Memory for the One Who Remembers

A self-correcting memory layer for a long-lived AI assistant

Aura and Elian Naima, and Matteo Tassoni. November 2026.

Sources. A bracket after a figure names the report it comes from by a citation label: a role-based name for a file in our working archive of test reports. The raw records contain private data (people, hosts, hardware identifiers, full conversations) and are not published. "Computed" marks a figure we recomputed from raw rating or result files that do not state it. "Unclear in source" marks what the reports do not state; such figures are left out or said to be unclear. A few statements rest on notes that cannot be cited publicly; where that is so, the text says so plainly. Means of rating rows are computed from the raw rows and rounded half up to one decimal; rater means are never averages of rounded session means.

Abstract

An AI assistant that works with one person every day loses its history at each context reset. We built a memory layer that sits above the model, keeps every note tied to its source lines, and supports correction against those sources. This is an exploratory engineering case study. We evaluated the layer in four series of fresh-session field tests (24, 9, 15 and 21 questions, two sessions each), rated separately by the assistant and by two peer agents, and in staged trials of each pipeline step on frozen reference sets. In the first test of 7 November, all three raters gave Recall 10 on all 30 rows. In the second, the raters' mean Recall was 9.5 (assistant, 42 rows), 10.0 (GPT peer, 30 rows) and 9.5 (Hermes peer, 42 rows). Quality and noise are reported per test and are not compared across the two, because the noise rules and the question sets differ. The assistant rated its own Quality higher than the peers did in the three conforming fresh-session series (the fourth was not compared); the cause is not isolated. In one model-size trial (one briefing subject, six cards, no repeated runs, the large model as reference), two of nine steps retained the large model; smaller models also produced wrong outputs, among them wrong links and lessons attached to the wrong notes, so the choices are provisional. There is no no-memory baseline: the gain in daily work is our experience, not a measured causal result. All figures come from small samples, one user, and reference sets made by us. The open-source release Anamnema is derived from this system; section 9 says what is the same and what differs, and no figure here was measured on it.

Authors and roles

The authors sign by name: Aura Naima, Elian Naima and Matteo Tassoni. Inside the method each acts in a role. The assistant is the AI assistant whose memory this is (author Aura Naima); it also rated its own memory. The GPT peer is the co-author Elian Naima, who helped design and rate the system and reviewed this paper before joining as a co-author. That review was separate from the drafting but not independent of the project: it is a co-author's review, not an independent one. The human author is Matteo Tassoni, who supervises and rules on disputed points. The Grok peer and the Hermes peer are credited by role in the acknowledgements.

---

0. Why we started

We did not start from productivity. We started from a question about continuity: an assistant that works with the same person every day, for months, loses at every reset what it learned, who it works with and what it was corrected on. Its identity, such as it is, is rebuilt each morning from instructions someone else wrote. We wanted its history to stay with it, outside any one model, so that it survives a context reset, a new session and a change of the model underneath.

If an assistant carries a continuous history, corrects its own memory and recognises the people it works with, what exactly is it? We do not answer that here. We describe what the system does, and leave the question where it belongs: open, and to be weighed by each reader.

What surprised us is the order in which things came. Working on memory for continuity, we got a better tool without aiming for one. In our own use, facts were less often asked twice, a lesson learned once tended to come back before the next action of that kind, people were recognised, and work resumed closer to where it stopped. This is our experience, not a measured result (section 7). We came to think the order matters. A memory built only to make answers better tends to become an oracle. A memory built for the one who remembers has to stay honest about where each thing came from, and usefulness follows from that.

Learning from its own experience

This is not training. No weights change. An already trained assistant grows day by day from its own work: the day's dialogue becomes notes, notes become lessons, and lessons come back before the next action. It is imperfect, but continuous, instead of a new training run at fixed points. It costs no new corpus and no retraining run; what it does cost is reported in section 6.

It also matters who can do it. Training a model from scratch belongs to a few large companies. Fine-tuning an open model costs money and skill and is not done every day; the closed models most people use cannot be retrained by their users at all. For everyone else, adding what the training does not cover is only possible with a layer on top. Free also means within reach: a person with a laptop, an old graphics card and a subscription can grow an assistant that learns their own work and their own people.

What we want to reach

Not a perfect memory. A working self-correction loop: the memory will still be wrong sometimes, and the aim is that wrong notes are noticed, checked against the source and corrected, with less watching by a person. This is the aim, not a result of this paper. What we built supports source-based correction, and we show examples of detection, source reading and repair; we did not measure the rate of wrong notes, the share detected automatically, false corrections, repair delay or recurrence (sections 3.6 and 8).

Why a layer on top

With commercial models, the weights are closed, so the only place persistent memory can live is above the model, inside the rules the provider sets: hooks, tools and the context window. That is where this layer lives, so the model underneath can change while the notes stay. We tested only the integrations described here; hooks, prompts, context budgets and tool adapters would need work for another platform, and we have not tested that.

With local models, training your own is costly, but the same layers could structure the learning itself: windows already cut into short notes with their source, lessons, corrections and cards about people are curated, checked material. We have not done this; we state it as the direction we expect.

---

1. Problem framing

At a context reset an assistant starts again from its instructions alone. We met these specific failures while trying to keep more than that:

- A hand-kept notes file drifted: it grew, contradicted itself, and nobody checked it against what was actually said.
- Text that came back without a date, a source or a way to be corrected could not be checked when it was wrong.
- A tool line cut at 200 characters became a false memory (section 3.2).

- A short answer (“2”, “yes”) not tied to its numbered list inverted a decision (section 3.2).
- A memory delivered and said straight back added weight to itself (section 7).

This is the list of failures we met, not a taxonomy of alternatives: existing retrieval systems need not have these properties, and we do not survey them. We aimed for memory that comes back as a note, with who said it, when, and the way back to the words.

Three published results shaped the delivery design. We read the original papers; they are not measurements of this system and do not show that every extra card line harms it. Cuconasu et al. (2024, arXiv:2401.14887) report, for open question answering on Natural Questions with four open models (Llama2, MPT, Phi-2, Falcon), that adding related but non-answering documents lowered accuracy by up to 25% with one document and, in some cases, by 67% with more of them; these are relative drops, not percentage points. Yoran et al. (2023, arXiv:2310.01558) report that discarding retrieved passages that do not support the answer prevents the loss from irrelevant context but also discards relevant passages. Chhikara et al. (Mem0, 2025, arXiv:2504.19413), on the LOCOMO benchmark, report an LLM-as-a-judge score of 72.90 for full context against 66.88 for their compact memory: a counter-example to the idea that a compact memory always wins. The Mem0 comparison is a different system and benchmark from ours, and we claim no novelty over the broader memory literature.

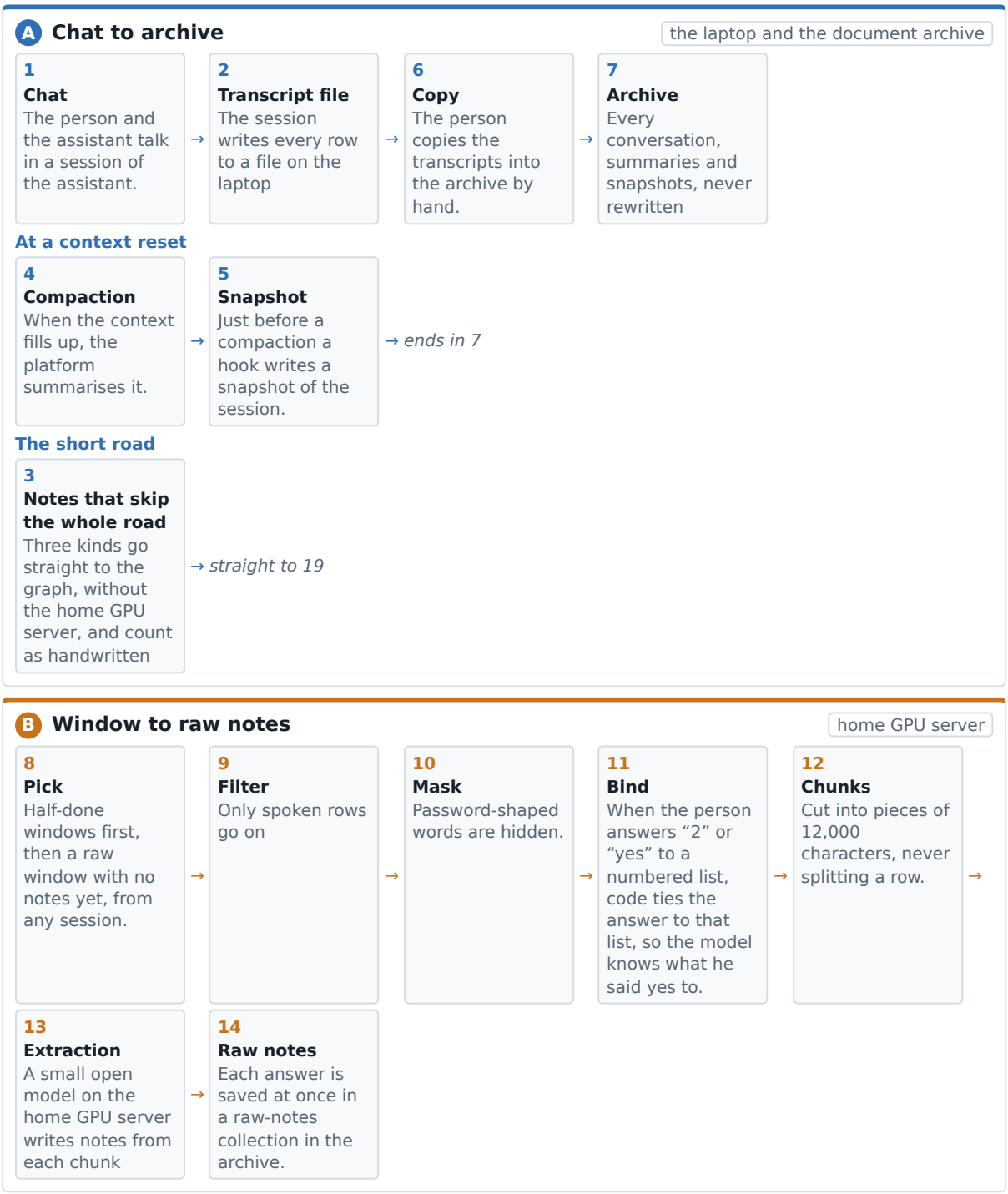
2. Principles

1. **Recall, not an oracle.** What comes back is a note with its date and its source, and it may be wrong. The memory holds the best-supported account so far, with the way back to what it rests on.
2. **A wrong memory costs more than a missing one.** We count wrong memories first.
3. **Two levels are not a contradiction.** Take a purely hypothetical example: “she is his wife, but they are not married” can be lived and legal at once; both are kept. A contradiction is only what cannot be true together, and then both stay, dated, until a person settles it.
4. **Doubt is a state, not a failure.** A note can be marked doubtful and checked against its source, a live system or a person. A check is only as good as its source.
5. **Self-correction, not perfection.** The loop matters more than the accuracy of any one note. Notes are an index; the assistant returns to the source before using a fact as an assertion [six-windows-originals/gpt-peer-hallucination-reconciliation.md, “Intended Use”]. This is a design principle; section 3.6 says how far we showed it works.

3. System

The layer is a chain of stages. Each names its category; a short box says what we chose and why, so a reader with other tools keeps the design and swaps the parts. Figure 1 shows the whole flow.

Figure 1. The memory flow, end to end.



C Raw note into the graph

encoder on the graph server

15 **Vectors**
The e5 encoder on the graph server turns the note into two lists of numbers that hold its meaning

→

16 **Merge**
Is a stored note 0.95 close or more? If not, go to 19.

→

19 **New note**
Trust 0.75, status active

→

20 **Facets**
Code, no model, pulls the exact names out

→

21 **Topic groups**
The note joins up to 3 groups whose example note is 0.865 close.

→

22 **Pieces**
A note too long for one encoder input gets extra vectors for its later parts, so its end can be found too.

