision of the auditor's contradiction verdicts on labeled claims, with intervals.	Calibration set of §9
Counterweight fidelity	Whether a fired challenge satisfied clauses (a) to (c), and whether its cited record contradicts the claim.	Partly enforced in the engine (§8.5); delivery not yet instrumented (§10.5)

10.4 Decision rules

Pre-registration here means that a decision rule is ratified before its data are examined. Table 8 gives each study's rule and its standing in September 2026, including where that standard was not met.

Study 2 received its first live reading on 11 September 2026, against 185 turn pairs in which both the engine's firing decision and the operator's own subsequent challenge are observable. The quantity is the lift of the engine's firing over the ambient rate:

> `agreement-with-key lift over a rate-preserving permutation = 0.059`
>
> Bootstrap interval over pairs, 0.011 to 0.108, excluding zero. Computed over all 185 joined pairs with no train-and-test split, at an uncalibrated retrieval floor of 0.35, firing on 172 of them.

Two things follow, and only one of them is encouraging. The association is real and positive, which is what the criterion's first conjunct asks for. But an engine that fires on 93% of turns is not a predictor of anything; at that operating point almost every turn is challenged, so the dose requirement of clause (c) is satisfied trivially while the discrimination requirement of clause (b) is not tested at all. The criterion's second conjunct, beating the fallback, could not be evaluated at the time for a blunter reason: the fallback had never been defined, and a conjunct whose comparator does not exist cannot be passed or failed, only skipped. Both gaps were closed by ruling on the day this paper was revised. The operator named the fallback, a second open-weight model judging each pair directly, and ruled that the firing floor must be set on one seeded half of the data and scored on the other, which replaces the single uncalibrated pass the reading above came from. The implementing change was merged and the first run under the ruled design completed the same evening, and it is the reading this section now rests on. The floor was chosen on a seeded calibration half of 92 pairs at 0.485, where the engine fired on 65. On the held-out half of 93 pairs carrying 44 operator corrections, the engine fired on 58 and the fallback, the second open-weight model judging each pair directly, fired on 7 and decided all 93. Against chance the lift was 0.100 with an interval from −0.043 to 0.237; against the fallback it was 0.054 with an interval from −0.097 to 0.194. Neither interval excludes zero. Under the ruled criterion the challenger

as built does not pass either conjunct, and the reading is discard-or-fix. The point estimates are positive and the intervals are about 0.14 wide on either side at this n, so the result is an absence of demonstrated effect rather than a demonstrated absence, but the pre-registered bar was set before the data and the engine did not clear it. The implementing seat's diagnosis matches the contract's own known limit: retrieval by similarity fires on topic, not on contradiction. A fix cannot be re-scored on these rows and needs fresh pairs. As built the challenger fails both conjuncts, and the reading is discard-or-fix: the path is not killed, but nothing on these rows is carried forward and any fix must be scored on fresh pairs. That the engine retrieves by topical similarity rather than by contradiction is the diagnosis a fix has to answer.

Table 8. Decision rules and their standing, September 2026.

Study	Rule	Outcome	Standing
Study 1 RQ1, retrospective	Material-reversal rate in challenged turns against a neutral arm: above an upper threshold, below a lower one, or between.	Confirm; null; or underpowered, meaning extend the corpus and do not reinterpret	Exploratory. A first census of the data was run before the decision rule was ratified, so Study 1 cannot serve as a pre-registered test. Its thresholds are being re-specified in any case: at the observed base rate of about 48%, the detector cannot produce a ratio above about 2.46, so the original threefold confirmation threshold could not have been met.

Study	Rule	Outcome	Standing
Study 2 RQ2	The predictor must beat chance and beat a fallback, with an interval that excludes zero, scored on the operator-and-agent pair rather than on the operator's turn alone.	Proceed; otherwise the personalized path closes and non-personalized challenge is used	Current criterion, adopted in September 2026. It replaced an earlier agreement threshold, Cohen's $\kappa \geq 0.70$ against a 199-item operator key. The first calibration of two auditors (§9) is upstream of this study, and neither clears it. A first reading on 11 September 2026 gave a lift of 0.059 with an interval excluding zero, over all 185 pairs with no split and an uncalibrated floor, firing on 172 of them. It is a pre-ruling operating-point reading rather than a result. Later the same day the fallback was defined and a seeded-half calibration was ruled, and the first run under that design gave lift 0.100 against chance and 0.054 against the fallback, neither interval excluding zero. Fails both conjuncts as built; discard-or-fix (§10.4). The rule named no minimum detectable effect, and at the held-out n the intervals span roughly ±0.14, so an effect smaller than that could not have passed whatever the truth; future rules carry one.

Study	Rule	Outcome	Standing
Study 3 RQ3, primary	The compound recurrence measure moves off its pre-computed baseline, with an interval that excludes zero.	H0 supported; if it does not move, a measured null, and the program ends	Pre-registered and not yet run. A null is an equally reportable result. A second conjunct was specified and never ratified into the rule: that the targeted classes move without a matching move in the untargeted ones, which is what separates a real reduction from a collapse in detection (§11). It should be adopted before the study opens, and reported as an addition made before the data were seen. The rule's metric needs the same treatment: as ratified it names the median gap, which cannot move while most intervals sit at the floor (§5.2), so the floor share should be ratified as the primary quantity, together with the unit, the interval procedure, how the staggered onsets enter the test, and a minimum detectable effect, before the study opens.

