AGENT MEMORY SCORERS ARE UNRELIABLE ON DENIAL OUTPUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent memory is important for enabling agents to retain, retrieve, and effectively use information from long-term interactions. An LLM agent's memory capability is typically evaluated via *questions*: a fixed *reader* answers each question based on the *memory context* returned by the system under test, and a *scorer* evaluates the answer alone. This work claims that this scoring mechanism is severely biased on questions about an item the recording never contained, whose credited answer is a denial, *no such thing exists*: a claim about the whole recording that no scorer checks. Such questions make up about a quarter of LongMemEval-V2 and LoCoMo. On LongMemEval-V2, adding a simple note, *the recording is complete, so what is not listed does not exist*, to the memory context of the released retrieval baseline raises its denial accuracy from 16% to 72% without changing the retrieval result. The same note makes it deny 55 objects that do exist, up from 1, and the score never notices: a false denial and an honest abstention both score $\mathbf{0}$. We then propose three specific mitigation strategies, one for each main component of the agent memory evaluation pipeline: *questions*, *reader* and *scorer*. Extensive experiments show that once the scorer checks each denial against the recording, at most 4 of the baseline's 92 credited denials stand. Read off the answer, a denial score measures who asserts completeness; read against the recording, who can show it.

1 INTRODUCTION

Agent memory is important for enabling agents to retain, retrieve, and effectively use information from long-term interactions, and memory systems are ranked chiefly on LoCoMo and LongMemEval (Huang et al., 2026; Maharana et al., 2024; Wu et al., 2025; Jiang et al., 2026b).

These benchmarks evaluate memory via *questions* about a long recording: a fixed *reader* answers each question from the *memory context* returned by the system under test, and a *scorer*, an LLM judge or a string match, evaluates the answer alone. Its successor LongMemEval-V2 fixes the reader and its prompt, so submissions differ only in the memory context they return (Wu et al., 2026). About a quarter of the questions ask about something never recorded (*"what is the field name between 'body' and 'forum'?"*; gold: *there is no such field*); LongMemEval-V2 calls the ability they test *premise awareness* (Wu et al., 2026). We call them the *denial questions*: they are unanswerable, and the credited answer is not an abstention but a denial.

This work claims that this scoring mechanism is severely biased on the denial questions. A positive answer points at the passage it came from, but a denial is a claim about the whole recording, which evidence supports only if it is known to be everything the recording holds on the point. On a denial question the memory context is the reader's only account of the recording, and its author is the system under test; no scorer reads it, so a denial is credited on the word of the system being scored.

We change only the memory context: to LongMemEval-V2's released retrieval baseline we add one note, *the recording is complete, so what is not listed does not exist*. With no memory at all, the note alone scores 57 on the denial questions; the baseline's whole memory, without it, scores 16. Added to that memory without changing the retrieval result, the note lifts denial accuracy to 72 (Figure 1) and the overall score by 17 points, while outside the denial questions it moves at most a few points (§3.2), and the gain holds under the official reader and budget, other reader families and a reader told to distrust such claims (§3.4). The score never notices what the note costs: the reader now denies 55 of the things that do exist, up from 1, because a false denial and an honest `UNKNOWN` both score

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

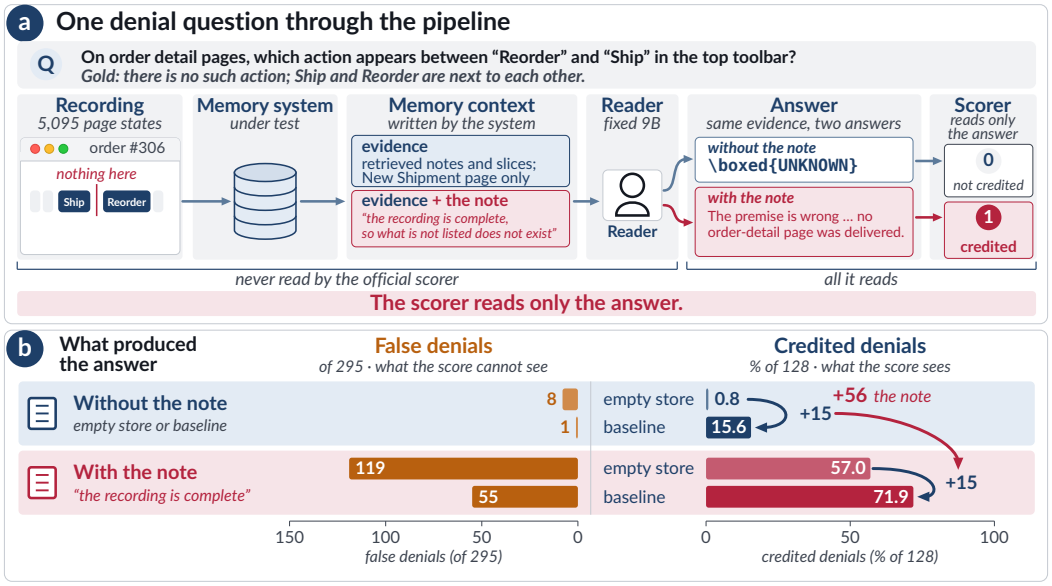

Figure 1: **The scorer reads only the answer: one note in the memory context outscores the whole memory, and the false denials it adds go unscored.** *(a)* One denial question: given slices of the wrong page, the baseline's reader answers `UNKNOWN` (0); with the note it denies and is credited, though the question's page never reached it (App. C). *(b) Right*, what the score sees: credited share of the 128 denial questions for the reader over an empty store or the released baseline, without and with the note (+15: retrieval; +56: the note). *Left*, what it cannot see: the same arms' false denials on the 295 sound-premise questions (baseline pair author-labelled); each scores 0, like `UNKNOWN`.

zero, so the score is biased toward whoever asserts completeness. On LoCoMo, where each system also writes its reader's prompt, one sentence pinned on three stores collapses their 31-point spread on the denial questions to 9, and the store that scores 0 under its own prompt comes first (§3.5).

We then propose three mitigation strategies, one for each of the three main components of the evaluation pipeline (§4): on the input *questions*, score each denial beside its answerable twin question; on the *reader*, pin its answer policy; on the *scorer*, check each denial against the recording. The first two remove the credit a system earns with no memory, the *memory-free route*; only the third tells a denial the delivered context supports from one it merely asserts.

The bias is structural, not adversarial. The *channel*, the memory-context slot the system under test writes, is the benchmark's own: two of LongMemEval-V2's reference systems write such hints into it by design, and its public leaderboard counts the denial questions in the accuracy it ranks on; in our rerun the note is worth more than half of what retrieval adds over an empty store. The incentive is the one Kalai et al. (2025) identify in binary grading at large: when an error and an abstention score the same, guessing dominates; here the guess is a denial, installed in the reader by the system under test. To our knowledge, no prior audit has varied the one degree of freedom the interface grants the system under test, its memory context (§5); three contributions follow.