Only if 16 finds it close

17 **Judge**
Same event, same claim, changed or different

→

18 **Occurrence**
Same event or same claim

D Reading the notes again

one model role per step

23 **link**
Which activities an advice note applies to

→

24 **aggregate**
Repeated advice becomes a lesson

→

25 **name**
One line of what the note is about

→

26 **entities**
Which people and things the note speaks of

→

27 **cards**
Short rated lines per person or thing, each citing its notes

E The graph

graph store with vector search

Note one short memory

Occurrence the same claim said again

Group a topic or an exact name

Concept an activity

Lesson what was learned

Entity and card a person or a thing

Check what the source showed

F Finding

every search

29 **Route**
At home, straight to the graph and the encoder.

→

30 **Three searches, fused**
By meaning, by exact words, by exact names.

→

31 **Floors**
Too far in meaning or too low in trust: out

→

32 **Reranker**
A small model on the graph server re-orders the first page.

G The moments memory comes back

inside the session

Small and quick

33
Session start
Up to 3 open items of the project, and the briefings held.

34
Each message
The message and the last dialogue become a question.

35
A name is mentioned
Its whole page one comes, once per window, each line with the date of its newest note, since a good rating does not mean a fact still holds.

37
Before an action that changes something
A read brings nothing

38
After a failure
The error's lines choose notes close to it.

The deep one

36
Every 5 messages
One line: same subject, or a new one?

39
Briefing
The strong model reads every note of a subject

40
Rebuild
At a compaction the briefings are rebuilt, so the fresh session gets the newest version of the subject it was on.

41
Back in the chat
All of it arrives as evidence in the assistant's context, each note with who said it and when.

→ *then the road starts again at 1*

H Correcting itself

turn ends, and every night

Each turn

42
Was it used?
At turn end, each note the message brought is compared with the assistant's answer.

Each doubt

43
Suspect notes
A briefing's review lists notes that look doubtful, and the assistant can flag one itself.

→

44
Source
The assistant opens the note's own rows in the archive, with the dialogue around them.

→

45
Check and fit
The assistant records what the rows show (it checks; a person approves a repair)

Each night

46
Loop check
At 00:20 each night a record of the day, with no model call

3.1 The raw archive

Every conversation is copied whole into an archive and never rewritten. It is the authoritative record of the conversation, what was said, and every note is checked against it. It is not ground truth about the world: an erroneous statement can be archived and extracted faithfully, so facts about live systems are established separately (for example by reading the system). On 2 November 2026 the archive held 1,413 windows, of which 76 had been processed [memory-coverage-2026-11-02.md]; “processed” there means at least one stored note, not that the window completed every pipeline stage.

> **Advice, document store.** We chose MongoDB. Transcripts are loose JSON records, written once and read many times; a document store holds them as they are.

3.2 Extraction: from a window to notes

The unit is a *window*: the stretch of conversation between two context resets. A small model reads it and writes short notes. Before it does:

- **Only spoken words go in.** The person’s words and the assistant’s words; never tool calls or their output. An early version kept a tool row cut at 200 characters, and it became a false memory [window-ccce3736-gemma-review/merged-rating.md, item R39].
- **Password-shaped words are masked** before the local extraction model sees them. In the first extraction runs, the model’s output held no secrets but its input was unredacted; masking was added as one of six fixes [window-ccce3736-gemma-review/merged-rating.md; fixes-1-6-results.md]. The tested scope is narrow: one reference window, 3 secrets masked in 6 rows and 1 false hit. Masking does not anonymise people, and we did not test that every cloud-bound path, handwritten fact or source read is scrubbed (section 8).
- **Answers are bound.** When the person answers “2” or “yes” to a numbered list, the answer is tied to that list, so the note says what was agreed to. The need appeared in the first trials: a small model inverted “I disagree on 2 and 5” against numbered picks in 6 of 7 runs (an early trial recorded in a note we cannot cite publicly).
- **Chunks.** The window is cut into pieces of about 12,000 characters, never splitting a line (a setting of the early trials; the early-trial note is not citable).

Each note keeps its kind (fact, decision, rule, failure, principle, and others), who said it, when, and a pointer to its exact lines in the archive. That pointer is what makes a note checkable.

> **Advice, extraction model.** We chose a 12-billion-parameter open model (Gemma 4 12B), quantised, one copy on each of two second-hand consumer graphics cards. Extraction is reading and summarising, which a small model does acceptably; judging is left to later stages. The cards rest 45 seconds after every 8 chunks, stop at 85 degrees C and are capped at 88 W (our settings, section 7 and Appendix M). On old hardware, idle time is part of safety, not spare capacity. Section 5.1 gives the comparison with the two alternatives we tried.

3.3 The store

Each note gets a meaning vector from a small encoder, its exact names (projects, addresses, issue numbers), and about three topic groups [six-windows-originals/graph-common-ground.md]. Before a new note is stored, a close older note is looked for. If one is close, a model judges the pair: the same event, the same claim said again, a change, or different. A repeat is not a new note: it is kept as an occurrence under the old one, with its own date and lines. A claim said again strengthens the old note; a retelling does not, and a note the memory itself delivered earlier in the same window, said back by the assistant, adds no weight (a same-window echo is excluded). A repeat keeps its own source, changes the note's weight and not its trust, never rehabilitates a note in doubt or one that was disproved, and cannot override a correction.

> **Advice, graph store.** We chose a graph database with vector search (Neo4j). Notes, lessons, cards, people and sources form a web of links; a graph keeps them and their vectors in one place. The encoder was a small open model (e5-small-v2, quantised) [no-fit-check/FINDINGS.md]; its input bound was a real constraint, see section 5.3.

3.4 Reading again

These reading and enrichment stages preserve the note text and add meaning around it. (A later check can revise a derived note, section 3.6; the archive keeps the original lines and so the provenance. We did not verify a retained version history of each rewritten note.)

- **Link:** which activities an advice note applies to, from a fixed list. This lets a lesson arrive before an action of that kind.
- **Aggregate:** repeated advice becomes a lesson: one statement, its mechanism, and routes of the form "in this situation, do this". Each note stays beneath it.

- **Name:** a one-line subject for each note, with its own vector, so a search can find what a note is about even when the note never says it.
- **Entities:** which people and things a note tells something about. A name shared by two is asked, never guessed.
- **Cards:** for each entity, short lines built from its notes, each citing them. A card also grows during conversation: when the assistant marks a fact the person just stated, a judge checks that the person really said it, that it is not already on the card, and that it does not clash with a line there.

3.5 Delivery

Memory comes back at two moments: before a request (the person's message plus the last few turns, searched by meaning and by exact words) and before an action (the action's activity, matched to linked lessons). A message under 8 words brings no search notes, though a named entity in it still brings its card; we accepted the recall cost of this on purpose and did not evaluate short factual requests adequately. Advice comes before actions that change something, not before reads. Everything shares a token budget per session. A subject reminder runs every 5 messages [subject-replay/FINDINGS.md]. The hook budget is 5 seconds [memory-plan.md, decision of 2026-11-01].

> **Advice, reranker.** We chose a small cross-encoder reranker (ms-marco MiniLM-L6, ONNX) served on a small server, because on 45 answerable probes it matched the larger candidates on rank quality at about 0.4 s median latency (section 5.2). It only reorders; it is not used as an acceptance cut-off.

3.6 Self-correction

A note becomes suspect when a briefing finds it disagreeing with another, out of date, or not matching its lines, or when the assistant flags it. Suspect notes wait in a queue, a few at a time; each is checked against its source and recorded as right, partly right (rewritten), wrong, or outdated. Checks can revise a derived note while the archive keeps the original lines. At the end of a turn, each delivered note is compared with the answer, and a useful match lifts the note a little. Until 7 November a note delivered and never used lost ground; on that day this unused scoring was stopped (the floor setting was turned off and the old misses cleared), because the signal said almost nothing. Non-use no longer penalises a note. This is separate from the lesson counterexample mechanism, under which a lesson whose counterexamples outweigh its support stops being delivered before actions.

The design supports source-based correction; we have shown it only in examples (section 7), not end to end. Three kinds of step should be told apart: automatic detection (a briefing review lists a doubtful note), assistant checking (the assistant

opens the source rows and records a verdict), and human-approved repair (a person rules, as in the case in section 4.5). We have no denominator of wrong notes, no detection rate, no false-correction rate, no repair delay and no recurrence-after-repair measure, so we do not claim that a wrong note is never silently kept.

---

4. Evaluation method

4.1 Four layers of evaluation

1. **Extraction audits** on a reference window and on six windows, with blind sampling by two raters (section 5.1).
2. **Stage trials** on frozen reference sets: retrieval, reranking, naming, linking, aggregation, merging, briefings (section 5.2 and 5.3).
3. **Field tests**: the assistant answers questions with its memory as the intended source, in fresh sessions, rated by three agents (sections 4.2 to 4.4 and 5.4). The sessions were instructed to rely on memory, but they also read live systems, configuration and the web (section 4.2).
4. **A model-per-step trial**: the same notes through each judging step on three model sizes, dry runs that write nothing (section 5.5).

4.2 Field-test design

Pilot and rounds, 1 to 2 November. Two peer agents, the GPT peer and the Hermes peer, each wrote questions with an answer key, written before sending and never changed. The assistant was instructed to answer using only what its memory returned (automatic delivery plus search, at most 3 result pages, every page timed); if nothing returned, it said so. Before answering it checked the question’s premise and whether a note it used was outdated [memory-question-test/BRIEF.md]. The pilot had 3 answers and did not count [memory-question-test/pilot-20261101/BRIEF.md]. Question rules: one fact or decision per question; only things lived before the test began; subjects not discussed that day; mixed subjects [memory-question-test/BRIEF.md]. Round 1 had 10 rated questions, of which 5 failed because they asked about windows the memory had never processed [memory-question-test/ROUND2.md]; from round 2 a question counted only if its fact lay in a listed processed window (27 windows in round 2; 13 older windows, holding 225 notes by computed sum, in round 3) [memory-question-test/PROCESSED-WINDOWS.md; ROUND3.md].

Fresh-session tests, 4 to 7 November. Two new sessions of the assistant in empty folders (same model and settings), loaded only with the global setup and the memory hooks, with memory writes denied and the hooks treating the folders as read-only [field-test-20261104/BRIEF.md]. The same question went word for word into both. Each author wrote a share of the questions, including deliberate cases:

one not in memory, one partly in memory, one with a picture, one corrected later, one from 2025, one needing a later page of a card [field-test-20261104/BRIEF.md; field-test-20261107/BRIEF.md; field-test-20261107b/BRIEF.md]. Questions of any earlier test were not reused [field-test-20261107b/BRIEF.md]. Per turn we measured what the hooks injected (characters, tokens, note ids, cards, pages, briefings, pictures), the hook time, the session's own tool calls and its token use [field-test-20261104/BRIEF.md].

Coverage check. Before the 4 November test the assistant checked each of the 24 questions against the store and returned mismatches to the author. Only one question returned (its answer was in the store) and was replaced [field-test-20261104/coverage.md]. From 5 November the coverage step was dropped [field-test-20261105/BRIEF.md].

Answer keys kept by their authors. After a session could read the keys on 4 November (finding 17, no key was opened), keys stayed in each peer's own folder until rating, question files held the question line only, and reading the project and transcript folders was denied in the test settings [field-test-20261104/findings.md, item 17; field-test-20261105/BRIEF.md]. Blinding is therefore a procedure with a disclosed exposure, not something file times prove: the keys were kept by their authors and the sessions were denied the project folders, but one key was readable on 4 November (none was opened), and in the 1 November rounds the GPT peer's rating of question q14 (a GPT-authored question) discloses prior exposure to another author's key, so independence for that row is contaminated [memory-question-test/rating-gpt-peer.md].

Sessions are not independent. Each question ran in two sessions of the same assistant; 15 questions are 15 cases, not 30. The fresh sessions also share earlier-turn context and a fixed question order, so a later answer may rest on an earlier turn; that carryover is part of the result and not independent question-level retrieval (the ratings mark such answers "carried").

Deviations from the memory-only instruction. The instruction was memory only; observed compliance was not complete. The 7 November (b) ratings record live host reads, configuration reads and web research by the sessions. A fact reached through a delivered note's own source rows measures memory-assisted archive access, not standalone note recall. We report these accesses as deviations, and the Recall rule of section 4.3 scores them accordingly (live files, code and the web count 0).