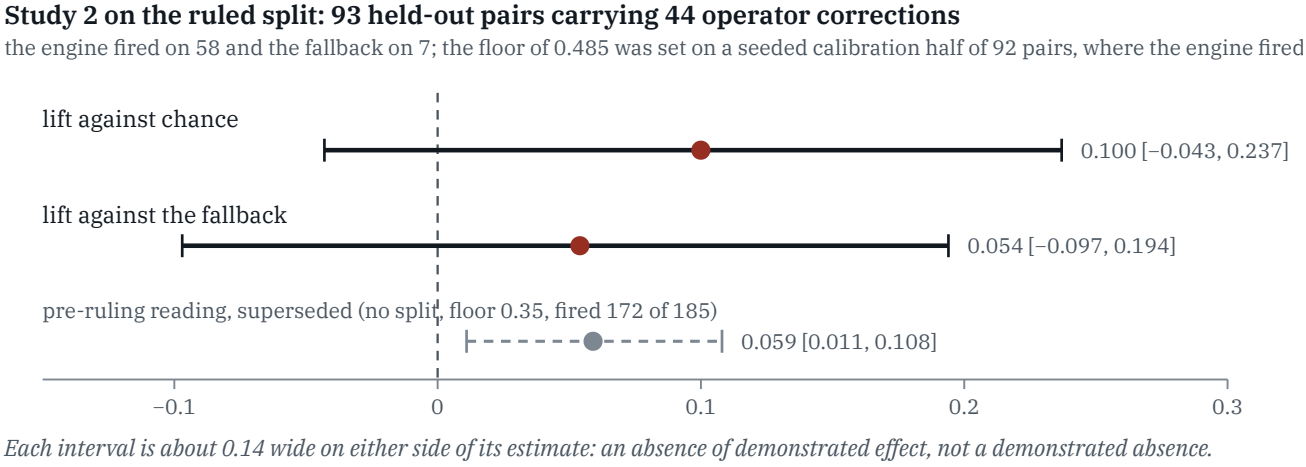

Figure 16. Study 2 on the ruled split (§10.4). Both ruled lifts are positive and both intervals include zero, so the challenger as built passes neither conjunct. The earlier reading of 0.059, which did exclude zero, was computed over all 185 pairs with no split at an uncalibrated floor and is superseded, not a result. The intervals are wide enough that this is an absence of demonstrated effect rather than a demonstrated absence. **Measured.**

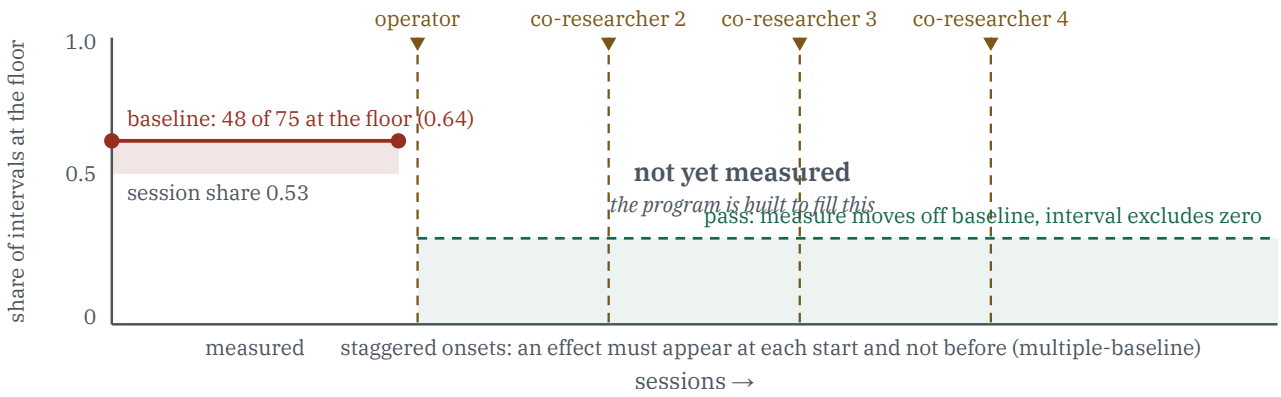

Figure 17. The quantity Study 3 should bind to, drawn empty because the study has not run. The baseline is a stamped reading, 48 of 75 intervals at the floor and a session share of 0.53, board render of 11 September 2026 at 18:57 UTC, with a same-evening re-read returning 74 sessions. The pre-registered rule requires the measure to move off that baseline with an interval excluding zero, at each staggered onset and not before. One amendment is owed before the study opens: the ratified rule names the median gap, which §5.2 shows cannot move at this baseline, so binding the rule to the floor share drawn here is a pre-data amendment that is the operator's to make (Table 8). Everything right of the first onset is what the program is built to fill; a null there closes it. **Rule pre-registered; plotted metric pending amendment.**

10.5 Fidelity

A null is uninterpretable without a fidelity measure, because it cannot distinguish a counterweight that does not work from one that was never delivered as specified. Four checks are required before the interventional arm opens. A delivery check logs render events and joins each to a read-after-render test; its pre-registered falsifier is that twenty renders with zero reads after render means the delivery mechanism failed, independent of any outcome. A clause check scores each fired challenge for externality, for whether it named a checkable source, and for whether it fired at a detected stopping moment; the engine now enforces the second by construction. A dose check records the firing rate against the ambient error rate for the same window, which is how the rate hypothesis of clause (c) gets tested rather than assumed. And a precision check adjudicates a sample of fired challenges for whether the cited record actually contradicts the claim. The evidence contract guarantees that a challenge cites a record, not that the record contradicts anything, since retrieval by similarity cannot tell contradiction from agreement about the same topic. A wrong but cited challenge would pass every other check, count as dose, and carry the operator's authority under clause (d), which is the exact condition under which H3 predicts folding rather than correction.

