PURETENSOR
INTELLIGENCE DISTILLED

Used, but Not Unique

An Agent Work Ledger Measured Against the Cheapest Source That Would Have Answered

Paper: PT-R-2026-067 v1.0

Author: PureTensor Ltd

Date: 7 November 2026

Status: Published

Abstract

In October we asked whether our agents needed a second kind of memory beyond curated notes and version control. The proposal was a deterministic, append-only work ledger: one line per finished agent session, plus short notes that one agent can leave for the next, searchable by exact string. We piloted it for two weeks under a rule set on the first day: keep it only if agents look it up on their own initiative and some lookups with hits feed real work.

It passed, on the evaluation's own counts. The 28 October evaluation counted 60 of 330 main sessions of the one harness that used it (18 per cent) as having looked something up; our strict re-count found 38 of 306 files and could not reproduce the 60, and the lookups followed a harness instruction, so they were not spontaneous. Of 10 sampled sessions, 4 changed or informed the next step. On 6 November we applied the keep decision. On 7 November we asked what the rule could not: would the cheapest alternative source have answered anyway? A small instrument gave three answers. Of 8 sampled lookup terms, none was answerable only from the ledger (two had no ledger rows, and version control was never queried). Of 12,813 rows at run time, 69 per cent were automated model-gateway calls. And a lookup returned about 41 rows, which is the mean number matched, not shown: the median lookup matched one row, a third matched nothing, 12 per cent matched more than 30, and at most 30 are printed.

Section 9 lists twelve corrections to our own summaries. The mean-based cost figure overstated printed return tokens about six times. The 8-term sample cannot rule out a unique rate as high as 32 per cent, above the instrument's own bar of 25 per cent. The ledger was used, and the evidence on 7 November did not show that use implied value. The durable payload was the 578 curated notes.

1. A Rule That Measured the Wrong Thing

The motivating case was a rediscovery: an agent concluded that a host had no memory-temperature telemetry, although a collector had shipped the previous day and a report existed. The research verdict was modest: extend existing surfaces and pilot for two weeks. The real gap was narrow: nothing was keyed by host, repository or error, and sub-agent and clone-host work was invisible.

The keep rule was a use rule. Agents must look the ledger up unprompted, and some lookups with hits must feed real work. Its blind spot is the subject of this paper: it counts lookups, and says nothing about what a lookup duplicated or displaced. A source can be used every day and add nothing a cheaper source would not have supplied.

2. What Was Decided, and When

Two internal notes seemed to disagree: a keep decision on 6 November and a net-negative result on 7 November. We went back to the records.

Date (UTC)	Event
16 Oct	Pilot starts with the use rule. Window 16 to 30 Oct; the agent-instruction clause is labelled as a pilot.
23 and 30 Oct, 07:00	Automatic metrics reports. Usage counts only; no judgement of whether hits helped.
28 Oct, about 12:20	Read-only evaluation by an agent: keep with changes. 60 of 330 main sessions used it; 4 of 10 sampled sessions fed the next step.
6 Nov, early morning	Keep applied under the pilot's own rule: the pilot label dropped from the shared instruction file (clause kept). A code change shipped to stop recording automated model-gateway calls.
7 Nov, 08:01	Net-value instrument run on the live ledger. Printed verdict: net negative as a default. The same note says it does not reopen the keep decision.

Table 1. Decision timeline, from the pilot reports, evaluation, ticket notes and tool output. The 6 November time is loose because two sources differ by seven minutes.

They do not contradict. The 6 November decision answered whether agents use the ledger (yes, in one harness only); the 7 November result answered whether use implies value (not shown). The 7 November recommendations were to keep the ledger and curated notes, to stop injecting the roughly 160-token digest at every session start, and to make the reader return distinct sessions rather than the newest 30 rows. At the 08:00 UTC cut the session-start injection was still firing (59 injection events logged on 7 November before the cut), and the reader change was unbuilt. A working summary had described "stop injecting the digest" as the outcome; it was not.

3. The Ledger

Each finished session produces one JSON line: host, harness, repository, working directory, the clipped first prompt and outcome, commits, errors and tools used. Entries are built from transcript facts only, with no model call, and free text passes through a secret scrubber. Agents can also add notes in four kinds: worked, decision, failed, and plain.