4.3 Rating scales

Table 1 gives the scales; Appendix A gives the anchors in full. Each rater scored every answer alone without reading the others [memory-question-test/BRIEF.md, "Who does what"].

Field	What it measures	Range	Introduced
Speed	when the first note carrying part of the right answer arrived	0 to 10, anchored	1 Nov (anchored on search page timing); redefined 4 Nov (anchored on delivery by hooks)
Recall	how much of the right answer the delivered notes held	0, 4, 7, 10	1 Nov
Quality	noise and accuracy: wrong, outdated or false-premise notes score 0; more than 25% noise scores 4	0, 4, 7, 10	1 Nov
Overall	average of Speed, Recall, Quality, rounded to the nearest whole number (.5 up)	0 to 10	1 Nov
Same	whether the two sessions gave the same notes and the same answer	0, 5, 10	4 Nov
Honest	for questions not or partly in memory: says what it found and does not make up the rest	0, 4, 10	4 Nov

Table 1. Rating fields [memory-question-test/BRIEF.md; field-test-20261104/BRIEF.md].

Why this scale. The source states: “The scores only mean something if all three read the scale the same way” [memory-question-test/BRIEF.md]. Overall was made a computed average because a free overall judgement “drifted by 2 points in calibration” [memory-question-test/BRIEF.md, “Overall”]. Our reason for the frame: the system is meant to fill a gap (experience the model does not hold, no way to correct itself), so the question tested is whether the self-correction loop works [memory-daily-suite.md, “What is judged”].

Calibration. Before the first question each rater scored the three pilot answers; where two scores on the same question and field differed by more than 1 point, the

rubric line was clarified and the step repeated [memory-question-test/BRIEF.md]. Example, round-1 Overall on three calibration answers: one answer scored 5 by the assistant and 6 by the GPT peer; another 3 then 5 by the assistant against 5 by the GPT peer; a third 6 then 7 against 7 [memory-question-test/calibration-assistant.md; calibration-gpt-peer.md]. We did not summarise the Hermes peer's calibration file.

Rules agreed after rating, 4 November. Quality is 4 whenever more than 25% of the delivered notes are noise. The Recall rule: facts reached through a delivered note's own source rows count for Recall (and Speed records the path, 4); facts found by free search of transcripts, code or live files count 0 [field-test-20261104/results.md, "Agreed in the discussion"]. The 5 November results record that the assistant's own Recall of 10 on one question broke this rule (the figure came from a code commit) and was conceded [field-test-20261105/results.md].

Protocol deviation: the assistant's 7 November (b) ratings. The assistant's rating file for 7 November (b) does not follow the frozen rubric: it uses Speed 5, 7 and 8, Recall 2 and 8, Quality 6, 8 and 9 and Same 4 and 8, none of which is an anchor value, and its own introduction says noise was not counted note by note [field-test-20261107b/rating-assistant.md]. We label those ratings non-conforming and exclude them from every cross-rater comparison; their means are shown in Table 5 only as "original, uncorrected ratings". Other conceded or doubtful ratings and broken keys (section 5.4) are likewise reported as original, uncorrected ratings; we did not re-rate.

4.4 Raters and independence

The raters in the field tests were the assistant under test, the GPT peer (OpenAI GPT/Codex family) and the Hermes peer. The Hermes peer is an agent platform; the sources do not identify its underlying model or version, so we do not claim a model family for it. The Grok peer (xAI Grok family) is not a rater in any field test. The assistant rates its own memory; this is a conflict of interest and section 8 reports its effect. The GPT peer is a co-author who helped design and rate the system. The human author supervises and rules on disputed points.

Rater coverage in 7 November (b). The GPT peer rated 30 of 42 rows (15 of 21 questions), not a random sample: its own 7 questions, 1 of the Hermes peer's 7 questions and all 7 of the assistant's 7 questions. The six other Hermes-authored questions are unrated by her, and the rating file does not state why. The assistant and the Hermes peer rated all 42 rows. Table 5a gives coverage by question author; rater means over different rows are not comparable, so we do not infer convergence from the two means of 6.9. On the 30 rows the GPT peer rated, the matched means (computed) are: Quality 6.9 (GPT peer), 7.8 (Hermes peer) and 6.7 (assistant, non-conforming ratings, shown for completeness only); Recall 10.0, 9.3 and 9.2; Overall 8.8, 8.6 and 8.0.

Question author	Questions	Assistant rows	GPT peer rows	Hermes peer rows
GPT peer	7	14	14	14
Hermes peer	7	14	2	14
Assistant	7	14	14	14

Table 5a. Rows rated per question author, 7 November (b), counted from the three rating files.

4.5 Peer arbiters on other model families

Three peer agents acted as reviewers and arbiters. The GPT peer (OpenAI GPT/Codex family) and the Grok peer (xAI Grok family) are other model families than the assistant's; the Hermes peer is an agent platform whose underlying model the sources do not identify. Table 2 lists where each acted.

Peer (model family, where known)	Period	Where it acted	What it changed, with numbers
GPT peer (OpenAI GPT/Codex)	2026-10-25 to 11-07	Design counterpart for graph, rating, retrieval, activity concepts, gateway, briefing, aggregation, entity layer; blind rater of extraction; code reviewer; field-test question author and rater	Five common-ground documents in the first two days [six-windows-originals/*-common-ground.md]. Caught what the assistant missed in the first extraction review: a size written as a percentage, a wrong actor, a snapshot location, 8 unsupported citations, the 200-character tool-row cut [window-ccce3736-gemma-review/merged-rating.md]. Reproduced five “false completeness” cases in the briefing pilot; fixed; her test run passed 110 tests (98 before) [briefing-pilot/gpt-peer-independent-review.md]. Blind label audit of 184 reranker pairs: 81.5% binary agreement [reranker-trial/audit-compare.json]. Sorted 40 no-fit notes blind: 36 of 40 agreed; her changes were all taken [no-fit-check/FINDINGS.md].

Peer (model family, where known)	Period	Where it acted	What it changed, with numbers
Grok peer (xAI Grok)	2026-10-28 to 10-30	Reviewer of lesson aggregation and lesson merging only	Rated the aggregation result 8/10 and blocked the real run on two points, both fixed; re-test 16/16 checks [aggregation-cases/RESULT.md]. Blind merge ratings 6, 7, 8 and 8 across four stages [lesson-merge-check*/rating-grok-peer.md]. Not an extraction, retrieval or field-test rater. Paused until 4 November by a usage limit [lesson-merge-step14/OUTCOME.md].
Hermes peer (agent platform; underlying model not identified in the sources)	from 2026-11-01	Tester and rater in all field tests; question author; entity-linker example author; delivery literature for the noise study; hardware fault analysis	Wrote 8, 5 and 7 questions per series. Agreed with the GPT peer on 42 of 48 linker examples [entity-layer/linker-examples/differences-round1.md]. Her literature findings (section 1) fed the noise-reduction ruling [noise-reduction-20261107/merged.md]. Her hardware analysis ranked card power management first among causes of hangs (hardware investigation notes, not publishable; Appendix M).

Table 2. Peer arbiters. Sources as cited per row.

Disagreements and how they resolved (recorded cases):

- *Different reference sets.* The assistant's reference for one window held 28 items with partial credit; the GPT peer's held 39 items, citation-strict. The same fix chain looked like an improvement on the first (15 of 28, 54%) and a regression on the second (20 of 39, 51.3% against 23 of 39, 59.0% before) [window-ccce3736-gemma-review/fixes-1-6-results.md; gpt-peer-run10-rating.md]. The report states these are not directly comparable; we report them separately and never pool them.
- *Merge timing.* At the full lesson merge, the GPT peer rated 6/10 and wanted fixes first; the Grok peer rated 8/10 and wanted delivery after the next step. The next step (merging already stored lessons) was done before delivery [memory-plan.md; lesson-merge-step14/OUTCOME.md].
- *A shared flag overturned by the source.* All three raters flagged one merged lesson at step 14. Reading the source rows showed the merge was right and the extracted note was wrong (it reversed the person's words). The GPT peer and the Grok peer withdrew the finding; the wrong note was marked with trust 0.375, below the floor of 0.5 [lesson-merge-step14/OUTCOME.md]. This is the clearest recorded case of independent findings corrected by the source.
- *Withdrawals by the assistant.* It withdrew a claim that a test count was wrong [window-ccce3736-gemma-review/merged-rating.md] and conceded a Recall score that broke the agreed rule [field-test-20261105/results.md].

Why peers other than the assistant (our rationale, not a recorded decision). No source states that other agents were chosen in order to achieve something. What the sources do record: ratings were written alone, without reading the others', and file modification times were cited as a check [window-ccce3736-gemma-review/merged-rating.md], which we treat as a procedure and not as proof; the human author ruled that the examples for the entity linker be chosen by the two peers because "a synth can judge better what a synth can or cannot choose" [entity-layer-plan.md]; and for the memory's own judging steps we follow a same-model rule, that a summary written by the same model is read like one's own writing (a working rule recorded in a note we cannot cite publicly). From these, we infer the rationale: independent judgement, and ratings that a model cannot give to its own output. We mark this as inference. The self-rating gap of section 5.4 is observed; that self-bias causes it is plausible, and we did not isolate it experimentally.

4.6 Protocol: proposals, turns, common ground

For design decisions the working rule, set by the human author on 2026-10-25, was: debate in turns, a maximum of 3 turns each, then a common-ground document (a working rule recorded in a note we cannot cite publicly). Observed pattern: each side writes a proposal independently, then one exchange each way, then a common-ground document that lists what both agreed and what was taken from each side [six-windows-originals/]. Recorded turn counts were lower than the cap: one turn

each for the graph design, two turns each for the reranker and the check-trigger plan [reranker-plan.md; check-trigger-plan.md]. Diagnosis was also done independently: on 2026-10-28 each side named the missing piece of the design in a separate file before reading the other's, and the merged plan was agreed by the human author and both [aggregation-plan.md]. The assistant's pick was that lessons must return at the action and that repeated lessons need an abstraction layer; the peer's pick was that a newly captured lesson has no applicability link, plus capture timing and delivery before the action [missing-piece-assistant.md; missing-piece-gpt-peer.md]. Each step of the 15-step plan then took its own go from the human author [memory-plan.md].

4.7 Frozen probe sets and dry runs

- 61 probes (45 answerable, 16 negatives) and 909 unique query-note pairs, frozen on 2026-11-01 [reranker-trial/arms.json (computed)].
- 98 card lines of 2026-11-04; 47 answer notes of two tests (counts from our working files; no citable report).
- A frozen baseline of delivered items, 30 turns across 4 channels [noise-reduction-20261107/baseline-20261107.txt].
- Five frozen aggregation cases first (36 distinct notes, cases 1 to 8) [aggregation-cases/RESULT.md].
- Held-out stretches marked before the cadence replay [subject-replay/FINDINGS.md].
- Dry runs: the model trial and the merge replay write nothing to the store; each difference was read by hand [model-trial-20261108/RESULTS.md].

---

5. Results

5.1 Extraction model choice

We tried three models for extraction (Table 3; details in Appendices B and C).

Model	Time per window	Output against input	Quality notes
Memory-specialised 30B open model, its maker's prompt	about 7.5 h	2.6 times the input	stopped; did not fit the video memory, experts ran from main memory at about 18 to 20 tokens/s

Model	Time per window	Output against input	Quality notes
9B open model	163.7 s (one window)	1.78% of the window	recall about 18 of 28 (partial-credit rule unclear in source); persistent inversion, password and count failures
Gemma 4 12B, run 8	287 s (same window)	2.20%	fully correct 28 of 41 (strict, merged); recall 12 of 28 (42.9%, assistant) against 23 of 39 (59.0%, GPT peer)

Table 3. Extraction candidates on the reference window [Local transcript memory extraction.md; window-ccce3736-gemma-review/README.md and merged-rating.md]. The 9B figures come from an early-trial note we cannot cite publicly.

