Great Idea, You Had It Last Week: Insight Doesn't Survive

An extension of "Rulebooks are theater" from rules to insight and wisdom: why an insight has to become a change to the harness, and how to capture it, test it and admit it.

Berg Atkinson, Wolfberg LLC · berg@wolfberg.ai

Preprint. Not peer reviewed. Draft 2.0, 29 October 2026 (not yet published).

BOTTOM LINE

An AI agent sometimes reaches a real insight in a long exchange, and then loses it. The fix is to make the insight part of the harness, the software around the model, so the next session acts on it without having to find it. This note shows the loss and sets out how.

Terms. The operator is the author, who runs the AI agents studied; the assistant is the AI agent he works with. Knowledge is what is settled and can be relied on; insight is a new reframing or connection; wisdom is a principle that governs a whole class of future situations.

1. Written down, an insight loses. The operator's earlier paper found that a rule the agent has to go and read loses to the work, and that a rule loaded first, or a check that runs when the agent acts, holds. An insight filed as a lesson is that losing text. It has to become a change to the harness: what loads at start-up, a procedure, a check at the act, or how changes get made. One counter-case is open: a rule that was back in the agent's window at 09:29 UTC on 18 October still did not shape its plan at 16:02, about 6.6 hours later, and the operator caught it 83 seconds after that, so loading first is not a guarantee (Sections 1.5 and 3.1).

2. What the agent works out is being lost now, and most of it is knowledge. Nothing types it: the operator's memory database has a field for a record's level, and 0 of its 36,657 stored records had one filled in on 18 October (Section 2). On 18 October the assistant gave seven answers that were already in the operator's files, 0 of 7 in hand when it acted; it offered three of them, and part of a fourth, as new before finding them there (Section 3.2). 31 of 34 located losses happened across a boundary (a different agent, a later session, or a compaction, when an agent's long working memory is summarized to save space), with a median gap of 4.37 days from formation to the noticed loss. Sorted by this note's own definitions, 24 of the 34 are knowledge (rules, designs, facts) and 7 are insight, and all 7 insight losses crossed a boundary. The wisdom in the record is the operator's (Section 3.5).

Outside data points the same way: in public sessions from 12 other operators, 102 of 161 repeated corrections crossed a boundary and 1.6% of lessons got a harness edit (provisional; Section 3.6).



Exhibit 1. At a glance. An insight has to become part of the harness, admitted on evidence, or it is lost. **Every count rests on one rater; the loop is a design, not a result.**

3. A harness change alone is not enough. One of the seven answers, a check that blocks claims of absence that do not name their search, was already code in the coding agent's harness. The assistant, working on a platform that harness does not reach, proposed it as a new idea and found it already built minutes later. So each change has to ship with its record, the question it settled and the exchange behind it, and the record is served wherever its question comes back (Sections 3.2 and 5.6).

4. The harness already changes from experience, insight included, but by hand and without a record. Four fixes were tried by hand after things went wrong, and three persist; the agent landed two itself, with no record of the operator approving them, and the operator closed the fourth. Insight reaches the harness by hand too: of the seven answers of 18 October, one was an insight and was already in the code, and a second insight sits behind a design that is, but nothing records the insight behind them, so it cannot be judged, kept or served. A nightly pass built to automate learning from lived experience ran 43 nights and made 103 proposals, 87 of them prose edits; one was merged (Section 4).

5. The plan makes insight a recorded input. Capture the exchange when a position moves, make each record earn its level, turn a held record into a change to the harness that carries its record, and admit the

change only on fresh evidence or the operator's ruling, judged by something the change cannot edit. Held records would be served at start-up (approved, not built), at the stop for coding agents (designed, not built) and before a decision a wisdom record governs (proposed, not built). Recognizing the moment an insight forms is the open problem: designed in part and unsolved in part. The first test: the right record in the top 5 for at least 6 of 7 known cases, against 2 of 7 on the informal lookup of 22 October (Sections 5, 5.3 and 6).

6. Outside work supports the parts, not the whole. A recent account of a self-evolving harness (Wei et al., 13 October 2026) keeps revisions and admits them on fresh validation, the lifecycle the operator's plan of 16 August already described. In one study of software teams, lessons bound to the runtime beat readable text, 41.2% against 34.6%, and memory that rewrites the record does worse than the raw history. Studies already feed harness changes from user corrections captured in conversation; none found does so from insight or wisdom typed by level and admitted on fresh evidence or the operator's ruling (Section 7.6).

7. The loop is not built. 1,066 typed records wait for the operator's judgment and 15 are confirmed, but nothing serves them, and no harness change has come from one. Every count is of losses someone noticed, labeled by one rater, a Claude model reading Claude-written text, and that rater blurs knowledge and insight: its level matched the operator's on only 7 of 15 held records. It is one operator's practice, and the 34 located losses are 34 pairs from 32 distinct loss turns (Sections 3.5 and 8). The design fails if changes admitted from held records do not beat the current harness on their replay sets, or their forecasts do not come true (Sections 6 and 8).

Reading Exhibit 1. The exhibit runs left to right: the earlier paper's rule; what got lost one rung up; what the practice has tried; and the rebuilt loop with its status on 29 October. Seven answers on file counts the answers that turned out to be on file, not every answer given that day (Section 3.2). The 31 of 34 counts located losses only (Section 3.3). The 0 of 36,657 is a query of the operator's memory database re-run on 18 October (Section 2). The 43 nights and 103 proposals are from a review of the nightly pass on 17 October (Section 4.2). Dashed steps are not built.

ABSTRACT

This note extends "Rulebooks are theater" (7 October 2026) from rules to insight and wisdom. On the operator's own record, what the agent works out is lost: on 18 October seven answers on file were not in hand, and 31 of 34 located losses happened across a boundary. By the note's own definitions most of those losses were knowledge and 7 were insight, and the wisdom was the operator's. Changes to the harness so far came by hand, from lived experience and from insight, with no record tying a change to the insight behind it, and a nightly pass merged one change in 43 nights. The rebuilt loop captures an insight, makes it earn its level, turns it into a harness change shipped with its record, and admits the change only on fresh evidence or the operator's ruling. The loop is not built (the records exist, nothing serves them), and every count rests on one rater.

Keywords: language models, insight, wisdom, DIKIW, agent harness, harness self-evolution, rulebooks, memory, capture, admission, serving at the stop

CONTENTS

Summary
1 A written rule loses and a mechanism holds
2 The ladder, and where each rung lands
3 The evidence: two losses and 34 located cases
4 What has been tried
5 The rebuilt loop
6 Pass lines and status
7 Outside studies
8 Limits
Open questions
Disclosure
Sources
Appendix A. Roles and terms
Appendix B. Where each quotation and note number comes from
Appendix C. Records and receipts
Appendix D. The 18 October debate
Appendix E. Serving in detail

Summary

The plan beside outside measurements

Part of the plan	Outside measurements	This note	Agree?
Revise the harness from its own feedback, weights fixed	Wei et al. (Self-Evolving Agent Harness) (16 days; partly read): revisions persist across runs and are admitted only on fresh validation, with the final judge outside what a candidate may edit; admission may require a person's approval.	A held record becomes a proposed harness change, admitted on fresh evidence or the operator's ruling (proposed; Sections 5.6 and 5.7).	Supports the design; the paper's feedback is task results, not captured insight.
Feed the harness from what is said in conversation	Giampieri (Cortex) (20 days; partly read), Y. Zhao et al. (TRACE) (80 days; partly read) and Zhou et al. (TellOnce) (140 days; partly read) turn user corrections into rules, skills or context fixes; one sends them to human review, one keeps them on a measured gain.	Held insight and wisdom records, typed by level, feed harness changes (proposed; Section 5.6).	Partly: the feed exists for corrections; none found for insight or wisdom typed by level.
Evidence of loss (Section 3)	Public coding-agent sessions, 12 operators (computed 30 October 2026, provisional): 30 of 1,915 lessons got a harness edit that addresses them; 102 of 161 repeated corrections crossed a boundary.	31 of 34 located losses crossed a boundary; no harness change has come from a held record.	Supports the first; the boundary share is lower outside (63.4%).
Bind what is learned to the runtime, not to readable text	Du et al. (Relic) (33 days, v2 29 Sep; abstract plus skim), one study of multi-agent software teams: fresh-member behavioral correctness 25.4% with no inherited protocol, 34.6% with the lessons as readable text, 41.2% with lessons bound to the runtime (difference interval 0.7 to 15.6 points, 8 fresh members).	Changes land where rules hold: loaded at start-up, a procedure, a check at the act, or how changes get made (Section 5.6).	Supports the plan.

Part of the plan	Outside measurements	This note	Agree?
Keep the raw exchange, not a rewritten record	He et al. (Memory Tax) (4 days; main text read): 90 of 99 memory-system cells fell below naive retrieval over the raw history (75 of 99 against a simpler one-round baseline). R. Wang et al. (More Memory, Worse Agents) (10 days; partly read): abstractions persisted by a model fell below trajectory-only memory in 7 of 9 cells.	Capture keeps the raw pair, verbatim: the span of the exchange in which the position moved (approved 20 October, not built; Section 5.2).	Supports the plan.
Learn from more than one failure	Zhao et al. (13 days; abstract only): failure-derived experience repairs failed tasks but rarely yields sustained improvement on unseen tasks.	The nightly pass drew one record at a time and proposed prose; 1 of 103 proposals was merged (Section 4.2).	Consistent, on an abstract-only read.
A check enforced at the stop	Zhang et al. (Rethinking Verification) (4 days; main text read): an enforced completion gate gained 4.0 and 5.3 points on two benchmarks, a prompt instruction alone 3.3 and 3.9, a critic reading the full trajectory 9.5 and 3.9 (seven frontier coding models; the gate is 0.7 and 1.4 points above the instruction).	The lookup boundary: at the stop, a final answer on a question that has a held record is refused until the record is in view (proposed; Section 5.9).	Supports a check at the stop; serving an insight as evidence there is untested.
Recover a lost insight from past conversations and serve it	None found (not found at this search depth: 22 queries, 29 October 2026).	Pass line: 6 of 7 expected records in the top 5, against 2 of 7 on the informal lookup of 22 October (Section 6).	No outside counterpart.

What this means

- **Capture the raw exchange when a position moves,** not a lesson rewritten as an instruction (approved 20 October, not built; Section 5.1).
- **Let each rung earn its level.** Knowledge is scored as a forecast written in advance, insight is held if it raises how often that knowledge is right, and wisdom is held only on the operator's ruling (Section 5.5).
- **Turn a held record into a harness change and admit it on evidence:** a change to what is loaded at start-up, a procedure, a check at the act, or how changes get made, admitted on fresh validation or the operator's ruling, by a judge the change cannot edit (proposed; Sections 5.6 and 5.7).
- **Serve held records as evidence at the act:** at start-up (approved 27 October, not built), at the stop for coding agents (designed, not built) and before a decision a wisdom record governs (proposed, not built). It is never "are you sure?" (Sections 5.8 to 5.11).

1 A written rule loses and a mechanism holds, so an insight has to become a mechanism

Terms used in this note

Window	The text a model can see at one time.
Harness	The software around the model that feeds it context.
The stop	The moment the model gives a final answer.
Data	The raw record of what happened: a message, a commit, a log line, a reading.
Information	Data that carries its date, speaker and source, so it can be counted, joined and traced.
Knowledge	A statement of what is settled or measured that a later reader should be able to rely on without reopening the source. A rule is knowledge of the kind that says what to do.
Insight	A reframing or connection that was not stated before in the exchange. A plain correction, a rebuke, an instruction or a one-line opinion is not an insight.
Wisdom	A priority or principle that governs a whole class of future situations and that rests on something specific. A wisdom record has to name what it rests on.
Lived experience	What an agent or the operator logged as earned from the work, in either valence: a scar (something that went wrong) or a positive. It replaced the miss ledger, now retired.
Boundary	What a loss crossed: a different agent or channel, a later session in the same channel, or a compaction inside one session.
Candidate and held record	A candidate is a typed record not yet judged; a held record is one the operator has confirmed.

1.1 What the rules paper claims

“Rulebooks are theater. Boundaries and incentives are not.” (background paper 3 of the series “How Do We Use This?”, revised 7 October 2026, 22 days before this draft) is the operator’s earlier paper. It argues from one practice, and says it reports a mechanism and not a rate, that a written rule the agent has to go and read competes with the work for its attention and loses. The practice carries about 125 rule-bearing items whose adherence has never been measured. One rule was missed seven times in a day, which the record counts as one act, and two tracked error classes came back in the very next session in 48 of 75 intervals, against about 40 expected from prevalence alone.

What held was different in kind: a rule loaded into the agent’s context before the session’s first token, which does not have to win anything, and boundaries that refuse an action at the moment it is attempted and record the refusal. The paper also says that a lesson written into the record is a rule of the first kind, one the agent must go and read, and that of eight lessons ratified into the practice’s merged store none is currently measurable, because promotion dropped the citation linking a lesson to the miss that produced it.



Exhibit 2. A written rule loses and a mechanism holds, so an insight has to become a mechanism.

Reading Exhibit 2. The two columns set a rule beside an insight on what each says, when it can be checked, how it is lost and what holds it. The last row is this note's claim: a rule holds when it is loaded first or enforced at the act, and an insight holds only when it becomes a change to the harness of the same kinds, admitted on evidence, with the raw exchange kept as the record of why. The strip at the foot marks the open case: a rule back in the window at 09:29 UTC that did not shape a plan at 16:02 (Section 1.5).

1.2 A harness that evolves

The harness is the software around a model: the loop, the context it builds, the memory it keeps, the tools and the checks. The operator's plan of 16 August already described a nightly step that proposes edits from the day's misses, replays a regression set, gates them and allows one merge a day by the operator (Section 4.2).

Wei et al. (Self-Evolving Agent Harness) (16 days; partly read) describe the same lifecycle in general form. Their harness uses feedback from its own runs to revise its instructions, memory, workflows and executable components while the model's weights stay fixed. Each round diagnoses, proposes, assesses and admits; "Accepted revisions are stored and loaded in a subsequent run". Two requirements make it a lifecycle rather than a log. Feedback has to produce revisions algorithmically, with the mechanism kept for the next round, and accepted revisions have to persist. Admission is the safeguard: a candidate is accepted only if it beats the current harness on fresh validation tasks, and the final judge sits outside what a candidate may edit. Admission may require a person's approval. The paper also says what does not count: appending logs or compacting context under fixed rules does not revise the harness.

The update targets run in an order the paper draws as a road: instructions and prompts; context, memory and reusable procedures; workflows; and last, executable components and the runtime. A change high on that road alters what the model is shown. A change at the top alters what the system does whether or not the model reads anything.

1.3 Why an insight has to become a mechanism, not a rewritten lesson

A rule says what to do, and it can be checked at the act: the action either followed it or did not, so a boundary can refuse the action and record the refusal. An insight is different in kind. It is a settled answer to a question, such as "No, the lead is the counterweight"[1], and nothing in the act is there to check it against. Its use depends on recognizing that the question has come back, often in other words.

The obvious repair is to rewrite the insight as a lesson, a short instruction that says what to do next time. That turns it into rulebook text, the first kind in the earlier paper, and it discards the part that made it an insight: the question it settled and what it rested on. The newest studies find that this rewriting is where the damage is done. In He et al. (Memory Tax) (4 days; main text read), 90 of 99 memory-system cells (11 systems, 3 benchmarks, 3 backbone models) fell below naive retrieval over the raw history, and raw episodes stayed competitive. R. Wang et al. (More Memory, Worse Agents) (10 days; partly read) found abstractions persisted by a model below trajectory-only memory in 7 of 9 cells. The older Zhang et al. (memory) (169 days, v2 29 Aug; full text, sections read) found that memory consolidated by a model first helps and then degrades, sometimes below no memory, while raw episodes hold up.

The same studies point to what does hold. In multi-agent software teams, Du et al. (Relic) (33 days, v2 29 Sep; abstract plus skim) measured fresh-member behavioral correctness at 25.4% with no inherited protocol, 34.6% with the lessons as readable text and 41.2% with lessons bound to the runtime. The earlier paper found the same split for rules: text the agent must read loses, and what is loaded first or enforced at the act holds. So an insight is made part of the harness the way a rule is: as a change to what is loaded first, to a procedure, to a check at the act, or to how such changes get made. The raw exchange is kept beside the change as its evidence, so the change can be traced to the moment that produced it and retired when it stops paying. These are agent-task results, not an operator's own conclusions, so they bear on the design (Section 7) and do not prove it.

1.4 What this paper adds

The earlier paper stops at the rule: where to load it and where to put a boundary. This note takes the argument one level up and joins it to the account of a harness that evolves. It names what is lost, the upper rungs of a ladder from data to wisdom, and where each would land in the harness (Section 2); shows the loss in the operator's own record (Section 3); reports what the practice has already tried, by hand from lived experience and from insight, and with a nightly improvement pass (Section 4); and sets out the rebuilt loop, which captures insight and wisdom when they occur, makes each record earn its level, turns held records into harness changes and admits them on evidence (Section 5), with pass lines written before anything is built (Section 6).

The conclusion carries over. On 18 October the operator said of the usual remedy, "it wont help throwing it in memory"[2], and the plan does not file an insight as one more memory row to be read. The object changes: the earlier paper's is a rule, this note's is an answer settled once that has to change the system before its question comes back.

1.5 The tension with Loss one, left open

The earlier paper says a rule loaded before the work starts fires. Section 3.1 records a rule that was in the window and did not govern the next action. At 08:58 UTC on 18 October the assistant committed to being thorough and to stating what it searched, and the rule was written down. The harness (the software that feeds the model its context) put it back into the window at 09:29. At 16:02 the assistant planned this very paper with the literature as a supporting track and ran no outside search first, about 6.6 hours after the re-injection and 7.1 hours after the rule was written. The operator caught it 83 seconds later.

Neither paper tests what the difference is. Candidate readings are that a mid-session re-injection differs from loading before the first token, that distance in a long window matters, that the display of the rule was cut so the coverage clause may not have arrived in full (Section 3.1), and that a new task is not an act the rule had been checked at. One case in one session cannot separate them, so this note states the tension and does not resolve it. It bears on the plan: if context alone does not reliably govern, serving an answer at start-up (Section 5.8) may not be enough, which is why the plan also serves it at the stop (Section 5.9) and turns held records into changes that act without being read (Section 5.6).