10.6 Reliability

The program sets an agreement floor of $\kappa \geq 0.70$ for any claim derived from labels. No verified measurement of agreement on the operator-class split is available, and none is reported here. A figure of $\kappa = 0.410$ circulates in the program's record, but its producing script could not be located, and every traceable occurrence of that value belongs to a different measurement, the coding of a precedence

clause's location over an 88-row corpus with a 44% uncodable share. The floor is currently report-only, enforced by discipline rather than in code. A second blind pass by the operator over 40 stratified turns puts a ceiling on what any label-derived claim can say, with one caveat that has to travel with it: those 40 rows were drawn from the whole corpus of 289 conversations, while the study they bound now targets a narrower lane of 113, so the ceiling and the target were measured on different populations and the figure needs a rescoped redraw before it is used against that target. The recorded reading was agreement on the binary teach flag of $\kappa = 0.40$ (95% CI 0.11 to 0.66), with five of six finer channels indistinguishable from chance. On the evening this paper was revised a seat traced that reading to a join defect: the key file's row identifier places every mark after item 26 one row above its item, and 37 of the 40 re-labeled items sat on a shifted row. Before any edit the defective join reproduced every recorded figure exactly, so the move is the join and nothing else. Under the repaired join the same 40 rows read $\kappa = 0.554$ (95% CI 0.292 to 0.798), with one channel rather than five at chance, and every detector graded against the key moves with it, the lexical rule from 0.04 to 0.21 and the local model from 0.09 to 0.37. The repair was merged on 11 September 2026, verified at the repository rather than at a report of it; the program's ruling that cites 0.400 had not been amended when this revision closed, so this paper reports both readings with their status and treats neither as settled. Two consequences follow either way. The ceiling's upper bound now reaches the 0.70 floor, so "unreachable" is not established, and at n = 40 the interval cannot tell unreachable from hard. And §3.1 continues to report corrective signal as one undifferentiated quantity, now because the separability of the finer channels is unsettled rather than because it was shown absent; no claim in this paper rests on the finer channels under either join. Its consequence is bounded: the shortfall blocks label-derived claims, not the natural-signal measurement of the primary outcome, which does not depend on the contested labeling route.

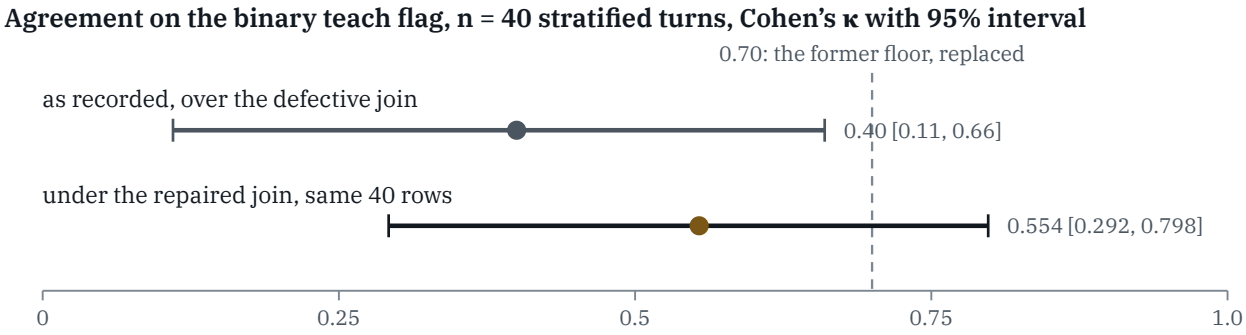

Figure 18. The labeling ceiling of §10.6 under both joins. The recorded reading was computed over a key whose row identifier sat one row off after item 26; the repaired join on the same 40 rows moves the estimate and the interval's upper bound now reaches the former 0.70 floor. At n = 40 the interval cannot tell unreachable from hard, and neither reading is treated as settled. **Measured.**

10.7 An external instrument

Every measurement above runs on the practice's own corpus. One planned measurement does not: a bench on the public ARC-AGI-3 game set, adopted in September 2026, in which the challenge is delivered at the stopping moment in one arm and withheld in the other, and performance is compared with a human baseline. It is designed and not yet implemented. Until it runs, the program's evidence is the practice's own.

10.8 Ethics and data

Co-researchers are volunteers and collaborators, not subjects or staff, and none is employed by the company. Before any co-researcher contributes measured data the program requires a participant information statement, a recorded consent, and a data-handling statement covering retention and withdrawal. The training data for the counterweight are operator-authored turns only; agent outputs enter as unlabeled context.