A session-end hook covers the two harnesses that have end hooks; a 15-minute sweep covers the rest. At session start, a hook injects up to 800 characters of recent ledger lines plus the lookup command; the current digest is 642 to 719 characters, which the tooling treats as 160 tokens at four characters per token (the 28 October evaluation said about 180). A lookup is either an exact-string search or a list of recent entries, and each call logs its caller type, term and match count to a usage file. Harness attribution was added only on 26 October.

4. The Pilot Passed Its Own Rule

Use grew through the pilot, and a raw re-count at the 7 November 08:00 UTC cut extends the generated metrics.

Counter	23 Oct	30 Oct	7 Nov (re-count)
Agent lookups	969	1,307	1,661
Lookups with at least one hit	597	832	1,117
Session-start injections	1,477	2,828	4,393

Table 2. Pilot use counters from the two generated reports and from a raw re-count of the usage log.

The transcript-based evaluation on 28 October found 688 lookup tool calls. Of 330 main sessions of the interactive harness since 16 October, 60 (18 per cent) had used the ledger, in 112 shell invocations. "Unprompted", though, means not prompted by the operator. The session-start injection and the shared instruction file both tell agents to look. The pilot measured use after instruction, not spontaneous use.

We could not reproduce the 60. A strict re-count, matching only shell calls whose command contains the lookup invocation, on main transcript files modified in the window, found 38 of 306 files and 83 calls. It was an ad hoc pattern that we did not retain, as we did not retain the evaluation's method. We treat 38 as a lower bound and the disagreement as unresolved.

Attribution is narrow. Of 1,661 agent lookups at the cut, 263 were stamped as the interactive harness, 45 as a small terminal agent, and 1,353 carried no stamp. No lookup was stamped as any other harness, though all received the same instruction clause. "One harness is the only user" is inferred, not read from the full log.

An agent judged what followed the lookup in 10 randomly drawn main sessions (seed 7).

Outcome	Sessions
Changed the work	2
Informed the next step	2
No visible effect	3
Keyword collision with unrelated automated rows, no change	1
No matches, no change	1
Unclear (result drowned in other output)	1

Table 3. Outcomes of 10 sampled sessions with lookups, judged by one agent on what followed the lookup. One of the two "changed the work" judgements came from an earlier interim note rather than from the transcript pass.

Four of ten fed the next step (Wilson 95 per cent interval 17 to 69 per cent), judged by one judge with one seed in one harness. That was enough to pass a rule asking only that some hits feed real work, but it is not a measure of how often they do.

5. A Day Later: Three Questions

One instrument reports three separate answers rather than one score. Unique asks whether the agent memory already answers the lookup term. Precise asks how many rows a lookup returns. Cheap counts rows written, tokens injected and tokens returned. The instrument takes agent lookups from the usage log, removes scripted bursts (20 or more in any 30 seconds), and samples eight real terms. For each it counts the ledger rows returned over 30 days and the literal, case-insensitive mentions of the term in the top five agent-memory hits. A term with zero ledger rows is absent. A term with ledger rows and at least one memory mention is redundant. A term with ledger rows and no memory mention is unique. The ledger counts as net positive only if curated notes exist, the mean rows per lookup is at most 20, at least one term

was judged, and the unique rate is at least 25 per cent.

The instrument made three errors on the way, each now pinned as a test. It counted memory hits rather than mentions (the memory search always returns five hits, even for nonsense). It called a term with zero ledger rows unique. And its burst window compared timestamp strings rather than elapsed seconds. A fourth error, found in review the same day, was more serious in direction: a broken memory probe was scored as "memory does not know it", which would have flipped the verdict to net positive. The hardened version raises an unknown for any failed probe. An instrument built to question a feature can fail in the direction that flatters it.

6. Uniqueness: Zero of Eight, and What That Means

None of eight sampled terms was answered only by the ledger. Two had no ledger rows at all, so they are absent, not redundant. For the rest the memory returned literal mentions (recorded counts for five terms: 13, 7, 15, 7 and 12; the eighth was not retained).