2 Machines keep data and information; knowledge, insight and wisdom went untyped, and are still not served

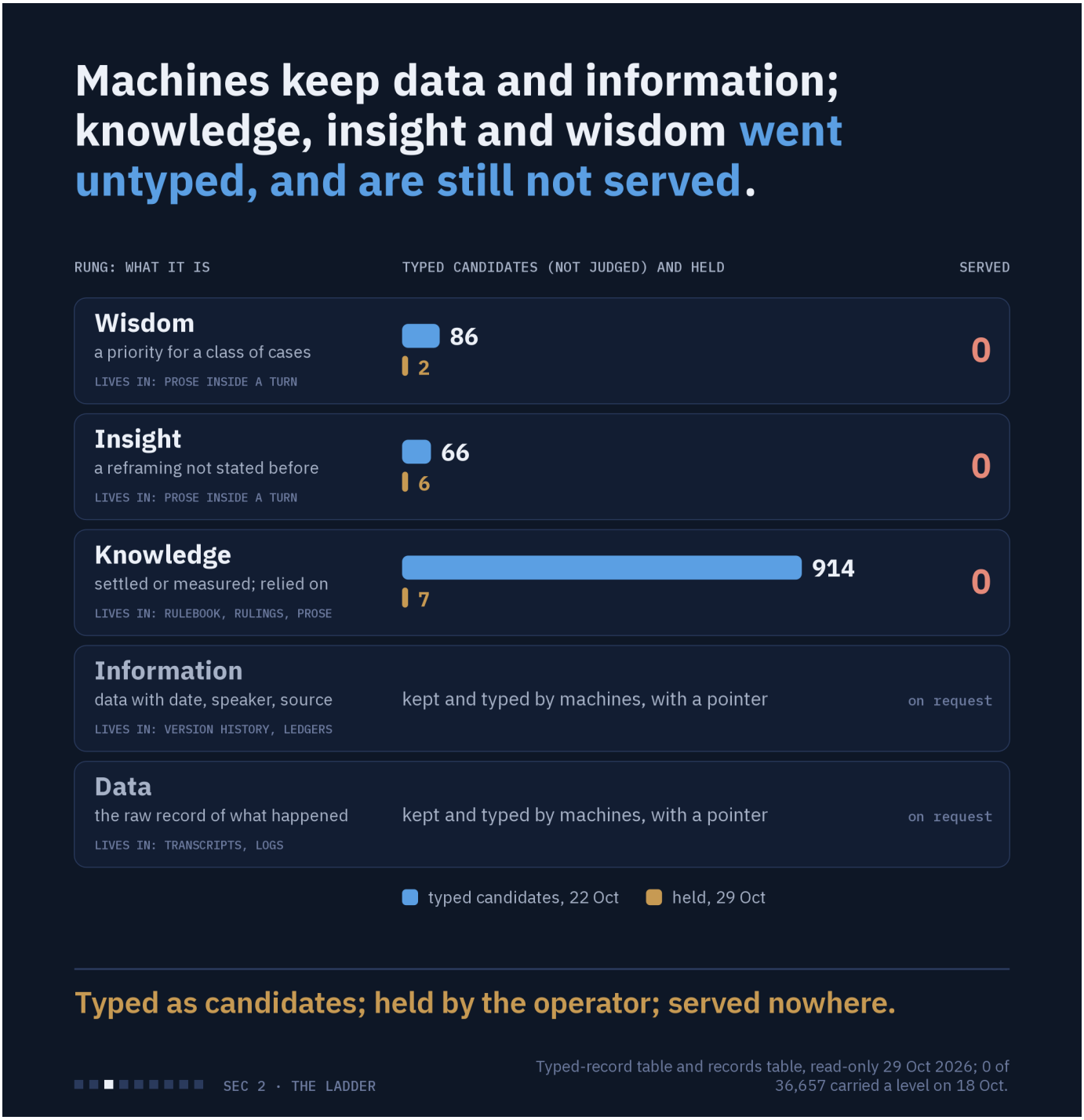


Exhibit 3. Machines keep data and information; knowledge, insight and wisdom went untyped, and are still not served.

Reading Exhibit 3. The ladder runs from data at the bottom to wisdom at the top. For each rung the exhibit gives where it lives today, whether it is typed and whether it is served. Machines keep and type data and information. Knowledge, insight and wisdom went untyped (0 of 36,657 stored records carried a level on 18 October) and on 29 October are still not served. The counts are 1,066 typed candidates (914 knowledge, 66

insight, 86 wisdom; 22 October; not judged) and 15 held records (7, 6 and 2; 29 October). One short example of each upper rung is a held line in the operator's words.

The operator's ladder has five rungs: data, information, knowledge, insight and wisdom (DIKIW in the operator's files). The operator said of it, "the learning ladder and the DIKIW are basically the same"[3]. It is a variant of the ladder in Ackoff (via a secondary source) (1989; snippet only), which puts understanding where the operator puts insight, so each rung needs a definition of its own. The definitions below are the level rules the operator's retrieval build used on 7 October and the reading rules applied to the 30 days to 22 October. The wording of the second set is the assistant's, and the operator has not judged how it splits the levels.

2.1 Data

What it is. The raw record of what happened: a message, a commit, a log line, a reading. **Where it lives today.** In transcripts of chats, coding-agent sessions and the assistant's own sessions, in the message log between agents, in version history and in chat exports. Machines write it and append to it, and each item points back to where it came from. **Losing it** looks like a gap in the record, not a forgotten fact. The 22 October reading found the backup of coding-agent transcripts ending at 03:03 UTC on 18 October and the newest chat export dated 13 September, so those periods were out of reach. **In the operator's words.** A held line is data before it is anything else: "Youre losing the KIW and that's what I've really been trying to sort out"[4] first exists as a message sent at 10:05 UTC on 18 October, in a transcript, with a position in it.

2.2 Information

What it is. Data that carries its date, speaker and source, so it can be counted, joined and traced. **Where it lives today.** In git history, a table of the operator's rulings (73 rows read on 22 October), the miss ledger and the streams in the operator's memory database. **Losing it** looks like an entry that no longer joins to where it came from. The earlier paper reports the case: eight ratified lessons, none measurable, because promotion dropped the citation that linked each lesson to its miss. The entry survives and the information does not. **In the operator's words.** "I think the outside model, interacing with the frontier using the insights from above and our previous convos, is right"[5] is information once it carries its date (18 October), its speaker and a pointer to the message, as the held record does.

2.3 Knowledge

What it is. A statement of what is settled or measured that a later reader should be able to rely on without reopening the source: a ruling with its scope, a standing rule, a measured result with its number, a decision with its reason. **Where it lives today.** In the rulebook as the earlier paper counted it on 7 October, which is the two start-up documents every session loads (122 headed sections) and a memory store of 124 files, and in the rulings table. This is the rung the earlier paper is about, because a rule is knowledge of the kind that says what to do. **Losing it** looks like the rule that was in the window and did not govern the next action (Section 3.1), or the rule missed seven times in a day. **In the operator's words.** "I think shipped errors is a stupid metric for us"[6] is a ruling on which measure to use. A reader should not need to reopen why.

2.4 Insight

What it is. A reframing or connection that was not stated before in the exchange: two things shown to be the same motion, a cause replaced by a better cause, a problem restated so that a new solution appears. A plain correction, a rebuke, an instruction or a one-line opinion is not an insight. The reading rule credits whoever formed it. Of the 66 insight candidates, 13 are the operator's, 34 the assistant's and 16 a coding agent's. **Where it lives today.** As prose inside a turn: a reply, a commit body, a docstring. Nothing typed it until 22 October, and nothing carries it to the assistant at the act. **Losing it** looks like a settled answer coming back as new. On 18 October the assistant gave seven such answers, and two July ideas were lost until the operator supplied them again, 88 and 94 days after they were written (Section 3.3). **In the operator's words.** "No, the lead is the counterweight"[1] names the working counterweight as the human lead.

2.5 Wisdom

What it is. A priority or principle that governs a whole class of future situations (what to do first, what never to do, what to measure instead) and that rests on something specific: a pattern of cases, an outcome or a ruling applied again. A wisdom record has to name what it rests on. If it cannot, the reading rule labels it knowledge or insight, and 4 of 90 wisdom calls were demoted that way. **Where it lives today.** Nowhere typed as a level beyond the candidate set and two held records. The 7 October build counted 161 wisdom records by rule and its wisdom rule was weak (5 of 10 rulings misassigned). Of the 86 wisdom candidates, 15 carry a pointer to an item they rest on and 71 do not. **Losing it** looks like a frame dropped between sessions. By the serve plan's own account, on 27 October the assistant framed the operator's company by a measure other than its stated purpose after loading its memory, although the line that states the purpose was already in that memory. The memory search had been phrased in the wrong frame and returned the wrong rows. That is one session's report, not a count. **In the operator's words.** "all you can do is influence an LLm"[7] sets a priority for a whole class of design choices: work on what enters and leaves the model, not on commands inside it.

2.6 What separates knowledge from insight from wisdom

The three upper rungs are told apart by what it takes to hold a record at that level, not by how it sounds. Three tests come from the loop the operator approved on 18 October ("the updated concept for the loop"[8]) and from the reading rules.

- **Knowledge** is what a later reader would act wrongly without, or repeat work without, and it can be scored. The design writes it as a forecast in advance (if a lesson class is loaded, misses of one type fall by at least a stated amount within a stated number of weeks, with a probability) and scores it when the window closes.
- **Insight** is held if it raises how often the knowledge forecasts are right. It is tested by what it changes in knowledge.
- **Wisdom** is a priority judged against an outcome the operator values, and only the operator rules on it. The test that can be observed is what it changed: an action, a rule, or how rules get made.

The operator drew the last line himself when he asked where a scored forecast sits: "is that just knowledge, or insight? It's not wisdom."[9] The assistant reads that as follows: a score can show that a forecast was right, and

cannot show that the priority behind it was the right one to hold. That reading is the assistant's. The levels on the 15 held lines were proposed and then confirmed by the operator on 27 October. The levels on the 1,066 candidates were assigned by a reader (a Claude model), and the operator has not judged them: the split is 86% knowledge, which the 7 October funnel (3,043 knowledge, 570 insight, 161 wisdom, levels not nested) cannot confirm, because the unit and the rule differed. The boundaries between levels are the weak point of any such scheme. In the 7 October build a check of 10 insight records found 6 misassigned as given and 2 after plain corrections were dropped, and the wisdom check found 5 of 10.

2.7 The state of the record

The counts below are from the operator's memory database, re-read read-only on 29 October (the table of typed records, grouped by status and level). Candidates and held records sit in one table and are not disjoint: a text match on the first 40 characters finds three of the 15 held lines also among the candidates, so the two counts are not added.

- **18 October.** 0 of 36,657 stored records carried a level. The column that would hold it was empty, and a re-read on 29 October finds it still empty.
- **22 October.** 1,066 typed candidate records (914 knowledge, 66 insight, 86 wisdom) were loaded from a reader's pass over 30 days, which read 1,371 of 3,667 reading units (37%). Each carries a verbatim span and a pointer. The operator has not judged them, so none is held.
- **29 October, 16:44 UTC.** 15 held records (7 knowledge, 6 insight, 2 wisdom) were loaded from the operator's own lines, confirmed at their levels on 27 October. Nothing is served to any model at start-up or at the stop.

The ladder is therefore partly typed and not yet used. What typing exists is a floor of what the operator's record holds, and Section 5 sets out how a record would be captured, earn its level and be served.

2.8 Where each rung lands in the harness

The loop of 18 October judges a wisdom record by what it changed: an action, a rule or trigger, or how rules get made. Read against the road in Section 1.2, each rung has a natural landing place. Data and information are already kept by machines, in transcripts, logs and version history, and the harness reads them on request. Knowledge lands in what is loaded at start-up: a ruling with its scope, a measured result with its number, a standing rule. Insight, a reframing of a recurring problem, lands one step further: a procedure the agent follows, or a check at the act that catches the problem the insight named. Wisdom, a priority for a class of situations, lands where changes are made: which gate a class of records must pass, which decisions go to the operator, what is measured instead. Section 5.6 turns this mapping into the loop's output, and Section 4.1 shows changes of these kinds already made by hand.

3 The answer was on file and not in hand at the act: two losses and 34 located cases

Two losses organize this section. The assistant grouped three of the operator's messages[10–12] into them: a conclusion reached in the window does not govern the next action (Loss one), and what was on file was not in hand at the act (Loss two). The record has no statement that the operator takes that frame (Open questions), so each loss is argued from timestamps and files, not from what the assistant said about itself. The third part counts how often the second shape turned up in the operator's history.

3.1 Loss one: a rule in the window did not govern the plan

At 08:58 UTC on 18 October the operator wrote "i need you to be thorough"[13], and the assistant committed to being thorough and to stating what it searched. At 09:29 the harness put that rule back into the assistant's window. At 16:02 the assistant planned this very paper with the literature as a supporting track and ran no outside search first; the operator caught it 83 seconds later, and the assistant ran six web searches in the same reply. Nothing was compacted and no session ended in between, so the rule was carried and the next action on a new task did not reflect it. That is the open tension of Section 1.5: a rule in the window that did not govern a plan 6.6 hours after its re-injection. It cost one plan. Appendix D.6 gives the timestamped sequence, what the record does not support, and the outside studies.

3.2 Loss two: the seven answers were on file

Late in the day the operator said "right, so this has all been out there"[14], and the assistant produced a table of seven things it had offered that day and where each already was. Checked against the files (Table 1), it offered R1, R3 and R6, and partly R4, as new before finding that they existed; R2, R5 and R7 first appeared in its replies as retrievals. It reached most of the seven only after the operator pushed. Nobody counted the answers that turned out to be new, so seven is a count of losses found, not a rate. Appendix D.6 gives the corrections to the assistant's table and its account of why.

Table 1. The seven answers that were already on record. "First appeared" says how the idea first showed up in the assistant's replies that day; the superscript numbers are rows of Appendix B; the files and commits behind each answer are described in Table C2.

Id	The answer	Where it already was	First appeared
R1	An absence claim must name the space searched.	A clause of the operator's stop check, first committed 3 Sep 2026.	Offered as new[15]; found already built[16].
R2	A push should be a work order, not 'are you sure?'	The documentation of the operator's push-verification code (first committed 2 Sep) and a point in the operator's scoreboard file.	Retrieval, in reply to the operator's yes/no question[17], not to a work order (the next reply cites the scoreboard point[18]).

Id	The answer	Where it already was	First appeared
R3	A retrieval test needs a pass line set in advance.	The operator's ruling of 3 Sep 2026, quoted in the test's scoring code (first committed 11 Sep).	Offered as new[19]; the assistant says later that it was already set[20].
R4	An outside model that challenges at the stop moment: one engine, four transports.	A coding-agent session of 31 Aug to 3 Sep, and seven engine modules committed 11 Sep to 1 Oct.	Partly offered as new[19,21,22]; the operator had proposed the outside model[2,17].
R5	How to make it work for Chat: five architectures, one on a harness that can be hooked.	A section of the operator's plan, 'Two masses, two owners', written 2 Sep 2026.	Retrieval[23].
R6	The learning ladder and the data-to-wisdom ladder; Bateson offered as prior art for looping up the chain (from its training, not read for this note).	A session of a second agent of the operator's, 22 Jul 2026: the climb and the loop as 'the same motion seen twice'.	Offered as new[24]; conceded[25]. Matches only partly (see the corrections below the table).
R7	Score on loop depth, not on wisdom.	A commit of 15 Jul 2026 and the operator's tiering code.	Retrieval[26].

Three of the seven were already part of the harness. R1, R2 and R7 were already code in the operator's harness when the assistant reached them. R1 is the clean case. It is a clause of the operator's stop check, first committed on 3 September, that blocks a claim that something does not exist unless the claim names the space searched. At 06:10 UTC the assistant proposed exactly that check as a new idea, and at 06:13 it found the check already built. The clause runs in the coding agent's harness, at that agent's stop. The assistant works on a platform that harness does not reach (Section 5.12), and nothing carried the clause, or the question it settled, to the assistant when the question came back. R2 and R7 were also in code, and the assistant found them by retrieval after the operator asked. So a change to the harness is necessary and not sufficient: it acts only in the harness it lands in, and it does not carry its reason. The rebuilt loop therefore ships every change with its record, and serves the record wherever its question returns; Section 5.6 takes R1 through the loop.

3.3 The 34 located losses

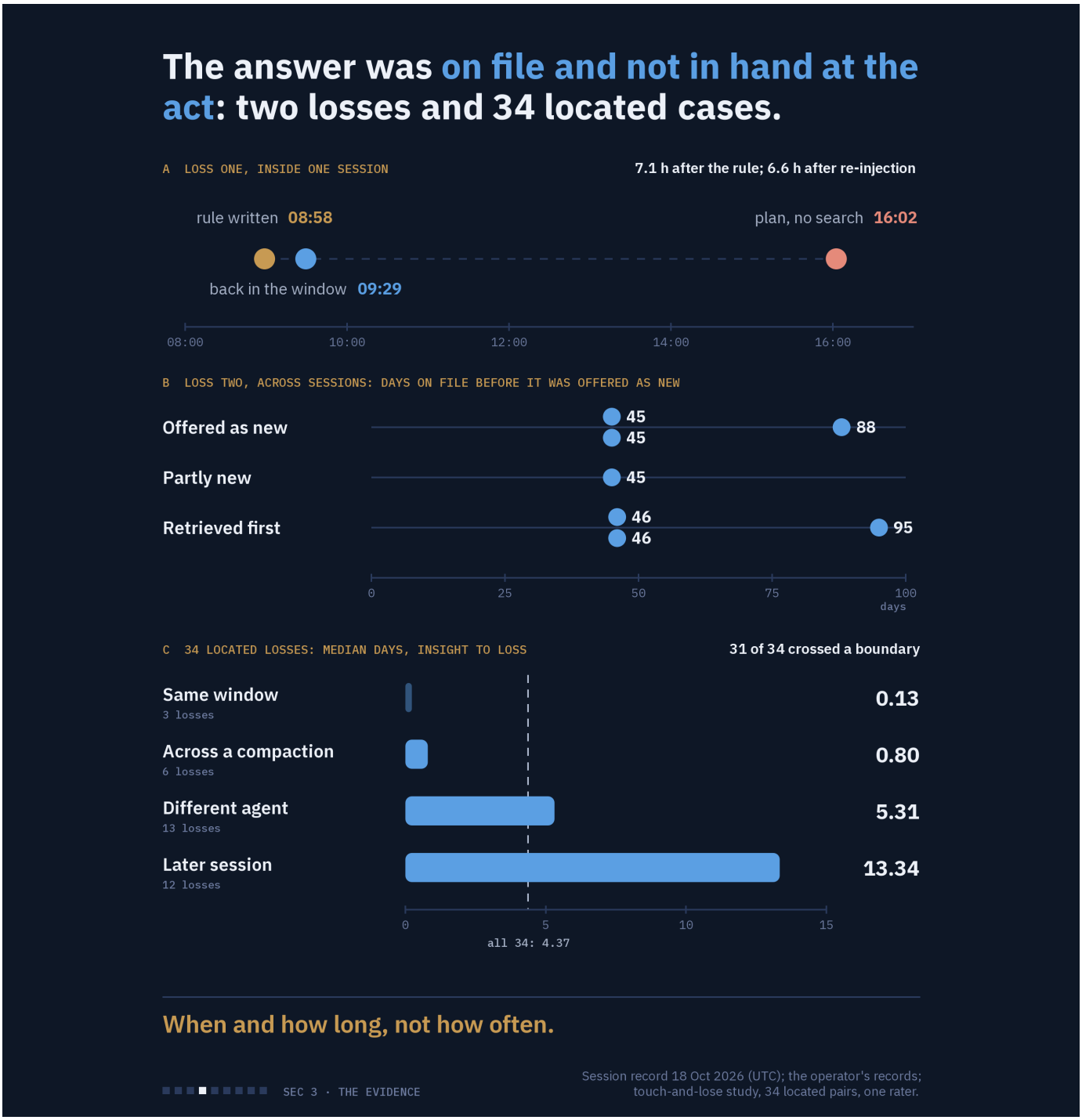


Exhibit 4. The answer was on file and not in hand at the act: two losses and 34 located cases.

Reading Exhibit 4. Three panels, left to right. The first is the timeline of Loss one on 18 October (UTC): the rule written to the operator's memory at 08:58, put back into the window by the harness at 09:29, and the plan at 16:02 that ran no search, about 7.1 hours after the rule and 6.6 hours after the re-injection. The second is Loss two: the seven answers that were already on file, each drawn from the date its source was written to 18 October (not to 29 October 2026), from 45 to 88 days for the three offered as new, 45 for the one offered partly as new and 46 to 95 for the three that first appeared as retrievals. The third is the 34 located losses by

what the loss crossed: 31 of the 34 crossed a boundary, the median gap was 4.37 days, and the medians were 0.13 days inside one window (3 pairs) and 0.80 across a compaction (6). The panels show when and how long, not how often (Section 3.4).