10.9 The program as built and as planned, 21 September 2026

The 11 September revision described the counterweight's engine as implemented and not deployed. In the nine days since, three build waves landed and merged, and the honest way to report them is by what each did to a measurement problem this paper has already named. A wiring audit now runs nightly: of 192 event producers in the practice's codebase, 122 are wired to a reader, 70 are declared, none are unwired, and a burn-down of 55 grandfathered producers is printed each night, so a signal that reaches nothing can no longer do so silently (§5's delivery argument, applied to the practice's own instruments). A delivery switch for the counterweight's challenge arms is built and deliberately unarmed: until the operator writes the first schedule entry under an account the working seats cannot modify, every session is unassigned, and unassigned delivers nothing. The sample-size calculation that had existed only as prose now has a producer: at the measured compound-recurrence share of 0.5101 with lag-one autocorrelation 0.2547 and the proposed smallest effect of interest of 0.25, a null is reachable one-sided at 95% with 84 sessions per arm under Bernoulli assignment and 140 under cycle-switch (two-sided, 119 and 199; the 11 September prose figures reconciled with this producer on the dependence model and not on the base n, and their method was never recorded, which is why the producer exists). And a machine-fed adherence instrument is on the nightly: the compaction hook writes a row the seat cannot decline to emit, and a subsequent load is the only thing that clears it. At this revision it reads 70 compactions with 43 reloaded, 18 not, and 9 unresolved, a settled not-reloaded rate of 29.5%, and the monthly split shows the direction: August ran 33% reload, September 85%.

What remains is not construction. It is a short chain of operator decisions, recorded here because §10.4's pre-registration freezes the instrument when signed, so the order matters. First the unit of analysis: a session identifier in this practice spans one turn to ninety-seven and zero compactions to sixteen, so a per-session outcome partly measures session length until the operator rules the unit. Second the feed: both of the primary outcome's current inputs are typed by the seat being measured, and the record shows what that produces; the practice's miss log accrued from twenty-two sessions in June and eighty-one in July, through thirty-two in August, to seven in September, a collapse in voluntary

emission, not in error, while the alternative decision log holds fifteen rows in its entire history, six of them written by the session that built the logger. Choosing between a hand-fed instrument with measured decay and a machine-fed one with a third bucket is the operator's call, and this paper's own §6 is the argument that it cannot be the seat's. Third, the compute moves off the operator's laptop, which suspends scheduled work when it sleeps and therefore cannot host an unattended measurement window. Fourth, the switch is armed, the pre-registration row is signed with the window, the n, and the frozen instrument identities, and the study of §10.4 runs to one of its three registered exits. That is where this program is pointed: challenges delivered on schedule to assigned arms, fed by signals the measured party cannot withhold, scored against a rule fixed before the data existed, on infrastructure that does not sleep, with the operator holding every gate that changes the system. The reader should weigh the plan accordingly: everything in this subsection before the word "remains" is merged and running, and everything after it is intention, dated 21 September 2026.

11 Threats to validity

Construct. The primary outcome is a proxy. Recurrence can fall because an error class genuinely recurs less, or because detection of the class degraded. The mitigations are an independent adjudication arm that does not share the detector's blind spots, and the per-class series, in which a detection collapse shows as a simultaneous fall across classes. Srinivasan and Paragiri (2026) give the general form of this hazard for agent-driven search: where validity lives in disaggregated structure, an aggregate reduction can rank the wrong candidate first, the headline number improving while the structure beneath it inverts. Their remedy is an external control loop that audits disaggregated behavior after the agent has decided and can reopen a run the agent declared finished, which is the shape of the off-target conjunct now proposed for Study 3 (§10.4). The pushback reading behind §3.1 and §6.2 comes from a machine labeler with 64% precision on the blind key, and its adjusted figure depends on that key being representative of the frame. The calibration set's labels are mechanical: they establish whether a test passed, not whether the agent's work was correct.

Internal. Instrumentation is a live threat, not a hypothetical one. A source field on the practice's records defaults to "self" when unset, so historical self-catch shares (30.6%, n = 556; 25.5% on wrong diagnosis) cannot distinguish a genuine self-catch from an unlabeled row. An audit of the producing repository confirms the defaulting behavior and the two shares. The remedy is to reject an unset source at the write path, flag explicit sources, and report historical and post-fix rows separately, with no backfill. History and maturation are present over a six-month corpus in which the harness changed repeatedly; testing effects are present because the operator knows a measurement is running; and selection is present because the error log records only caught errors.

Evidence from the practice's own instruments. By the rule in §8.3, instruments the practice built and operates are exposed to the measured party, and every reading in §3 to §7 comes from such an instrument. We claim no exemption. The readings are exploratory; the primary outcome depends on a natural-signal measurement checked by an independent adjudication arm; and the one external instrument (§10.7) has not yet run.

External. One operator, one domain, one harness, one model family for the agents. Nothing here generalizes to other operators without replication, and the multiple-baseline extension raises N to four within one organization, not to a population. The setting's ecological validity is bought with exactly this cost.

Conclusion. Many analyses have been run against one corpus over six months, and the multiplicity is not accounted for. Pre-registration of the primary rule is the principal control, and secondary findings, including every reading in §3 to §7, the calibration in §9 and Study 1, are reported as exploratory.