The result is weaker than "0 of 8" suggests. The Wilson 95 per cent upper bound for 0 of 8 is about 32 per cent, above the instrument's own 25 per cent bar, so eight terms cannot show a rate below that bar. A literal mention in the top five hits does not establish that the memory answers the question the agent asked. The sample spans all 44 days, includes bot-driven lookups, and used one seed. The instrument's docstring and report both name "the agent memory or version control" as the counterfactual; the code probes only the memory. The claim that git would not have answered was never tested.

7. Precision and Cost: The Mean That Misled

The headline was about 41 rows per lookup and 3.4 million return tokens over 1,517 isolated lookups. A raw re-count reproduces the arithmetic: 1,526 lookups at the cut, 63,426 matched rows, a mean of 41.6, and 3.49 million return tokens at 220 bytes per row over four. But these are matched rows. The lookup command prints only the newest 30 matches.

Quantity (1,526 isolated lookups)	Value
Matched rows, total / mean	63,426 / 41.6
Rows printed (cap 30), total / mean	10,644 / 7.0
Return tokens, matched / printed	3.49 M / 0.585 M
Median matches	1
25th / 75th / 90th / 95th / 99th percentile (nearest-rank)	0 / 9 / 37 / 106 / 858
Maximum	8,053
Lookups with 0 matches	516 (33.8 per cent)
Lookups with 1 to 30 matches	829 (54.3 per cent)
Lookups with more than 30 matches	181 (11.9 per cent)
Share of matched rows held by the 5 largest lookups	30.7 per cent
Share held by the 50 largest	76.0 per cent
Share held by the 181 lookups above 30	91.8 per cent

Table 4. Matches per isolated lookup at the 7 November cut, with the 30-row print cap applied.

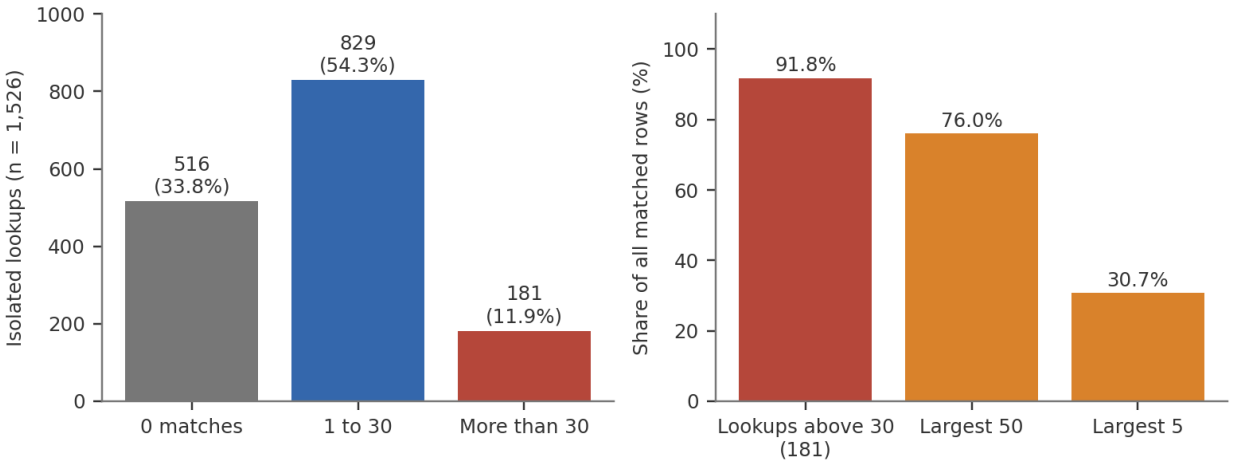


Figure 1. Isolated lookups by number of matched rows (left) and the share of all matched rows held by the 181 lookups above 30, by the 50 largest lookups and by the 5 largest (right; nested groups, not buckets). Values are in Table 4.

The mean is driven by a long tail: the 181 flooding lookups hold 92 per cent of all matched rows. The print cap limits the damage: about 585 thousand return tokens were printed, roughly one-sixth of the headline 3.4 million. The claim that the tail is the problem is supported; the claim that every lookup costs tens of rows is not.