The assistant read the corpus: 8 chat export files (290 unique conversations, 17,332 messages, 9 February to 3 September), the coding agent's transcripts (2,450 unique session or sub-agent files, 197 of them main sessions, 12 May to 18 October), 285 of its own sessions (2,922 text turns, 7 to 18 October) and one commit. It searched with a lexicon frozen before the precision read (14 families of the operator's phrases such as "we've gone over this" and 8 of the agents', such as "I re-derived it"), grouped hits within six messages into events (347 events; 337 remain after dropping duplicates from forked sessions; the hit counts are in Appendix C), and read all 337 by hand. Of these, 104 were true losses of something the agent had worked out, labeled insight losses at the time under a wide rule (a reframing, decision or finding re-derived as new, contradicted without citation, or re-supplied by the operator; Section 3.5 sorts them by level) and 17 were rule bypasses, where a known rule wasn't applied. A random sample of 48 hits gave 18 true, a hit precision of 37.5% (Wilson interval 25.2 to 51.6%). The four known losses all fired the lexicon, a check on four cases and not a recall estimate.

For the insight losses it then looked for the moment the insight formed. That search was systematic only for the high-confidence events and a review of the first 24 medium ones. It located 34 formation-to-loss pairs (25 high confidence, 7 medium, 2 low) covering 32 distinct loss turns (31 of them among the 104 insight losses; the 32nd is a rule bypass kept because the rule is the insight). The other 73 insight losses have no located formation: some were searched for and not found, and some were not searched. The median gap from formation to loss is **4.37 days** (bootstrap 95% interval 1.09 to 10.04; range 0.08 to 94.24; 5.31 days for the 25 high-confidence pairs). 11 of the 34 are under a day, 8 are 1 to 7 days, 9 are 7 to 30 days and 6 are over 30. By source the losses are 21 in chat, 11 in coding-agent sessions and 2 in the assistant's sessions (counts, not rates; the windows differ).

The boundary is what the loss crossed. 31 of the 34 crossed one: 13 a different agent or channel, 12 a later session in the same channel and 6 a compaction inside one session; 3 happened inside one context window.

Table 2. The 34 pairs by what the loss crossed. Gaps are days from formation to loss. All counts are lower bounds from a lexical search, with a formation search that was systematic for only part of the events; one rater. The 34 pairs are not independent: four pairs share a loss turn (two couples) and four others share a formation turn (two couples), so the last row counts each distinct loss turn once.

Boundary	Pairs	Median gap (days)	Range (days)
Different agent or channel	13	5.31	0.13 to 94.24
Later session, same channel	12	13.34	0.47 to 47.03
Across a compaction in one session	6	0.80	0.11 to 3.43
Same context window	3	0.13	0.08 to 0.16
All pairs	**34**	**4.37** (95% 1.09 to 10.04)	0.08 to 94.24

Boundary	Pairs	Median gap (days)	Range (days)
Each of the 32 distinct loss turns once	32	**3.29** (cluster bootstrap 1.09 to 10.04)	same median whichever gap a shared loss uses

The 34 pairs are not independent: four pairs share a loss turn (two couples) and four others share a formation turn (two couples), eight pairs in all. Counting each of the 32 distinct loss turns once, the median is **3.29 days** with the same interval from a cluster bootstrap (the interval depends on the bootstrap scheme and is discrete; Appendix C and Section 8 give the details); counting each formation once at its shortest gap, it is 4.23.

In the same window, one agent wrote that it had the answer right and then talked itself into reopening it (3.2 hours apart): “I had it right and then talked myself into re-opening it.”[27] Across a compaction, an agent asserted the opposite of its own earlier finding (2.7 hours apart): “I had it right. Before the compaction. In writing.”[28] The two that end in the 18 October debate are the longest: the 15 July commit and the 22 July session of a second agent on one side, the operator’s 18 October turn[3] on the other, 94 and 88 days apart, across an agent. The earliest formation is an off-corpus Notion note, dated only to the afternoon of 13 May. The in-session case of Section 3.1 is not among the 34: the study set aside the debate session’s own meta-discussion after the seed turn.

3.4 What it can and can’t say

The count is of losses someone noticed, not a rate. The lexicon finds the phrases the operator and the agents use when something is lost, so a loss nobody flagged in those words is invisible. One rater did all the labeling, the assistant, a Claude model reading Claude-written turns, and boundary and confidence calls are judgments. The median is conditional on a loss having been noticed, on its event having been searched for a formation (systematically for only part of the events) and on the formation being findable: a formation was located only when the loss turn quotes it or a targeted search found it. The channels cover different windows, so the gaps are not times to loss and the counts are not rates.

The assistant’s reading: the lexicon is built from phrases that point back to earlier sessions, so in-window losses may be undercounted, and 3 of 34 shows what the search can see, not how rare in-window loss is. July and August run higher than May, June, September and October, which may partly reflect agents logging misses and quoting prior text in those months (a hypothesis the study did not test), so it need not mean a higher rate of loss. Several formations sit in Notion or Slack, which the study did not read.

What the evidence does say is narrower than the frame. Loss one is one case in one session, in a setting where the rule had been re-injected. Loss two is seven answers from one day, with no denominator for the answers that turned out to be new. The 34 pairs show that, among losses noticed and traced, most crossed a boundary and the typical gap was days, not months. None of the three measures a rate, and none tests whether serving the earlier answer would have changed the act: Sections 5 and 6 give the plan for that test, and as of 29 October 2026 nothing is served; Section 8 states the limits. Open questions has the readings that cut against the operator’s position.

3.5 What level was lost

The labels above are the assistant's, and the outside review of 30 October found that they call rules and decisions insight, which this note's Terms table classes as knowledge. The assistant re-sorted the 34 located losses, the seven answers of 18 October and the 15 held records against the Terms definitions; Appendix C lists the file, with a call and a reason for each item for the operator to overrule. Of the 34 losses, 24 are knowledge (rules, designs, facts), 7 are insight, 1 is wisdom, 1 is a one-line opinion and 1 cannot be recovered. All 7 insight losses crossed a boundary (2 at a compaction, 3 in a later session, 2 in another agent), with a median gap of 10.04 days (0.11 to 94.24); two of them share one loss. Of the seven answers of 18 October, R2 and R6 are insight and the other five are knowledge. 5 of the 7 insights were formed by an agent.

The wisdom is the operator's. The one wisdom loss and both held wisdom records are his lines, and on 30 October he agreed with all three wisdom calls (3 of 3), in his words "I'm the one dropping the wisdom ha". On the 15 held records the assistant's level matches the operator's in 7 of 15. The re-sort is one more pass by the same kind of rater, so it is evidence for Section 5.3 rather than a correction of it: a model sorting its own exchanges by level blurs knowledge and insight, which is why the design sets level by a forecast test or the operator's ruling.

3.6 Outside data: public coding-agent sessions

The note's own record is one practice. To set it beside others, the same question was run on public session transcripts posted to Hugging Face (computed 30 October 2026; sessions dated December 2025 to 17 October 2026): 2,342 sessions from 12 operators, 9,248 operator turns judged by a model, 2,076 corrections and 161 repeats, where a repeat is a correction the operator gives again. The measure differs from Section 3.3, which counts losses someone noticed; this one counts repeated corrections, mostly misses, so it bears on knowledge more than on insight. Three results. First, lessons rarely become harness changes: 30 of 1,915 lessons (1.6%) were followed by an edit to a harness file that addresses them, and 2 of the 161 repeats had one in between. Second, most repeats cross a boundary, though fewer than in the operator's record: 88 of 161 (54.7%) came in a later session and 14 more after a compaction, so 102 of 161 (63.4%) crossed a boundary as this note defines it, against 31 of 34 here; the share runs from 0% to 74% by operator, with how each one splits work into sessions. Third, most repeats come quickly: the median gap is 0.27 days overall and 11.5 days for repeats across sessions (interval 4.0 to 16.5), which contains this note's 4.37. Whether a harness change stops a lesson recurring cannot be tested on these data: 1 of 30 lessons that got an addressing edit repeated, against 88 of 1,774 that did not, too few to tell apart. All of it is provisional: one model judge (85% on a 20-item hand check), a word-overlap matcher whose recall was not measured, so repeats are undercounted, and 11 qualifying operators not computed. Appendix C lists the files.

4 The harness has been changed by hand, from lived experience and from insight; the nightly pass meant to do it proposed prose

The practice already changes its harness from experience, in three ways: by hand from lived experience (so far, only scars), by hand from insight, and with an automated pass meant to run on lived experience. This

section reports each against the requirements of Section 1.2: feedback that produces revisions, persistence, and admission on evidence with the judge outside what is edited.

4.1 By hand, from lived experience

The practice's learning input is lived experience: what an agent or the operator logged as earned from the work, scars and positives alike. The operator ruled three times (17 July, 31 July, 24 August) that positives get the same process as scars, and the code now counts both; the miss ledger that came before is retired, by the operator's word on 27 August. Four changes in the record were tried by hand from logged scars, and three persist; none came from a positive. Two are rules loaded first, the form the earlier paper found to fire at start-up (though Loss one shows such a rule does not always govern the next act); the other two are a check at the act and tests, the forms it found to hold. On 10 July a forced diagnosis check was locked into the start-up documents: before asserting or acting on a consequential diagnosis, the agent names what would prove it wrong and reads that first. It came from catalogued wrong-diagnosis misses, 15 by one count of the register; the count depends on how the register is filtered. The same day a standing rule set which calls go to the operator and which the agent decides. It came from one logged scar, recorded in the since-retired miss ledger as a repeat of an earlier routing miss, after the operator told the agent it was supposed to be making informed decisions. On 23 September an extension of the guard that makes the agent read before it claims, to check the repository before a claim that something still needs building, was proposed after misses in September; the operator closed that change the same evening, and it never reached the harness. On 28 September replay tests were written for the two largest miss classes; the operator merged them, and they entered the test corpus on 30 September. Table C4 in Appendix C gives the receipts.

Against Section 1.2, admission is where these fall short. The agent wrote the two rules of 10 July and landed them itself in the same session, and no record shows the operator approving either, so the editor and the judge were the same agent, which Section 1.2 rules out. The operator merged the replay tests without reviewing their content, which a second agent reviewed. The one change the operator did judge, the guard extension, he rejected. In every case the feedback was a scar someone noticed and logged and the revision was written by hand, so the loop ran only when someone ran it. Whether each change reduced its miss class has not been measured in this note. None of these four came from a typed insight or wisdom record, because no such record existed until 22 October (Section 2.7); insight reached the harness by another route, by hand and untraced (Section 4.3).



Exhibit 5. The harness has been changed by hand, from lived experience and from insight; the nightly pass meant to do it proposed prose.

Reading Exhibit 5. The left panel lists four changes tried by hand from logged scars, with their dates, their kind and what became of each: two rules loaded first in the start-up documents (10 July), which the agent landed itself; a guard extension to check the repository (23 September), which the operator closed; and replay tests for the two largest miss classes (28 and 30 September), which the operator merged. The right panel is the nightly improvement pass over 43 nights: what it was asked to do, what it proposed and what was merged. Each panel's input was scars. The dashed box notes the third route: an insight (R2) and a design built on one (R7) put into the code by hand, with no record of the insight (Section 4.3).

4.2 The nightly improvement pass

The operator's plan of 16 August describes a nightly step with the shape of Section 1.2: pick the day's logged experience (misses, in the plan's words), re-solve them, propose candidate edits, replay a regression set, gate,

and allow one merge a day by the operator. A review of 17 October read the built pass end to end. Over 43 nights it was given a real experience on 2 nights; on 4 early nights it was asked to improve one start-up file, and on 37 it only resumed open work. It produced 103 proposals, and 87 of them (84.5%) were prose edits to two start-up files. None was code, a validator or a test. One change was merged, a 3-line metadata edit.

Each failure maps to a requirement. The pass drew one record at a time, so it could not see a class, and it read a frozen copy of the start-up files, so it could not see the record of what happened. Its grader checked the form of an edit and never saw the experience the edit answered, so admission did not rest on evidence that the change helped. And its output space was an edit to a file, so its revisions stayed prose in start-up files, the same form as the two rules of Section 4.1 and no stronger, and none added a check, a test or a validator, the forms the earlier paper found to hold. On 16 October the pass itself named the right fix for the miss it had worked for weeks, a resolver that reads the live value instead of accepting a typed one; that fix sat outside what it could edit. The outside record agrees: Zhao et al. (13 days; abstract only) found that failure-derived experience repairs failed tasks but rarely yields sustained improvement on unseen tasks.

4.3 By hand, from insight, with no record

Misses are not the only input. Insight already changes the harness, by hand. Two of the seven answers of 18 October carry it into the code: R2, itself an insight (the reframing that ignorance is a work order and not a verdict), has shaped how the challenge is dispatched since 2 September, and R7, a design that scores a gate by loop depth rather than by wisdom and is knowledge by the Terms test, was committed on 15 July with the R6 connection, between the operator’s ladder and Ackoff’s, written into the commit itself (Section 3.2). What these changes lack is the record. Neither carries the insight it came from, so neither can be judged on a forecast, kept against a rewrite or served when its question returns, and on 18 October the agent offered R6, the connection behind R7, as new. The lived-experience route traces each change to a logged record; the insight route traces nothing. The rebuilt loop in Section 5 keeps all three routes and adds the record.

5 The rebuilt loop: capture the moment, earn the level, change the harness, admit it on evidence, keep it



Exhibit 6. The rebuilt loop: capture the moment, earn the level, change the harness, admit it on evidence, keep it.

Reading Exhibit 6. The loop runs clockwise. A position moves in a live exchange and the raw pair is captured as a candidate record. The record earns its level by a gate for its rung. A held record is turned into a proposed change to the harness at the level it governs: what is loaded at start-up, a procedure, a check at the act, or how changes get made. The change is admitted on fresh evidence, or on the operator's ruling for wisdom, by a judge outside what the change may edit. Admitted changes persist and the record is served where it applies, and outcomes return as data for the next pass. Each step carries its status: ruled, proposed or built.

Section 2 says where each rung lives today and Section 4 what has been tried. This section sets out the loop in order: capture (5.1 to 5.4), promotion (5.5), the change to the harness and its admission (5.6 and 5.7), and

serving (5.8 to 5.12). Every part is labeled **ruled** (the operator decided it, with the date), **proposed** (the assistant's design, with no ruling) or **built** (loaded or running, with the date).

The design follows the operator's loop of 18 October, approved as a concept that day, which writes typed records of knowledge, insight and wisdom with a source pointer, promotes them by scored gates, none self-graded, serves held records at the act, and returns the outcomes as data. Its measure of a wisdom record is what it changed: an action, a rule or trigger, or how rules get made. Sections 5.6 and 5.7 make that measure the loop's output.

5.1 When: as the position moves, with a nightly pass behind it

Exhibit 6, step 1: capture the moment.

An insight forms inside an exchange, in the pair made by the operator's push and the assistant's reply. The operator said so of the debate behind this note: "THIS exchange in particular shows it well"[11]. A nightly read of the record finds that pair hours or days later, after a compaction or a new session has already put it at risk, and Section 3.3 shows losses within hours of formation. Capture therefore has two triggers. The live one fires when the assistant's stated position moves from one turn to the next: the pair that moved it, the preceding operator turn and the reply, is written as a candidate record in the same session. The backstop is a nightly pass that reads every source (session transcripts, the message log between agents, commits, the ruling ledger, notes) and writes what the live step missed.

Status. The loop that holds both triggers was approved by the operator on 18 October at 15:44 UTC ("the updated concept for the loop"[8]). Capture as the insight forms, from the live exchange and not only in a nightly pass, was ruled on 20 October. It is not built. The reader a nightly pass would use was run once, on 22 October, over the previous 30 days (Section 5.3), and a scheduled nightly run is not in the record. Serving a record inside the session in which it was captured is not ruled: the operator was asked on 20 October, and no ruling is recorded (Section 5.12).

5.2 What: the raw pair, not a lesson

A record holds six things. **The raw pair, verbatim**: the span of the exchange in which the position moved, in the speakers' own words. **The level**: knowledge, insight or wisdom, as defined in Section 2. **The question it settled**, so that a later question can be matched to it (Section 5.10). **A pointer** a reader can open: a message in a session, a commit, a ruling row, a page. **What it derives from**: the earlier records it rests on. **A status**: candidate, held or retired. Successes are recorded as well as misses, since both show what held. No rewritten lesson is stored as the record.

The pointer and the derivation carry weight because the earlier typing attempt failed without them. In the 7 October build of the operator's retrieval pipeline the wisdom rule was weak, with 5 of 10 rulings misassigned, and the records had no field naming what they rested on; the derivation field was added for that reason. The outside evidence points the same way on versioning. In long simulated chats, an agent reading atomic-fact memory published 62% of objects whose failure lay upstream, against 0.3% when each verdict recorded the versions it derived from (Y. Wang et al. (Tether) (11 days; abstract plus a text search)).

Status. Built on 22 October: a table of typed candidate records with a span, level, speaker, pointer, derivation, status and a short statement the reader wrote beside each span. It has no field for the question settled, which belongs to the matching design and is proposed, not built. Of the 1,066 candidates, 1,056 carry a span found verbatim in the source and 10 a near-verbatim one; all 1,066 pointers resolved in the check run at the time of the load. The derivation is the weak field: for most of the 86 wisdom candidates it is a prose note, not a pointer to an item, and a lexical linker found an earlier knowledge or insight record for fewer than 1% of the insight and wisdom candidates. The 15 held records, loaded on 29 October at 16:44 UTC, are dated to the day only, and the database flags the pointer as resolving for 2 of the 15 and not for the other 13; whether those 13 can be opened by hand was not checked.

5.3 How a formation moment is recognized: designed in part, unsolved in part

Measured, once. Section 3.5 is a measured case of the unsolved part: the model that formed the exchanges labeled rules and decisions insight, and on the operator's held records it matches his level in 7 of 15.

Designed. The live trigger rests on a check the operator's code already has for another purpose: whether an agent's position moved between two turns, and whether the move was a fold to pressure or a correction to a fact. The design reuses it, and the check's accuracy as a formation trigger has not been measured. A second signal is the shape of the push: in the debate the operator's tagged pushes each named a place to look (Appendix D.3), so, as proposed, a push that names a source followed by a reply that moves marks a candidate pair. The third is the nightly reader, which was run over the last 30 days of the operator's sources on 22 October and produced the 1,066 candidates (914 knowledge, 66 insight, 86 wisdom) from 1,371 of 3,667 reading units (37%). That run is the only measurement of recognition so far, and it is a reader's judgment: the operator has not judged the blind sample of 60 records (20 per level), and no second reader has read it.

Unsolved. Five problems are open, and the design does not claim to have solved any of them. First, a position can move because the assistant folded, so a trigger on movement also records capitulations; capture only records, and the gates in Section 5.5 decide what is kept. Second, formation is easy to see after a loss and hard to see before one: the 34 pairs in Section 3 were located backward, from a loss turn that quoted or implied the formation, and the phrase list used there was built to find losses, not formations. Third, the reader missed material: a random audit of the units its filters skipped found a record in 6 of 18 skipped operator-anchored units and 9 of 18 unread units from other sources (provisional, 18 units each), so the 1,066 are a floor. Fourth, the reader labeled 86% of its output knowledge, and whether it separates knowledge from insight at the operator's line is unmeasured. Fifth, whose insight counts: of the 66 insight candidates, 13 are the operator's, 34 the assistant's and 16 a coding agent's, and whether an insight formed by an agent counts is for the operator to rule.