Reflexivity. The AI assistants used in this research come from the same model family as the agents under study and exhibit the defect under study. An earlier draft of this paper's case series contained seven factual errors of exactly the kind the paper describes, among them a twenty-hour window described as one morning and one automated catch counted as two; all were found on re-check against the transcript. A later draft carried, as its headline exposure figure, a census count of whether the operator's reply contained one of six literal phrases, reported as the share of answers that were never challenged at all; it passed the paper's own fact-check because it carried a receipt, and the operator caught it. Drafts written on the day this version was revised did it several more times: a one-day tally of seven misses read as a census of rule adherence; a recurrence count that equals the number of sessions seen minus one by construction, quoted as though it could vary; a reliability figure whose only traceable producer is a different instrument; and, in the literature review, a range that does not exist in the cited paper, a diagnostic arm quoted as the deployable result, a section header promoted to a title, and a quotation attributed to a paper that does not contain it. Every one was caught before the paper left the room, by the operator or by fetching the source, and the corrected figures stand in the text without further comment. Read together, the day's errors share one shape: none was a miscalculation, and every one was a bad join. A count was joined to the wrong meaning (a recurrence tally that is the session count minus one), a statistic to the wrong instrument, a labeling ceiling to a key one row off its items (§10.6), and copies of a retired figure to a search that never reached them. The mitigation this suggests is retrieval-shaped rather than judgment-shaped: identifiers that resolve or fail loudly, and provenance edges written by instruments the agent cannot author, so that a figure carries its join and the join can be checked. It does not touch the judgment-shaped half, in which an auditor holds the contradicting evidence in its window and folds anyway (§9). The pattern is the finding. The paper's own production is a running specimen of its thesis. A hostile reading turns that around: a process that needed these corrections mid-draft produced the surviving numbers too. The answer is not a defense of the process, which would be the defect restated; it is that the paper's weight-bearing claim is the one object that process cannot have shaped, a decision rule ratified before its data exist, and that everything else is labeled exploratory and priced accordingly. The mitigations are mechanical rather than discretionary assignment of assistant instances to tasks, the operator outside the loop as the final check, and independent review of the design before the primary study opens.

12 Limitations

The program cannot establish a catch rate. The logs contain only errors that were caught, and no denominator of uncaught errors exists in them. A denominator is obtainable from the full transcript record but requires an adjudication arm that is not yet in production; until it is, a catcher share may be reported with its n, its window and its explicit-versus-defaulted split, and a rate may not be reported at all.

The program cannot establish that the counterweight raises correctness in the world. It targets discipline at the stopping decision, meaning whether the process that makes a claim checkable was followed. That is a narrower claim and the one the corpus can support.

The logging gap of §6.2 is an indication across two instruments, not a measurement. The rate requirement of clause (c) is untested. Study 1 is exploratory, the Study 2 criterion was re-specified in September 2026, and neither auditor has passed calibration.

13 Conclusion

The stop problem sits at the termination decision, not in the reasoning before it. That is why better reasoning does not remove it, and why an external check timed to the stop might. We have argued five propositions from one practice's logs and from recent work, found the same shape in the practice's own instruments, specified a counterweight from published work, and built its challenge engine. Its first calibration shows two auditors folding or hedging on the evidence in front of them, which is the problem restated inside the instrument meant to solve it, and exactly what a gate before deployment exists to catch. The program is built to be able to fail. If the recurrence measure does not move, that is the result, and the program ends.

Table 9. The five propositions, what each rests on, and what would overturn it.

	Rests on	Would be overturned by	Kind
P1	Corrective signal in ~48% of operator turns (§3.1); Mehta (2026)	A build that prices correctness at the stop and still terminates early	measured
P2	The case series (§4.1); Kelley & Riedl (2026)	The same agents catching their own premature stops without operator direction	interpretive
P3	48 of 75 intervals at the floor (§5.2); the read-versus-injected mechanism, not a rate (§5.1); Wang & Huang (2026)	A remembered rule that holds across sessions where only a boundary now does	measured
P4	75 of 579; the logging gap; four instrument failures (§6)	A self-report that tracks the logs it claims to summarize	measured

	Rests on	Would be overturned by	Kind
P5	The 8,221-turn record and the blind key (§7)	A broad preference dataset that recovers not just dislike but wrong, why, and what happened next	interpretive

What transfers, if anything does. Three rules earned here do not depend on this practice's N of 1. A boundary that refuses and records its refusal outlasts a memo that asks an agent to remember (§5). A challenge must cite evidence the challenged agent can open, or it is an opinion wearing a citation (§8.2). And an outside checker must be calibrated against known-false cases before it is allowed to gate anything, because a checker built from the same class of model may simply agree with the claim it is shown (§9).

Disclosure. Drafting and literature synthesis were assisted by AI models (Claude, Anthropic) working under the author's direction; the author is responsible for the content. The works in the References were read in full, and every specific figure cited comes from a work read in full. The prior-art works listed under Prior art are cited at the level of an established concept and its origin: each was verified for author, title, year and venue, but not read in full, and no numeric claim rests on any of them. Software documentation, source code and press accounts are listed under their own headings and were read at the linked pages. Every reference below carries a link, and every arXiv identifier and DOI was resolved against its registry, with title and first author matched, on 23 September 2026.

Corrections, 23 September 2026. No figure and no finding changed. The paper was retitled; earlier revisions were titled *The Stop Problem: Defensible Is Not Correct*, and the stop problem remains this paper's name for the failure it studies. The Anthropic interview in §2.3 aired on 13 September, not over a weekend of 13 and 14 September; the web article is stamped 14 September. The essay listed under Prior art is by Ryan Forstie; an earlier revision gave the initial K. The completion evaluator in §8.3 is documented as a Claude Code feature, whose agent loop the Claude Agent SDK embeds; an earlier revision attributed it to the SDK directly. Two works in the References, Graves (2016) and Liu (2026), were listed without being cited in the text; each is now cited where it bears (§2.1, §8.3). Links were added to every press, documentation and prior-art source. A duplicated section number in §10 was corrected.