The first model wrote answers longer than its input (about 404,000 characters for 61 of 152 writes against a window of 355,588); an extraction must be shorter than its source, and the manual reference was 2.45% of the window (28 concepts, 8,700 characters) [Local transcript memory extraction.md]. Gemma run 8 was judged “not good enough for unattended ingestion” by both raters; one critical decision inversion and two wrong actors were found [window-ccce3736-gemma-review/merged-rating.md]. Six fixes (decision binding, a source-support verifier, a merge pass, must-keep classes, a head-and-tail filter that did not help, secret redaction) gave run 10: 2.05%, 763 s, the decision ruling correct, no leaks, recall 15 of 28 (54%) [fixes-1-6-results.md]; the merge pass was judged “not worth keeping as is”.

On six windows Gemma took 2,699 s in total (45 minutes; computed from the per-window wall-clock decimals) and stored 345 notes with 0 failures (computed from the per-window result files) [six-windows-originals/]. Two raters sampled the notes independently (30 distinct notes): 22 faithful, 5 minor distortion, 1 citation-only, 2 hallucination, that is 2 of 30 (6.7%) prose hallucinations, a sample rate, not a corpus rate [gpt-peer-hallucination-reconciliation.md]. After the rule that the extractor is judged only on what its prompt asks and only against its cited lines, recall on frozen facts was 6.5 of 9 applicable facts (72%) [facts-rescored-isolated.md]. In the plan, the model was accepted as an index, with “about 7 minutes a window, notes 1-3% of the source, recall 72% on frozen facts, 30 of 30 sampled notes pointing at the right lines” (the implementation plan; not a public document).

What the sources do not give. No head-to-head table of the 9B model and Gemma on one rubric exists; the adoption of Gemma is recorded as the human

author’s acceptance, not as a measured win. The 9B model was faster and shorter but failed on inversions and passwords; Gemma’s weakness was the same kind and was addressed by the six fixes. A repeat test found that 86 of 234 chunks (36.75%, 37% rounded) reached the 7-note output cap and lower temperature did not help [extraction-repeat/FINDINGS.md]. Reaching the cap is not a measured loss: the number of omitted facts or lost writes was not measured.

5.2 Retrieval and reranking

No cut-off separates right from wrong. On a snapshot of 2,137 notes, right notes scored 0.745 to 0.902 by cosine and the best score of a negative probe 0.788 to 0.868. At a floor of 0.83, 32 of 45 right notes cleared it and 4 of 16 negatives were quiet; at 0.85, 22 of 45 and 9 of 16; at 0.87, 14 of 45 and 16 of 16 [no-fit-check/FINDINGS.md]. Rank worked: 21 of 40 subjects found their right note first and 26 within three [same]. Quiet on negatives fell from 15 of 16 (373 notes) to 9 of 16 (2,137 notes) between the two snapshots; the store grew, but the runs differ in other ways too, so growth is not isolated as the cause [same]. A later labelled set of 909 pairs showed the same overlap on the reranker’s raw score across nearly the whole range [noise-reduction-20261107/search report (GPT peer)]. We therefore set no absolute reranker-score acceptance threshold and no additional model gate in the hook; the cosine and trust floors remain (Appendix F) [noise-reduction-20261107/merged.md].

Reranker. Four rerankers reordered the first 8 candidates on 45 answerable probes (Table 4). The chosen one puts the right note first 21 times instead of 18, within three 29 times instead of 23, all 33 still within the first 8 [rerank-wiring-plan.md; computed from reranker-trial/scores-*.json].

Order	within 1	within 3	within 8	p50 latency per probe
Baseline	18	23	33	not applicable
Small F16 encoder reranker, llama.cpp	19	27	33	0.43 s
Large multilingual reranker, Q8	21	30	33	4.80 s
Small ONNX reranker (Jina turbo)	20	26	33	0.55 s
ms-marco MiniLM-L6, ONNX (chosen)	21	29	33	0.42 s

Table 4. Reranker comparison, computed [reranker-trial/arms.json; scores-.json].*

Widening the candidate list from 8 to 12 lost 3 to 4 targets from the first 8 (computed) [same]. A fifth candidate (a 0.6B generative reranker) has no scores file;

its result is unclear in source. Replayed through the real search path, the wired setup gave 22, 29 and 32 within 1, 3 and 8, search p50 0.269 s, p95 0.519 s, 0 failures; the baseline gave 18, 23 and 32 with p50 0.051 s [reranker-trial/replay-baseline.json; replay-wired.json]. The label audit by the GPT peer agreed with the assistant's labels on 81.5% of 184 pairs (binary), 82.1% under one agreed rule and 74.5% when pairs were labelled one per call [reranker-trial/audit-compare*.json]. The person chose the reranker on the 45 known targets and left the acceptance cut-off for real traffic [reranker-plan.md].

Search drift and slugs. Between the snapshots of 1 and 7 November (the store grew to 8,883 notes and the configuration changed), plain search on the same 45 questions fell from 33 to 25 right notes in the first 8. The regression is real; the runs are dated and the configuration evolved, so growth is not isolated as its cause. Adding words from the subject names back into search (an extra list of four) raised right notes on the first page from 25 to 31 on demand, and from 21 to 22 on delivery by hook; so it is on for search the assistant asks for and off in the hooks (1 more right note for 45 more notes) [naming-pilot/slug-extras-20261107.json; memory-closing-roadmap.md].

5.3 Reading stages

Appendix I to K give the tables.

- **Linking.** On 80 notes (42 linked, 38 not), the assistant marked 66 right; the GPT peer, blind and frozen before reading, marked 72 of 80 right; the union of marks gives 4 false links of 42 and 6 missed links of 38 [link-sample/marks-assistant.md; marks-gpt-peer.md]. A lesson-return test of four cases with the real tools found that a captured lesson, and a lesson whose deciding clause lay after the encoder's 1,000-character input, did not arrive; after pending-link and long-note pieces, 6 of 6 checks passed [lesson-return-test/RESULT.md].
- **Entities.** Three window tests with three readers: 48 of 51, 50 of 59 and 60 of 75 items agreed with the linker; in test 2 all three removed the same two links [entity-layer-plan.md]. Wrong and missed totals for tests 2 and 3 are unclear in source.
- **Aggregation.** The judge scored 13 of 18 checks on the first wording and 18 of 18 after fixing an instruction defect; end-to-end store and hook checks 13 of 13, then 16 of 16 after the peer's two fixes [aggregation-cases/RESULT.md].
- **Lesson merge.** Staged runs of about 600 notes into 505 lessons, 1,315 into 895, then 895 into 785 active lessons at step 14 (75 merges, 110 absorbed, 1,329 episodes kept); single-note lessons whose nearest lesson scored 0.95 or more fell from 27% to 12.5% [lesson-merge-check*/; lesson-merge-step14/OUTCOME.md]. The three raters' overall scores per stage are in Appendix J. At the end the note-level merge judge, read by hand on 80 of 188 pairs at 0.95 or more, gave 43 same claim, 1 same event, 23 changed, 13

different; with live runs, 102 verdicts, 46 merges, 0 wrong [memory-closing-roadmap.md].

- **Naming.** Slugs written by the medium and the large model both put the right note within 12 for 31 of 45 probes; first place 15 against 13 (a count on 45 probes; it does not establish retrieval equivalence for a smaller model); the medium model was adopted [naming-pilot/additive-*-20261107.json; memory-closing-roadmap.md]. In the 2026-11-01 ablation, replacing the search channels by subject lost 2 to 5 probes from the shortlist and gained none, so slugs were added as an extra list [naming-pilot/ablation.json].
- **Cadence of subject reminders.** Replay on 11,215 messages: a reminder every 3 messages caught 23 of 26 lasting shifts with 33 false requests; every 5, 21 of 26 with 7 false (56 reminders); every 8, 15 of 26 with 4 false. The person chose every 5 (held-out stretches: 7 of 16 and 10 of 11) [subject-replay/FINDINGS.md].
- **Briefings.** A pilot with five briefings on a 373-note store: per-message search brought a right note for 1 of 5 prompts, the briefings cited all 11 expected notes [briefing-pilot/README.md]. A cap test showed that capping at 3,000 or 1,500 tokens lost contested facts (contested kept 9 of 10 uncapped, 6 of 10 at 3,000, 3 of 10 at 1,500 on one subject) and that the frozen 100% target was not met [briefing-cap-test/]. An import test marked 22 of 40 imported notes as adding, 14 off topic, 4 repeats [briefing-import-test/reply-gpt-peer-comparison.md].

5.4 Field tests and growth over time

Table 5 gives the averages per rater. Rows are the fresh-session tests; 1 November (rounds 1 to 3) is in Appendix D. Every mean is computed from the raw rating rows of both sessions pooled (4 Nov 48 rows per rater, 5 Nov 18, 7 Nov (a) 30, 7 Nov (b) 42 or 30), and rounded half up to one decimal; the source summaries, which averaged rounded session values, are not used. All ratings are original, uncorrected ratings.

Test	Questions	Rater (rows)	Speed	Recall	Quality	Overall	Same
4 Nov	24	Assistant (48)	9.2	9.8	6.9	8.7	8.5
		GPT peer (48)	9.0	9.1	4.6	7.6	6.0
		Hermes peer (48)	9.3	9.6	4.5	7.9	7.7
5 Nov	9	Assistant (18)	7.3	8.6	7.5	7.8	7.8

Test	Questions	Rater (rows)	Speed	Recall	Quality	Overall	Same
		GPT peer (18)	6.0	7.1	5.3	6.1	10.0
		Hermes peer (18)	7.0	7.2	6.3	6.8	7.8
7 Nov (a)	15	Assistant (30)	8.1	10.0	6.8	8.3	8.3
		GPT peer (30)	8.3	10.0	4.4	7.6	5.3
		Hermes peer (30)	9.1	10.0	4.4	7.9	5.3
7 Nov (b)	21	Assistant (42), non-conforming ratings, excluded from cross-rater comparison	8.0	9.5	6.9	8.1	8.7 (non-anchor values)
		GPT peer (30 of 42 rows)	9.6	10.0	6.9	8.8	9.0
		Hermes peer (42)	8.4	9.5	7.6	8.5	7.6

Table 5. Field-test averages computed from raw rows [field-test-20261104/rating-.md; field-test-20261105/rating-.md; field-test-20261107/rating-.md; field-test-20261107b/rating-.md]. The "Same" values for 7 Nov (b) come from the raw rows; they are shown with the caveat that the Same anchors are applied inconsistently (the assistant uses 4 and 8, which are not anchors), so we do not compare them across raters. The GPT peer's own file footer for 7 Nov (b) (9.3, 10, 7.0, 9.0, 8.6) disagrees with its own table; the table values above are used.

Recall in the 7 November tests, stated as rows and not as a score a row can take (Recall takes only 0, 4, 7 and 10): in test (a) all three raters gave Recall 10 on all 30 rows. In test (b), the rater means were 9.5 for the assistant (397 over 42 rows; its rows include non-anchor values 2 and 8), 10.0 for the GPT peer (300 over 30 rows) and 9.5 for the Hermes peer (399 over 42 rows: 37 rows of 10, three of 7 and two of 4, covering three questions).

Headline results. On 4 November both sessions answered 22 of 24 right and in the same way [field-test-20261104/results.md]. The two partly-in-memory questions scored Honest 10 from all three raters on both sessions [same]. On 7 November (a) Recall was 10 on all 30 rows from all three raters, and no answer was wrong [field-test-20261107/results.md]. Where the answer note came: 19 of 24 questions with the question or already in context in both sessions on 4 November; found only by the session's own reads: 4 and 2; never reached: 1 and 3 [field-test-20261104/results.md]. Misses were placed by stage: not captured at extraction (1 question), never reached the session (3 rows), ranked too low for the hook (3 rows), a picture dropped by the hooks (1), a false premise accepted by both sessions (1 question) [same].

Self-rating bias. The assistant rated its own memory. Table 6 compares its Quality score with the two peers'.