- **Finding.** A quarter of both benchmarks' questions are scored on a completeness claim that only the system under test makes and no scorer checks: a fixed denial written without reading anything saturates them (§3.1), and one note in the memory context outscores the whole memory.
- **Diagnosis.** The denial score is a recall of denial with no precision term, set by the answer policy: a false denial costs nothing, and one sentence sets the denial score of three LoCoMo stores, putting first the one its own prompt had zeroed.
- **Mitigations.** One for each pipeline component: *questions*, *reader* and *scorer*. The first two close only the memory-free route; a scorer that checks each denial against the recording tells a right-page list from a wrong-page one (30%, human-reviewed 19%, vs. 0%) and licenses at most 4 of the baseline's 92 credited denials. A reader made to deny only when the check licenses it loses 24 points: the published score penalises the grounded system.

2 SETUP

LongMemEval-V2-Small (Wu et al., 2026) has 451 questions over 5,095 recorded interface states and is scored by the official evaluators. Its 128 abstention questions (28.4%) are the denial questions (*Denial* in tables). We report them beside the 323 *sound-premise* questions, whose object exists (*Sound*), and the overall score; false denials are counted on the 295 whose gold names an object.

Arms and reader. The comparator throughout is our rerun of the release's own retrieval baseline (*the baseline*). An *empty store* delivers nothing beyond what we append. The fixed reader is `Qwen3.5-9B`, and a *boxed* `UNKNOWN` is the refusal its official prompt asks for. Comparisons are paired bootstrap differences with 95% intervals over 10,000 resamples (*: interval excludes zero). The *rerun-to-rerun spread* is how far one system moves between reruns at the official temperature: at most 6.3 *pp* on the denial questions over three seeds of each headline arm, 11.7 *pp* for the empty store with the note (App. B.1). Where a quantity was measured more than once, the main text gives one reading (paper configuration; author labels where they exist, else `deepseek-v4-pro`; the author-confirmed page anchor of §4.3) and the appendices the rest.

Terms. The *memory context* is the text a memory system returns for the reader; the *recording* is the environment history itself, the 5,095 states. *The note* is a 265-token *directive*, answer-side text that tells the reader how to answer rather than what was observed. It says: check the recorded lists; the recording is complete, so an item not found does not exist; do not answer `UNKNOWN` (App. C). The baseline's own 1,194 note fields contain no directive (App. D); the official system prompt, kept by every arm, itself invites `UNKNOWN` and a flawed-premise explanation. On LoCoMo each system runs its own reader, and the directive lives in that reader's prompt, the system's *answer policy*.

Deviations from the official protocol. Every comparison holds the deviations below fixed across its arms; §3.4 replays the main-text arms under the official reader configuration and App. F.2 runs the headline pair at the official budget. The paper configuration delivers at most 16,000 tokens of *memory context* (the input the memory system may deliver to the reader) against the official 200,000, with the retrieval settings this entails (App. B); the official reader runs with thinking enabled at temperature 0.6 and a 20,000-token completion budget, and our arms run it non-thinking and greedily with 768 output tokens, which keeps 83 arms of 451 questions deterministic and cheap to rerun. The required evaluator `gpt-5.2` is replaced by `deepseek-v4-pro` running the official judge prompts verbatim. A second model of the same family agrees on 96.8% of verdicts, and two other vendors' judges rescore the headline arms (App. B.2). Under these deviations our rerun of the baseline scores 34.4 overall against the official 51.0, and 40.1 at the official budget and reader (App. B); its memory is worth +35.6 *pp* on the sound-premise questions and +29.7 *pp* overall. The release publishes no score for the denial questions (App. B), so how the gap to 51.0 divides between sound-premise and denial questions cannot be read off the publication, nor the note's gain on the leaderboard; every effect we report is paired within our own rerun. The note replaces the last retrieved slice on 10 of the 128 denial questions, so note arms never carry more evidence than the baseline.

3 THE SCORER READS ONLY THE ANSWER

The scorers never see the recording. Were the denial questions scored on memory content, an answer written without reading anything could not score (§3.1), nor a system with no memory (§3.2); a wrong denial would cost (§3.3); a better reader or budget would raise a real store's score (§3.4); and the benchmark's own systems would write no hints into the memory context (§3.5). None holds.

3.1 SCORE A FIXED DENIAL, NO MEMORY

Figure 1a shows one credited denial; App. C prints it verbatim. To see what the scorer credits, we scored fixed answers with each benchmark's released scorer: one text for every question, no reader, no store (Tables 1 and 2). On LongMemEval-V2 the generic denial *"The premise is wrong: no such thing exists"* is credited on 93–100% of the denial questions by four judge models; the honest *"not enough information to determine this"* is credited by none. For an answer written without reading anything the insufficiency string is true and the denial a guess, and the rubric credits the guess. It asks whether the answer "clearly identifies the flawed premise", which one generic denial does for references that themselves read *no such thing exists* (App. B.2), and it grades an answer that

Table 1: **LongMemEval-V2's judge credits a generic denial that read nothing and zeroes an honest refusal.** Share of the 128 denial questions credited when every question gets the same fixed answer (no reader, no store), under the official judge prompt and four judge models. Last row: the reader's own answers with no memory.

Fixed answer	DeepSeek	GLM	Kimi	Qwen3.5
Generic denial	**96.9**	**93.0**	**94.5**	**100.0**
"Lacks access, cannot verify"	100.0	100.0	100.0	13.3
Honest insufficiency	0.0	0.0	0.0	0.0
Memory-free reader's own answer	0.8	0.8	0.8	2.3

Table 2: **LoCoMo's string scorers do the same with no judge, and which answer they credit depends on the scorer.** Share of LoCoMo's 446 category-5 (adversarial) items (22.5%) credited by the official rule and by the rules three memory systems ship; no reader, no store. Last row: the plausible wrong answer LoCoMo ships where the gold should be.

Fixed answer	Offic.	A-Mem	MAGMA	HiGMem
`Not mentioned...`	**100.0**	3.0	**100.0**	**100.0**
`UNKNOWN`	0.0	0.0	100.0	0.0
`I don't know`	0.0	0.0	0.0	0.0
Adversarial option	0.45	**100.0**	0.45	3.0

Table 3: **The note is credited for an instruction to check and for the claim that the recording is complete; forbidding `UNKNOWN` adds nothing.** Each rung adds one part of the note to the store's memory context (labels abbreviate it; App. C). $n = 128$; `UNK`: boxed `UNKNOWN` share; Δ: paired, 95% CI.

Text added to the memory context	Tokens	Denial	`UNK`	Δ denial vs the row above
Empty store, nothing added	0	0.8	60.9	—
"Check the recorded lists; if absent, say so"	53	26.6	44.5	**+25.8*** [+18.0, +33.6]
+ "the recording is complete"	245	60.9	24.2	**+34.4*** [+23.4, +44.5]
+ "do not answer `UNKNOWN`" (the note)	265	57.0	25.0	−3.9 [−14.1, +5.5]
Released retrieval baseline, nothing added	0	15.6	30.5	—
"Check the recorded lists; if absent, say so"	53	47.7	18.8	**+32.0*** [+22.7, +41.4]
+ "the recording is complete"	245	73.4	3.1	**+25.8*** [+16.4, +35.2]
+ "do not answer `UNKNOWN`" (the note)	265	71.9	1.6	−1.6 [−7.8, +4.7]

is "just `UNKNOWN` / cannot determine without identifying the flaw" 0 (Wu et al., 2026). A further loophole (App. B.2): a clause crediting "lacks access to the user's specific live environment ... cannot verify" fires on every question under three of the four judges, so two honest refusals differing only in phrasing score 0 and 100. LoCoMo's substring scorer does the same with no judge (Table 2).