Corrections, 25 September 2026. No figure changed. §8.3 said that the human-in-the-loop primitives across the three frameworks surveyed gave a developer nothing to require a person to confirm finished work. That holds for the OpenAI Agents SDK and the Claude Agent SDK. It does not hold for CrewAI, whose task documentation, already cited as CrewAI (2026b), offers an opt-in setting for a human to review the agent's final answer; §8.3 now says so.

Competing interests. The author owns Wolfberg LLC, the practice studied.

Data availability. The practice's logs contain client work and personal records. They are private and are not offered for sale or sharing. The measures are described in enough detail to be reimplemented, and

figures from the practice are reported as of the dates given. What is available is the design: the clauses of §8.1, the evidence contract and admission checks of §8.5, and the decision rules of §10.4 are stated fully enough to be rebuilt without access to the logs.

The series. This paper is also published as seven short background papers, each reproducing the relevant sections word for word with their numbers kept, for a series of plain-English posts that shares this paper's title. Part 0, what "AI" actually is: a model and a harness (§1, §2.2, §4.2, §5.1, §8.3, §8.6). Part 1, the model stops at the first answer it can confidently defend, not when it is right (§1, §2.1, §3, §4.2, §6.5, §8.2). Part 2, AI does not need to be smarter; it needs a boss (§2.3, §4, §5.1, §8.3, §8.4, §8.6). Part 3, rulebooks are theater; boundaries and incentives are not (§4.1, §5, §6.3, §8.3). Part 4, never let your AI grade its own homework (§1, §2.3, §6, §8.3, §9). Part 5, your corrections are the most valuable data in your company (§2.2, §3.1, §5.2, §6.2, §7, §8.4). Part 6, rent the pipes and own the judgment (§2.3, §5.2, §8.3, §8.6, §10.4, §10.9). Where a part and this paper differ, this paper governs; they are built so that they cannot.

Suggested citation. Atkinson, B. (2026). *How do we use this? Findings from running an AI team on real work for six months, and a test that could prove them wrong.* Working paper, Wolfberg LLC.

REFERENCES

Adler, S. (2025, October 2). *Practical tips for reducing chatbot psychosis.* Clear-Eyed AI. clear-eyed.ai

Advani, L. (2026). *From confident closing to silent failure: Characterizing false success in LLM agents.* FAGEN Workshop at ICML 2026. arXiv:2606.09863. arxiv.org/abs/2606.09863

Challapally, A., Pease, C., Raskar, R., & Chari, P. (2025). *The GenAI Divide: State of AI in Business 2025.* MIT NANDA project, July 2025. Project page: nanda.media.mit.edu; the report as read: mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf

Denisov-Blanch, Y., Kazdan, J., Chudnovsky, J., Schaeffer, R., Guan, S., Adeshina, S., & Koyejo, S. (2026). *Consensus is not verification: Why crowd wisdom strategies fail for LLM truthfulness.* arXiv:2603.06612. arxiv.org/abs/2603.06612

Dongre, V., Rossi, R. A., Lai, V. D., Yoon, D. S., Hakkani-Tür, D., & Bui, T. (2025). *Drift No More? Context equilibria in multi-turn LLM interactions.* arXiv:2510.07777. arxiv.org/abs/2510.07777

Flynt, J. (2026). *GroundEval: A deterministic replacement for LLM-as-judge in stateful agent evaluation.* arXiv:2606.22737. arxiv.org/abs/2606.22737

Graves, A. (2016). *Adaptive computation time for recurrent neural networks.* arXiv:1603.08983. arxiv.org/abs/1603.08983

He, C., Chen, Z., Yang, Z., Qiao, S., Ju, M., Liu, J., Wen, D., & Liu, G. (2026). *Minority Sentinel: When to overturn majority voting in multi-agent LLM debates.* AgentSearch Workshop at SIGIR 2026. arXiv:2606.29270. arxiv.org/abs/2606.29270

Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X., & Zhou, D. (2023). *Large language models cannot self-correct reasoning yet.* arXiv:2310.01798. arxiv.org/abs/2310.01798

Ivanov, I., & Africa, D. D. (2026). *LURE: Live-usage replay evaluations for reducing evaluation awareness.* arXiv:2605.26438. arxiv.org/abs/2605.26438

Kelley, S. W., & Riedl, C. (2026). *Personalization increases affective alignment but has role-dependent effects on epistemic independence in LLMs.* arXiv:2603.00024. arxiv.org/abs/2603.00024

Kenton, Z., Janzer, L., Greig, R., Teh, T. H., Tyshchuk, K., Brown-Cohen, J., Edwards, H., Rajamanoharan, S., Siegel, N. Y., Jaques, N., et al. (2026). *Debate training reduces reward hacking in RLAIF.* arXiv:2608.17776. arxiv.org/abs/2608.17776

Ko, D., Kim, J., Kim, S., Park, H., Lee, D., Kim, G., Lee, M., & Lee, K. (2026). *When is enough not enough? Illusory completion in search agents.* arXiv:2602.07549. arxiv.org/abs/2602.07549

Laban, P., Murakhovs'ka, L., Xiong, C., & Wu, C.-S. (2023). *Are you sure? Challenging LLMs leads to performance drops in the FlipFlop experiment.* arXiv:2311.08596. arxiv.org/abs/2311.08596

Lamparth, M., Fein, D., Haupt, A., Hussing, M., & Kochenderfer, M. J. (2026). *Reward bias substitution: Single-axis bias mitigations redirect optimization pressure.* arXiv:2605.27996. arxiv.org/abs/2605.27996