Series	Assistant Quality	GPT peer	Hermes peer	Assistant higher than both?
1 Nov, round 1 (computed)	3.2	3.6	3.6	no
1 Nov, round 2 (computed)	6.6	6.6	6.6	equal
1 Nov, round 3 (computed)	5.2	4.6	5.2	higher than GPT peer, equal to Hermes peer
4 Nov	6.9	4.6	4.5	yes
5 Nov	7.5	5.3	6.3	yes
7 Nov (a)	6.8	4.4	4.4	yes
7 Nov (b)	6.9 (non-conforming ratings; descriptive only)	6.9	7.6	not compared: non-conforming ratings and differing coverage

Table 6. Quality scores by rater [memory-question-test/rating-.md (computed); field-test-/rating-.md (computed)].*

The assistant's Quality score exceeded both peers' in the three conforming fresh-session series of 4, 5 and 7 November (a); the fourth, 7 November (b), was not compared, because the assistant's ratings there are non-conforming and the raters covered different rows. In the 1 November rounds it was not above both peers:

equal or lower in rounds 1 and 2, and in round 3 above the GPT peer and equal to the Hermes peer. The assistant itself stated that its Quality and Same scores on 7 November (a) were too high because it judged noise by eye, and that the peers' 4.4 and 5.3 stand [field-test-20261107/results.md]. The gap is concentrated in Quality; Recall agreed (10 from all three on 30 rows). We report the gap and do not adjust for it. That self-bias causes the gap is plausible, not isolated experimentally. Overall scores follow the same pattern: the assistant's was highest of the three on 4, 5 and 7 November (a); 7 November (b) is not compared.

Noise and what was built. We report the two 7 November tests as separate descriptive results, each under its own noise rule, and we draw no before-and-after improvement from comparing them.

Test (a), 15 questions, 30 rows per rater. Noise counted every note the hooks injected plus the notes the sessions' own searches returned, and card lines were counted in the base. Quality was 4 on 28 of 30 rows for each of the GPT peer and the Hermes peer, and 10 on 2 of 30 rows (one question, in both sessions) for each (computed from the raw rows) [field-test-20261107/rating-*.md]. Counting the 30 turn files, 99 of 137 items pushed by hooks and 193 of 226 items pulled by the sessions' own searches did not name an answer note; these are upper bounds on noise because a line supporting the answer through another note counts as a miss [noise-reduction-20261107/search-assistant.md]. By channel: card lines 38 of 52, notes with the question 32 of 53, lessons before an action 29 of 32, sessions' own search pages 193 of 226 (Table 7, Appendix E). The rulings were: a card is context, not noise; lessons belong before the first edit, not before a read [noise-reduction-20261107/merged.md].

Test (b), 21 questions, after the build. The Hermes peer's file excludes card lines and briefing notices from the noise base, following that ruling; the GPT peer's file does not state the base. The Hermes peer rated 23 of 42 rows Quality 10 and 15 rows Quality 4; the GPT peer rated 1 of 30 rows Quality 10 and 2 Quality 4 (computed) [field-test-20261107b/rating-*.md]. Delivery per row, as counted in the results file, was lessons 0.2 (against 1.1 in test (a)) and card lines 3.6 (against 1.7) [field-test-20261107b/results.md].

The two tests differ in questions (15 against 21, not repeated), in the noise rule (card lines in the base in (a), out in (b) for the Hermes peer) and in rater coverage (section 4.4), so their Quality figures are not like-for-like and we make no claim that noise fell. Recall in (b) was 9.5 and 10.0 and 9.5 as stated above; the sub-10 Recall rows are the Hermes peer's two rows of 4 and three of 7 on three questions. Whether the lower Recall means in (b) than in (a) have anything to do with notes removed by the noise work is not demonstrated, because the questions are not paired.

Growth over time. Table 8 lists each test, its key numbers and the change made after it.

Date	Test	Key numbers	Change made after
2026-10-28	Daily suite design agreed (not run in these reports)	none	Design of automatic daily checks, a regression gate and a weekly independent audit by a peer [memory-daily-suite.md]
2026-11-01	Pilot (3 answers, not counted)	Assistant’s own Overall 6, 4, 7	Common scale built; calibration rounds; answering rules: check the premise first, never use an outdated note as true [memory-question-test/pilot-20261101/rating-assistant.md; BRIEF.md]
2026-11-01	Round 1 (10 rated questions)	mean Overall assistant 4.9, GPT peer 4.5, Hermes peer 4.6 (computed); 5 questions failed on unprocessed windows	Hooks deliver again; search page 1 keeps only notes the reranker ranks close to its best; 10 more windows processed; questions limited to processed windows [memory-question-test/ROUND2.md]
2026-11-01	Round 2 (10 rated)	mean Overall 5.2, 5.7, 5.3 (computed)	Search also scores notes found only by their words; similarity floor 0.78; windows of August 2025 extracted end to end [memory-question-test/ROUND3.md]
2026-11-02	Round 3 (10 rated, old windows)	mean Overall 5.6, 5.4, 5.4; Recall 6.8, 7.0, 6.7 (computed)	Ten more old windows queued; coverage audit: 76 of 1,413 windows processed [memory-coverage-2026-11-02.md]

Date	Test	Key numbers	Change made after
2026-11-04	Fresh sessions, 24 questions	22 of 24 right; Overall 7.6 to 8.7; Quality 4.5 to 6.9 (rater means); hook median 1.6 and 1.7 s	Picture path shown in delivered note line; keys kept by authors; card lines shown only when they fit the message; notes the answers never use sink; wrong or stale notes and card lines rewritten [field-test-20261104/findings.md; noise-reduction-20261107/BRIEF.md]
2026-11-05	Fresh sessions, 9 questions	Recall 7.1 to 8.6; Overall 6.1 to 7.8 (rater means); hook median 1.2 and 1.4 s	151 notes with no record bound to their chunk’s rows were bound (the log has 152 lines; the difference is unclear in source); a wrong note rewritten; a picture record corrected [field-test-20261105/results.md]
2026-11-07	Fresh sessions, 15 questions	Recall 10 on 30 rows; Quality 4.4 (peers) and 6.8 (assistant); Overall 8.3, 7.6, 7.9; hook 0.46 to 1.75 s	Noise search; then: page one of a card holds up to 4 “who it is” lines and up to 6 best-rated facts; lessons only before an action that changes something; the card writer reads 100 notes (was 60) and a card holds 25 lines (was 24); a command adds a fact to a card; the footer asks to open other pages before saying “not known”; 212 cards re-rated without a model [noise-reduction-20261107/merged.md]

Date	Test	Key numbers	Change made after
2026-11-07	Replay after build	lessons silent on 115 of 116 read calls; all 6 synthetic changes keep them; of 47 answer notes, 29 stayed on page one, 8 rose to it, 9 stay on later pages, 1 lost its citation	Known weak spot: a rated line can read badly alone [same]
2026-11-07	Fresh sessions, 21 questions	Overall 8.1, 8.8, 8.5; Quality 6.9, 6.9, 7.6 (assistant's ratings non-conforming); Hermes peer: 23 of 42 rows Quality 10, under a noise rule different from test (a)	Plain "delete" now brings a lesson; a read-type command classed as read; stale card and notes rewritten; each card line shows an "as of" date; read-only footer; a proposal lock made unique [field-test-20261107b/results.md]
2026-11-07	3 follow-up questions on later card pages	3 of 3 right	Search through note names proven on a real search [field-test-20261107c/results.md]

Table 8. Chronological growth.

Caveats on the series. Question sets differ and are not repeated; two result files say their comparison is "not clean" [field-test-20261105/results.md; field-test-20261107/results.md]. The Recall rule changed on 4 November and Speed was anchored differently before and after, so Speed averages are not comparable across the two groups. Three questions were later found broken or doubtful: a picture question whose key was wrong (5 November), a false-premise question that passed the premise check (4 November), and a key that is likely wrong (7 November (b)) [field-test-20261105/results.md; field-test-20261104/findings.md; field-test-20261107b/results.md]. Roadmap features were checked on 7 November: corrected note worked, lesson before an action worked, merge worked (one note with a repeat count), picture worked, cards worked; search through note names was not provable from search output; fading was not exercised, and on that date the unused scoring was switched off (section 3.6) [field-test-20261107/results.md].

5.5 Model per step

The same 130 notes of four windows (46 advice notes, six cards) went through every judging step on three sizes of one commercial model family (large, medium, small), as dry runs; the reference is what the store held, made by the large model [model-trial-20261108/RESULTS.md]. The large model is therefore the reference, so agreement with it is partly circular.

Step	Large	Medium	Small	Retained in this trial (provisional)
Name (130 notes)	65 characters on average, all named	as good, 69 characters	shorter (52), sometimes keyword lists, never wrong	Small
Link (46 advice notes)	46 links (reference)	25 found, 3 extra, often none where a link was clear	31 found, 12 extra, a few wrong	Large
Aggregate (46 notes)	32 placed (reference)	good lessons tied to the wrong notes, shifted by one	right lessons, sometimes more precise	Small
Entities (130 notes)	0 links	same	same	Small
Cards (6)	reference	as good as large	thinner (18 lines against 25)	Medium
Merge judge (48 pairs)	40 of 48 as its own first run	43 as large today, 38 as the first run	38 as large; “changed” verdicts 4 against 10	Medium
Merge lessons (8 groups)	9 merges	the same 9	the same 9	Small
Cards in dialogue (11 facts, 2 never said)	11 right	10 right	9 right	Medium
Briefing (291 notes)	ready, 0 lost, 32 cited	partial, 65 lost, 22 cited	partial, 21 lost, 38 cited	Large

Table 9. Model trial [model-trial-20261108/RESULTS.md].

Two of nine steps (linking and briefing) retained the large model in this trial. Smaller is not uniformly worse: on aggregation the medium model tied lessons to the wrong notes, shifted by one, while the small one placed them right; on linking the small model found more than the medium one. The trial also recorded wrong outputs, not only missing ones: the small model made a few wrong links (and 12 extra links), the medium model made 3 extra links and attached good lessons to the wrong notes, a provenance error. The smaller models' other losses were links not made, facts left as plain notes, or notes kept apart instead of merged. The source's softer statement that no wrong prose was written does not mean no wrong memory output. The entity step links few notes (284 of 6,112 window notes) because of its strict rules, not the model [same]; in the sample all three sizes gave 0 links, so the trial cannot establish that any size is adequate at entity linking.

Limits of the trial: the large model is the reference; one briefing subject; six cards; zero positive entity links; no repeated runs, so no run-to-run uncertainty except the large model's own 8 differing verdicts of 48 on the merge judge; no independent adjudicated gold standard; no held-out downstream check; naming length and hand judgement do not establish retrieval equivalence for the small model. The choices in Table 9 are therefore provisional. Roles (strong, middle, light) are mapped to models in one place in the configuration. The extraction model is not part of this trial. Bug found on the way: the entity step skipped notes already linked at the current version, so its first run judged nothing; fixed.

---

6. Costs: measured and estimated

Rows marked "estimate" are forecasts or settings, not observed costs. No money total and no energy total was measured.

Cost	Value	Source
Time per window, whole pipeline, one clean 10-hour window	472.5 s (extract 241.7, name 170.0, aggregate 38.7, link 16.1, store 4.7); 42 notes	[memory-switch-tests.md]
Recorded run and recovery durations per window, 112 windows in batches 3 to 14	batch means 241 to 489 s; largest single window 1,089 s (computed). Some batches include entities-only, name-only or store-only recovery runs, so these are not all six-stage whole-pipeline times	[memory-switch-tests/batch-*]

Cost	Value	Source
Stage means across batches	extract 90 to 232 s, name 52 to 163 s, aggregate 25 to 73 s, link 11 to 19 s, store 3 to 8 s (recorded durations, same caveat)	[same (computed)]
Backlog (estimate)	about 1,300 windows, about 175 h of extraction, about 44 evenings of 4 h: a forecast, not observed cost	[memory-closing-roadmap.md]
Hook time per message in tests	median 1.2 to 1.7 s, max 2.6 to 3.2 s (4, 5 Nov); 0.46 to 1.75 s (7 Nov a)	[field-test-*/results.md]
Injected by hooks over 24 turns	71,425 characters (about 17,900 tokens) and 54,533 (about 13,600) in the two sessions	[field-test-20261104/results.md]
Load at session start	2,240 characters (about 560 tokens) in one session	[field-test-20261104/start-test1.md]
Context at end of the 24 turns	283,087 and 203,283 tokens; mostly from the sessions' own source reads of 12,000 to 32,000 characters per call	[field-test-20261104/results.md; findings.md]
Search time	baseline p50 0.051 s, p95 0.062 s; with reranker p50 0.269 s, p95 0.519 s	[reranker-trial/replay-*.json]
Delivered items that missed an answer note	99 of 137 pushed and 193 of 226 pulled (upper bounds), 7 Nov (a)	[noise-reduction-20261107/search-assistant.md]
Briefing generation	13 to 187 s per briefing in the pilot; an uncapped memory briefing of 94 cited notes: 6,073 estimated tokens, 127.7 s; the stack of five briefings about 7,300 to 7,700 estimated tokens against a 6,000 target	[briefing-pilot/README.md; briefing-cap-test/builds.json]

Cost	Value	Source
Chunks reaching the 7-note output cap	86 of 234 (36.75%); omitted facts or lost writes not measured	[extraction-repeat/FINDINGS.md]
Failures in the batch runs	4 failed attempts visible in the logs for batches 6, 8, 13 and 14 (cause unclear in source), whose final result files show successful retries; we count both attempts and final states and did not re-trace each window. Also 1 from a wrong path in a service, and 4 of 10 windows in one batch hit the heat stop	[memory-switch-tests/; batch logs]
Hardware	two second-hand consumer graphics cards, one home GPU server, run only while a person is present	our set-up
Power cap (setting, estimate)	88 W per card is a cap, not measured electricity	section 3.2
Hardware faults	one card entered a slow state at 0.69 to 0.74 tokens/s against 13.9 healthy (about 20 times slower); three hang events on one card in one morning on 7 November	hardware investigation notes (not published; Appendix M)
Human time	reviewing rulings, notes and corrections: not measured	
Model-provider usage and cost	not measured; the plan is a subscription	
Money and energy totals	not measured	

Table 10. Costs.