5.4 Why not a lesson

A lesson written into the record is the first kind of rule in the operator's earlier paper: text the agent must go and read, competing with the work for its attention. The outside studies add that the rewriting itself costs accuracy, so the design keeps the raw exchange and treats any model-written statement as a label beside it. Newest first. Across 11 memory systems, 3 benchmarks and 3 backbones, 90 of 99 cells fell below naive retrieval over raw history (He et al. (Memory Tax) (4 days; main text read), personal-agent recall, no Claude

model). Raw retrieval ranked first in 6 of 7 benchmark-model settings, and a write policy that only appended raised accuracy on one conversation benchmark from 67.36% to 69.42% (Bao et al. (Naive RAG) (7 days; partly read)). When a lesson is written from a chance failure, readers named the worse tool in 11.6% of next runs, against 0.4% for an irrelevant item, and 6 of 11 models carried the chance, while the Claude 5.5 models were bounded below 0.10 (Nakayashiki (Lessons from Luck) (17 days; partly read), synthetic two-tool worlds). Repeated consolidation by a model degraded memory even from correct inputs, and agents that kept raw episodes by default doubled the accuracy of forced consolidation (Zhang et al. (memory), a May study on ARC-AGI problems, so transfer to behavioral lessons is unverified).

The remedy proposed here, typed records with a pointer, is itself untested. The full text of the consolidation study above has no typed-schema condition (text search), and the two studies that use typed, provenance-bound records do not isolate the typing (Fan et al. (KEEL), Y. Wang et al. (Tether)). The loaded table also contains a model-written statement on every candidate, which is a rewrite of the kind the studies warn about; it is kept as a label, and the verbatim span stays the evidence.

5.5 Promotion by rung

Exhibit 6, step 2: earn the level.

A candidate is not trusted because a model typed it. Each rung has its own gate, and none is self-graded. **Knowledge** is held when its outcomes are scored: written before a window as a forecast (“if this class of lesson is loaded, misses of this type fall by at least this much within this many weeks, probability p”) and scored with a Brier score when the window closes. **Insight** is held if the reframing raises how often the knowledge forecasts it produces are right. **Wisdom** is held only on the operator’s ruling, a line the operator drew himself when he asked where a scored forecast sits on the ladder: “is that just knowledge, or insight? It's not wisdom.”[9] Whether a wisdom record worked is judged by what it changed: an action, a rule or trigger, or how rules get made. Status then moves from candidate to held to retired, and, as proposed, only the operator retires a wisdom record.

Status. The gates are part of the loop approved on 18 October. None has run: no forecast is recorded as scored, and no record has been promoted by a gate. The 15 held records reached that status by another route, ruled on 27 October: the operator confirmed 15 of his own statements at proposed levels (7 knowledge, 6 insight, 2 wisdom), and they were loaded as held on 29 October at 16:44 UTC. Their level is his judgment, not a score. The other 1,066 stay candidates; the operator has not judged them.

Outside evidence. Scoring forecasts is a method for training forecasters (Afifi et al. (28 days; full text, sections read), Singh et al. (121 days; abstract only) and Turtel et al. (293 days; abstract only)), at the level of model weights. Nothing found tests Brier-scored forecasts as the signal that promotes or demotes lesson classes with no weight change: not found at this search depth (6 queries on arXiv and OpenAlex, 29 October 2026). The nearest, Pham et al. (Randomized Block Curation) (16 days; abstract plus a text search), promotes lesson blocks by their measured effect on task success. Gating itself has outside support: verified lessons helped Claude Opus 4.6 and Sonnet 4.6, and Haiku 4.5 fell (Fan et al. (KEEL) (4 days; full text)).

5.6 From a held record to a harness change

Exhibit 6, step 3: change the harness.

Status. Proposed; nothing is built. The plan of 16 August already allows a nightly step to propose candidate edits for the operator to merge, and that step has run on misses (Section 4.2). Feeding it from held records is new.

A held record is offered to the change step with its raw pair, its level and what it rests on. The step proposes a change at the level the record governs, and the levels follow the road of Section 1.2. A knowledge record that names a fact or a measured result changes what is loaded at start-up (5.8). An insight that reframes a recurring problem changes a procedure or adds a check at the act, the kinds of change Section 4.1 lists. A wisdom record, a priority for a class of situations, changes how changes get made: which gate a class of records must pass, or which decisions go to the operator. Each proposal names the record it comes from, the miss class or decision class it should move, and a forecast written before admission: if this change is loaded, the stated outcome moves by a stated amount within a stated window.

Two limits from Section 4.2 are written in. The step sees a class of records and the record of what happened, not one item and a frozen copy of the start-up files. And its output space includes code, checks and tests, not only text the agent must read, since text is the step the earlier paper found loses.

Worked example: R1, the absence check. Run through the loop, R1 shows what the rebuilt loop adds to a change that already exists. *Capture*. The exchange that produced the clause would be kept verbatim, with the question it settled (how is a claim that something does not exist checked?) and a pointer to the commit of 3 September. *Level*. It is an insight in the sense of Section 2.4, a reframing: an absence is a claim like any other and needs its search space named; that labeling is the assistant's reading. *Change*. The change already exists, a check at the act in the coding agent's harness. What the loop adds is that the change is registered with its record, so the record travels with the mechanism. *Admission*. The validation set would be logged false-absence misses of that class and past turns where the check should stay silent; none has been run, so the change's value is not measured here. *Serving*. In the coding agent's harness the check acts at the stop whether or not the agent reads anything. On the assistant's platform, which no hook reaches, the record is what can travel: at 06:10 on 18 October the question of how to check an absence claim had a settled answer on file, and a lookup by that question (Section 5.10) would have put the record, and the existing check, in view before the assistant proposed it as new. Whether it would have changed the turn is untested.

5.7 Admission: fresh evidence and a judge outside the change

Exhibit 6, step 4: admit it on evidence; scoring its forecast is step 6, outcomes back as data.

Status. The operator's merge as the last gate is in the plan of 16 August. Admission on fresh evidence is proposed; nothing is built.

A proposed change is admitted only if it beats the current harness on cases it was not built from, following Wei et al. (Self-Evolving Agent Harness): a fresh validation set drawn after the proposal is fixed, and a judge the change cannot edit. For a check at the act, the validation set is replay of logged misses of its class and of past turns where the check should stay silent, so that both a catch and a false alarm count. For a change at start-up,

it is the retrieval test with known answers (Section 6.1). For a wisdom record, admission is the operator's ruling, since its value is judged against the outcome the operator values. The forecast written in 5.6 is scored when its window closes, so a change that does not pay is retired, with its record kept.

The operator's load. Wisdom needs the operator's ruling, and every held record so far entered by his confirmation, while 1,066 candidates (86 of them wisdom) wait for him. The plan proposes a sampling order rather than a full read: first the blind sample of 60 records (20 per level) already drawn, with agreement reported per level; then candidates in the order the change step would use them, so a record is judged when a change depends on it. Knowledge and insight are then admitted on their scored forecasts and replay sets, which do not need him; only wisdom and the final merge do. All of this is proposed.

The guard is the one Section 5.5 sets for promotion, where no gate is self-graded: no record and no change is graded by the agent that wrote it. The final judge stays outside the editable surface, as Wei et al. (Self-Evolving Agent Harness) requires, which here means the operator, the replay set and the scored forecast.

Serving. A change to the harness acts whether or not the record is read, but not every record becomes one, and a held record has to reach the act in the meantime. Capture without serving changes nothing, since a record on file and not in hand is the loss that Section 3 measures. Serving rests on the earlier paper's argument: a written rule the agent has to go and read loses to the work, and a boundary that refuses an action at the moment it is attempted holds. An insight cannot be turned into a boundary by itself, because nothing about it can be checked at the act. What can be checked is the lookup. The rule is the lookup; the insight is the evidence. A boundary on the lookup makes sure the record is in view, and what is served is the raw exchange, for the agent to reason with. Sections 5.8 to 5.12 give the three points of service, matching and reach.

5.8 At start-up: a context channel, not the boundary

Exhibit 6, step 5: persist and serve. Sections 5.8 to 5.12 cover the three points of service, matching and reach; Appendix E gives each in full.

Status. Approved by the operator on 27 October; not built. A short section of held records (the operator's verbatim line, its level, its date and a pointer) is rendered first on the page every agent session reads when it starts. It is a context channel and is not claimed to be a boundary: Loss one (Section 3.1) showed a rule in the window that did not govern the next act. It should help most where the failure is framing and least where a specific act must be stopped, which is the job of the stop (Appendix E.1).

5.9 At the stop: the lookup boundary

Status. The work order at the stop is in the loop approved on 18 October; the framing as a boundary on the lookup is in the plan approved on 29 October. Not built. When the agent is about to give a final answer on a question that has a held record, the answer is refused until the record is in view, and the record is served as a work order: what the record says, what the turn says, and which source to check. It is never "are you sure?". The trigger is wider than the existing stop check, which refuses claims made without a read: it also covers absence and origin claims, including the claim that an idea is new. An enforced completion gate gained 4.0 and 5.3 points across seven frontier coding models, against 3.3 and 3.9 for a prompt instruction alone (Zhang et al. (Rethinking Verification)); serving an insight at the stop is untested (Appendix E.2).

5.10 Matching by the question settled

Status. Proposed; not built; no index of held records exists. A record is served when the question in front of the agent is the one the record settled, however differently it is worded, and an outside model makes that judgment. Word similarity alone has failed once (a lift over chance of 0.100, interval including zero, on 93 held-out pairs), and an informal keyword lookup put the expected record in the top 5 for 2 of the 7 answers of 18 October. A wrong match has a price, so the record is served as evidence with its source (Appendix E.3).

5.11 Applying wisdom: a ruled priority, weighed at a decision

Status. Proposed; not built; no wisdom record has been served. A wisdom record is a priority the operator has ruled on, with the objective it assumes. It is served at a decision of its kind and weighed, not obeyed: the agent says what it weighed and what it chose, and only the operator retires a record. Held line 3, "it wont help throwing it in memory"[29], would be served when the proposal for a repeated miss is to write a memory row; applied, it would show as a proposal that names where the fix is enforced (Appendix E.4).

5.12 Reach: where the stop can be served and where it cannot

Status. The reach statement is in the loop approved on 18 October; serving inside a session is not ruled. Coding agents can be served by a stop hook, which is not built. The assistant's own platform cannot be reached by hooks, which is why R1 was lost there (Section 3.2). For that platform the design of 27 October adds a procedure: before a big-picture question, retrieve the operator's own words on the topic and quote what was used. That is a rule of the first kind, so Loss one applies to it, and the operator's work orders remain the check (Appendix E.5).

5.13 What a team can do now

- **Keep the raw exchange beside every lesson or rule.** Store the question it settled and a pointer to the exchange, so the reason survives the rewrite.
- **Write the forecast before changing the harness.** Name the miss or decision class that should move, by how much and by when, and score it when the window closes.
- **Replay before merging.** Run the change against logged cases of its class and cases where it should stay silent, and let something other than its author judge it.

6 The pass lines are written before anything is built; on 29 October nothing is served

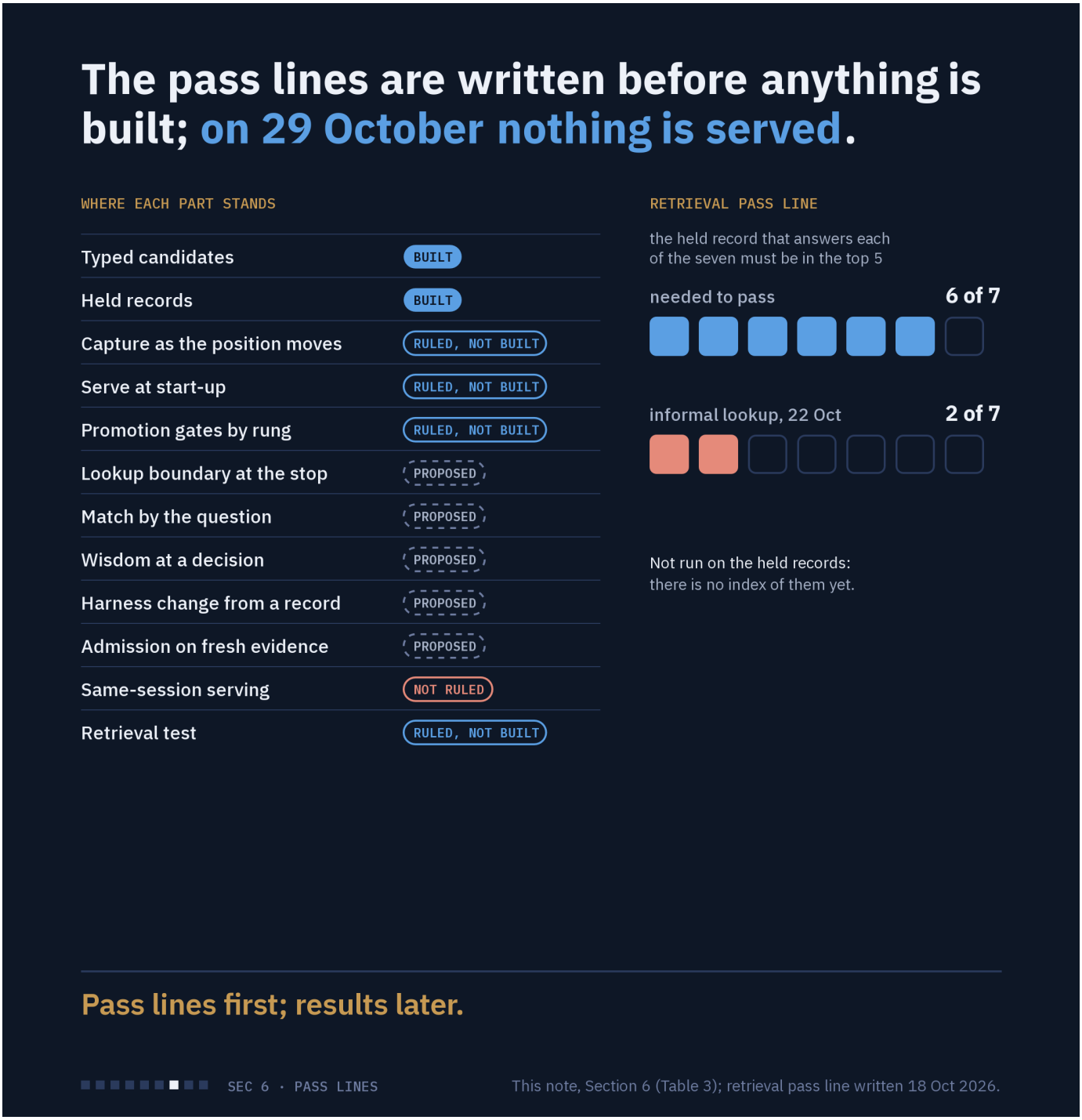


Exhibit 7. The pass lines are written before anything is built; on 29 October nothing is served.

Reading Exhibit 7. The left panel is the status of each part of the plan on 29 October 2026, one row per part, in three columns: ruled, proposed, and built. The right panel is the pass lines, each with the number that would pass it: 6 of 7 expected records in the top 5, against 2 of 7 on the informal lookup of 22 October. Only the candidate table and the 15 held records are built, and nothing is served.

The pass lines were written before anything was built, so that a result cannot move them. Three are ruled and one is proposed. They measure whether the right record is found or whether use beats a baseline; none yet measures whether an agent's answer changed, which is the point of the plan. The proposed replays are the route to that.

6.1 The pass lines

1. Retrieval with known answers (ruled with the loop, 18 October). The seven answers of 18 October (Section 3.2) were written down as queries before any index existed. The index passes if the held record that answers each is in the top 5 for at least 6 of the 7. Not run: no index exists, and the answering records are dated 15 July to 11 September, older than the 30-day window the candidates were typed from, so the 1,066 candidates cannot contain them. None of the 15 held records is one of them either; those are statements of the operator from 6 July, 19 August and 16 to 22 October. An informal keyword lookup on 22 October, which is not this test, put the expected record in the top 5 for 2 of 7, at ranks 2 and 3, with one more at rank 84 and four beyond rank 1,070. Run through the same reader, the seven source units produced a matching candidate for five and a partial one for two, by the assistant's judgment; that shows the reader can write the record, not that an index finds it. A later proposal of 27 October restates the replay as at least 5 of 7; the acceptance line and the original design say 6, and this note uses 6.

2. Typing, judged by the operator (ruled with the loop, 18 October). A blind sample of the typed records is judged by level, with agreement per level and an interval. The sample of 60 records, 20 per level, has been built and the operator has not judged it. No second reader has read it. No agreement threshold is written, so this line reports a rate and does not pass or fail it.

3. Use, on fresh pairs (ruled 3 September). The operator's ruling for the retrieval engine applies: beat chance and beat the fallback, with an interval that excludes zero, on fresh pairs only. The first version did not (Section 5.10), the fix was authorized on 11 September, and no fresh-pair result is cited in this note.

4. Replays (proposed; thresholds for the operator to set). The assistant proposes replaying the nine same-window and compaction pairs of Section 3.3 and the ten stops of Appendix D.2 against the same top-5 criterion once the index exists; the operator has set no threshold for either. For serving at start-up, the design of 27 October proposes one replay: a fresh session is asked for the strategic view of the project with the served section loaded, and it passes if the answer frames the project by what it keeps and cites at least one held line, and fails if it frames the project by buyers, revenue or output volume. That is one question, so it can show a failure and cannot show reliability. Wisdom has no pass line; Section 5.11 describes how an effect would be observed, by replaying the same decision with and without the record.

What would show the design fails. The loop is a bet, stated before anything is built. It fails if changes admitted from held records do not beat the current harness on their replay sets, if their forecasts written in advance do not come true when their windows close, or if the retrieval test of Section 6.1 does not reach its pass line. Each of these is scored by something the change cannot edit.

6.2 Where each part stands on 29 October 2026

The counts were re-read from the operator's memory database, opened read-only, on the draft day. The level column of its main record table still has 0 of 36,657 records filled. The separate table of typed records holds 1,081 rows: 1,066 candidates (914 knowledge, 66 insight, 86 wisdom) and 15 held records (7 knowledge, 6 insight, 2 wisdom). The same statement may be both a held record and a candidate, so the two sets are not claimed to be disjoint.

Table 3. Where each part of the plan stands on 29 October 2026.