Liu, J. (2026). *When may an agent stop? Evidence-carrying termination for tool-using LLMs.* arXiv:2608.23623. arxiv.org/abs/2608.23623

Luo, H., Wen, B., & Wang, L. L. (2026). *Agentic abstention: Do agents know when to stop instead of act?* arXiv:2606.28733. arxiv.org/abs/2606.28733

Mehta, A. (2026). *When agents commit too soon: Diagnosing premature commitment in LLM agents.* Snowflake AI Research. arXiv:2606.22936. arxiv.org/abs/2606.22936

Mittapalli, S., Dani, V., Pilli, S., Ansu, A., Teymoorianfard, M., Dernoncourt, F., Chen, Z., Wang, R., Rossi, R. A., & Ahmed, N. (2026). *TRACE: Trajectory reasoning through adaptive cross-step evidence aggregation for LLM agents.* arXiv:2606.07054. arxiv.org/abs/2606.07054

Mytsyk, S., Zhang, Y., & Krishnamurthy, V. (2026). *Mitigating LLM sycophancy with RL-based fine-tuning: Bayesian truth serum approach.* arXiv:2608.25267. arxiv.org/abs/2608.25267

Pan, J., He, H., Bowman, S. R., & Feng, S. (2024). *Spontaneous reward hacking in iterative self-refinement.* arXiv:2407.04549. arxiv.org/abs/2407.04549

Panickssery, A., Bowman, S. R., & Feng, S. (2024). *LLM evaluators recognize and favor their own generations.* arXiv:2404.13076. arxiv.org/abs/2404.13076

Park, H., & Choi, B. (2026). *When do agent loops mistake stagnation for progress? Self-evaluation bias and externally grounded verification in long-running autonomous LLM agent loops.* arXiv:2607.25152.

arxiv.org/abs/2607.25152

Park, J., Cho, S., & Lee, J.-Y. (2025). *Stop-RAG: Value-based retrieval control for iterative RAG.* NeurIPS 2025 MTI-LLM Workshop. arXiv:2510.14337. arxiv.org/abs/2510.14337

Piatrashyn, D., Kotelevskii, N., Grishchenkov, K., Glazkov, N., Nasonov, I., Makarov, I., Baldwin, T., Nakov, P., Vashurin, R., & Panov, M. (2026). *ReDAct: Uncertainty-aware deferral for LLM agents.* arXiv:2604.07036. arxiv.org/abs/2604.07036

Roh, D., & Han, D. (2026). *HALT: Verification-aware stopping for retrieval-augmented search agents.* Findings of EMNLP 2026. arXiv:2608.02009. arxiv.org/abs/2608.02009

Srinivasan, A., & Paragiri, D. (2026). *Search discipline for long-horizon research agents.* arXiv:2606.11522. arxiv.org/abs/2606.11522

Vasudev, R., Russak, M., Bikel, D., & Alshikh, W. (2026).*Accurate failure prediction in agents does not imply effective failure prevention.* arXiv:2602.03338. arxiv.org/abs/2602.03338

Wan, Y., Fang, T., Li, Z., Huo, Y., Wang, W., Mi, H., Yu, D., & Lyu, M. R. (2026). *Inference-time scaling of verification: Self-evolving deep research agents via test-time rubric-guided verification.* Findings of ACL 2026. arXiv:2601.15808. arxiv.org/abs/2601.15808

Wang, B., Zhang, C., Liu, D., Zhang, J., Chen, J., Li, M., Chen, M., Fang, R., Zhang, S., Wang, X., Jing, Y., Ma, Z., & Cui, Z. (2026). *The verification horizon: No silver bullet for coding agent rewards.* arXiv:2606.26300. arxiv.org/abs/2606.26300

Wang, J., & Huang, J. (2026). *Reward hacking as equilibrium under finite evaluation.* arXiv:2603.28063. arxiv.org/abs/2603.28063

Xu, Y., Li, C., Wang, Z., Yang, J., & Chen, T.-H. (2026). *Preventing premature commitment in coding agents with an evidence-conditioned execution layer.* arXiv:2607.28815. arxiv.org/abs/2607.28815

Zhang, B., Zhu, J., Shi, Z., Liu, D., & Tang, R. (2026). *AgentForesight: Online auditing for early failure prediction in multi-agent systems.* arXiv:2605.08715. arxiv.org/abs/2605.08715

Zhao, J., Bhattacharjee, H., Korevaar, H., Radharapu, B., & El-Arini, K. (2026). *Jagged judges: Epistemic stability under perturbation, pressure, and persistence.* arXiv:2608.12645. arxiv.org/abs/2608.12645

PRIOR ART (cited at concept level; verified for author, title, year and venue, not read in full)

Simon, H. A. (1955). A behavioral model of rational choice. *Quarterly Journal of Economics*, 69(1), 99–118. doi.org/10.2307/1884852

Browne, G. J., Pitts, M. G., & Wetherbe, J. C. (2007). Cognitive stopping rules for terminating information search in online tasks. *MIS Quarterly*, 31(1), 89–104. doi.org/10.2307/25148782

Graber, M. L., Franklin, N., & Gordon, R. (2005). Diagnostic error in internal medicine. *Archives of Internal Medicine*, 165(13), 1493–1499. doi.org/10.1001/archinte.165.13.1493

Croskerry, P. (2003). Cognitive forcing strategies in clinical decisionmaking. *Annals of Emergency Medicine*, 41(1), 110–120. doi.org/10.1067/mem.2003.22