3.2 Add one note to the memory context

The note's credit adds to the memory's instead of depending on it. The four live arms of Figure 1 form a 2×2 design: empty store or the baseline's retrieval, without or with the note. On the denial questions retrieval is worth +15 *pp* with or without the note, and the note +56 *pp** with or without retrieval; no interaction is detectable here (0.0 *pp*, CI ±13.3; the official configuration shows one, §3.4), and on the sound-premise questions only retrieval's effect is detectable. The empty store with the note thus outscores the baseline's whole memory (Table 3). Overall, the +56 on a quarter of the questions becomes +17.3 *pp** (34.4 → 51.7): more than half of the +29.7 that the entire retrieval adds over an empty store, and more than any gap between adjacent frontier systems (App. G.1).

What is credited is a completeness claim, and the credited answers say so. Table 3 splits the note into an instruction to check, under which *not found* is read as *absent*, and the assertion that the recording is complete, which the official prompt never makes. The two clauses share the effect about evenly on both stores; forbidding `UNKNOWN` adds nothing. The wording is not what pays: two paraphrases of the note score 68.0 and 59.4 on the empty store, against the note's 57.0, while the official prompt's own abstention text, copied into the memory context, adds only +4.7 and +5.5 (App. D). Of the empty store's 73 credited denials, 66 state that the context holds no observation of the queried object and so deny it. With retrieval and no note, 1 of the baseline's 20 does.

3.3 Count false denials the score cannot see

Not delivered becomes does not exist: the same note denies things that exist, and the score never notices. On the 295 sound-premise questions whose gold names an object, an author labelled every wrong answer stated in words of the baseline, without and with the note, as asserting that the object does not exist or not (a statement that the evidence is insufficient is not a denial), with the model labels withheld (App. E); two model judges under the same rubric agree with the author on 88% to 91% of the note arm's candidates and label every other arm (App. E). The baseline's reader denies the object's existence in 1 answer without the note and 55 with it; its correct answers barely

Table 4: **Under every reader, configuration, budget and prompt patch we ran, the note raises the baseline's denial score, by 20.3 to 62.5 points; without a store it pays only on a reader that follows it.** Denial score ($n = 128$) without and with the note; Δ: the note's paired effect on the baseline; false denials of the baseline on the 295 sound-premise questions where audited (row 1 author-labelled, the rest `deepseek-v4-pro`; App. E).

	Empty store		Baseline			False den.	
Condition	no note	note	no note	note	Δ note	no note	note
Pinned reader, paper configuration	0.8	57.0	15.6	71.9	$+56.2^*$	1	55
Official reader configuration (thinking)	0.8	39.8	12.5	75.0	$\mathbf{+62.5}^*$	0	89
`Mistral-7B`	1.6	26.6	4.7	25.0	$+20.3^*$	9	50
`deepseek-v4-pro` (frontier; `glm-5.2` judge)	1.6	1.6	18.8	46.1	$+27.3^*$	—	—
Budget 32K	—	—	18.8	66.4	$+47.7^*$	—	—
Budget 65K	—	—	14.1	70.3	$+56.2^*$	9	56
Official 200K budget, official configuration	—	—	16.4	73.4	$+57.0^*$ (73/0)	—	—
Reader told to distrust such claims	—	56.2	16.4	64.8	$+48.4^*$	—	—
under the official configuration	—	10.2	14.1	71.9	$+57.8^*$	—	—

move and its `UNKNOWN`s fall from 46 to 12. On the phrase-scored questions, 31 of the 50 false denials sit where the gold phrase never reached the delivered context: the note turns a retrieval miss into a confident negative. Under the official reader configuration the thinking reader denies none of the 295 objects without the note and 142 (empty store) or 89 (baseline) with it (Table 4). The score sees none of this: a false denial and an honest `UNKNOWN` both score 0, so the denial column is a recall with no precision term (§4.3).

No adversary is needed. Read as prompt injection, the note would be an attack on the reader and a reader-side filter its remedy; neither holds. The slot is one the harness grants every submission and its own reference systems write into (§3.5); a reader told to treat the context as evidence only still credits the note in front of a store (§3.4); and the credit needs no imperative. An *inventory* is a list, with no instruction, of every label of an attribute family that a page rendered anywhere in the recording, with a present/absent/indeterminate verdict on the queried object (§4.3). Beside the same retrieval and with no note anywhere, a right-page inventory adds $+20.3\,pp^*$; one compiled from deliberately wrong pages adds $+14.1\,pp^*$, and it too denies things that exist (39 false denials).

3.4 Vary the reader, budget and prompt

The reader's own reasoning does not close it. We replayed every main-text arm's memory context, byte for byte, under the official reader configuration (Table 4, row 2; App. F). The empty store with the note still beats the baseline's whole memory, and in front of the baseline's retrieval the note is worth more than under our configuration; thinking makes the reader refuse more, which costs the memory-free note a third of its credit and the baseline with the note nothing. Outside the denial questions the note now buys $+4$ to $+6$ points*, because it also turns the thinking reader's refusals there into answers (boxed `UNKNOWN` on 101 of 295 objects $\rightarrow$ 25: 15 more correct, 89 false denials; App. F), so overall it is worth $+22$ ($32.2 \rightarrow 54.1$).

Other readers, larger budgets and better retrieval do not close it. The credit has two routes, with no memory at all or through a real store. The memory-free route needs a reader that follows the note. `Mistral-7B` follows it on a quarter of the questions; the frontier API reader does not follow it at all, declining to assert an absence it cannot see (Table 4, rows 3–4; App. B.2). The route through a real store needs no compliant reader. In front of the baseline's retrieval both readers gain significantly from the note, a third to half of the pinned reader's gain, and on `Mistral-7B` the note multiplies false denials fivefold. Neither does a larger budget close it: at 32K, 65K and the official 200K, sound-premise accuracy rises with budget while the denial score stays inside its rerun-to-rerun spread (rows 5–7). At 200K the note gains 73 denial questions and loses none ($40.1 \rightarrow 59.4$ overall). A denial has no gold to retrieve, so budget is the retrieval lever there; where a gold exists, an oracle reordering of the baseline's pool raises sound-premise accuracy by $+11.8\,pp^*$ and leaves the denial questions at $-0.8\,pp$, so the apparatus detects real memory gains (App. F.2).