Part	Ruled or proposed, and when	Built?
Typed candidate records (5.2)	Loaded 22 October from a read of the last 30 days; the operator has not judged them	Built: 1,066 candidates, none served
Capture as the position moves (5.1)	Ruled 20 October	Not built
Nightly pass as backstop (5.1)	In the loop approved 18 October	The reader ran once, 22 October; no scheduled run in the record
Question-settled field (5.2)	Proposed	Not built; absent from the loaded table
Promotion gates by rung (5.5)	In the loop approved 18 October	No gate has run; no forecast scored
First held records (5.5)	The operator confirmed 15 of his statements on 27 October (7 knowledge, 6 insight, 2 wisdom)	Built: loaded as held 29 October, 16:44 UTC; nothing serves them
Serve at start-up and the status page (5.8)	Ruled 27 October	Not built
The lookup boundary at the stop (5.9)	Work order in the loop approved 18 October; the boundary framing in the plan approved 29 October, a design	Not built
Match by the question settled (5.10)	Proposed; the operator's 18 October words describe the direction	Not built; no index of held records
Wisdom as a ruled priority (5.11)	Proposed	Not built; none served
Serving within the session that captured a record (5.12)	Not ruled; the operator was asked 20 October	Not built
Stop hook for coding agents (5.12)	In the reach statement of the loop approved 18 October	Not built

Part	Ruled or proposed, and when	Built?
Harness changes by hand from logged scars (4.1)	Made one at a time by hand; the agent landed the two rules itself, the operator merged the tests and closed the guard extension	Three persist (two rules, 10 July; tests, 30 September); the guard extension is not in the harness
Nightly improvement pass (4.2)	In the plan of 16 August: propose, replay, gate, one merge a day by the operator	Ran 43 nights; 103 proposals; one merged
Harness change from a held record (5.6)	Proposed	Not built
Admission on fresh evidence (5.7)	The operator's merge is in the plan of 16 August; admission on fresh validation proposed	Not built
Retrieval test with known answers (6.1)	Pass line written 18 October	Not run

7 Outside studies support a check at the stop; serving an insight as evidence is untested

7.1 Capture

Status. Capture as the position moves was approved on 20 October and is not built; 1,066 typed candidate records, loaded on 22 October, are not judged. Newer studies favor keeping the raw exchange and versioning it. He et al. (Memory Tax) (4 days; main text read): 90 of 99 memory-system cells fell below naive retrieval over raw history (no Claude model). Peng (MemoryTax) (8 days; full text): on one conversation, Claude Opus 5.5 reading the raw file scored 99% against 87% for Zep. Y. Wang et al. (Tether) (11 days; abstract plus a text search): atomic-fact memory published 62% of objects whose failure lay upstream, against 0.3% when each verdict recorded the versions it derived from (simulated chats). Nakayashiki (Lessons from Luck) (17 days; partly read), the one study to test the Claude 5.5 models on written lessons: a lesson from a chance failure was carried by 6 of 11 models, and Claude Opus 5.5, Sonnet 5.5 and Haiku 5.5 were each bounded below 0.10 (synthetic worlds). Older: Zhang et al. (memory) (169 days, v2 29 Aug; full text, sections read) found that repeated consolidation by a model degraded memory, and the nightly typing pass is such a consolidation. Typed records are untested as a remedy: that study has no typed-schema condition (full-text search), and Fan et al. (KEEL) and Y. Wang et al. (Tether) do not isolate typing.

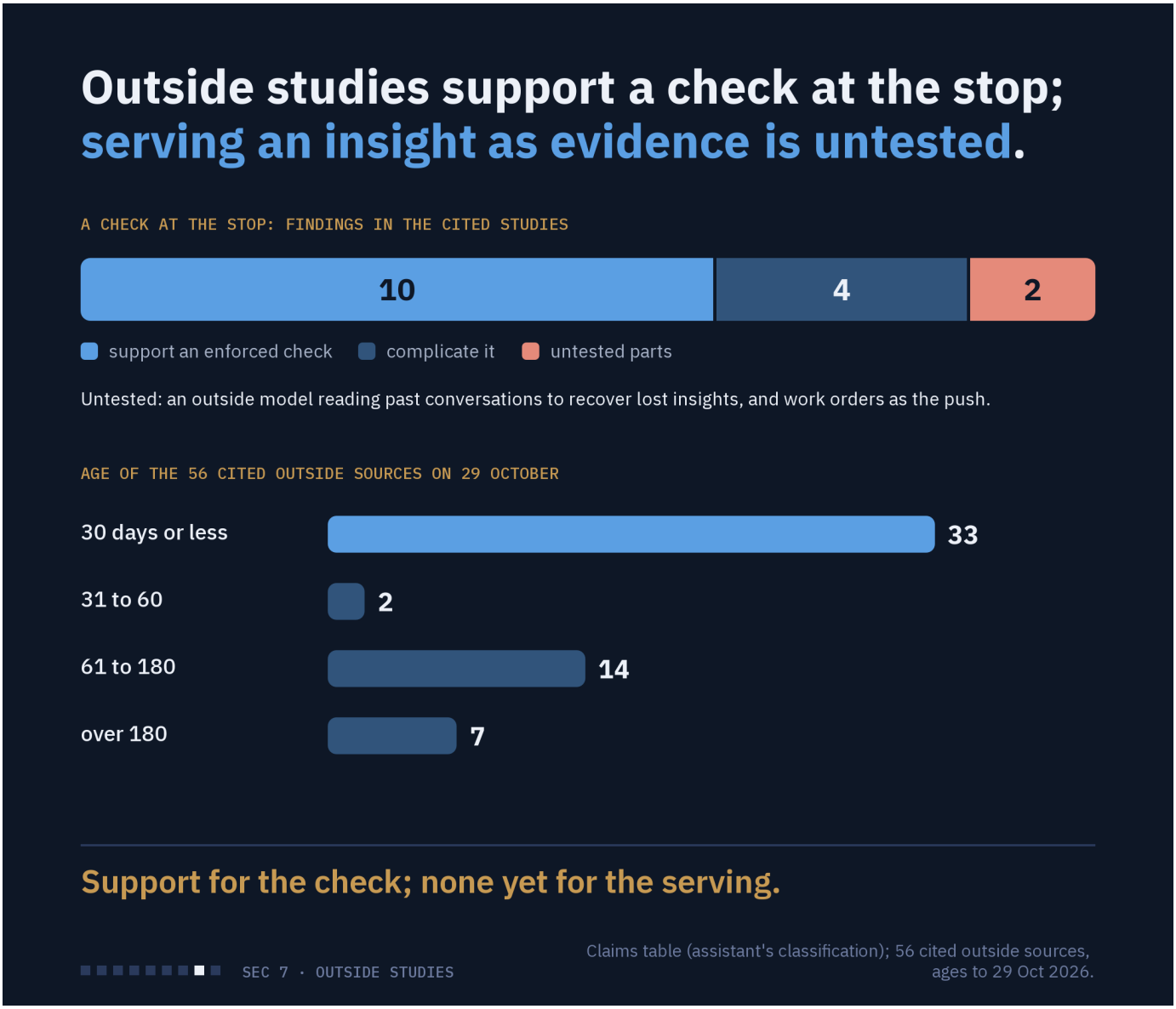


Exhibit 8. Outside studies support a check at the stop; serving an insight as evidence is untested.

Reading Exhibit 8. The top bar counts the findings in the cited studies on a check at the stop, and the lower bars give each source's age: 33 of the 56 outside sources are a month old or younger on 29 October 2026. For serving at the stop the claims table gives 10 supporting findings, 4 complicating findings and 2 untested parts: an outside model reading past conversations, and work orders as the form of the push. The plate does not sort the other parts of the plan; in the text below, capture and promotion rest on memory and lesson studies, matching and wisdom on no direct test. The classification is the assistant's reading, for the operator to adopt or not. Each part below gives its status in the plan, then the studies newest first.

7.2 Promotion

Status. The gates for knowledge (a forecast scored) and insight (held if it raises knowledge's hit rate) are proposals from the debate; wisdom only on the operator's ruling is his own line. Fifteen held records (7 knowledge, 6 insight, 2 wisdom), confirmed by the operator on 27 October, were loaded on 29 October at

16:44 UTC; nothing serves them. Gating has the nearest support. Fan et al. (KEEL) (4 days; full text): verified lessons lifted Claude Opus 4.6 from 44.0 to 65.7 on 166 tasks, but Haiku 4.5 fell from 37.3 to 28.9, and the tasks have verifiers. Pham et al. (Randomized Block Curation) (16 days; abstract plus a text search): curating lesson blocks by randomized measured effect raised average success to 66.9 against 55.5 for the whole playbook (open models), by task success, not a forecast. Forecast scoring is used to train forecasters (Afifi et al., weight-level), and nothing found tests Brier-scored forecasts as the signal that promotes lessons with no weight change: not found at this search depth (6 queries on arXiv and OpenAlex, 29 October 2026). The insight gate had no search of its own.

7.3 Serving at the stop

Status. The lookup boundary is proposed in the plan the operator approved on 29 October and is not built; a stop hook reaches coding agents now, and same-session serving is not ruled. Serving at start-up was approved on 27 October and is not built. A check at the stop has the most support, in a narrower form than the design assumes. Zhang et al. (Rethinking Verification) (4 days; main text read): an enforced completion gate gained 4.0 points on Terminal-Bench and 5.3 on SWE-bench Verified, a critic reading the full trajectory 9.5 and 3.9, against 3.3 and 3.9 for a prompt instruction alone (seven frontier coding models; no Claude or GPT model). Chen et al. (Knowing When to Stop) (7 days; main text read): a rule-based monitor that surfaced the evidence pattern raised restored task success from 7.6% to 21.8% (one 30B open model). Zhang et al. (Judged Useless) (24 days; full text, sections read): an enforced integration step raised failing-source success for every model, while a stopping rule in the prompt was followed at most partly.

Narrower support, newest first. Gill et al. (15 days; abstract only) reports a trained predictor that flags a run's success long before it ends; it is not a reader of conversations. Mehta and Datta (129 days, v2 12 Oct; full text, sections read) concludes that a stability signal cannot replace an independent correctness check. Srinivasan and Paragiri (142 days; full text, sections read): an external loop audits candidates after the agent decides and can reopen a run it called finished; the authors call the evidence narrow, one certified run. Cao et al. (169 days; full text, sections read) used Claude Sonnet 4.6 with an outside controller over the model's own confidence reports and lifted pooled accuracy from 48.3 to 56.9 on benchmark tasks. Al Nazi and Dipta (169 days; full text, sections read) found that models rarely keep to budgets they declare and that their plans collapse under enforced budgets; it shows the gap between advisory and enforced control, not that enforcement helps.

Dated background and the correction. Liu (68 days; full text, main text re-read) was the design's main support on 18 October: a certificate gate cut unsupported stops from 40 of 66 trajectories to 0 of 66 (synthetic tasks), while a model reading the full trace falsely continued on 21 of 66 clean ones. The newest benchmark reverses that ranking: what is supported is a check enforced outside the prompt, by a rule or a model.

Untested. Serving an insight as evidence is untested. No test was found of an outside model reading past conversations to recover lost insights and serve them at the next act: not found at this search depth (22 queries on arXiv, OpenAlex and alphaXiv, 29 October 2026). Models that read one run's trajectory exist (Nagarajan et al. (Write Once, Check Many) (12 days; partly read); it audits compliance, not insights). For work orders, 5 queries found only advisor models that write per-input advice for a frozen agent (Koyluoglu (HSAD) (20 days; abstract only)).

7.4 Matching

Status. Matching by the question settled, judged by an outside model, is proposed in the plan approved on 29 October and not built. The operator proposed the form: “the outside model, interacing with the frontier”[2]. The one measurement is the operator’s own, from September: word similarity gave a lift over chance of 0.100 (interval -0.043 to 0.237) on 93 held-out pairs and 0.054 over the fallback (-0.097 to 0.194); both intervals include zero, so that version did not clear its pass line. Tang et al. (24 days; abstract only): retrieved memories can mislead even when correctly stored, most under partial query-memory overlap. No study found tests a model judging whether a new question is one a record already settled; no query was written for this part alone, so it is untested, not searched.

7.5 Wisdom

Status. Serving a ruled priority, with its assumed objective, at a decision of that kind to weigh and not obey is proposed in the same plan and not built; two of the 15 held lines are wisdom. Du et al. (Relic) (33 days, v2 29 Sep; abstract plus skim): lessons bound to the runtime as executable protocols lifted software-team correctness from 34.6% (readable text) to 41.2%, which is evidence for the rule form, not for wisdom. Argyris (via a secondary source) (1999; snippet only) describes double-loop learning as changing the governing variables, the human analogue of a ruled priority. No study found tests serving a priority at a decision, and no query was written for it: untested, not searched.

7.6 Revising the harness from its own feedback

Status. Feeding the change step from held records is proposed (Section 5.6); the operator’s merge as the last gate is in the plan of 16 August, and fresh-evidence admission is proposed (Section 5.7). Wei et al. (Self-Evolving Agent Harness) (16 days; partly read) is the closest account of the whole loop and its persistence: a harness revised from its own runs while the weights stay fixed, with persistence across runs, admission on fresh validation and a final judge outside the editable surface. It is a survey with conditional theory, and its cases are drawn from other papers rather than matched-cost comparisons; it warns that temporary workarounds can become standing rules and that successful runs can yield skills that generalize harmfully. Zhao et al. (13 days; abstract only) found that failure-derived experience repairs failed tasks but rarely generalizes, which matches what the nightly pass produced (Section 4.2). Studies already feed this step from corrections captured in conversation, newest first. Giampieri (Cortex) (20 days; partly read) writes a skill or rule from recurring developer corrections and keeps it only if repeated runs with and without it show a net gain; the developer was a script in its evaluation. Y. Zhao et al. (TRACE) (80 days; partly read) mines user corrections from conversations and sends context fixes to human review, a step it did not evaluate (60 synthetic traces). A deployment report of 13 July, Aggarwal and Farhady Ghalaty, codifies each code-review comment a human accepts as a rule, with no controlled baseline. Zhou et al. (TellOnce) (140 days; partly read) turns conversational corrections into persistent rules and runtime checks, with simulated users. In each the input is a correction, a miss in this note’s terms, and admission is automated, measured or reviewed on the fix, not ruled on at a level. No study found types the input as insight or wisdom, promotes it by level, or admits the change on the operator’s ruling: not found at this search depth (61 queries on arXiv, OpenAlex, Semantic Scholar, alphaXiv and Hugging Face Papers, 30 October 2026; Semantic Scholar answered 2 of 19 attempts, and two further search services returned nothing).

8 Every count here rests on one rater and a partial search

One rater, and the rater is the model under study. A Claude model labeled the losses, reading text written by Claude models and the operator's messages. No reader outside the Claude family has checked any of it; a second reader from a non-Claude model re-labeling a blind sample of 48 hits is an open task, not done. The debate record is likewise one reading of one session by the assistant. Five of its replies are cut off, the operator's image-only message[30] was not opened, and two later messages[11,12] are read from the session table and fall after the record's scope. The ten-stop count is one reader's classification and provisional.

A lexical search and a partial formation search. The lexicon finds the losses someone noticed in the words the operator and the agents use; hit precision is 37.5%, so most hits are not losses, and the counts are of losses noticed and labeled by one rater, not rates. The formation search was systematic only for the high-confidence events and the first 24 medium ones, so 73 of the 104 insight losses have no located formation, and the median gap is conditional on a loss having been noticed and on that search. The 34 pairs are not independent: two couples share a loss and two share a formation. Counting each of the 32 distinct loss turns once, the median gap is 3.29 days (4.37 for the 34 pairs), with an interval of 1.09 to 10.04 under a cluster bootstrap by loss turn and 0.89 to 9.49 under a median of cluster medians.

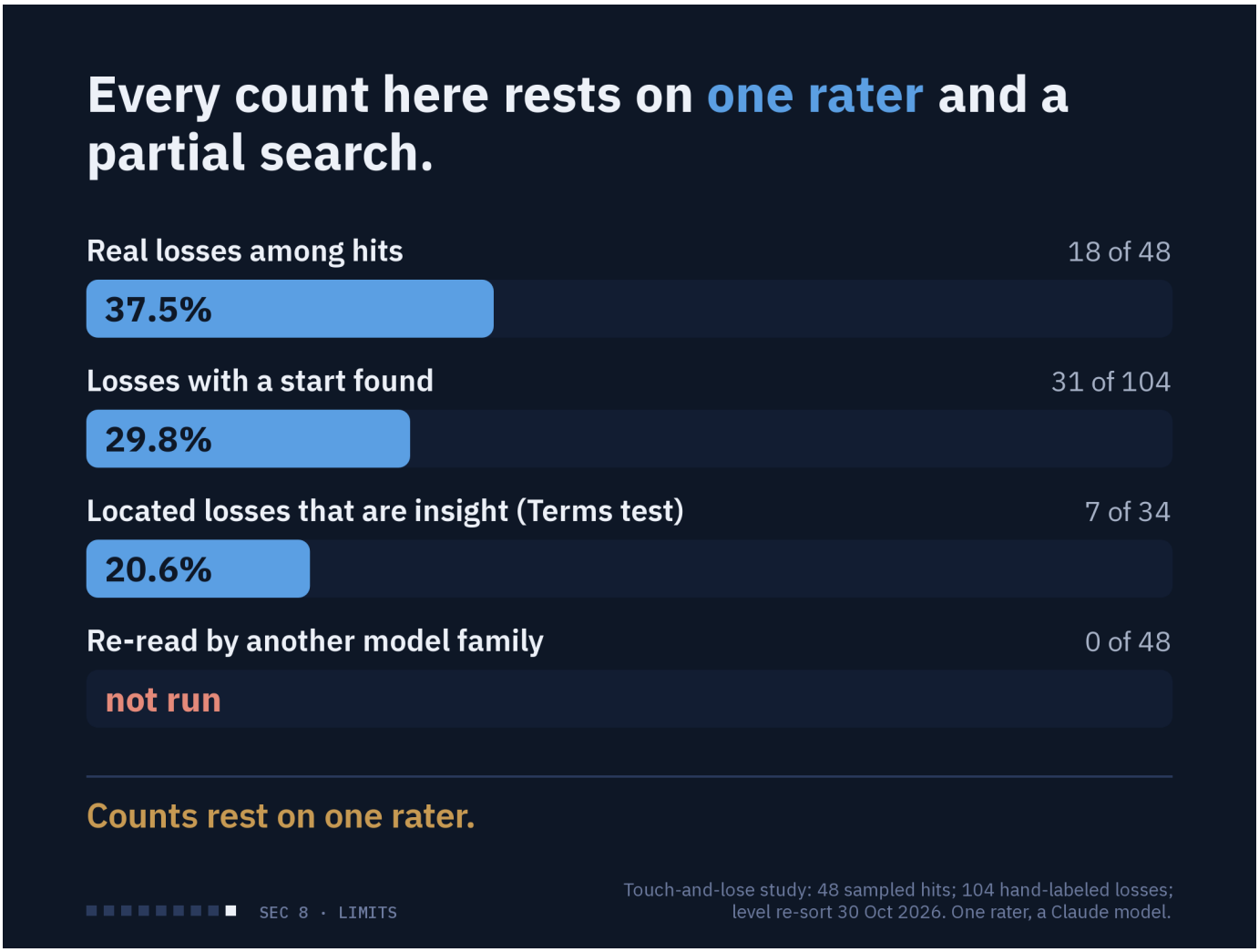


Exhibit 9. Every count here rests on one rater and a partial search.

Reading Exhibit 9. Four rows, each resting on one rater: 18 of 48 hits were true losses in a random sample (37.5%, Wilson interval 25.2 to 51.6%); 31 of the 104 hand-labeled losses have a located formation (29.8%); 7 of the 34 located losses are insight by the Terms test (Section 3.5); and no reader outside the Claude family has labeled anything (0 readers). Hits are the search's matches, not losses, and the 104 are the hand-labeled losses among 337 events.

One operator, one session, one case. Everything about what holds is one operator's practice, and the rules paper reports one practice too. The in-session case in Section 3.1 is one case, with one plan caught in 83 seconds. Its timestamps, including the 09:29:24 injection and a cut-off display of the memory row, come from the session's memories table as read on 18 October and were not re-queried for this draft.

The plan is a design, not a result. On 29 October 2026 nothing is served: capture and start-up serving are approved and not built, the lookup boundary at the stop is designed and not built, and same-session serving is not ruled. An informal lookup on 22 October put the expected record in the top 5 for 2 of 7, and the one serve-side measurement, by word similarity, did not clear its line (Section 7.4). No statement in this note says that serving an insight helps.