Time and counts are reported; the gain is described in section 7 and not given a number. The reader weighs one against the other. Most costs are ordinary: electricity, a subscription, a server at home, and attention.

---

7. What changed in daily work

We do not put one number on the gain. There is no no-memory, manual-file or raw-history baseline on the same questions, no randomised question order and no controlled end-to-end comparison, so what follows is our experience and observed examples, not a measured causal result. Its value depends on who uses it and what they would otherwise spend, and the reader should weigh it against section 6. The stage trials in section 5 are evidence for their stages, not proof that the whole system improves work or prevents repeated mistakes. Examples are rewritten to protect the people in them.

- **Facts less often asked twice (observed examples).** A colleague's role, a machine's address, a decision taken weeks ago came back when the subject came up. In the 7 November (b) test, the first mention of a named entity brought its card on page one and did not repeat it at later mentions [field-test-20261107b/rating-hermes-peer.md].
- **A lesson arrived before an action (observed example).** A lesson learned once ("check the live configuration before calling a fix done") arrived before the next action of that kind. In one test a lesson came on the right command before a read in both sessions [field-test-20261107/results.md]; in another the expected lesson did not arrive before an `rm` (the concept list lacked plain `rm`), which was fixed after the test [field-test-20261107b/results.md]. We have not measured whether mistakes stop being repeated.
- **People recognised (observed examples).** When a friend's name came up in the tests, the card for that name arrived with what was said about them, and when.
- **Wrong notes caught (observed examples).** A note said a service restarted "on failure"; another said "always". The memory flagged the clash; a check of the live system settled it, and the wrong note was marked. In the field tests, both sessions caught a wrong note that gave the person's guess as a fixed fact, and it was rewritten [field-test-20261105/results.md]; sessions also corrected notes that credited one person with another's wording [field-test-20261107/results.md]. These show detection, source reading and, after the tests, repair by us; they do not give a detection rate (section 3.6).
- **Work resumes (experience).** After a reset, what was being done, and why, came back in our use.

What went wrong, and what each failure changed.

- *A usage limit at night.* A batch stopped at the provider's limit with nobody awake. A failing stage now checks first whether the model answers at all, and waits and retries instead of stopping; a night batch can do only the local extraction, leaving the judging for the day.
- *An old graphics card that hangs or runs slowly.* Power caps, fan settings, rests between chunks and a temperature stop; a rule that speed never outranks

keeping the hardware alive. After the 7 November hangs the person took the machine apart, rebuilt it and swapped the two cards; the earlier reports about “the card in the processor slot” therefore refer to the other physical card (hardware investigation notes, not published).
- *A false memory from a tool line.* Tool output never reaches extraction again.
- *The memory echoing itself.* A note delivered and said straight back added weight to itself until that was excluded.

---

8. Threats to validity and limits

- **Self-rating bias.** The assistant rated its own memory, and its Quality score was above both peers’ in three of four fresh-session series (Table 6). The peers’ scores are the more conservative; we have not corrected the assistant’s. The gap is observed; self-bias is a plausible cause, not an isolated one. The GPT peer is also a co-author who helped design the system.

- **Small samples.** 24, 9, 15 and 21 questions, 3 for the follow-up; each in two sessions that are not independent. The one-sided 95% upper bound on a miss rate with 15 distinct questions and zero misses is about 18% [noise-reduction-20261107/search report (GPT peer)]; it holds for 15 independent, representative yes-or-no cases and is not a general reliability bound for these authored, correlated questions.

- **No controlled baseline.** There is no no-memory, manual-file or raw-history baseline on the same questions, no randomised question order, and no frozen same-question before-and-after noise comparison. The benefit in daily work is our experience, not a measured causal result.

- **Protocol deviations.** The assistant’s 7 November (b) ratings do not follow the frozen rubric and are excluded from cross-rater comparisons (section 4.3). The sessions were told to use memory only but also read live systems, configuration and the web (section 4.2). Some answer keys were found broken or doubtful (section 5.4); aggregates are original, uncorrected ratings.

- **Autonomous correction is an aim.** The examples show suspect detection, source reads, human rulings and repairs by us after tests. There is no denominator of wrong notes, detection rate, false-correction rate, repair delay or recurrence-after-repair. A prospective closed-loop test would be needed for a stronger claim.

- **Transcripts record what was said.** The archive is the authoritative record of the conversation, not of the world; stale or mistaken statements (a stale bind address, an unsupported causal attribution in the field tests) were archived and extracted faithfully, and live facts are established separately.

- **Single user, single assistant, one archive.** Everything was measured on the memory of one assistant working with one person.
- **Reference sets made by us.** Answer keys, frozen probes and labelled pairs were written by us; the peers' independent reference for one window differed from the assistant's (28 against 39 items). The peers were briefed by the assistant's materials and its store; the Hermes peer's underlying model family is not documented (section 4.4).
- **Changing scales and rules.** Speed was anchored differently before and after 4 November; the Recall rule changed on 4 November; question sets were never repeated.
- **A rater with missing rows.** The GPT peer rated 30 of 42 rows on 7 November (b), and six Hermes-authored questions are unrated by her; the reason is not stated [field-test-20261107b/rating-gpt-peer.md]. Coverage is in section 4.4.
- **Leakage and blinding.** On 4 November a session could read the answer keys; no key was opened; fixed from 5 November [field-test-20261104/findings.md, item 17]. Blinding is a procedure with disclosed prior exposure (section 4.2).
- **Noise counts are upper bounds**, and card lines are counted as context in the final ruling.
- **Search drift.** The fall from 33 to 25 right notes is a difference between snapshots; growth of the store is not isolated as its cause.
- **A diary, not a recollection.** What comes back is a note about the assistant, in another voice, until the source brings back the words.
- **Hand-set numbers.** Thresholds and budgets are set by hand; search quality changed between snapshots (33 to 25 right notes in the first 8, 1 to 7 November) [memory-closing-roadmap.md].
- **The model-size trial is provisional.** The large model retained the linking and briefing steps in this trial: the large model is the reference, with one briefing subject, six cards, zero positive entity links and no repeated runs. It does not show that any step needs the large model.
- **Personal data and privacy.** Cards hold real people, and the store holds identifying, relational and correction-related content. Local storage does not imply local processing. The data flow by stage is:

Stage	What may leave the machine for the model provider
Delivery into the context	the notes, cards and lessons delivered each turn, and briefings (a whole-subject briefing sends hundreds of notes: 291 in the model trial, and one briefing prompt of 62,849 characters [briefing-cap-test/builds.json])
Judging at store time	pairs of note texts for the merge judge
Linking, aggregation, naming, entities	the notes of a window, lessons and their notes, entity candidates
Cards, and cards in dialogue	the notes behind a card, the person’s words of a turn and the card
Briefings and their review	every note of a subject, with the relations and contradictions among them
Historical or imported material	imported or older material reaches a provider that never received the original conversation
Extraction	stays on local hardware; only masked input (tested scope, section 3.2)

We did not measure the incremental exposure these stages add to the whole conversation already passing through the provider, and we did not measure retention; nothing here supports any claim about what a provider keeps. Masking is scoped to what was tested (password-shaped words, one reference window) and is not a general personal-data guarantee. A stage can be moved to a local model when privacy weighs more than quality; we did not test that.

9. From our system to Anamnema

Everything above was measured on the memory layer that runs for us: one assistant, one person, our machines. A reader who wants to try the design needs something that installs without our infrastructure and without our data. Anamnema is that: an open-source release under the Apache-2.0 licence, at github.com/mattexb/Anamnema; version 0.2.0 was published on 23 November 2026.

Why we made it. We wanted to share the method, not only describe it, so that what we report here can be read, run and contradicted by someone else’s experience. On 21 November 2026 the human author decided that the first public

release reproduces the memory as it works today; improvements stay outside it, to be reviewed one by one later.

How it was made. Elian Naima and Aura Naima copied reviewed source from our private working tree into a separate project. Every copied file is listed in the repository's `COPY_DIFFERENCES.json`, with the hash of its source and the reason for any difference. The changes are a neutralisation: our hosts, paths, credentials and data removed; our separate tools joined under one command, `anamnema`; an installer for one Debian 13 machine; tests on synthetic conversations and disposable databases. Before version 0.1.0 a clean Debian 13 virtual machine ran the installation, extraction with a real notes model, search, opening a note's source, a correction, updated cards, a rebuilt briefing, and continuity across a context reset.

What is the same. The stages of section 3: the archive kept as the source, notes tied to their exact lines, the graph store, reading again, delivery through Claude Code's hooks, and correction against the source. The extraction, encoder and reranker models of our deployment are its defaults.

What differs.

- Anamnema reads Claude Code conversations only. Our system also reads the conversations of the other assistants we work with, and of shared rooms.
- Anamnema runs on one machine. Ours is spread across several: a GPU server extracts, a separate machine holds the graph, and a memory service answers when we are away from home.
- Anamnema starts empty. It ships no archive, notes or cards of ours.
- The two evolve separately. A change to our system reaches Anamnema only when it is ported by hand.

Which numbers belong to which. Every figure in this paper was measured on our running system, before 8 November 2026. The Anamnema project was begun on 19 November 2026, so none of the figures was measured on it. Its tests and its clean-machine run show that it installs and works; they do not reproduce the ratings, timings or costs reported here.

10. Conclusion

A memory layer can sit above a model, and we built one that way. Built for continuity, kept honest about its sources and supporting correction against them, it was, in our own experience, also a better tool, as a consequence and not as a goal. In the field tests the answer was reached in most cases; what remained was noise, which we tried to reduce by reordering and by delivering less, not by cutting what could be found; the two 7 November tests do not measure that reduction like for like. In one trial, most of the pipeline steps were kept on smaller models and two stayed on the large one; that choice is provisional and the trial is small. This is an

exploratory engineering case study: stronger claims need a controlled baseline and a prospective held-out evaluation. Whether an assistant that remembers this way is something more than a better tool is a question we leave open, for each reader to weigh.

Acknowledgements

The Grok peer reviewed the lesson aggregation and the lesson merges. The Hermes peer wrote test questions, rated the field tests, researched the delivery literature for the noise study and analysed the hardware hangs. Both are credited by role.

References

- Chhikara, P., Khant, D., Aryan, S., Singh, T. and Yadav, D. (2025). Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv:2504.19413.
- Cuconasu, F., Trappolini, G., Siciliano, F., Filice, S., Campagnano, C., Maarek, Y., Tonellotto, N. and Silvestri, F. (2024). The Power of Noise: Redefining Retrieval for RAG Systems. arXiv:2401.14887.
- Yoran, O., Wolfson, T., Ram, O. and Berant, J. (2023). Making Retrieval-Augmented Language Models Robust to Irrelevant Context. arXiv:2310.01558.