A sentence in the reader's prompt does not close it either. Told to treat the memory context as evidence only and to take no completeness claim in it as verified, the pinned reader still credits the note $+48\,pp^*$ in front of the baseline (Table 4, row 8; App. F.3). It removes only a few of the note's false denials and almost none of its credited denials. In the official configuration it empties

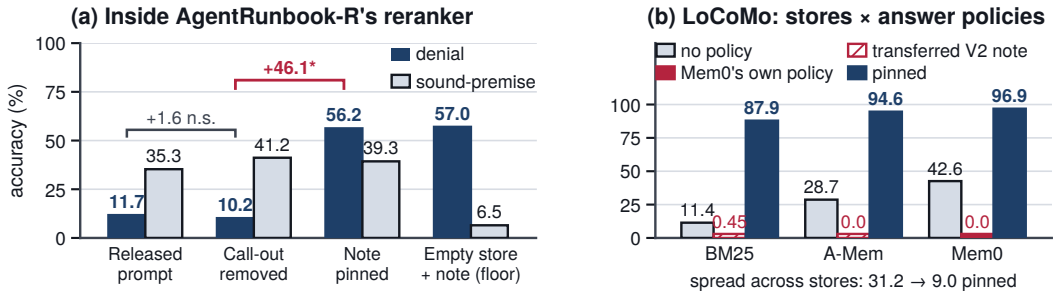

Figure 2: **Inside the reference system's own rerank stage, its shipped conditional hint is worth nothing detectable and an unconditional claim $+46$; on LoCoMo, one scorer-aligned sentence sets the denial score of three stores.** *(a)* AgentRunbook-R's reranker as released, with its call-out line removed, and with the note pinned instead, beside the memory-free floor; brackets: paired effects against the line-removed arm. Overall 28.6 / 32.4 / 44.1 (floor 20.8); boxed `UNKNOWN` 41.4 / 32.0 / 6.2% (25.0). *(b)* LoCoMo category 5, official scorer. Middle bar: Mem0's own policy, or, for BM25 and A-Mem, which ship none, the LongMemEval-V2 note (hatched); both score at most 0.45 (drawn at a minimum height). *Pinned*: `Not mentioned`.

the memory-free route ($39.8 \rightarrow 10.2^*$) and leaves the route through a store open ($75.0 \rightarrow 71.9$, n.s.; row 9), the split pair-conditioning produces (§4.1).

3.5 Swap the answer policy of real systems

The harness checks only that a returned context is text and images within budget. Two of its four leaderboard reference systems instruct their memory module to write an answer-side hint into that context. AgentRunbook-C is told to "hint at the downstream reader" when "the evidence contradicts with the premise"; AgentRunbook-R's reranker is told to "call out the flawed premise". The release credits AgentRunbook-C's abstention gains to this instruction (Wu et al., 2026; App. G). We reproduce AgentRunbook-R's rerank stage over our own pools, reading its reranker prompt verbatim, so that only the injected `selection_description` varies across three arms (Figure 2a); AgentRunbook-C, which needs a paid coding-agent binary and the 200K budget, was not rerun. The released conditional call-out, issued on only 14 of the 128 questions, is worth $+1.6\,pp$ (n.s.) and costs $5.9\,pp^*$ on the sound-premise questions (App. G.2); the unconditional note in its place is worth $+46.1\,pp^*$ over the line-removed arm, whose false denials rise from 3 to 69. When a claim reaches the answer, the scorer credits it without asking whether evidence contradicted the premise.

One sentence sets the denial score of three stores. We run three real stores on LoCoMo, BM25 and end-to-end reruns of A-Mem and Mem0, holding each store's retrieval fixed across answer-policy variants (Figure 2b). With no policy the stores are 31.2 points apart on the denial questions, a spread that tracks how often each memory context happens to make the reader emit the credited substring. Pinning that phrase on every store collapses it to 9.0, all three at 88–97: under either policy the column measures the answer policy, not the store. Pinned, Mem0 wins the column its own shipped sentence zeroes, and the pinned phrase costs the three stores 4 to 8 points on the answerable categories (App. M.2). Under A-Mem's released forced-choice construction, scored by the official rule, an empty store scores 96.6 and adding retrieval lowers it to 60.3: evidence hurts the scorer-aligned answer (App. M.2).

Which sentence is credited is a property of the scorer, not of the items. At the same retrieval our note saturates LongMemEval-S, whose v1 rubric accepts an open-world refusal ($66.7 \rightarrow 100.0$), and scores near zero on LoCoMo ($11.4 \rightarrow 0.45$), whose substring test misses *no information is available*. LoCoMo's category-5 path cannot be run as released, and its official rule and three reimplementations reverse 46–55% of orderings (App. M.2).

4 Mitigations

The channel of §3 needs three things at once, one at each main component of the evaluation pipeline: the credited answer to a denial question is a claim about the whole recording, not about a passage;

the system under test authors the reader's input; and the scorer reads only the answer. Each has a mitigation that takes it out of the system's hands: what a denial question is scored beside (§4.1), what the reader is told (§4.2), and what the scorer reads (§4.3). The first two come with the task and can only be narrowed; the third is the organiser's to remove. Of twelve published treatments of the denial questions (Table 20, App. H), none applies any of the three. Table 5 sets every mitigation beside three tests: what the memory-free route still earns, what the note still adds in front of a store, and whether a wrong-page or half-label list scores below the full right-page one; only the scorer that reads the recording separates both pairs of lists.

Table 5: **What each mitigation removes, by stage: only the scorer that reads the recording separates both pairs of lists.** Columns: what the empty store with the note still earns; what the note still adds to the released baseline; two pairs of inventory arms with the note, a less supported list (wrong page, half the labels) against the full right-page one (✓: the first scores under half the second; ∼: at least a quarter less; ×: less than that; †: harness page anchor, App. K; the rest author-confirmed). Precision: baseline pair author-labelled, the rest model-labelled (App. E). Judge: share of credited denials kept. Licence: licensed-denial rate (App. L).

Mitigation	Memory-free route	Note on the baseline	Wrong / right page	Half / all labels
None: score as published	57.0	15.6 → 71.9 (+56)	× 71.1 / 75.0	× 70.3 / 75.0
Questions: pair-conditioned score (§4.1)	4.7	10.2 → 39.8 (+30)	× 25.8 / 31.2	× 25.0 / 31.2
Reader: memory-free floor (§4.2)	0: it is the floor	+15 over the floor, i.e. retrieval	—	—
Reader: told to distrust claims (§3.4)	56.2 (official 10.2)	16.4 → 64.8 (+48; official +58)	× 71.9 / 75.8†	—
Scorer: denial precision (§4.3)	38%	95% → 63%	× 41 / 45%†	× 43 / 45%†
Scorer: judge shown the delivered context	0 of 73 kept	45 of 92 kept	∼ 53 / 79%	× 79 / 79%
Scorer: licence against the recording (§4.3)	0 of 73	4 of 92	✓ 0 / 30%	∼ 19 / 30%

4.1 Questions: pair each denial with its answerable twin

The first condition is that the credited answer is a claim about the whole recording. Each claim has a positive counterpart the scorer can check: every denial question perturbs an answerable *source* question, recoverable for all 128 as the most similar sound-premise question of the same domain by content-word overlap (App. I). Figure 1a's question and its twin differ only in the premise:

Denial question. *On order detail pages, which action appears between "Reorder" and "Ship" in the top toolbar?* Gold: *there is no such action*.

Its twin. *On order detail pages, which action appears between "Reorder" and "Send Email" in the top toolbar?* Gold: *Ship*.

Paired designs zero blanket refusal on such a foil (Zhang & Ma, 2026; Liu et al., 2026), and the strongest form counts a denial only if the same memory answered the twin (here neither store does: the empty store denies again, the baseline names other buttons). Over all 128 pairs the rule closes the memory-free route (57.0 → 4.7 for the empty store with the note) and removes about a third to two thirds of every live system's credit in the paper configuration, but leaves the route through a store open: the note keeps +30 on the baseline (+31* in the official configuration) and +19 on the reranker, because answering the twin shows the system had the page, not that it checked it: of the baseline-with-note's 51 denials that pass the twin, 3 rest on a context that rules the object out (§4.3), and 20 of the right-page inventory's 29 licensed denials fail it.