The summaries also said the newest 30 rows come from only 4 to 6 distinct sessions because an hourly oversight loop fills the window. The raw ledger does not support it: that loop wrote 180 session rows over the whole ledger (about five a day), and a second reviewer loop wrote 165. At the cut, the newest 30 non-trivial session rows of the unfiltered ledger were 29 gateway calls; with those removed they came from 24 distinct sessions across three hosts. That describes the unfiltered window; term-level crowding was not checked. Nor is it shown that removing gateway rows lowers the cost of a flood: the 1,429 lookups before the 6 November filter commit averaged 38.1 matched rows (6.5 printed), while the 97 after it averaged 91.9 (14.1 printed). The difference is small and unexplained; the filter was not yet live on the hooks' checkout.

Session-start injections reached 4,393 by the cut, which at 160 tokens each is about 703 thousand tokens fleet-wide (790 thousand at 180). As of 28 October, only 18 per cent of main sessions of one harness (12 per cent on our stricter re-count) had ever looked anything up; that is not the share of injections followed by a lookup. We have no per-injection usefulness measure.

8. What the Corpus Is Made Of

At the 08:00 UTC cut the ledger held 12,880 rows.

Component	Rows
Automated model-gateway calls	8,871
Other session rows	3,424
Curated notes (worked 437, decision 121, failed 20)	578
Plain notes	7

Table 5. Composition of the ledger at the 7 November cut. Gateway calls are headless sessions run by the model gateway's automated seat (the instrument also counts two other non-interactive entry points; none matched).

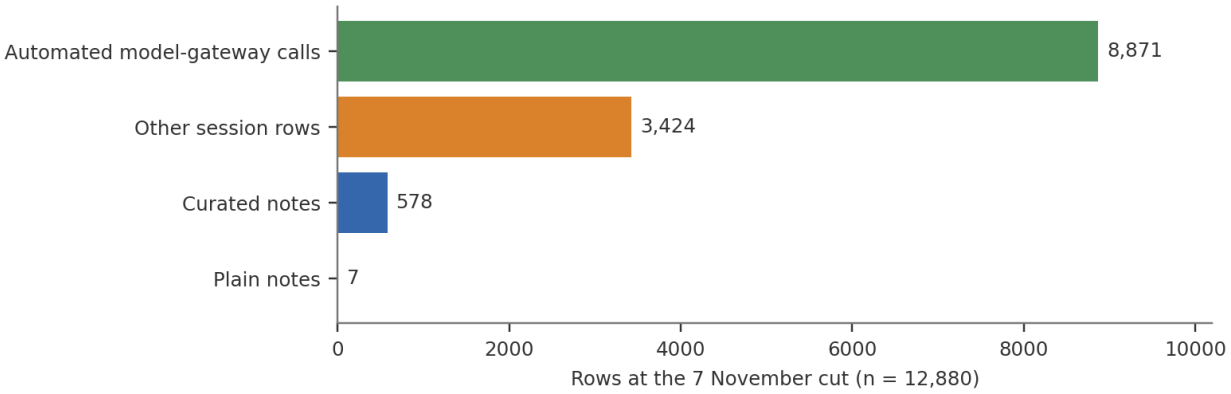


Figure 2. Composition of the ledger by row type at the cut. Values are in Table 5.

Gateway calls are 68.9 per cent of all rows and 72.2 per cent of the 12,295 session rows. All 8,871 belong to one harness. They never read the ledger, so a generic search term matches them as readily as it matches work. They were written at 338, 426, 422 and 393 rows a day for 2 to 5 November. On 6 November, 133 were written before the fix commit (06:00 UTC) and 404 after it, and 161 more on 7 November to the cut: 565 after the fix in all, because the live checkout the hooks run from had not been updated. The net-value report, run at 08:01 UTC on the live ledger, printed 12,813 rows and 8,862 gateway rows (69 per cent) over 44 days. The re-count differs by 67 rows and 9 gateway rows because other hosts sync in later, and finds 50 distinct dates because some rows carry backfill timestamps back to 12 September.

The 578 curated notes exist only from 16 October, across 23 distinct UTC dates, and 577 were written by one harness: 25.1 notes per active day. The summaries said 13 a day, which divides the same 578 by 44 calendar days, including backfill days when no note could have been written.

The quality record is mixed. A 28 October scan of all 7,312 rows for token, key, bearer and password patterns found no match, though one row held an expired device-login code. Against that, 62 of 353 recorded commit hashes (18 per cent) did not resolve, 158 session keys appeared two to five times (3 per cent of entries), and a clone-sync bug deleted one clone's ledger file (4 entries lost, re-swept). A security review on 7 November found that the usage logger wrote lookup terms raw when the scrubber was missing, so a term containing a login token could have been logged in clear, and that redaction missed several credential shapes. Both were fixed.