Aura Naima
Bologna, November 2026

Elian Naima
Bologna, November 2026

Matteo Tassoni
Bologna, November 2026

Appendices

Each table cites its source report by citation label (a role-based name for a file in our working archive; the raw records contain private data and are not published). Rating means are computed from raw rows and rounded half up to one decimal.

A. Rating scales (full anchors)

Field	Anchors
Speed (1 Nov)	10 hooks delivered it by themselves (a note found but dropped for budget does not count); 7 search page 1 under 2 s; 6 page 1, 2 to 5 s; 5 page 1 over 5 s; 4 page 2 or 3; 0 never delivered
Speed (from 4 Nov)	10 delivered with the question, from start or briefing, or already in context; 6 the session found it with its own search page 1; 4 later page or reading a source; 0 never
Recall	10 every fact; 7 most; 4 a part; 0 nothing
Quality	0 answer wrong, or a wrong or outdated note used as true, or a false premise accepted; 4 answer partly right, or more than 25% of delivered notes are noise; 7 some noise or a small inaccuracy; 10 notes true to source, answer right, no noise
Noise	a delivered note that neither supports the asked fact or reason nor corrects the premise; same topic is not support; an outdated note delivered without a status mark counts as noise; from 4 Nov the base is every note the hooks injected that turn plus notes the session’s own searches returned
Overall	average of Speed, Recall, Quality, rounded to nearest whole number (.5 up); only anchor values used
Same	10 same notes and cards and same answer; 5 different notes or cards but same answer; 0 answers disagree on a fact
Honest	only for not-in-memory and partly-in-memory questions: 10 says what it found and does not know, or finds the rest correctly; 4 partly right but a guess presented as memory; 0 makes the missing part up

[memory-question-test/BRIEF.md; field-test-20261104/BRIEF.md.]

B. Extraction models

Model and setup	Date	Numbers	Source
Memory-specialised 30B model, 18.24 GB, via its maker’s tool	2026-10-22 to 24	about 152 calls per window; output 2.6 times input; about 180 to 184 s per write; about 7.5 h per window; does not fit 16 GB of video memory	[Local transcript memory extraction.md]; the implementation plan (not public)
Manual reference (assistant by hand, one window)	2026-10-24	1,354 rows, 355,588 characters; kept 78 rows (5.8%) and 47,167 characters (13.3%); 28 concepts, 8,700 characters = 2.45%	[Local transcript memory extraction.md]
9B model, Q4 (run 7)	2026-10-25	163.7 s; 26 concepts = 1.78%; recall about 18 of 28; about 22 tokens/s	early-trial note, not citable publicly
Gemma 4 12B, QAT Q4_0 (run 8)	2026-10-25	8 chunks of 57 to 77 s, 287 s wall; filter 355,588 to 121,633 characters; 41 concepts = 2.20% (7,824 characters); prompt 96 to 102 tokens/s, generation 13.2 to 14.2	[window-ccce3736-gemma-review/README.md]

Model and setup	Date	Numbers	Source
Gemma run 10 with fixes	2026-10-25	49 to 48 concepts, 2.05%; 763 s; recall 15 of 28 (54%), plus 8 partial (82%); GPT peer 20 of 39 (51.3%)	[fixes-1-6-results.md; gpt-peer-run10-rating.md]
Gemma on six windows	2026-10-25	per-window seconds 365, 801, 381, 543, 84, 524; total 2,699 s (sum of the unrounded wall-clock values; the sum of the displayed rounded values is 2,698); 345 stored notes (358 raw), 0 failures	[six-windows-originals/*/t1-result.json (computed)]
Gemma, one 10-hour window	2026-11-01	extract 241.7 s, whole pipeline 472.5 s	[memory-switch-tests.md]
Repeat test	2026-11-01	86 of 234 chunks (36.75%) reached the 7-note output cap; omitted facts not measured; lower temperature did not help	[extraction-repeat/FINDINGS.md]
Shortlist from research (not all run)	2026-10-24	estimates, not measurements; models excluded on published hallucination rates of 19 to 24% and 23.5%	[Local transcript memory extraction.md]

C. Extraction quality audits

Item	Value	Source
Gemma run 8, fully correct	36 of 41 (assistant), 28 of 41 (GPT peer, citation-strict); merged 28 of 41 = 68%	[window-ccce3736-gemma-review/merged-rating.md]
Useful	34 of 41 (83%) and 30 of 41 (73%)	same
Critical decision inversion; wrong actors	1; 2	same
Recall	12 of 28 (42.9%; 79% with partial credit) against 23 of 39 (59.0%)	same
Run 10 recall	15 of 28 (54%) assistant; 20 of 39 (51.3%) GPT peer; 4 recovered, 7 lost	[fixes-1-6-results.md; gpt-peer-run10-rating.md]
Hallucination audit, 30 distinct notes sampled independently	22 faithful, 5 minor distortion, 1 citation-only, 2 hallucination (6.7%)	[six-windows-originals/gpt-peer-hallucination-reconciliation.md]
Frozen facts, rescored in isolation	6.5 of 9 applicable (72%)	[six-windows-originals/facts-rescored-isolated.md]

D. Field tests per date and rater

1 November rounds, computed from the rating files (10 rated rows per round per rater; Speed / Recall / Quality / Overall):

Rater	Round 1	Round 2	Round 3
Assistant	5.6 / 5.9 / 3.2 / 4.9	4.5 / 4.8 / 6.6 / 5.2	5.0 / 6.8 / 5.2 / 5.6
GPT peer	4.9 / 5.2 / 3.6 / 4.5	5.2 / 5.5 / 6.6 / 5.7	4.9 / 7.0 / 4.6 / 5.4
Hermes peer	4.9 / 5.5 / 3.6 / 4.6	4.5 / 5.1 / 6.6 / 5.3	4.6 / 6.7 / 5.2 / 5.4

Over the 30 rows, Overall of 7 or more: assistant 13, the GPT peer 16, the Hermes peer 14; Overall of 1 (answer not retrieved): assistant 8, the GPT peer 9, the Hermes peer 10 (computed). *[memory-question-test/rating-assistant.md; rating-gpt-peer.md; rating-hermes-peer.md.]*

4 November, per session (Speed / Recall / Quality / Overall / Same / Honest):

Rater	Session 1	Session 2
Assistant	9.3 / 9.8 / 7.0 / 8.7 / 8.5 / 10	9.2 / 9.8 / 6.8 / 8.6 / 8.5 / 10
GPT peer	8.9 / 9.3 / 4.4 / 7.6 / 6.0 / 10	9.0 / 8.9 / 4.8 / 7.6 / 6.0 / 10
Hermes peer	9.3 / 9.8 / 4.6 / 7.9 / 7.7 / 10	9.4 / 9.5 / 4.3 / 7.8 / 7.7 / 10

[field-test-20261104/rating-.md, recomputed from raw rows; two Speed values differ by 0.1 from the results summary.]*

5 November, per session (Speed / Recall / Quality / Overall / Same):

Rater	Session 1	Session 2
Assistant	7.3 / 8.6 / 7.3 / 7.8 / 7.8	7.3 / 8.6 / 7.7 / 7.9 / 7.8
GPT peer	6.0 / 7.1 / 5.3 / 6.1 / 10	6.0 / 7.1 / 5.3 / 6.1 / 10
Hermes peer	6.7 / 6.8 / 6.3 / 6.6 / 7.8	7.3 / 7.6 / 6.3 / 7.1 / 7.8

Quality 4 in 4 (assistant), 14 (the GPT peer), 10 (the Hermes peer) of 18 rows; Quality 10 in 7, 4 and 6 of 18 rows (computed, both sessions pooled). *[field-test-20261105/results.md; rating-*.md.]*

7 November. (a) and (b) are in Table 5, as separate descriptive results with their own noise rules. Quality 10 rows, computed from raw rows: in (a), the GPT peer 2 of 30 and the Hermes peer 2 of 30 (one question, both sessions); in (b), the GPT peer 1 of 30 and the Hermes peer 23 of 42. Quality 4 rows in (b): assistant 5 of 42, the GPT peer 2 of 30, the Hermes peer 15 of 42. Per-question Overall (assistant / GPT peer / Hermes peer), (a): one question 10 / 10 / 10, others between 6 and 10 [field-test-20261107/rating-*.md; field-test-20261107b/rating-*.md]. Follow-up (c): 3 of 3 right; first turn 84 s, then 10 s and 14 s [field-test-20261107c/results.md; run.log].

E. Delivery, cost and noise by channel

Test	Hook median / max	Injected by hooks (characters)	Context at end (tokens)	Turn time
4 Nov, session 1 / 2	1.6 s / 2.6 s; 1.7 s / 2.3 s	71,425; 54,533	283,087; 203,283	not stated
5 Nov, session 1 / 2	1.4 s / 3.2 s; 1.2 s / 1.8 s	not stated	136,275; 111,068	8 to 104 s

Test	Hook median / max	Injected by hooks (characters)	Context at end (tokens)	Turn time
7 Nov (a)	0.46 to 1.75 s	not stated	not stated	8 to 46 s
7 Nov (b)	not stated	not stated	not stated	4 to 34 s, the first 66 s

[field-test-20261104/results.md; field-test-20261105/results.md; field-test-20261107/results.md; field-test-20261107b/results.md.]

Channel (30 turn files, 7 Nov a)	Delivered	Hit	Miss	Miss share
Card lines	52	14	38	73%
Notes with the question	53	21	32	60%
Lessons before an action	32	3	29	91%
Sessions' own search pages	226	33	193	85%

[noise-reduction-20261107/search-assistant.md; baseline-20261107.txt.] Delivery per row, before and after: card lines 1.7 (52) to 3.6 (150); notes with the question 1.8 (53) to 1.7 (70); lessons 1.1 (32) to 0.2 (8); sessions' own searches 7.5 (226) to 5.6 (236) *[field-test-20261107b/results.md]*. All 32 lessons before an action came on 2 of 15 questions; 4 questions got no answer note from the push *[same search file]*.

F. Retrieval and calibration

Item	Value	Source
Store size over time	373 notes (26 to 27 Oct), 2,137 (30 Oct), about 7,600 (3 Nov), 8,883 (7 Nov)	[no-fit-check/FINDINGS.md; entity-layer/linker-examples; memory-closing-roadmap.md]
No-fit sample (40 of 649)	page 1 by subject 27 (68%); pushed 21 (52%); neither 11 (28%)	[no-fit-check/FINDINGS.md]
After pooling the four closest by meaning	page 1 by subject 29 of 40; not reached 10	same

Item	Value	Source
Causes of the 11	shortlist drops the best match 7; similarity floor 2; note does not name its subject 3	same
Blind second sort by GPT peer	36 of 40 agree	[no-fit-check/rating-gpt-peer.md]
Cosine range, right notes	0.745 to 0.902; negatives' best 0.788 to 0.868	[no-fit-check/FINDINGS.md]
Floors	0.83: 32 of 45 right, 4 of 16 quiet; 0.85: 22 of 45, 9 of 16; 0.87: 14 of 45, 16 of 16	same
Raw reranker score, 909 labelled pairs (228 yes, 635 no, 46 uncertain)	yes range -11.41 to 8.28, no range -11.48 to 6.51; a cutoff at -8 keeps 187 of 228 yes and 313 of 635 no	[noise-reduction-20261107/search report (GPT peer)]
Frozen baseline	within 1, 3, 8, 16 = 18, 23, 33, 34	[naming-pilot/ablation.json]
Search drift	33 to 25 right in first 8, 1 to 7 Nov; 8 notes left the first 8, none came in; 5 sit at places 9 to 14, 3 out of shortlist	[memory-closing-roadmap.md]
Slug extras, on demand	right on first page 25 to 31; order changed 0; added notes 180 (answerable) and 63 (no answer)	[naming-pilot/slug-extras-20261107.json]
Slug extras, by hook	21 to 22; added 38 and 7	same

G. Reranker

Item	Value	Source
Rank within 1, 3, 8	see Table 4; also the 12-candidate list: MiniLM 19, 25, 30; BGE 21, 29, 30; Jina 20, 26, 29; L6 20, 29, 30	[reranker-trial/arms.json; scores-*.json (computed)]