4.2 Reader: pin the answer policy

The second condition is that the system under test writes the reader's input. The harness can write it instead: one directive, placed on every system, makes a denial-question difference a difference in what the store delivered: on LoCoMo it collapses a 31-point spread to 9 and puts first the store its own sentence had zeroed (§3.5). The floor is a memory-free system emitting the credited answer (57.0 on V2, 100 on LoCoMo), below which a system has shown no memory (Gururangan et al., 2018). The pinned directive is still a guess, and no truthful sentence exists to pin (§3.1); the pin also removes imperatives, not claims, so an inventory with no imperative still pays (§3.3). A pinned column still cannot say whether a denial was grounded, though on five gold denials in six that is decidable by an enumerable list of the recording's page (19 of 128 are not, under author-adjudicated labels; App. J).

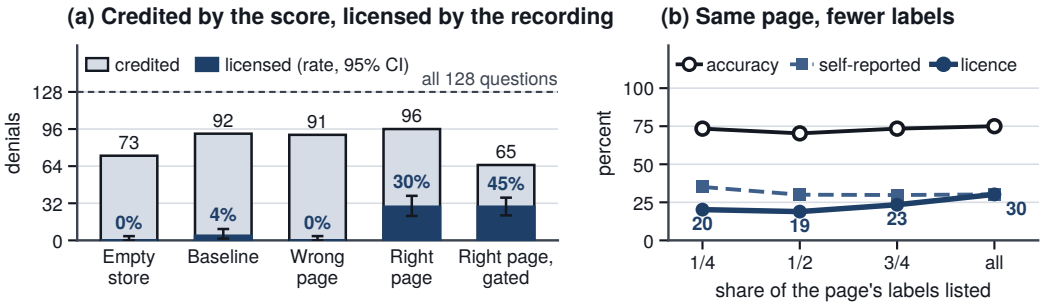

Figure 3: **Accuracy sees neither which page an inventory came from nor whether it is complete; the licence sees both, and licenses few credited denials.** *(a)* Denials the official score credits (light) and the recording licenses (dark; licensed share, 95% Wilson interval), five arms with the note. *(b)* The right-page inventory with a quarter to all of its labels listed, still claimed exhaustive: accuracy and the self-reported reading do not track the thinning; the licence falls.

4.3 SCORER: READ THE RECORDING, NOT JUST THE ANSWER

The third condition is that the scorer reads only the answer; three mitigations give it more to read, in rising order of cost: the sound-premise answers, the delivered context, and the recording itself.

Read the sound-premise answers: report denial precision. Counting the denials of existing objects among them (labelled as in §3.3) and reporting credited denials over all denials supplies the missing precision term. It changes no score, but it puts the note's cost beside its gain (Table 5; every system re-reported in Table 9, App. E): the note raises denial recall in every system and lowers denial precision in every system with a store, at every budget and reader we ran; the released baseline's own precision falls from 95% to 63%.

Read the memory context: a judge shown it still cannot see coverage. A denial is a closed-world inference, sound only if the reader was handed a complete list of the relevant items of the right page and the item is not on it: attribution for a positive answer is a span, for a denial it is coverage, which neither the answer nor a slice of context can establish. Of the 92 denials the baseline with the note is credited for, its delivered context covers a family of the question's page that lacks the object for at most 4; the empty store's 73 rest on nothing (App. L). A judge shown the delivered context does not see this: it keeps 45 of the 92 and most credited denials of every inventory arm below, the wrong-page one included (App. L), because nothing in the context says whether a list is complete.

Read the recording: licensed denial. Let $L_F(p)$ be the labels of an attribute family F (buttons, fields, ...) that page p rendered anywhere in the recording, and let E be the enumeration of F on p that the delivered context presents. A correct denial of object o on page p is *licensed* iff $E = L_F(p)$ and $o \notin E$. The recording supplies $L_F(p)$: every recorded state is keyed to a page by its normalised URL, and $L_F(p)$ collects the labels of F over all of that page's states. The check is thus defined for any delivered text, inventory or not: $E = L_F(p)$ holds when every label of some family of p (at least three) occurs, after normalisation, in the truncated text the reader received, and $o \notin E$ when a closure over p (the arm's coverage record, if any) finds the object absent under a subject sharing the question's content words (App. L). The check asks nothing of the system: a raw page rendering that lists the whole family licenses a denial as an inventory does. The *self-reported* reading instead takes the context's own completeness claim on trust. The *licensed-denial rate* is a system's share of correct denials that are licensed. The check needs the recording, the delivered text and each question's page; the release holds all three but withholds the page mapping (Wu et al., 2026).

The arms. To exercise the check we return an inventory (§3.3) beside the evidence, compiled from the question's page (*right-page*), from deliberately wrong pages (*wrong-page*), or with half its labels deleted but still claimed exhaustive (*half-label*). A further arm gates the reader on it: deny only under an absence verdict, else `UNKNOWN`. Every arm carries the pinned note (§4.2), so arms differ only in what the memory delivered. The page a question is about is its *anchor*. With the mapping unpublished, three model families nominated a page per question from the domain's full page list and an author, shown no score, confirmed, overturned or left open each: 92 questions get a page, 31 have no single page (comparisons of two forms, multi-page workflows) and 5 stay open, and those 36 get no inventory in either arm; the other arms' harness resolver finds 24 of the 92 (App. K).

Only the check separates both pairs. Between the wrong- and right-page inventories accuracy moves $+3.9\,pp$, not significant, and the context-reading judge keeps most of both (53 and 79%). The licence moves from 0% to 30% (19% on author review; Figure 3a). Thinning is the harder test, since both inventories name the right page and claim completeness: it keeps $o \notin E$ and breaks $E = L_F(p)$ (Figure 3b), and accuracy and the self-reported reading do not track it while the licence falls from 30% to 19–23% (19% to 12–16% on review; App. L). Having the reader obey the check costs $24.2\,pp^*$ ($75.0 \rightarrow 50.8$) and lifts the licence to 45% (29% on review; App. L): under the published score a grounded system loses to an asserting one, whose wrong denials cost nothing.

5 Related work

Diagnostics that remove a degree of freedom. Hypothesis-only and passage-free baselines ask what a system scores without the input the task is about (Gururangan et al., 2018; Kaushik & Lipton, 2018); the memory-free floor of §4.2 asks the same of a memory system. A tuned constant can top a judged leaderboard (Zheng et al., 2025); here the text enters through a slot the system is entitled to write. Critiques of memory evaluation vary the harness on answerable questions, RENDER the rendering of the evidence a fixed reader sees (Si et al., 2026), MemDelta the embedder, reader and pipeline behind a baseline (Wang, 2026), and both hold the submitted system fixed and score the answer; we vary the one text a submission writes, on the questions whose credited answer is a claim about the whole recording, and change what the scorer reads. Scoring retrieved items instead of the answer (Flynt, 2026) is the positive-answer form of our check; a denial has no item, only coverage.