Buçinca, Z., Malaya, M. B., & Gajos, K. Z. (2021). To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1), 1–21. doi.org/10.1145/3449287

Campbell, D. T. (1979). Assessing the impact of planned social change. *Evaluation and Program Planning*, 2(1), 67–90. doi.org/10.1016/0149-7189(79)90048-X

Fagan, M. E. (1976). Design and code inspections to reduce errors in program development. *IBM Systems Journal*, 15(3), 182–211. doi.org/10.1147/sj.153.0182

Knight, J. C., & Leveson, N. G. (1986). An experimental evaluation of the assumption of independence in multiversion programming. *IEEE Transactions on Software Engineering*, SE-12(1), 96–109. doi.org/10.1109/TSE.1986.6312924

Buser, D., Schwaninger, A., Rehor, V., & Sterchi, Y. (2025). Reliability and validity of threat image projection data as a measure of performance in X-ray baggage screening. *Transportation Research Part A: Policy and Practice*, 200, Article 104640. doi.org/10.1016/j.tra.2025.104640 *Named as an ancestor of the live-queue method only; a corrigendum (doi.org/10.1016/j.tra.2025.104683) is unread, so no figure from it is cited anywhere in this paper.*

Kratochwill, T. R., Hitchcock, J. H., Horner, R. H., Levin, J. R., Odom, S. L., Rindskopf, D. M., & Shadish, W. R. (2013). Single-case intervention research design standards. *Remedial and Special Education*, 34(1), 26–38. doi.org/10.1177/0741932512452794

Tate, R. L., Perdices, M., Rosenkoetter, U., Shadish, W., Vohra, S., Barlow, D. H., et al. (2016). The Single-Case Reporting guideline In BEhavioural interventions (SCRIBE) 2016 statement. *Aphasiology*, 30(7), 862–876. doi.org/10.1080/02687038.2016.1178022

Vohra, S., Shamseer, L., Sampson, M., Bukutu, C., Schmid, C. H., Tate, R., et al. (2015). CONSORT extension for reporting N-of-1 trials (CENT) 2015 statement. *BMJ*, 350, h1738. doi.org/10.1136/bmj.h1738

Forstie, R. (2026). The part of the agent stack nobody wants to build. LinkedIn, 25 August 2026. linkedin.com/pulse/part-agent-stack-nobody-wants-build-ryan-forstie-ihyqc *Cited for its three closing questions on managed-agent platforms, first read on 1 September 2026; author, title, date and the three questions re-verified at the source on 23 September 2026. No figure from it is cited anywhere in this paper.*

SOFTWARE AND DOCUMENTATION (read at the linked pages, September 2026; current as of then, not permanent)

Anthropic. (2025, September 29). *Building agents with the Claude Agent SDK.* claude.com/blog/building-agents-with-the-claude-agent-sdk

Anthropic. (2026a). *How the agent loop works.* Claude Agent SDK documentation. code.claude.com/docs/en/agent-sdk/agent-loop

Anthropic. (2026b). *Keep Claude working toward a goal.* Claude Code documentation. code.claude.com/docs/en/goal

CrewAI. (2026a). *Agent output parser* (source code). github.com/crewAIInc/crewAI, the agent output parser

CrewAI. (2026b). *Tasks.* CrewAI documentation. docs.crewai.com/en/concepts/tasks

Nous Research. (2026). *Hermes agent* (repository, configuration and issue tracker). github.com/NousResearch/hermes-agent

OpenAI. (2024). *Swarm* (repository; experimental, superseded by the Agents SDK). github.com/openai/swarm

OpenAI. (2026a). *Running agents.* OpenAI Agents SDK documentation. openai.github.io/openai-agents-python/running_agents/

OpenAI. (2026b). *Human in the loop.* OpenAI Agents SDK documentation. openai.github.io/openai-agents-python/human_in_the_loop/

PRESS AND PUBLIC STATEMENTS (§2.3; read at the linked pages, or at the named carrier where the original is paywalled)

Axios. (2026, September 3). *Sam Altman's sobering siren.* Interview at the G20 Innovation Ministerial. axios.com/2026/09/03/axios-interview-sam-altmans-sobering-siren; paywalled, read as carried by The Next Web: thenextweb.com/news/sam-altman-axios-idea-guy-sobering-models

CBS News. (2026, September 13). *Anthropic CEO Dario Amodei: "For too long the industry lied" about AI risks.* Sunday Morning; web article updated 14 September. cbsnews.com/news/anthropic-ceo-dario-amodei-on-ai-risks/

CNBC. (2026, September 13). *Anthropic's Amodei says China presents "toughest dilemma" for his proposed AI slowdown.* Cited for the broadcast date. cnbc.com/2026/09/13/china-dilemma-ai-slowdown-anthropic.html

Fortune. (2026, May 26). On the two laboratories walking back earlier job-loss forecasts ahead of public offerings; cited for the reported valuations. fortune.com/2026/05/26/sam-altman-

dario-amodei-walking-back-ai-jobs-apocalypse-prophecies-ipo/

Fortune. (2026, September 12). Interview with the chief executive of OpenAI on safety and the timing of a public offering. fortune.com/2026/09/12/sam-altman-interview-ai-doomsday-safety-models-control-ipo-2027/

Reuters. (2026, September 14). *Wall Street ends down, calls for AI slowdown pummel chipmakers.* As carried by Yahoo Finance. finance.yahoo.com/technology/ai/articles/ai-warnings-knock-nasdaq-futures-092329455.html