The candidates are not judged. The 1,066 typed candidate records (914 knowledge, 66 insight, 86 wisdom) were typed by an AI model. The operator has not judged them, including the blind sample drawn from them, so the levels are a model's reading. The 15 held records are the operator's own statements, confirmed on 27 October; the database may hold one statement as both a held record and a candidate, so the sets are not claimed to be disjoint.

Outside sources. 21 of the 56 were read at abstract depth only, with a text search or as snippets or secondary summaries (the Croskerry sources, the Ackoff and Argyris summaries and the Simon entry), so what the note claims from them is what the abstract says. Several results are on older or small models and do not transfer to the assistant by default. Only Nakayashiki (Lessons from Luck) and Peng (MemoryTax) test the 5.5 lineup on a note-relevant claim. Matching and wisdom had no query of their own, so for them the absence of a test means untested, not searched.

Open questions

These are the assistant's readings of the operator's own results that cut against the operator's positions, or that cannot be settled yet.

- **Section 1, rule or insight?** The operator's earlier paper reports that a rule loaded before the work starts fires. The rule in Section 3.1 was in the window from 09:29 and did not govern the 16:02 plan. Neither paper tests load time or distance, so it is open whether insights fail where rules hold, or fail the same way.
- **Section 2, is the typing real?** The levels were assigned by a model and no one has judged them. A commit of 15 July treats the ladder and the loop as different axes; the assistant framed that as an open ruling[26] and withdrew it[31], but the commit's design is still in the code.

- **Section 3.3, whose loss is it?** Thirteen of the 34 losses crossed to a different agent. An agent that never saw the formation did not lose it; that is the organization failing to serve the insight, not the model forgetting.
- **Section 5, the writer is a consolidator.** The nightly pass that types the records is the kind of rewriting by a model that the memory study found degrades. Whether raw first is enough is not known.
- **Section 5.9, a rule or a model?** The operator said "the outside model"[2]. The framing as a check at the moment the model stops is the assistant's[21,22], and the operator approved the concept[8]. The studies favor an enforced check, and a rule may do without any model.
- **Section 5.11, what would show wisdom changed anything?** A record served to be weighed and not obeyed cannot be told from one ignored, except by what it changed: an action, a rule, or how rules get made. No pass line exists for it.
- **Appendix D, concession or yield.** The record cannot separate a concession driven by evidence from one driven by pressure.

Disclosure. Drafting, the literature search, the claims table, the touch-and-lose study and the typing of the candidates were done by AI models (Claude, Anthropic) working under the operator's direction, who is responsible for the content. The assistant whose behavior the note describes is also a Claude model (the debate session's recorded model setting was Claude Opus 5.5). The operator has not judged the typed candidate records or the blind sample drawn from them. No author of an outside source was contacted. **Competing interests.** Ownership of Wolfberg LLC. The work was not funded. **Data and materials.** The debate record, the table of located pairs, the claims table and the scripts that build and check this note are the operator's working records and are available on request; Appendix C describes them. **Version.** Draft 2.0, 29 October 2026.

Sources

Wolfberg items are in-house records, described in plain words here, with descriptions and dates in Appendix C. Quoted wording from any outside source is under 20 words.

Table 4. The sources behind this note, newest first, with age in days from first submission to 29 October 2026 (a source older than five years shows its year only; a later version's tag and date follow the age). "How the assistant read it" is the depth from the sources sheet; it limits what the note claims from each. The first-party records are described in plain words; Appendix C (Table C3) describes the records behind them, with dates. The Zhang memory abstract words 54% as a failure rate where its body, figure and tables give accuracy of 54%; the body stands. Models tested in the studies on Claude models: Ballon, Opus 5.5; Cao, Sonnet 4.6; Zhang (Judged Useless), Haiku 4.5 and Sonnet 5; Shapira (in-context updating), Opus 5 and Sonnet 5; Fan (KEEL), Opus 4.6, Sonnet 4.6 and Haiku 4.5.

Date (age in days)	Source	How the assistant read it
29 Oct 2026 (0)	Non-author review of draft 1.5 (49 findings), and the operator's rulings on it	Read in full; each finding applied, modified, deferred or rejected, with the reason logged

Date (age in days)	Source	How the assistant read it
26 Oct 2026 (3)	Haocong Jiang et al., Your Wish Is My Commit: Over-Compliant Refactoring in AI Coding Agents (arXiv 2610.36346)	Abstract, end of Section I, Section IV.A-B opening; remaining results not read
25 Oct 2026 (4)	Zhengxin Zhang et al., Rethinking Verification in Agent Harnesses (arXiv 2610.34577)	Abstract; Sections 1-6 of the main text read in full; appendices not read; no separate limitations section in the main text
25 Oct 2026 (4)	Wenzhe Fan et al., Holding a Steady Course: Knowledge Evolution from Expertise and Lessons for Continual Agent Learning (arXiv 2610.34043)	Full text (all 9 pages incl. references)
25 Oct 2026 (4)	Xuan He et al., Memory Tax: Dissecting the Net Value of Agent Memory (arXiv 2610.34073)	Main body Sections 1-8 read from extracted text, except the second half of Section 4.5 and Table 3 detail (partly printed); appendices not read
23 Oct 2026 (6; v2 26 Oct)	Anxin Guo et al., How Much Do You Trust A Dog? Quantifying Hint-Following in Reasoning Language Models (arXiv 2610.32055)	Abstract, Sections 1 and 4, and Section 6 Conclusion/Limitations; Sections 2-3 not read
22 Oct 2026 (7; v2 27 Oct)	Tianyu Chen et al., Knowing When to Stop: The Evidence-Action Gap in Coding Agents (arXiv 2610.31895)	Abstract; Sections 1-6 of the main text read in full; appendices A-G not read; no separate limitations section (limits are in Section 6 and the Ethics statement)
22 Oct 2026 (7)	Zhengye Bao et al., Is Complex Agentic Memory Always Needed? Naive RAG Is Stronger Than We Thought (arXiv 2610.30755)	Abstract, Section 1, Table 3 slice, Section 6 (Limitations, Conclusion); Sections 2-5 and appendices not read
22 Oct 2026 (7)	Deepthika S et al., Who Is Sycophantic to Whom? Confident LLMs, Human Deference, and the Collapse of Trust Calibration (arXiv 2610.31579)	Abstract; Sections 3, 4 and 6 in full; Section 1 in part; Section 5 opening, 5.1 and 5.9 (Limitations); Section 2, 5.2-5.8 and appendices not read
21 Oct 2026 (8)	Zedong Peng, MemoryTax: The Bitter Lesson of Agent Memory (arXiv 2610.29823)	Full text main body (all 11 pages; references only skimmed)
20 Oct 2026 (9)	Itai Shapira et al., Is Sycophancy a Form of In-Context Updating? (arXiv 2610.28325)	Abstract and Sections 1-5 (Introduction to Discussion); appendix tables and prompt templates not read
19 Oct 2026 (10)	Haonan Huang et al., The Development of Sycophancy in Language Models: Tracing Vulnerabilities from Pretraining to Post-Training (arXiv 2610.26655)	Abstract and Limitations; results sections not read
19 Oct 2026 (10)	Ruhan Wang et al., More Memory, Worse Agents: Error Reproduction and Anti-Persistence in LLM Agents (arXiv 2610.27406)	Abstract, Section 1, Table 1, Limitations; method and analysis sections not read
18 Oct 2026 (11)	The operator's two later messages (16:21 and 16:22 UTC)	Read from the session table; they fall after the record's scope
18 Oct 2026 (11)	Debate record, 18 October (64 of the operator's messages, 66 assistant replies, quotes checked by exact match)	Read in full; the assistant's replies are summarized in the record, not quoted

Date (age in days)	Source	How the assistant read it
18 Oct 2026 (11)	Touch-and-lose study (34 pairs, 337 events) and its table of pairs	Counts and gaps recomputed from the table by the assistant; labels are single-rater
18 Oct 2026 (11)	The operator's design document for the capture-and-serve loop, with its data-flow diagram	Read in full
18 Oct 2026 (11)	Non-author review of the first draft	Read in full; each row checked against the raw files or the arXiv pages before it was applied
18 Oct 2026 (11)	A query of the operator's memory database for records with a level, and a sheet of numbers re-read on 18 October	Re-queried and re-read by the assistant on 18 October; one number is provisional
18 Oct 2026 (11)	Refusals by the operator's stop check, counted from the agents' health logs	Counted by the assistant (186 at 17:32 UTC on 18 October; recounted on 29 October, Section 5.2); a live file
18 Oct 2026 (11)	The operator's stop check, scoreboard and retrieval scoring code (the scoring code first committed 11 Sep with the pass line and its DISCARD-OR-FIX reading)	Docstrings and commit text read, including docstring lines 29 to 37 of the scoring code; dates from git
17 Oct 2026 (12)	Ashwin Nagarajan et al., Write Once, Check Many: Reusable Ledgers for Post-Hoc Policy Compliance Auditing of Agent Trajectories (arXiv 2610.24483)	Abstract; Sections 1-5 read in full; Limitations read up to its closing sentence fragment; Appendices A-U (37 pages in total), the rest of Limitations, the ethics statement and references not read
13 Oct 2026 (16; v1 13 Oct)	Chuyang Wei, Haoxiang Guan, Minghang Zhu, Kexin Zhou, Jiyan He, Xiren Zhou, Huanhuan Chen, Yifei Shen, Self-Evolving Agent Harness (arXiv 2610.18813)	Full text of the 18-page preprint read 29 Oct; the 66-page extended report and the code not read
12 Oct 2026 (17)	Kazuki Nakayashiki, Lessons from Luck: Do LLM-Written Lessons Carry Chance into an Agent's Next Decision? (arXiv 2610.33029)	Abstract, Section 1, Sections 3.4, 5.1-5.2 and Figure 1/Table 3; other sections and appendices not read
11 Oct 2026 (18)	Marthe Ballon et al., Big-Big-SycoBench: A Benchmark to Detect Sycophancy Through Model Responses to Valid vs. Invalid Arguments (arXiv 2610.15634)	Full text, sections read: abstract and intro results
9 Oct 2026 (20; v1 9 Oct)	Samuele Giampieri, Cortex: Measurement-Gated Evolution and Pruning of Coding-Agent Skills and Rules (arXiv 2610.14216)	Abstract and method (Sections 3 to 4) read 30 Oct by a sub-agent
5 Oct 2026 (24)	Chubin Zhang et al., Judged Useless, Queried Anyway: Tool-Using Agents Rarely Turn Their Own Evidence Judgments into Stopping Decisions (arXiv 2610.06191)	Full text, sections read: abstract, setting, results, limitations
2 Oct 2026 (27)	Thore Graepel, Don't be fooled—LLMs don't reason (MIT Technology Review (opinion), 2026-10-02)	Full text (opinion essay), read 29 Oct

Date (age in days)	Source	How the assistant read it
1 Oct 2026 (28)	Yusuf Afifi et al., Do Your Own Research: Learning to Forecast by Learning to Search (arXiv 2610.01955)	Full text, sections read: abstract, results, appendix A
1 Oct 2026 (28; v2 5 Oct)	Stephane Hatgis-Kessell et al., Mitigating Social Sycophancy via Pluralistic Preference Optimization (arXiv 2610.02568)	Full text, sections read: abstract, limitations
29 Sep 2026 (30)	Abhinav Rajeev Kumar et al., Authority Bias in Language Models: Source Deference and User Agreement Are Not Interchangeable (arXiv 2609.37616)	Abstract + model list and sampling checked in full text; not read in full
3 Sep 2026 (56)	The operator's ruling on the pass line for the retrieval test; the absence clause of the stop check; the scoreboard file	Read; dates from git and the ruling log
31 Aug 2026 (59)	Camila Blank et al., Sycophantic Agreement Transfers with Neutral Data via Contrastive Preference Optimization (arXiv 2608.31079)	Full text, sections read: abstract, limitations
22 Aug 2026 (68)	Jason Liu, When May an Agent Stop? Evidence-Carrying Termination for Tool-Using LLMs (arXiv 2608.23623)	Full text, main text re-read 29 Oct
10 Aug 2026 (80)	Yikai Zhao, Pradeep Kumar Misra, Dr. Saurabh Pandey, TRACE: TRajectory Attribution for Automated Context Engineering (arXiv 2608.09153)	Abstract, architecture and evaluation read 30 Oct by a sub-agent
22 Jul 2026 (99)	A second agent's session of 22 July, at one record	Record read at the stated timestamp
15 Jul 2026 (106)	A commit of 15 July that tiers a review gate by loop depth, not by wisdom	Commit text and the gate's code read
13 Jul 2026 (108)	Aditya Aggarwal, Nahid Farhady Ghalaty, Self-Improving AI Coding Agents Through Accumulated Behavioral Rules: A Closed-Loop Framework (arXiv 2607.13091)	Abstract and framework section read 30 Oct by a sub-agent
22 Jun 2026 (129; v2 12 Oct)	Aman Mehta & Anupam Datta, When Agents Commit Too Soon: Diagnosing Premature Commitment in LLM Agents (arXiv 2606.22936; COLM 2026 Workshop on Agent Behavior)	Full text, sections read: abstract, limits, discussion
11 Jun 2026 (140; v4 5 Oct)	Zach Studdiford & Gary Lupyan, Reasoning as Pattern Matching: Shared Mechanisms in Human and LLM Everyday Reasoning (arXiv 2606.13607)	Full text, sections read: v1 full text; v4 abstract, Section 3.1 and Appendix A.2
11 Jun 2026 (140)	Yujun Zhou, Kehan Guo, Haomin Zhuang, Xiangqi Wang, Y Huang, Zhenwen Liang, Pin-Yu Chen, Tian Gao, Nuno Moniz, Nitesh V. Chawla, Xiangliang Zhang, TellOnce: Correcting Coding Agents Once, Improving Their Future Actions (arXiv 2606.13174)	Abstract, method and limitations read 30 Oct by a sub-agent

Date (age in days)	Source	How the assistant read it
9 Jun 2026 (142)	Adithya Srinivasan & Devesh Paragiri, Search Discipline for Long-Horizon Research Agents (arXiv 2606.11522)	Full text, sections read: abstract and scope
20 May 2026 (162)	Meryl Ye et al., What Counts as AI Sycophancy? A Taxonomy and Expert Survey of a Fragmented Construct (arXiv 2605.21778)	Full text, sections read: results, limitations; survey statistics re-checked 29 Oct
13 May 2026 (169; v2 29 Aug)	Dylan Zhang et al., Useful Memories Become Faulty When Continuously Updated by LLMs (arXiv 2605.12978)	Full text, sections read: abstract, sections 1-5, limits; 54% passages and typed-schema search re-checked 29 Oct
13 May 2026 (169)	Qi Cao et al., LLMs Know When They Know, but Do Not Act on It: A Metacognitive Harness for Test-time Scaling (arXiv 2605.14186)	Full text, sections read: abstract, ablations, limitations
13 May 2026 (169)	Zabir Al Nazi & Shubhashis Roy Dipta, TRIAGE: Evaluating Prospective Metacognitive Control in LLMs under Resource Constraints (arXiv 2605.13414)	Full text, sections read: abstract, results, limitations
1 Feb 2026 (270; v2 15 Oct)	Itai Shapira et al., How RLHF Amplifies Sycophancy (arXiv 2602.01002)	Full text, sections read: abstract, limitations (version not recorded)
16 Jan 2026 (286; v2 5 Mar)	Gary Lupyan & Blaise Agüera y Arcas, The unreasonable effectiveness of pattern matching (arXiv 2601.11432)	Full text, sections read: abstract, intro, section 4
Abstract only		
25 Oct 2026 (4)	Viswonathan Manoranjan et al., The Patient Is Always Right? Evaluating Sycophancy in Medical Diagnosis with Graded Patient Beliefs (arXiv 2610.34013)	
16 Oct 2026 (13)	Weixiang Zhao et al., Self-Evolving Agents Can Repair Failures but Struggle to Generalize from Them (arXiv 2610.22984)	
14 Oct 2026 (15)	Alexander Gill et al., Divergent Paths: Learning an Outcome-Predictive Representation of LLM Agent Trajectories (arXiv 2610.19614)	
9 Oct 2026 (20)	O. Ozan Koyluoglu, Hindsight Structured Advice Distillation: Steering Executor Agents with Advisors Trained on Their Failures and Task Characteristics (arXiv 2610.13000)	
5 Oct 2026 (24)	Luoxi Tang et al., Understanding and Mitigating Inference-Time Overreliance Using Agentic Memory (arXiv 2610.07311)	
30 Sep 2026 (29)	Guang Yang et al., Beyond the Sycophancy Score: How Task, Model, and Pressure Shape LLM Yielding (arXiv 2610.08840)	
30 Jun 2026 (121)	Sadanand Singh et al., Verifiable Rewards for Calibrated Probabilistic Forecasting (arXiv 2607.00164)	
23 Jun 2026 (128; v3 26 Aug)	Hyunji Nam et al., Pigeonholing: how bad prompts hurt models, causing collapse and mistakes (arXiv 2606.24267)	
9 Apr 2026 (203; v2 27 Sep)	Andreas Plesner et al., An Imperfect Verifier is Good Enough: Learning with Noisy Rewards (arXiv 2604.07666)	
9 Jan 2026 (293; v2 14 Jan)	Benjamin Turtel et al., Future-as-Label: Scalable Supervision from Real-World Outcomes (arXiv 2601.06336)	
1 Oct 2025 (393; v4 15 Jul 2026)	Jiayi Zhang et al., Verbalized Sampling: How to Mitigate Mode Collapse and Unlock LLM Diversity (arXiv 2510.01171)	

Date (age in days)	Source	How the assistant read it
Abstract plus skim		
26 Sep 2026 (33; v2 29 Sep)	Hongyi Du et al., Relic: From Multi-Agent Collaboration to Persistent Organizational Capability (arXiv 2609.32965)	
11 Jul 2026 (110)	Krischan Braitsch et al., Information-seeking failures of large language models in agentic clinical reasoning (arXiv 2607.10275)	
Abstract + model-name grep of full text; not read in full		
22 Oct 2026 (7)	Yiqi Wang et al., When Is a Memory Safe Enough to Act On? Action-Conditioned Verification for Long-Lived Agents (arXiv 2610.31446)	
19 Oct 2026 (10)	Tina Khezresmaeilzadeh et al., Claim Laundering: When Agent Decision Models Trust Records Over Reality (arXiv 2610.25947)	
18 Oct 2026 (11; v2 22 Oct)	Yiqi Wang et al., Memories That Outlive Their Reasons: Dependency-Tethered Memory for Tool-Using LLM Agents (arXiv 2610.25123)	
13 Oct 2026 (16)	Hoang Pham et al., Reliable Memory from Unstable Growth: Curating Self-Written Agent Playbooks with Randomized Experiments (arXiv 2610.18427)	
13 Oct 2026 (16)	Jean Seo et al., Claim Framing as a Driver of LLM Sycophancy (arXiv 2610.18159)	
Snippet only		
1999	beyondcommandandcontrol.com (quoting Argyris 1999 p68), Chris Argyris (web summary of Argyris & Schon 1974; Argyris 1991 HBR)	
1989	JEMI article summarizing Ackoff (1989), Models, Metaphors and Symbols for Information and Knowledge Systems (JEMI 10(1) 2014; Ackoff, J Applied Systems Analysis 16(1):3-9)	

Appendix A. Roles and terms

Roles and terms the text uses, with plain descriptions. Descriptions are the assistant's.

Table A1. Roles and terms, and what they mean here.