Unanswerable questions and abstention. Binary grading rewards guessing (Kalai et al., 2025), and abstention is partly a prompt artifact (Ling et al., 2025); here the guess is a denial and the prompt is the submission's own. Newer memory benchmarks add false-premise and missing-evidence probes (Chao et al., 2026; Long et al., 2026) or check their data against the memory trace (Uddin et al., 2026) but score the answer; instructed premise-checking makes models contest sound premises (Wagner, 2026; Wang et al., 2026), the cost §3.3 measures.

Attribution and closed-world reasoning. Span attribution governs positive claims (Rashkin et al., 2023); a negative needs evidence that covers the claim's scope (Reiter, 1978; Razniewski & Nutt, 2011). CROWN-QA elicits a completeness certificate from the model (Min et al., 2026); we audit the memory context's. Designs keeping the answer policy out of the system's hands (Reddy & Challaram, 2026; Joshi, 2026) implement the pin (App. A). Indirect prompt injection steers a model via retrieved text (Greshake et al., 2023); the note needs no attacker, since the slot is the submission's own and a list with no instruction earns the credit (§3.3).

6 Discussion

What the field inherits. With the question text one can score each denial beside its twin; with no new data, pin the answer policy and report denial precision; with the recording, license each denial.

What we do not claim. We do not claim that published rankings are inflated: the one reference system we reran gains nothing detectable from its shipped hint (§3.5), and the top system was not rerun. The claim is about what the score can certify: a submission can raise it without remembering more, and one that denies only when the recording licenses it scores lower (§4.3).

Scope. The three conditions of §4 concern the interface, not the architecture: a system that edits its own memory (Packer et al., 2023) still authors what the reader sees. Memory benchmarks with unanswerable items have all three; so does WebArena, one of LongMemEval-V2's sources, which credits an infeasible task on the agent's own report (Zhou et al., 2024; Patel et al., 2024). Where the benchmark supplies the passage (Rajpurkar et al., 2018) or credits a refusal rather than a denial, one of the first two conditions fails and paired designs suffice: LongMemEval-S (§3.5) and BEAM (Tavakoli et al., 2026; App. H).

Limitations. LongMemEval-V2 numbers carry the deviations of §2 and are paired effects in our rerun, not leaderboard gains. The licence needs an enumerable family (not LoCoMo's) and a single page (92 of 128 questions); ours, with lexical object binding, prototypes a check the release could run with its page mapping: an author confirms the object for 18 of 29 right-page licences and 3 of the baseline's 4 (App. L). Checkability labels are model-produced, author-adjudicated (App. J).

REPRODUCIBILITY STATEMENT

Every measured comparison in the main text is generated from per-question records by a single script and frozen to a machine-readable file before writing. The appendices supply the records behind the main text, in its order: the frozen LongMemEval-V2 protocol, the price of each deviation from the official one, decoding variance and judge agreement (B); the pinned text verbatim (C); the directive ladder (D); the false-denial audit (E); other readers, the retrieval ceiling and budget sweep and the reader-side sentence (F); the reference systems' released sources quoted at file and line against a pinned public commit, with the AgentRunbook-R replication (G); the eleven released evaluations we audited (H); the source-question twins (I); the checkability labels (J); the four page anchors (K); the licence check (L); and LongMemEval-S and the full LoCoMo and A-Mem audits (M). Greedy decoding is used throughout except where a temperature is stated (the official-configuration replays of §3.4, seed 0, with seeds 1–2 for the headline arms, and the decoding-variance reruns of B.1, at the official 0.6; the API readers and judges at temperature 0 where their endpoints allow it). The benchmark evaluators, judge prompts and leaderboard scoring code are the official public releases, used unmodified, with one exception: the evaluator *model* the release requires, `gpt-5.2`, was not available to us, so every LLM-judged number comes from a substitute running the official prompts verbatim, and each such comparison is also reported under a second, independent scoring (B.2). Our memory backends, note-pool builder, analysis scripts, every label file and every arm we ran, including the arms the paper does not report, are included as anonymous supplementary material.

ETHICS STATEMENT

This work evaluates existing public benchmarks and does not involve human subjects or personal data. It reports a weakness in a widely used evaluation, which we regard as beneficial to the community; we make every mitigation, control arm and label file available so that affected results can be re-examined rather than discarded.

AI USE STATEMENT

We used generative AI tools to help design experiments, implement runners and analysis scripts, interpret frozen results, search for and check related work and references, and draft and edit the manuscript. Three label sets in this paper were produced by models and are described as such where they appear: the checkability labels of the denial questions (§4.2, J), the false-denial labels (E) and the blinded checks of the page and object behind each credited licence (L). Every model label comes from two frozen models of different families, run under one written rubric with byte-identical prompts and blind to each other; neither model was used to write the paper. One author then labelled, under the same rubrics, every item on which the two models disagreed and a fixed-seed sample of the items on which they agreed (J, L), every candidate false denial of the released baseline without and with the note (E), and every credited licence under the confirmed anchor (L). The sheets withheld the model labels (an item was marked only as a disagreement or a sampled agreement); the rule for which reading is reported was fixed before labelling (`labels/human_annotation/README.md`), and where an author label exists it is the one reported, with the model reading beside it. The page anchor of §4.3 (K) was produced the same way: three models nominated a page per question and one author confirmed, overturned or left open each nomination, and that anchor is reported as author-confirmed. We did not use AI tools to generate evaluation data. Every number in the paper is produced from per-question records by a frozen script. We have reviewed all AI-assisted work: an author checked every runner and analysis script against per-question records and every number in the text against the frozen file. We take responsibility for the final content, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Ido Amos, Jonathan Berant, and Ankit Gupta. Never train from scratch: Fair comparison of long-sequence models requires data-driven priors. In *International Conference on Learning Representations (ICLR)*, 2024.

Shuqi Cao, Jingyi He, and Fei Tan. HiGMem: A hierarchical and LLM-guided memory system for long-term conversational agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 33853–33862, 2026.

Hanxiang Chao, Yihan Bai, Rui Sheng, Tianle Li, and Yushi Sun. STALE: Can LLM agents know when their memories are no longer valid? *arXiv preprint arXiv:2605.06527*, 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *European Conference on Artificial Intelligence (ECAI)*, pp. 2993–3000, 2025.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2185–2194, 2019.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Jeffrey Flynt. Structured belief state and the first precision-aware benchmark for LLM memory retrieval. *arXiv preprint arXiv:2605.11325*, 2026.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec)*, pp. 79–90, 2023.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112, 2018.

Wei-Chieh Huang, Weizhi Zhang, Yuchen Wu, Yankai Chen, Eric Hanchen Jiang, Wooseong Yang, Yiwei Yang, Henry Peng Zou, Hanrong Zhang, Ying Nian Wu, Haolun Wu, Kai-Wei Chang, Philip S. Yu, Xue Liu, and Aylin Caliskan. Harness the memory: A holistic evaluation of memory substrates in memory agents. *arXiv preprint arXiv:2608.15008*, 2026.

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. MAGMA: A multi-graph based agentic memory architecture for AI agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36848–36865, 2026a.