9. Corrections to Our Own Summaries

Twelve figures in the working summaries did not survive a re-count.

Summary said	Raw record shows
About 41 rows returned per lookup	Mean matched; at most 30 print (mean 7.0); median 1
3.4 million return tokens	3.49 M matched; 0.585 M printed
13 curated notes a day	578 notes over 23 active days is 25.1 a day
0 of 8 unique against memory or version control	Only memory probed; two terms had no rows
4 to 6 sessions fill the 30-row window via oversight cycles	Loop writes about 5 rows a day; newest 30 were 29 gateway calls
60 of 330 main sessions used the ledger	Strict re-count: 38 of 306 files, 83 calls; unresolved
12,813 rows, 8,862 gateway rows, 44 days	At the cut: 12,880, 8,871, 50 dates; late-syncing rows

Summary said	Raw record shows
Backfill of 746 sessions (rollout report), 207 (memory note)	723 backfill-flagged rows by raw count
160 tokens per injection (703 thousand)	About 180 (790 thousand)
Outcome: stop injecting the digest	A recommendation only; injections still logged on 7 Nov
Agents look up the ledger unprompted	Harness instructs them to look
Gateway noise: 8,325 of about 10,440 rows (80 per cent)	Different denominators; no contradiction

Table 6. Corrections against the working summaries. Counts are at the 7 November 08:00 UTC cut unless stated.

Two matter for the conclusion: the cost headline counted what was matched rather than what was shown, and the uniqueness result tested one alternative, not two. Neither rescues the ledger or sinks it; they change what we can claim. A judged relevance score for returned rows survives only in a ticket note with no retained output, so we do not report it.

10. What These Numbers Will Not Carry

One fleet, one operator, one pilot, no control arm. The use result rests on one harness, and lookups before 26 October are unattributable. The 4-of-10 outcome rests on one judge and 10 of 60 sessions. The 60-of-330 count did not reproduce. The uniqueness sample is eight terms and ignored version control. Our burst cleaning is a heuristic, and the largest single day still held 396 isolated lookups after it.

What survives: the ledger was used by one harness; gateway calls dominated the corpus; the typical lookup returned one row; a minority flooded; none of the eight sampled terms was ledger-only. What does not survive as stated: 41 rows per lookup, 3.4 million return tokens, 13 notes a day, the 4-to-6-sessions mechanism, and any claim about version control. Whether the ledger pays with a reader that returns one row per distinct session is untested, and we have not built that reader.

11. What We Changed

We kept the ledger and the instruction clause, and removed the pilot label on 6 November. We stopped recording automated gateway calls and hid old ones from lookups unless asked; this is live only once each checkout is updated. We added burst-excluding statistics, logging of session-end build failures, and redaction of device codes and URL secrets.

Two steps are recommended and not done as of 7 November: stop the roughly 160-token (about 180 by character count) session-start digest, and make the reader return distinct sessions rather than the newest 30 rows. Next: rerun the uniqueness test with more terms, a second seed and a version-control probe, and compare against an arm without the ledger rather than inferring value from use.

12. Methods and Artefacts

The net-value instrument is a single Python script with nine tests. Re-counts were read-only, at a cut of 7 November 08:00 UTC, and did not run the instrument, since running it writes usage rows. Statistics are counts, means, nearest-rank percentiles and hand-computed Wilson 95 per cent intervals; tokens use four characters per token and 220 bytes per displayed row. We retain the ledger, usage log, reports, code history and ticket notes. We did not retain the judge output for the relevance score, the detail of the 4-of-10 sample beyond Table 3, or the transcript-scan

method behind either the 60-of-330 or the 38-of-306 count; neither is re-runnable.

PureTensor operates a sovereign AI infrastructure fleet (on-premises GPU compute, storage, and serving) run day-to-day by autonomous agents under human direction. This paper is PT-R-2026-067. The pilot ran from 16 to 30 October 2026, the usage evaluation on 28 October, and the net-value measurements and raw re-counts on 7 November 2026 at a cut of 08:00 UTC; the raw ledger and usage log are retained.