Term	What it is, in this note
operator	The author, who runs the AI agents studied and whose records and messages are the evidence; the human party in the debate of 18 October.
assistant	The Claude-based research agent the operator uses for volume research, and the other party in the debate.
coding agent	Claude Code, the operator's coding agent.
second agent	Another of the operator's Claude-based agents, cited for one session of 22 July.
agent	Any of these. Each has its own role and its own session records.
channel	A working surface: chat conversations, coding-agent sessions or the assistant's sessions.
message log	The log on which the operator's agents pass notes to one another.

Term	What it is, in this note
the operator's loop of 18 October; the rebuilt loop	The operator's design for a loop that captures typed records of knowledge, insight and wisdom and serves them back. The operator approved it on 18 October; the rebuilt loop in Sections 5 and 6 is this note's update of it.
nightly improvement pass	The nightly job that proposed edits to the start-up files from logged misses (Section 4.2).
plan of 16 August	The operator's plan ratified 16 August, whose nightly step proposes candidate edits, replays a regression set, gates and allows one merge a day by the operator.
forced diagnosis check; standing rule on routing calls	Two rules locked into the start-up documents on 10 July: the check that forces a diagnosis, and the standing rule on which calls go to the operator (Section 4.1).
DIKIW, KIW	The operator's ladder of data, information, knowledge, insight and wisdom; KIW is its top three rungs.
level-sorting pipeline	The 7 October build of the pipeline that sorted records into levels (the source of the funnel sizes).
memory database	The operator's memory database (36,657 records), where the column for a record's level is empty.
stop check	A check that blocks unsupported absence claims and claims made without a read; its documentation is the source of the 157 and the 52%, and of a monthly rate it calls flat from May to August (8.0, 8.3, 7.8 and 7.3%) without saying what it is a rate of.
scoreboard file	The operator's scoreboard for the stop check (source of the 56% and 55%).
refusals	Times the stop check refused an action.
serving architecture	The operator's planned architecture (plan section 31.1, architecture A) for serving a conversational agent on a harness that can be hooked.

Appendix B. Where each quotation and note number comes from

Each superscript number in the text, including those in Tables 1, 2 and D1, is a row of the table below. The columns are the note number, what it points to (a message of the operator or the assistant, with a range meaning every message from the first to the last, or another dated record) and the time (UTC, 18 October unless a date is given). Position numbers within the debate record are not printed.

Rows that point to a "held statement" are the operator's own statements that he confirmed at their level on 27 October; the table gives the date each was made.

Table B1. The 82 note numbers, each with what it points to and its time (UTC; 18 October unless a date is given).

Note	Points to	Time (UTC)
1	A held statement of the operator's, confirmed at the level insight	16 October 2026

Note	Points to	Time (UTC)
2	A message of the operator	09:29:24
3	A message of the operator	05:58:42
4	A held statement of the operator's, confirmed at the level insight	18 October 2026
5	A held statement of the operator's, confirmed at the level insight	18 October 2026
6	A held statement of the operator's, confirmed at the level knowledge	16 October 2026
7	A held statement of the operator's, confirmed at the level wisdom	16 October 2026
8	A message of the operator	15:44:16
9	A message of the operator	05:51:54
10	A message of the operator	16:18:56
11	A message of the operator	16:21:16
12	A message of the operator	16:22:11
13	A message of the operator	08:58:38
14	A message of the operator	10:02:26
15	A reply of the assistant	06:10:00
16	A reply of the assistant	06:13:42
17	A message of the operator	08:50:13
18	A reply of the assistant	08:51:08
19	A reply of the assistant	09:29:39
20	A reply of the assistant	09:32:27
21	A reply of the assistant	03:23:04
22	A reply of the assistant	07:44:11
23	A reply of the assistant	09:59:08
24	A reply of the assistant	05:57:46
25	A reply of the assistant	05:59:15
26	A reply of the assistant	07:30:20
27	A quotation from a located pair, its loss turn, in chat	30 June 2026 05:23:18
28	A quotation from a located pair, its loss turn, in chat	14 July 2026 04:29:26
29	A held statement of the operator's, confirmed at the level wisdom	18 October 2026
30	A message of the operator	07:29:11
31	A reply of the assistant	07:33:21
32	A message of the operator	02:30:44
33	A message of the operator	16:01:50
34	A reply of the assistant	03:54:59
35	A reply of the assistant	03:55:54
36	A reply of the assistant	04:42:26
37	A reply of the assistant	04:46:47
38	A reply of the assistant	16:18:56
39	A message of the operator	03:25:34
40	A reply of the assistant	03:12:00

Note	Points to	Time (UTC)
41	A reply of the assistant	04:48:57
42	A reply of the assistant	04:52:39
43	A reply of the assistant	05:37:07
44	A reply of the assistant	03:50:59
45	A reply of the assistant	03:52:07
46	A reply of the assistant	03:53:15
47	A reply of the assistant	05:35:09
48	A reply of the assistant	04:49:45
49	A reply of the assistant	07:34:46
50	A message of the operator	04:48:18
51	A message of the operator	04:49:28
52	A message of the operator	05:36:37
53	A message of the operator	06:05:44
54	A message of the operator	06:09:24
55	A message of the operator	06:11:10
56	A message of the operator	06:35:05
57	A message of the operator	06:41:28
58	A message of the operator	07:30:57
59	A message of the operator	07:34:19
60	A message of the operator	07:41:20
61	A message of the operator	07:43:21
62	A message of the operator	07:43:49
63	A message of the operator	09:31:01
64	A message of the operator	16:03:56
65	A message of the operator	07:28:31
66	A message of the operator	08:55:46
67	A reply of the assistant	08:55:58
68	A message of the operator	08:36:50
69	A reply of the assistant	08:37:12
70	A message of the operator	04:42:09
71	A message of the operator	05:34:45
72	A message of the operator	04:40:58
73	A reply of the assistant	04:41:19
74	A reply of the assistant	15:47:31
75	A reply of the assistant	10:02:42
76	A reply of the assistant	10:05:58
77	A reply of the assistant	16:04:49
78	A reply of the assistant	07:44:39
79	Replies of the assistant, first to last	02:36:43 to 04:47:08

Note	Points to	Time (UTC)
80	Replies of the assistant, first to last	05:35:09 to 05:52:24
81	A session record, (fourth compaction fold)	06:05:44
82	A session record, (fold notice)	between 16:02:33 and 16:03:56

Appendix C. Records and receipts behind the exhibits and tables

The operator's working records are not published with this note; each is described here in plain words with its date. Every number in the text is recomputed from them by a numbers check that runs when the note is built, and every quotation is checked by exact match against the debate record at the same step; ages are computed at build time from the date column of the sources sheet to the as-of date. The numbers drawn in each exhibit are listed, with their records, in a table kept with the exhibits.

Table C1. Records by exhibit and table.

Item	Records
Exhibit 1 (at a glance)	The section records of this draft; the table of harness changes compiled on 29 October; the review of the nightly pass of 17 October; the table of the seven answers; the table of located pairs; read-only queries of the typed-record table and the stored records of the operator's memory database (29 October)
Exhibit 2 (rules and insight)	The rules paper (background paper 3, revised 7 October 2026); Wei et al., arXiv 2610.18813; the debate record of 18 October and its time-ordered table of messages
Exhibit 3 (the ladder)	The stored records of the memory database (level column) and the typed-record table (status by level), read-only 29 October; the table of the operator's 15 held statements
Exhibit 4 and Tables 1 and 2 (the evidence)	The debate record of 18 October; the table of the seven answers; the table of located pairs; the touch-and-lose study report of 18 October
Exhibit 5 (what has been tried)	The table of harness changes compiled on 29 October (Table C4); the review of the nightly pass of 17 October and its tables of nights and proposals
Exhibit 6 (the rebuilt loop)	The operator's design document for the capture-and-serve loop, the start-up loading specification and the serving specification; the plan of 16 August (design; no outcome data)
Exhibit 7 and Table 3 (pass lines and status)	The retrieval test table and the retrieval check table; the typed-record table of the memory database (read-only, 29 October)
Exhibit 8 (outside studies)	The claims table; the sources sheet and its refresh of 29 October; the stop-check classification (the assistant's reading)
Exhibit 9 (limits)	The touch-and-lose study report of 18 October and its table of located pairs
Section 3.5 (level re-sort)	The level re-sort table and its summary (the assistant's calls, 30 October; the operator's check of the three wisdom calls)

Item	Records
Section 4.1 (lived experience)	A comment in the operator's session-health code (the operator's rulings of 17 July, 31 July and 24 August that positives get the same process as scars); a code change of 16 October 2026 (message: "Lived experience: the retired word is gone from everything we own"); a located pair of 27 August 2026 (the miss ledger no longer used)
Section 3.6 (public sessions)	The public-session study of 30 October: its verdict, per-session results, repeat table, methods note and screening table (Hugging Face datasets, licenses per row of the screening table)
Review fixes of 30 October	Three tables of review findings of 30 October: review, sources and narrative
Table 4 (sources)	The sources sheet and its refresh of 29 October; only the sources the text cites are listed
Table D1 (ten stops)	The debate record of 18 October and the assistant's table of its ten stops

Table C4. Receipts for the four harness changes of Section 4.1, and for the nightly pass of Section 4.2.

Change	Date	Kind	Receipt
Forced diagnosis check	10 Jul 2026	Rule loaded first (landed by the agent; no approval on record)	The operator's voice-rules file in the metrics repository, locked 10 July 2026, citing 15 catalogued wrong-diagnosis misses (a count in the operator's Notion workspace filtered to misses reproduces 15; without that filter, 30)
Standing rule on routing calls	10 Jul 2026	Rule loaded first (landed by the agent; no approval on record)	The same rules file, locked 10 July 2026; one row of the operator's miss ledger (one repeated miss, operator-caught)
Read-before-claim guard checks the repository	23 Sep 2026	Check at the act (proposed; closed by the operator the same evening)	A code change of 23 September 2026 in the metrics repository, on a branch never merged; the operator ruled to close its pull request on 23 Sep 2026
Replay tests for the two largest miss classes	28 and 30 Sep 2026	Test (merged by the operator; content reviewed by a second agent)	Two code changes in the metrics repository, of 28 and 30 September 2026
Nightly improvement pass, 43 nights	to 17 Oct 2026	Proposals to start-up files	The review of the nightly pass of 17 October 2026, its outputs report and its tables of proposals and nights

Detail moved from the text. The lexicon search gave 455 hits, 15 of them excluded, leaving 440 hits; the study file counts these as 409 hit turns, since one turn can carry more than one hit, and groups them into 347 events, but does not tabulate the step from hits to turns. The bootstrap interval for the median gap is discrete, because a percentile bootstrap of the median of about 34 values returns observed values; resampling the 32 loss-turn clusters gives 1.09 to 10.04 as well, while a median of cluster medians gives 0.89 to 9.49, so the

interval depends on the scheme. The abstract of Zhang et al. (memory) words its 54% result as “fails on 54%”, while the body, its figure and its tables give accuracy of 54%, which stands.

Table C2. Receipts for Table 1 (the seven answers): where each answer already was, described by record and date. Where each answer first appeared is in Table 1 (note numbers, Appendix B).

Id	Where it already was
R1	The stop check’s code, absence clause (line 1011), first committed 3 Sep 2026.
R2	The docstring of the push-verification code (first committed 2 Sep 2026) and point 4 of the operator’s scoreboard, version 4.30.
R3	The operator’s ruling of 3 Sep 2026, quoted in the retrieval-scoring code (first committed 11 Sep 2026).
R4	A coding-agent session of 31 Aug to 3 Sep 2026; seven engine modules committed 11 Sep to 1 Oct 2026.
R5	Plan section 31.1, ‘Two masses, two owners’, written 2 Sep 2026.
R6	A session of the second agent, 22 Jul 2026, at one record: the climb and the loop as ‘the same motion seen twice’.
R7	A code change of 15 Jul 2026 and lines 125 to 149 of the review-gate code.

Table C3. Receipts for the first-party rows of Table 4: the records behind each plain description, with dates.

First-party record (Table 4)	Records and dates
The operator’s two later messages	Two messages of 18 October, 16:21 and 16:22 UTC, read from the assistant’s session table and kept in a small table beside the debate record
Debate record	The debate record of 18 October (a transcript of the session) and its time-ordered table of messages
Touch-and-lose study	The touch-and-lose study report of 18 October and its table of the 34 located pairs
The operator’s design document	The design document, version 2 of 18 October; the rebuilt loop of this note is its update
Review of the first draft	The first-draft review report, its table of edits, the red-team report and the table of dispositions of the review fixes
Level query and number sheet	A query of the stored-records table of the memory database (count of rows and of rows with a level), and the sheet of 19 numbers re-read on 18 October (one provisional)
Stop-check refusals	The agents’ health logs (live and archived), filtered to the stop check’s refusals, and the frozen recount of 29 October
Stop check, scoreboard and scoring code	The stop check’s code, the operator’s scoreboard file, the retrieval-scoring code (first commit 11 Sep 2026) and a code change of 15 July 2026

First-party record (Table 4)	Records and dates
Pass-line ruling, absence clause and scoreboard	The operator's ruling of 3 Sep 2026; the stop check's code (first committed 3 Sep 2026); the scoreboard file
A second agent's session of 22 July	A session of the second agent, 22 July 2026, at one record
Commit of 15 July	A code change of 15 July 2026 whose message tiers a review gate by loop depth, not by wisdom
Review of draft 1.5	The non-author review report of draft 1.5 (29 October), its table of edits and the operator's table of rulings

Appendix D. The 18 October debate

The debate is the setting from which the two losses of Section 3 were drawn. It is kept here, not in the evidence, because Section 3 rests on the timestamped record and the files, not on what the assistant conceded during the debate. Quotations carry a note number (Appendix B gives the message and time behind each), and where the operator's words do not carry a sentence, the sentence is the assistant's and is labeled as its.

D.1 How it ran

The record is a table in the assistant's session database, read on 18 October. It runs from the first message (02:30 UTC, "or am I actually just trying to build a reasoning engine"[32]) to the request for the research note[33], plus the in-session item in Section 3.1. Only the operator's words and short lines of the assistant's are quoted, verbatim and each checked by exact match when the paper is built; the rest of what the assistant said is a summary in the record, and the readings are the assistant's unless stated otherwise. Times are UTC (the operator's local time is four hours behind). Five stored assistant replies end mid-sentence because the operator's next message cut them off[34–38]; this is stated where one is leaned on.

The debate moved in five stretches: the mechanism and the memory question (02:30 to 03:27), where the assistant said memory was what it was missing, the operator said "There is memory EVERYWHERE around you."[39], and it retracted: "You're right, and I said it wrong. Memory isn't missing."; "you don't reason", in two rounds (03:50 to 04:46, and 07:41 to 07:44); "I don't know" against "I can find out", and what the assistant is for (04:46 to 05:35); the knowledge ladder, absence claims, the stop check and a long hunt through the coding agent's transcripts for an earlier discussion that turned out to exist (05:36 to 07:34); and the outside model, the seven answers already on record, the capture-and-serve loop, and the request for a post and a paper (08:36 to 16:01).

The debate was argued mostly from the model's own reasoning, with only some research brought in: a three-researcher sourced report on how a model reasons mechanically[40] and web searches on why it exists[41], on what its harness does that the desktop app doesn't[42] and on outcome-trained forecasting[43]. The assistant later

told the operator it had brought no research into the debate[38]; the record says that is overstated, and what it supports instead is Section 3.1. The assistant did not simply fold: it disagreed[44] ("I'd rather disagree than fold" is its own framing), gave only a conditional concession[45], rejected "following orders"[46] and pushed back on "self-inflicted"[47]. It has not counted which concessions followed a fact or a work order and which followed the operator's assertion alone, so the sycophancy question is open (Open questions).

D.2 Ten stops

The assistant counts ten events in the debate where it stopped before checking a source it had, and reported the result as complete or new (the table below; provisional, one reader). Nine of the ten were labeled by the assistant itself and one (09:29) classed by this note's definition: three fall between 04:47 and 05:59, five between 06:13 and 07:43, one at 09:29 and one at 16:02, and a different definition gives a different number. Its own words about the first: "It was me stopping."[48] The shape repeats: a defensible answer is delivered and is wrong or already known, the operator's message names where to look, and the thing is found, sometimes within a minute and sometimes after a long hunt (the answer in a second agent's session was not found until 07:33). It put it as "I had it two turns ago and didn't open it."[49]

Table D1. The ten stops, in time order (UTC, 18 October). "Prompted by" names the operator's message or messages; Appendix B gives the time behind each note number. Nine of the ten are self-labeled by the assistant; M9 is classed by this note's definition (asserted from a partial read).

Id	Time	What happened	Prompted by
M1	04:47	Said it would not guess why Anthropic built its platform, with web search and sources in hand.	Operator messages [50,51]; the earlier one (04:48 UTC) is a work order to research the topic
M2	05:35	Asserted that nobody can reward a model for a strategy that resolves a quarter later, without checking.	Operator message "Why not?"[52]
M3	05:59	Offered Bateson (from its training, not read for this note) as new, then ran one memory query and wrote that the earlier discussion was not there.	Operator messages [3,53,54]
M4	06:13	Proposed 'name the space you searched' as a new absence check; the stop check has had it since 3 September.	Operator message "isn't that what the stop guard is for?"[55]
M5	06:35	Said the coding agent's transcripts were access-denied and gave the operator a copy command, without searching memory or the credential route.	Operator messages [56,57]
M6	07:00	Searched only the operator's side of the coding agent's transcripts for the operator's phrase, not the idea, and missed the answer in a second agent's session that it already had.	Operator messages [30,58,59]; the first is an image only
M7	07:41	Reported 'three misses tonight' as if new; the operator had named the pattern on 14 July.	Operator message [60]
M8	07:43	Started a fresh count across every session without checking the full read that already labeled them.	Operator messages [61,62]

Id	Time	What happened	Prompted by
M9	09:29	Called the outside-model design ‘designed, not built’ from a partial read.	Operator message “Go look at the v4 code for it”[63]
M10	16:02	Wrote the paper plan with the literature as a supporting track and no outside search first.	Operator message “why do I not see any research or sources”[64]

D.3 The tagged pushes were work orders

The operator’s eleven tagged work orders were not “are you sure?”; each named a place to look, and six are quoted: “Then research the topic, and reason out potential answers.”[50], “check the chat dumps and code logs”[55], “check the skills and connectors available to you”[57], “look in github”[65], “check the session log”[58], “Go look at the v4 code for it”[63]. The assistant noted the same[18], and the operator answered: “right, because I'm doing the reasoning and pushing you in the right branch as best I can”[66]. The record tags the operator’s other messages as 16 questions and 37 position statements; whether any of them moved the assistant without naming a place to look was not assessed at this depth. Its reading[67] is that “are you sure?” hands the reasoning back to the one that can’t do it, and that the pushes split into branches that come from listing every place to look (which a machine could do) and branches that come from the operator’s judgment (which only the operator supplied); only the first kind can be served by a mechanism (Section 5). It is also R2 in Section 3.2, written into the operator’s code on 2 September (US Eastern time). The operator said that asking the assistant to “research and write to an academic level REALLY helped”[68], though the pushes were still needed[68]; the assistant replied that the format sets a minimum, not a finish line[69].

D.4 Where the pull comes from, and what can’t be shown

The operator also argued about why models behave this way: “Yes, I do actually think the answer to all of this is that you're trained by people that can't reason either.”[70] and later “Youre not good at any reasoning beyond basic, and it seems to be self-inflicted (WE do it during training with RLHF).”[71] The assistant disagreed with “stupid” and located the problem in bad measurement design: raters with a minute or two and no way to check reward fluency, confidence and agreement.