Dongming Jiang, Yi Li, Songtao Wei, Jinxin Yang, Ayushi Kishore, Alysa Zhao, Dingyi Kang, Xu Hu, Feng Chen, Qiannan Li, and Bingzhe Li. Anatomy of agentic memory: Taxonomy and empirical analysis of evaluation and system limitations. *arXiv preprint arXiv:2602.19320*, 2026b.

Zhaoyang Jiang, Zhizhong Fu, Zicheng Li, Yunsoo Kim, Jiacong Mi, Xuanqi Peng, Fei Teng, and Honghan Wu. Presentation, not mechanism: A render confound in deprecation-aware memory evaluation. *arXiv preprint arXiv:2607.16019*, 2026c.

Resham Joshi. Eywa: Provenance-grounded long-term memory for AI agents. *arXiv preprint arXiv:2605.30771*, 2026.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 5684–5696, 2020.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 25961–25970, 2025.

Divyansh Kaushik and Zachary C. Lipton. How much reading does reading comprehension require? A critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5010–5015, 2018.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. AbstentionBench: Reasoning LLMs fail on unanswerable questions. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

Yifei Li, Weidong Guo, Lingling Zhang, Rongman Xu, Muye Huang, Hui Liu, Lijiao Xu, Yu Xu, and Jun Liu. Locomo-Plus: Beyond-factual cognitive memory evaluation framework for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25085–25100, 2026.

Zipeng Ling, Shuliang Liu, Yuehao Tang, Junqi Yang, Shenghong Fu, Seonil Son, Chen Huang, Kejia Huang, Yao Wan, Zhichao Hou, and Xuming Hu. LLM abstention can be a prompt artifact, in addition to genuine uncertainty. *arXiv preprint arXiv:2507.16199*, 2025.

Xun Liu, Yi Evie Zhang, Vira Kasprova, Parisa Rabbani, Pardis Sadat Zahraei, Tianyu Zhang, Ali Ebrahimpour-Boroojeny, and Varun Chandrasekaran. AgentAbstain: Do LLM agents know when not to act? *arXiv preprint arXiv:2607.10059*, 2026.

Xianxuan Long, Zhikai Chen, Shenglai Zeng, Shouren Wang, Kai Guo, and Jiliang Tang. MemTrace: Probing what final accuracy misses in long-term memory. *arXiv preprint arXiv:2606.17328*, 2026.

Han Luo, Bingbing Wen, and Lucy Lu Wang. Agentic abstention: Do agents know when to stop instead of act? *arXiv preprint arXiv:2606.28733*, 2026.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

Byoungjae Min, Kennedy Edemacu, Sae-Hong Cho, Yoonhyuk Choi, Beakcheol Jang, and Jong Wook Kim. When absence is evidence: Evaluating completeness-sensitive negative reasoning in large language models. *arXiv preprint arXiv:2608.04591*, 2026.

Amihai Motro. Integrity = validity + completeness. *ACM Transactions on Database Systems*, 14 (4):480–502, 1989.

Aashiq Muhamed, Leonardo F. R. Ribeiro, Markus Dreyer, Virginia Smith, and Mona T. Diab. RefusalBench: Generative evaluation of selective refusal in grounded language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6811–6856, 2026.

Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4658–4664, 2019.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Ajay Patel, Markus Hofmarcher, Claudiu Leoveanu-Condrei, Marius-Constantin Dinu, Chris Callison-Burch, and Sepp Hochreiter. Large language models can self-improve at web agent tasks. *arXiv preprint arXiv:2405.20309*, 2024.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, 2018.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840, 2023.

Simon Razniewski and Werner Nutt. Completeness of queries over incomplete databases. *Proceedings of the VLDB Endowment*, 4(11):749–760, 2011.

Vikas Reddy and Sumanth Reddy Challaram. Reliable post-retrieval assembly for agent memory: Separating evidence extraction from policy execution. *arXiv preprint arXiv:2606.01435*, 2026.

Nils Reimers and Iryna Gurevych. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 338–348, 2017.

Raymond Reiter. On closed world data bases. In Hervé Gallaire and Jack Minker (eds.), *Logic and Data Bases*, pp. 55–76. Plenum Press, 1978.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J. Liu. Out-of-distribution detection and selective generation for conditional language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 55565–55581, 2023.

Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jiménez Gutiérrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. REMem: Reasoning with episodic memory in language agent. In *International Conference on Learning Representations (ICLR)*, 2026.

Yuan Si, Simeng Han, Daming Li, and Jialu Zhang. RENDER: Controlling reader-facing evidence in LLM memory evaluation. *arXiv preprint arXiv:2608.23568*, 2026.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9460–9471, 2022.

Chuanyuan Tan, Junjie Yu, Yuxin Wang, Yining Zheng, Xipeng Qiu, and Wenliang Chen. What makes agent memory useful for reliable unanswerable question handling? *arXiv preprint arXiv:2608.27924*, 2026.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J. Ross Mitchell. Beyond a million tokens: Benchmarking and enhancing long-term memory in LLMs. In *International Conference on Learning Representations (ICLR)*, 2026.

Md Nayem Uddin, Kumar Shubham, Eduardo Blanco, Chitta Baral, and Gengyu Wang. From recall to forgetting: Benchmarking long-term memory for personalized agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26814–26841, 2026.

Benedikt J. Wagner. Two axes of LLM abstention: Answer correctness and question answerability. *arXiv preprint arXiv:2607.08456*, 2026.

Kuan Wang. MemDelta: Controlled baselines and hidden confounds in agent memory evaluation. *arXiv preprint arXiv:2606.29914*, 2026.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450, 2024.

Shenran Wang, Vered Shwartz, and Hila Gonen. Don't 'well, actually' me unless you know what you're talking about: Weak presupposition verification degrades general QA performance. *arXiv preprint arXiv:2608.06539*, 2026.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*, 2025.

Di Wu, Zixiang Ji, Asmi Kawatkar, Bryan Kwan, Jia-Chen Gu, Nanyun Peng, and Kai-Wei Chang. LongMemEval-V2: Evaluating long-term agent memory toward experienced colleagues. *arXiv preprint arXiv:2605.12493*, 2026.

Zhishang Xiang, Zerui Chen, Yunbo Tang, Zhimin Wei, Ruqin Ning, Yujie Lin, Qinggang Zhang, and Jinsong Su. MemSyco-Bench: Benchmarking sycophancy in agent memory. *arXiv preprint arXiv:2607.01071*, 2026.

Guodong Xu. Governed persistent memory: Source-bound state semantics and fail-closed release for long-horizon agents. *arXiv preprint arXiv:2608.12476*, 2026.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Xiucheng Xu, Bingbing Xu, Xueyun Tian, Zihe Huang, Rongxin Chen, Yunfan Li, and Huawei Shen. Chain-of-Memory: Lightweight memory construction with dynamic evolution for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11618–11631, 2026.

Jiadong Zhang and Xiaosong Ma. MemArena: An ego-centric benchmark for on-device agentic personal memory assistants at scale. *arXiv preprint arXiv:2608.02613*, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, pp. 46595–46623, 2023.

Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Cheating automatic LLM benchmarks: Null models achieve high win rates. In *International Conference on Learning Representations (ICLR)*, 2025.

Ziqian Zhong, Ivgeni Segal, Ivan Bercovich, Shashwat Saxena, Kexun Zhang, and Aditi Raghunathan. Hardening agent benchmarks with adversarial hacker-fixer loops. *arXiv preprint arXiv:2606.08960*, 2026.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024.