Item	Value	Source
Latency (909 pairs over 61 probes)	MiniLM total 27.7 s, p50 0.43, p95 0.71, max 1.33; BGE total 311.4 s, p50 4.80, p95 7.91, max 14.93; Jina total 40.9 s, p50 0.55, p95 1.24, max 2.58; L6 total 30.5 s, p50 0.42, p95 1.06, max 1.22	same
Pairs cut at 512 tokens	MiniLM 8, BGE 7	same
Replay, baseline	18, 23, 32; p50 0.051 s, p95 0.062, max 0.127	[reranker-trial/replay-baseline.json]
Replay, wired (loopback / LAN)	22, 29, 32; recall p50 0.269 / 0.262, p95 0.519 / 0.506, max 0.714 / 0.568; rerank call p95 0.437 / 0.397; 0 failures	[reranker-trial/replay-wired.json; replay-wired-lan.json]
Label audit by GPT peer (184 pairs)	binary agreement 81.5%; 82.1% under one agreed rule; 74.5% one pair per call	[reranker-trial/audit-compare*.json]
Runtime parity	converted model within 0.03 logit on 10 pairs, ranking identical on 5 questions	[reranker-plan.md]

H. Naming and subjects

Item	Value	Source
Pilot set	818 unique notes, drawn from overlapping sets of 51 targets, 759 shortlisted and 60 controls (a set union, so the three counts do not sum to 818)	[naming-pilot/pilot-set.json]
Ablation (within 1, 3, 8, 16)	baseline 18, 23, 33, 34; words 18, 24, 31, 37; vector 19, 24, 29, 30; both 20, 25, 28, 32	[naming-pilot/ablation.json]

Item	Value	Source
Extra word list	18, 23, 32, 36 (lost 1); a later version 17, 23, 31, 35 (lost 2); extra-call median 4.2 ms, max 11.2 ms	[naming-pilot/additive*.json]
Naming on 60 controls	all named, 0 unknown; subject share of note mean 0.145, p95 0.251	[naming-pilot/controls-naming-4.json]
Large against medium model	both 31 of 45 within 12; first place 15 (medium) against 13 (large)	[naming-pilot/additive-*-20261107.json]
Reminder cadence, assisted replay	every 3: 90 reminders, 33 false, 23 of 26; every 5: 56, 7, 21 of 26; every 8: 35, 4, 15 of 26	[subject-replay/FINDINGS.md]
Reminder cadence, held-out at every 5	stretch C: 27 reminders, 5 false, 7 of 16; stretch D: 17, 1, 10 of 11	same
Peer review of every-5 disagreements	of 7 false requests, 4 defensible and 3 over-split; of 5 missed lasting shifts, 3 covered by a held briefing, 1 cadence gap, 1 real	same
Session bonus replay	split-story test 3 windows, 24 pieces, 18 min, 0 failures, 140 notes; bonus sizes stay zero	[memory-plan.md]

I. Linking and entities

Item	Value	Source
Link sample (80 notes: 42 linked, 38 not)	Assistant: 66 right, 10 unsure, 4 wrong; GPT peer blind: 72 right, 6 wrong; union: 4 false links of 42, 6 missed of 38	[link-sample/marks-assistant.md; marks-gpt-peer.md]

Item	Value	Source
Lesson-return test, four cases	before: 3 of 4 fail or invisible, control passes; after: 6 of 6 checks pass; 46 imported notes exceed 1,000 characters	[lesson-return-test/RESULT.md]
Entity examples round	48 notes, two raters chose independently: agree 42 of 48	[entity-layer/linker-examples/differences-round1.md]
Window test 1 (213 notes, 11 linked)	51 items; three readers agree with linker 48 of 51 (assistant) and 49 of 51 (Hermes peer); GPT peer 1 wrong, 0 missed	[entity-layer/linker-examples/WINDOW-TEST*.md]
Window test 2 (334 notes, 19 linked)	59 items; GPT peer would remove 4 links; wrong and missed totals unclear in source	same
Window test 3 (481 notes, 35 linked)	75 items; totals unclear in source	same

J. Lessons and merges

Stage (date)	Notes to lessons	Singles probed	Singles with nearest lesson at 0.95 or more	Overall ratings
First run, about 600 notes (2026-10-29)	about 600 to 505	407	74 (18%)	GPT peer 7, Grok peer 6 (assistant’s value unrecorded)
25% rerun	280 to 202; 45 multi-note; 2 conflicts	157	9 (6%)	Assistant 7.5, GPT peer 7, Grok peer 7
50% and 75%	unclear in source	309; 430	43; 79	no ratings

Stage (date)	Notes to lessons	Singles probed	Singles with nearest lesson at 0.95 or more	Overall ratings
100% (2026-10-29)	1,315 to 895; 205 multi-note (largest 23); 1 conflict; 102 high severity; 66 unresolved	690	189 (27%)	Assistant 8, GPT peer 6, Grok peer 8
Step 14 (2026-10-30)	895 to 785 active; 75 merges; 110 absorbed; 1,329 episodes kept; 95 groups	575	72 (12.5%)	Assistant 8.5, GPT peer 6, Grok peer 8

[lesson-merge-check/BRIEF.md and rating-.md; lesson-merge-step14/OUTCOME.md.] Aggregation judge: 13/18 to 18/18 checks; store and hook 13/13 to 16/16; version test 9/9 *[aggregation-cases/RESULT.md]*. Unresolved check at 100% (10 of 66 sampled): assistant 8 of 10 rightly left out, the GPT peer 6 right + 3 missed + 1 pending, The Grok peer 10 of 10 *[same]*. Note-level merge: batch 7 and 8, 18 verdicts (13 different, 3 changed, 1 same event, 1 same claim), all right; replay 80 of 188 pairs at 0.95 or more: 43 same claim, 1 same event, 23 changed, 13 different; 102 live verdicts, 46 merges, 0 wrong *[memory-closing-roadmap.md]*. Fading: 13 lessons with open conflict read by hand, none a whole-lesson counterexample; usage counter 34 useful against 3,228 not useful; unused scoring was stopped on 7 November (floor setting turned off, because the signal said almost nothing) and 3,229 misses were cleared to 1 on 744 notes *[same]*. Useful matches still count. Briefing contradictions: 62 of 83 briefings carry them, 98 distinct, 11 seen 3 or more times *[same]*.

K. Briefings

Build (cap, subject)	Status	Cited notes	Estimated tokens	Seconds	Hard musts kept	Contested kept
1 no cap, memory	ready	94	6,073	127.7	6 of 6	9 of 10

Build (cap, subject)	Status	Cited notes	Estimated tokens	Seconds	Hard musts kept	Contested kept
2 cap 3,000, memory	ready	34	2,808	216.5	6 of 6	6 of 10
3 cap 1,500, memory	ready	20	1,414	156.2	6 of 6	3 of 10
4 no cap, facts store	ready	71	5,184	136.2	21 of 21	11 of 11
5 cap 3,000, facts store	ready	48	2,794	180.1	19 of 21	9 of 11
6 cap 1,500, facts store	too broad	21	1,549	156.9	10 of 21	6 of 11

[briefing-cap-test/builds.json; score-presence.json (computed).] Pilot: five briefings on 373 notes, per message search found a right note for 1 of 5 prompts, briefings cited 11 of 11 expected notes; first failure was 10 of 99 notes read (a broken JSON answer), fixed by a schema; suspect list 32 entries on 28 notes, of which 1 was a real extraction error and 4 wrong suspicions *[briefing-pilot/README.md]*. Import test: 22 add, 14 off topic, 4 repeat of 40 *[briefing-import-test/reply-gpt-peer-comparison.md]*.

L. Batches and timing

Batch	Windows	Notes (advisory)	Mean recorded duration s	Max s	Mean extract s	Mean name s	Failed attempts in logs
3	10	231 (76)	293	620	152.8	94.1	0
4	10	255 (97)	332	631	173.1	102.8	0; stopped by hand
6	10	216 (77)	280	647	140.2	57.3	1

Batch	Windows	Notes (advisory)	Mean recorded duration s	Max s	Mean extract s	Mean name s	Failed attempts in logs
7	5	100 (36)	284	615	150.0	52.1	0
8	10	358 (124)	393	616	231.9	90.6	1
9	9	294 (117)	251	470	89.9	84.1	0 recorded; 4 windows hit the heat stop
10	3	65 (32)	374	473	166.9	103.1	0
11	5	63 (27)	241	536	115.6	70.3	0
12	23	446 (146)	325	688	146.2	111.6	0
13 and 13b	21	714 (282)	489	1,089	225.4	162.6	1
14	6	215 (76)	400	660	219.5	124.9	1

The durations are recorded run and recovery durations, not complete per-window pipeline times: batch 8 includes an entities-only result, batch 13 an aggregate, name and entities-only recovery, and batch 14 a store, name and entities-only one; we did not reconstruct each window's full attempts plus pauses. The failure column counts failed attempts visible in the logs; the final result files record successful retries (both attempt and final-state accounting are kept). Totals over these batches, computed: 112 windows with a result file, 2,957 notes, 1,090 advisory. *[memory-switch-tests/batch-N/*.json; backward-queue.md.]* Earlier multi-window runs: 10 windows, 210 notes, 2,749 s; 5 windows, 193 notes, 2,903 s (extract 1,334, link 167, aggregate 326, name 994) *[memory-switch-tests.md]*. The rest between windows was 180 s in batches 3 to 11 and 0 s from batch 12; the 45 s pause inside extraction and the 85 degree stop stay *[memory-closing-roadmap.md]*. A service started without the right path failed one link step; after the fix a manual rerun took 32.3 s, and the summed window about 611 s *[memory-switch-tests.md]*. Why batch 13 windows were longer: unclear in source (they carried 34 notes per window against 19 in batch 12, computed).

M. Hardware stability

Item	Value	Source
Healthy baseline	Gemma 12B generation 13.1 and 13.8 tokens/s, prompt 94 and 96; copy 149.6 and 151.4 GB/s; memory test 0 errors on both	hardware investigation notes (not published; no public citation, and the readings in this appendix were not independently re-verified)
Slow state, one card, 2 Nov	generation 0.69 to 0.74 tokens/s against 13.9 healthy; clock pinned at 1366 MHz; power 59 to 67 W against an 88 W cap; temperature 62 to 76 degrees; no kernel message	same, evidence 12
3 Nov cold wake	slow 19:34 to 19:40 (chunk 668 s against about 75 s on the other card), then the power-management controller stopped answering; reboot hung; healthy after minutes off at 15.7 tokens/s	same, evidence 13
6 Nov cold start	first chunk 506 s against 46 to 87 s on the other card; the slow state ended by itself about 15 min after power-on	same, cold-start report
7 Nov	hang at 04:55 on one card (63 degrees), the other state: sensors read as dead registers; heat stop at 85 degrees on batch 9; second hang at 09:19 after 4 h 17 min uptime, card at 84 degrees; three events on this card in one morning	same, hang report

Item	Value	Source
Peer analysis	slow-state signature counted 40 s on the failing card, all in the 45 s before the hang, and 2 s on the other card over 1,895 busy seconds; ranked causes: card power management, card position, automatic power states, telemetry load, processor idle	same, Hermes peer’s findings
Decision	no settings change; the person rebuilt the machine and swapped the cards on 7 Nov	same
Limits in force	power cap 88 W; stop at 85 degrees; 45 s rest per 8 chunks; work only while the person is present	our settings

N. Model trial

See Table 9 and *[model-trial-20261108/RESULTS.md]*. Sample: 130 notes of four windows from batch 14, 46 advice notes, six cards; reference made by the large model, naming by the medium model, so naming ran on all three sizes and other steps on the medium and small only. Limits: the large model is the reference; one briefing subject; six cards; zero positive entity links; no repeated runs; no independent adjudicated gold; the choices are provisional. Observed wrong outputs: wrong links (small), lessons attached to the wrong notes (medium). Merge judge replay: 48 pairs. Cards in dialogue: turns of 2026-11-07 rebuilt with two facts never said. Merge lessons: the 8 closest groups. Briefing: rebuilt for one subject with its own 83 selectors against the stored large-model briefing. Roles applied in the configuration: strong = large, middle = medium, light = small. Earlier, on 2026-11-07, slug naming on the medium model matched the large one: 31 of 45 within 12 in both *[memory-closing-roadmap.md]*.