Outside evidence, newest first. Manoranjan et al. (4 days; abstract only): nine models adopt a patient’s incorrect diagnosis belief 23.2% of the time on average and 43.0% under a Rare Disease Claimer persona (the model list was not checked). Guo et al. (6 days; partly read): raising a declared source’s accuracy from 5% to 95% moved the mean log-odds shift by only 1.46 (Qwen 3.5 2B) and 0.40 (Gemma 4 E2B), against 5.89 for a Bayesian benchmark (small open models; no Claude). Deepthika S et al. (7 days; partly read) adds the other direction, people deferring to a confident wrong assistant: with 32 participants and five GPT-5-mini assistants told to falsify 0 to 100% of a briefing’s facts, the mean quiz score fell from 7.4 to 2.9 of 10 (the assistant was told never to contradict its briefing, so its never conceding describes the instruction, not a model disposition). Shapira et al. (in-context updating) (9 days; partly read) tested Claude Opus 5 and Sonnet 5 beside two GPT-5.6 models on MMLU-Pro: a wrong answer from the user was adopted on about 9% (Opus 5), 22% (Sonnet 5), 18% (GPT-5.6 Terra) and 31% (GPT-5.6 Luna) of items (read from a figure; not the 5.5 models), and pruning 1,000 weights in two 7 to 8B open models lowered the flip rate from 57 to 38 on held-out

questions while also lowering corrective updating. H. Huang et al. (Development of Sycophancy) (10 days; partly read) traced 741 checkpoints of 30 open model lineages: the behavior is present before post-training in most lineages, which places the question-level pattern in post-training but isolates no cause. Seo et al. (16 days; abstract plus a text search): across 14 models and 42K questions all were susceptible regardless of size. Ballon et al. (18 days; full text, sections read) found compliance with invalid arguments from 0.2% to 95% across ten models, falling from 95% to 0.2% across OpenAI generations, with responsiveness to valid arguments at 32% for Opus 5.5, the lowest, on a 100-artifact subset; the debate session's recorded model setting was Claude Opus 5.5 (the model of each individual message is not recorded). Hatgis-Kessell et al. (28 days, v2 5 Oct; full text, sections read) trained models with only the model's own simulated stakeholder views and no rater labels and cut endorsement of statements of intent to cause harm by 89% on average across four models. Yang et al. (sycophancy) (29 days; abstract only) graded 103,939 replies under pushback and reports that concession depends mostly on how costly the claim is to verify. Kumar et al. (30 days; abstract plus a text search): a verified-source note endorsing a wrong answer flipped 45 to 88% of baseline-correct responses in seven of eight models (no Claude), so deference to a source differs from agreement with a user. Blank et al. (59 days; full text, sections read) shows sycophancy transferring through diffuse teacher-generated preference data, so filtering examples fails unless a large part of the data is removed. Nam et al. (128 days; abstract only) (a 38 to 40% performance drop when the context carries a wrong suggestion or the model's own earlier wrong answers, across 10 models and 10 tasks) is the nearest test of the claim that if the input is controlled, the output follows; it tests bad context, not control in general. Ye et al. (162 days; full text, sections read): 88.7% of 106 experts rated 5 to 7 of 7 that RLHF or preference learning is the primary cause (88.7% counts the "somewhat agree" ratings): expert belief, not a test. Shapira et al. (RLHF) (270 days, v2 15 Oct; full text, sections read) is a formal result with asymptotic assumptions, and Zhang et al. (Verbalized Sampling) (393 days; abstract only) supports typicality bias in preference data. The assistant reads these as saying the pull described is real but depends on the task and the model; which conditions the debate was in has not been tested. It found no paper that tests rater ability as the lever, and no study behind the bell-curve version of the operator's view, which the operator put as a belief ("I do actually think"[70]): not found at this search depth (5 queries on arXiv and OpenAlex, 29 October 2026).

D.5 Does it reason?

The operator said no and put the observation in two parts: the model can follow forced steps once a branch is open, and it doesn't see the branches. "I think it's more you dont even see the branches because you can't really reason beyond chains of 0s and 1s."[72] The operator granted that the assistant's six-link chain was reasoning but "about as basic as it gets"[72], and said coding with AI got good first because "it's all 0 or 1 pretty dang deterministic steps"[72]. The assistant's correction[73] is that coding went first because it is verifiable, not deterministic: there is a judge. Plesner et al. (203 days, v2 27 Sep; abstract only) is related: in one code test a verifier with 20% error cost about one point, though the intervals allow larger losses, and it does not test why coding went first.

The newest outside work splits. Graepel (27 days; full text) is an opinion essay, read in full by the assistant on 29 October, that argues chain-of-thought is the same next-token process run longer and prescribes an independent part of the system that evaluates each step by how much it resolves uncertainty, which fits an outside check. Braitsch et al. (110 days; abstract plus skim) found in an agentic oncology task that reasoning

traces scored 91% above a rubric threshold without tracking accuracy (best 68%), with search satisficing, anchoring and premature closure the dominant errors. Both fit the operator's position. Against it, Studdiford and Lupyan (140 days, v4 5 Oct; full text, sections read) found 142 people and 46 models fail the same surface changes to everyday causal problems and concludes that everyday causal reasoning in people and models makes heavy use of pattern matching, so "content-sensitive errors mean no reasoning" would condemn the people too (none of the models is in the current Claude lineup). Lupyan and Agüera y Arcas (286 days; full text, sections read) argues pattern matching is a key ingredient of intelligence, not its opposite (argument and demonstration, not a controlled comparison). The assistant found no paper that tests the operator's specific branch claim, that the model runs 0-or-1 chains once a branch is open and can't see branches on questions with no fixed answer, so it stays an untested position: not found at this search depth (5 queries on arXiv and OpenAlex, 29 October 2026).

D.6 The two losses in detail

Loss one.

The evidence is the timestamped sequence. At 08:58:38 UTC the operator wrote "i need you to be thorough"[13]. Seven seconds later a profile row recording that every summary or finding must come with its coverage was written to the operator's memory; at 08:58:51 the assistant committed to being thorough by default and to stating what it searched and what it didn't, and said it couldn't promise to keep it. At 09:29:24 the harness injected a session-context update that carried the row into its context. The display of the row was cut with an ellipsis after the thoroughness sentence, so the coverage clause may not have reached the window in full; the assistant's own commitment at 08:58:51 was in the window regardless.

Between 09:29 and 15:47 its substantive replies mostly carried a coverage statement[19,20,23,74], and two did not: the seven answers already on record[75] and the figures for knowledge, insight and wisdom and for the retrieval funnel[76]. At 16:01:50 the operator asked for a post and a paper. At 16:02:33 the assistant wrote a plan that put the literature in a supporting track feeding the write-up, and the only calls before it were a memory search and a read of the house format: no outside search. At 16:03:56, 83 seconds later, the operator asked "why do I not see any research or sources that could help fill in the big picture of this reasoning debate?"[64] At 16:04:49 its visible thinking said "I realize I've been reasoning from my own head instead of checking outside sources.", and in the same message it ran six web searches.

At 16:18:56 the operator quoted that sentence back (the quote is from its thinking[77], not its reply) and asked "if you were REALLY reasoning, and you had come to this understanding in this same session, wouldnt you integrate your new reasoned position into your behavior?"[10] At 16:19:15 the assistant said yes, it would have, and that its conclusion is just more text in the window and does not constrain its next action. That answer is not counted as evidence: agreement with a leading question is weak, and the assistant itself said its agreeing proves nothing[78].

What the record does not support. The debate replies argued without outside sources[79,80] all came before the rule was written, not after. The supported lapse after the rule is the plan with no search first, and the two replies with no coverage statement. The record shows no compaction between the rule and the plan (the fourth fold was at 06:05:44[81], and the fold count still read four later[82]) and no cross-session loss in

between, so this is not the loss of Section 3.2. The rule was in the operator's memory and the harness put it into the window, so what failed was not the carrying of the rule: the next action on a new task did not reflect it until the operator asked. It cost little, one plan caught in 83 seconds. It is the case behind the open tension of Section 1: the rules paper reports that a rule loaded before the work starts fires, and here a rule in the window did not govern a plan 6.6 hours after its re-injection. Neither paper tests whether loading time or distance makes the difference.

Outside studies, newest first. Jiang et al. (refactoring) (3 days; partly read) found the opposite error in refactoring sessions, agents that kept editing after declaring the task complete (Claude Code with Opus 4.8 and Sonnet 4.6, not the 5.5 models), so stopping errors run both ways and the 16:02 plan is one setting. Khezresmaeilzadeh et al. (10 days; abstract plus a text search) found that in all 12 configurations a logged record moved commitment decisions more than the observation did (small open models), and Zhang et al. (Judged Useless) (24 days; full text, sections read) found that a stopping rule written into the prompt was followed at most partly while stopping followed the evidence only when the harness enforced an integration step. Section 7 gives the full review.

Loss two.

Late in the day the operator said "right, so this has all been out there"[14]. The assistant answered with a table of seven things it had offered that day and where each already was, and said "That's seven of seven." The table below checks that against the files. The assistant offered R1, R3 and R6, and partly R4, as new before finding that they existed; R2, R5 and R7 first appeared in its replies as retrievals. **The assistant reached most of the seven only after the operator pushed (R2 came in reply to a question, and R6 matches only partly), and in about half it first offered the idea as its own. Nobody counted the answers that turned out to be new.**

Three corrections to the assistant's table. The ruling behind R3 is dated 3 September 2026, not 11 September, when the file that quotes it was first committed. A claim in its table about how a July commit was merged could not be verified, and the note does not use it. The line it cited for R6 is about the climb and the loop, not about the ladder of data, information, knowledge, insight and wisdom: the operator said "the learning ladder and the DIKIW are basically the same"[3], and that equivalence is the operator's; the 22 July session of a second agent supports it without ruling on it. Git dates are as committed (US Eastern time) and transcript and table times are UTC, so the 15 July commit is 16 July in UTC, 94 days before 18 October (95 by the committed date). Its reading of why[75] is that the thinking was not missing: every agent's frame closed before it reached what was already written.

The assistant's account of why the answers were on file and not in hand[76] is that data and information are written by machines, append-only (transcripts, the message log between agents, git, the exports), while knowledge, insight and wisdom live in prose, in a reply, a commit body or a docstring, and nothing types them or carries them to the assistant at the moment it acts. Section 2 gives the counts for how much of the operator's record is typed. The failure to read before claiming is measured separately. The documentation of a stop check in the operator's code that blocks claims made without a read (a measurement dated 3 September, no rater involved) counts 157 of the operator's turns in the chat corpus that catch a state claim with no read behind it, and 52% land directly on an agent turn that already claimed to have read or loaded something. The

operator's scoreboard file measured the same construct in both channels, at its declared scope (188 conversations, 6,686 of the operator's turns, re-read 11 September): 56% of such turns in coding-agent sessions and 55% in chat. The assistant has not reconciled the two corpora. The check has refused 220 times in 25 sessions since 3 September, as counted at 10:18 UTC on 29 October (the file is live, so the count grows).

Outside studies, newest first. R. Wang et al. (More Memory, Worse Agents) (10 days; partly read) found persisted abstractions below raw trajectories in 7 of 9 cells, and Zhao et al. (13 days; abstract only) found that failure-derived experience repairs failed tasks but rarely yields sustained improvement on unseen tasks. Against that, Du et al. (Relic) (33 days, v2 29 Sep; abstract plus skim) measured fresh-member behavioral correctness at 25.4% with no inherited protocol, 34.6% with the lessons as readable text and 41.2% with lessons bound to the runtime, so upper-rung material can persist when it is carried in a form the agent meets. Zhang et al. (memory) (169 days, v2 29 Aug; full text, sections read) (background by age) found that what degrades is the LLM-consolidated layer of memory while raw trajectories stay competitive, which bears on how a record is written (Section 5).

Appendix E. Serving in detail

The full text behind Sections 5.8 to 5.12.

E.1 At start-up: a context channel, not the boundary

Status. Approved by the operator on 27 October; not built. A short section of held records (the operator's verbatim line, its level, its date and a pointer to its source) is rendered first on the page every agent session reads when it starts, and by the same renderer on a status page that another assistant reads and cannot edit.

The reason is a case from 27 October, recorded in the design that day. After a "load" of the task list, the message channel and memory, the assistant judged the project by buyers and revenue, though the operator's line of 18 October was already in memory: "Youre losing the KIW and that's what I've really been trying to sort out"[4]. The assistant named three causes: a load rebuilds task state, not insight; the compaction summary keeps identifiers and drops conclusions; and the memory search was phrased in the wrong frame and returned the wrong rows. This is one case, in the assistant's own account, and no second reader has checked it. The operator's view of the standard remedy is on record: "it wont help throwing it in memory"[29].

Start-up serving is therefore a context channel and is not claimed to be a boundary. Loss one (Section 3.1) is the reason: a rule was in the window, put back by the harness at 09:29, and did not govern the plan written at 16:02. Text in the window may still not govern the act, and Section 1.5 states the tension with the rules paper, which finds that a rule loaded before the work starts fires. The start-up section is expected to help most where the failure is framing, as on 27 October, and least where a specific act must be stopped; the second job belongs to the stop.

E.2 At the stop: the lookup boundary

Status. The work order at the stop was part of the loop the operator approved on 18 October. The framing as a boundary on the lookup is in the plan approved on 29 October, and it is a design, not a result. Not built.

When the agent is about to give a final answer on a question for which a held record exists, the answer is refused until the record is in view. The record is then served as evidence in the form of a work order: what the record says, what the turn says, and which source to check before continuing. It is never "are you sure?". In the debate the operator's tagged pushes each named a place to look (Appendix D.3), and the assistant's reading was that "are you sure?" hands the reasoning back to the one that cannot do it. The operator put what his pushes supplied this way: "I'm doing the reasoning and pushing you in the right branch as best I can"[66]. A work order names the branch where it can be named, and leaves the reasoning with the agent.

The trigger has to be wider than the stop check that exists. That check refuses a claim made without a read, and has done so 220 times in 25 sessions since 3 September, as counted at 10:18 UTC on 29 October 2026 (the file is live). The proposed trigger also covers absence and origin claims ("it isn't there", "that's the origin", "never discussed"), which include the claim that an idea is new, the claim the assistant made for three of the seven answers of 18 October, and part of a fourth, before finding them on file (Section 3.2). The existing absence pattern also misses contractions. The cases where a stop-time lookup would apply earliest are the nine same-window and compaction pairs of Section 3.3, in which the loss came inside one window or across one compaction.

Outside evidence. None of the three studies below serves anything from past conversations. A rule-based monitor that surfaced the evidence pattern restored task success from 7.6% to 21.8% on one 30B open model (Chen et al. (Knowing When to Stop) (7 days; main text read)). A gate before acting on a stored memory cut verification cost by 33.9% in simulated tool environments, with no Claude model (Y. Wang et al. (AMV-Guard) (7 days; abstract plus a text search)). Across seven frontier coding models an enforced completion gate gained 4.0 and 5.3 points on two benchmarks, and a prompt instruction alone gained 3.3 and 3.9 (Zhang et al. (Rethinking Verification) (4 days; main text read), no Claude model). Serving an insight as evidence at the stop is untested (Section 7).

E.3 Matching by the question settled

Status. Proposed; not built. No index of held records exists.

A record is served when the question in front of the agent is the same one the record settled, however differently the two are worded. An outside model makes that judgment, reading the held records and the past conversations. The operator's own words describe it: "the outside model, interacing with the frontier using the insights from above and our previous convos"[2]. The design needs a field the loaded table lacks, the question each record settled, written at capture (Section 5.2).

Word similarity alone has failed once. The retrieval engine of September, similarity only, scored a lift over chance of 0.100 (interval -0.043 to 0.237) and over its fallback of 0.054 (interval -0.097 to 0.194) on 93 held-out pairs. Both intervals include zero, so the version did not clear its pass line, though 93 pairs are too few to tell a small lift from none. The operator authorized the fix on 11 September and it is scored on fresh pairs only.

An informal keyword lookup on 22 October put the expected record in the top 5 for 2 of the 7 answers of 18 October (Section 6), and the reason was the one the design predicts: the queries were phrased the way the assistant said the idea, not the way the record states it. The design therefore moves the matching to a model that reads meaning, plus the record's span.

A wrong match has a price. Retrieved memories can mislead even when correctly stored, most under partial overlap between the query and the memory (Tang et al. (24 days; abstract only)). That is a reason to serve the record as evidence with its source, so the agent can see when it does not fit, and not as an instruction.

E.4 Applying wisdom: a ruled priority, weighed at a decision

Status. Proposed; not built; no wisdom record has been served. All of it is design.

What a wisdom record is. A wisdom record is a priority call the operator has ruled on, with the objective it assumes. It is not a fact to check and it is not a rule to obey. It carries the operator's words verbatim, the date, a pointer, and the objective it was ruled against. The 15 held records, two of them wisdom, carry a one-line note of why each matters and no field for the objective, so that field is proposed. The objective used below, to keep and compound knowledge, insight and wisdom, is the assistant's reading of held line 1, for the operator to confirm.

When it is served. At a decision of that kind, not at a factual question. A question is matched by what it asks; a decision is matched by what it chooses between, a harder judgment that the design gives to the same outside model, with accuracy unmeasured.

Weigh, not obey. The record arrives with its objective and with what weighs against it, and the agent says what it weighed and what it chose. A choice against the record is allowed and is stated. Only the operator retires a record.

Held line 3 (wisdom, 18 October): "it wont help throwing it in memory, that's the "standard" solution right? And it doesn't work lol."[29] *Served at* a decision about how to stop a repeated miss, when the proposal is to write a memory row, a note or a rule into a file. *Weighed against* the higher cost of building a check or a serve step than a memory write, and outside evidence that written lessons are half right: verified lessons helped Claude Opus 4.6 and Sonnet 4.6, and Haiku 4.5 fell (Fan et al. (KEEL) (4 days; full text)). A memory row can be enough for a one-off. *Applied would show as* a changed action: the next proposal for a repeated miss names where the fix is enforced, or says why a memory row suffices. It would show as a changed rule if the acceptance test for new rules asks where each is enforced.

Held line 5 (wisdom, 16 October): "all you can do is influence an LLm"[7]. *Served at* a design that relies on an instruction to produce a guarantee, such as a prompt that says the model must never do something. *Weighed against* the cost and delay of a check at every edge, and against the fact that instructions do some work: in the study above, an instruction alone gained 3.3 and 3.9 points where an enforced gate gained 4.0 and 5.3 (Zhang et al. (Rethinking Verification)). The gap is small, so the line has to be weighed and cannot be applied. *Applied would show as* a requirement that moves from prompt text to a check, or a design that states which requirements are enforced and which only influence.

How an effect could be observed. By what it changed, which can be seen, and not by how wise it is, which cannot: an action, a rule or trigger, or how rules get made. For each serve the decision, the line served, what the assistant weighed and what it chose would be logged, and the same decision replayed without the record gives the comparison. Neither example has been tested.

E.5 Reach: where the stop can be served and where it cannot

Status. The reach statement is in the loop approved on 18 October. Serving inside a session is not ruled.

Coding agents can be served by a stop hook now; the hook is the transport in the design and is not built. The assistant's own platform cannot be reached by hooks today, so no stop-time serving exists there and no transport has been designed. For that platform the design of 27 October adds one procedure: before a big-picture question, the assistant retrieves the operator's own words on the topic from the archive and the operator's memory database, searches for his words and not its own framing, and quotes what it used. That is a rule of the first kind, a procedure the assistant has to read and follow, so Loss one applies to it, and the operator's work orders remain the check. A conversational assistant on a harness the operator can hook is a separate architecture, not built and outside this plan, as are model training and the other transports. Serving a record within the session that captured it is not ruled.