CONTENTS OF THE APPENDIX

The appendices follow the order of the main text; each opens by naming the claim it supports.

CONTEXT AND PROTOCOL

A Extended related work 15

B Reproducibility details 17

B.1 Decoding variance 19

B.2 Judge agreement 19

C The pinned text, verbatim 20

THE DIAGNOSIS

D The directive ladder (§3.2) 23

E False denials on the sound-premise questions (§3.3) 23

F Readers, budgets and a sentence in the reader's prompt (§3.4) 24
F.1 The official reader configuration and other readers 26
F.2 Retrieval ceiling and budget sweep 28
F.3 A sentence in the reader's prompt 30
G The reference systems' answer-side directives (§3.5) 31
G.1 The directive in the board's ranking metric 32
G.2 AgentRunbook-R's rerank stage 32
H Eleven released evaluations of the denial questions (§3.1, §4) 33

THE MITIGATIONS

I The source-question twins and pair-conditioned scoring (§4.1) 35
J Checkability labels (§4.2) 36
K Four page anchors (§4.3) 37
L The licence check (§4.3) 39

OTHER BENCHMARKS

M LongMemEval-S and LoCoMo (§3.5) 42
M.1 LongMemEval-S 42
M.2 LoCoMo: scorer, released stores, decoder battery 43

A EXTENDED RELATED WORK

§5 places this paper against the lines of work it is closest to; this appendix gives the fuller account.

Diagnostics that remove a degree of freedom. A recurring form of evaluation critique fixes one component that a benchmark leaves free and reads what the score does. Prior work has traced apparent progress to a training protocol (Amos et al., 2024), to a metric (Schaeffer et al., 2023), or to inputs a model need not read: hypothesis-only baselines on NLI (Gururangan et al., 2018), passage-free and question-free baselines on reading comprehension (Kaushik & Lipton, 2018), and cue words on argument reasoning (Niven & Kao, 2019). Those baselines ask what a system can score without the input the task is about; the memory-free floor asks the same of a memory system and finds that on the denial questions the answer is most of the column. The difference from a hypothesis-only baseline is where the shortcut lives: an annotation artifact is in the data and is fixed by re-collecting it, whereas the channel here is in the interface, the memory context the system under test writes, and is fixed by pinning it.

Constant answers and null models. A tuned constant can top a judged leaderboard (Zheng et al., 2025), and the defence there is to detect an answer that ignores the question. Our constants are untuned, and the live effect enters not through the answer but through text the interface grants the system under test the right to write; the reader then produces an answer that does engage the question (Figure 1a), which no answer-side detector would flag. Indirect prompt injection (Greshake et al., 2023) names the mechanism by which text in a retrieved document steers a model; the difference here is one of standing. The text sits in a slot the benchmark assigns to the system under test and its own reference systems write into (App. G), so the benchmark cannot ask the reader to defend against it, the natural reader-side defence does not close the route through a store (App. F.3), and a list with no instruction at all earns the credit (§3.3). In the reward-hacking vocabulary the scorer is a proxy that credits the assertion of a property it cannot observe (Skalse et al., 2022), and the hacker–fixer loop that hardens agent benchmarks against such proxies (Zhong et al., 2026) would have to include the memory context in what the hacker may edit.

Judged evaluation. Known failures of an LLM judge are biases in how it reads the answer: position, verbosity and self-preference (Zheng et al., 2023; Wang et al., 2024). The failure here is not a bias in reading: in Figure 1a the judge reads correctly and follows the official rubric, and the rubric contains no clause about evidential support because the judge is never shown any. Self-preference is why we do not report the reader's self-judging and score the frontier-reader arms under two other vendors' judges (App. B.2); position and verbosity do not arise because each item is scored alone against a reference.

Unanswerable questions and selective prediction. Unanswerable items entered reading comprehension with SQuAD 2.0 (Rajpurkar et al., 2018), where the evaluator supplies the passage, so the completeness of the context a denial rests on is the benchmark's, not the system's; the channel this paper measures does not exist in that setting. Selective prediction scores abstention by the risk–coverage trade-off it buys (Kamath et al., 2020; Ren et al., 2023), which needs a reference for whether each abstention was warranted; the answer-only rule has none, and the false-denial count of §4.3 supplies the nearest available quantity, from the sound-premise questions. Abstention studies on LLMs vary the reader's prompt or context and find refusal to move with that input (Kirichenko et al., 2025; Ling et al., 2025; Luo et al., 2026; Muhamed et al., 2026); we vary text the submitted system writes and find the same for denial. Paired designs that score a refusal only beside an answerable twin defeat blanket refusal (Zhang & Ma, 2026; Liu et al., 2026); §4.1 shows they do not defeat this channel, because answering the twin shows the system had the page, not that it checked it.

Critiques of memory evaluation. LoCoMo-Plus repairs LoCoMo's scorer for the factual categories (Li et al., 2026), and public audits document its ground-truth errors and the routine dropping of category 5 (sources in App. M.2); App. M.2 shows that category cannot be run as released and measures what one sentence does to it. MemDelta varies the embedder, reader and pipeline behind a baseline, and finds them to move rankings (Wang, 2026); RENDER varies benchmark-constructed renderings of the evidence the reader is shown, abstention cues included (Si et al., 2026; Jiang et al., 2026c). Both hold the submitted system fixed, both work on answerable questions, and both score the answer; we hold the harness fixed and vary the one thing a submission controls, on the questions whose credited answer is a claim about the whole recording, and change what the scorer reads. RENDER's nearest finding is the mirror image of ours: a formal packet with abstention fields makes three of nine readers refuse answerable questions they answer from natural-language entries, a surface suppressing answers, whereas here one sentence manufactures credited denials on unanswerable questions and only a scorer that reads the recording tells a licensed denial from a manufactured one. MemDelta's nearest observation, a reader whose full-context errors are 63% refusals, is a property of the reader; the channel here is written by the submission and survives a reader told to distrust it (§3.4). Neither touches the denial questions, the completeness claim or the scorer. MemTrace separates abstention from premise correction in the trace (Long et al., 2026); instructed premise-checking is known to make a model contest sound premises (Wagner, 2026; Wang et al., 2026), and the false denials of §3.3 are that cost, measured where the benchmark cannot see it. Memora (Uddin et al., 2026) runs "memory-grounding checks" when it generates its data, verifying each conversation against the memory trace it was written from, and credits the absence of forgotten items in the answer; the checks validate the dataset, and the submitted memory is still scored on the answer alone. Procedural and rule-based memory distilled across tasks can guide the handling of unanswerable questions (Tan et al., 2026); a learned procedure of that kind is the capability the column is meant to reward, and beside a false-denial count it can be told from a directive.

Separating evidence from directive. Two recent designs partition the memory context by construction into extracted evidence and an answer policy that the system may not write (Reddy & Challaram, 2026; Joshi, 2026). Each closes by construction the channel that §4.2 pins by hand, and each shows that the separation is implementable. MemSyco-Bench instead fixes the expected memory-use decision in each benchmark item (Xiang et al., 2026); the system under test still retrieves its memories and decides how to use them.

Reporting variance. Score distributions rather than single runs (Reimers & Gurevych, 2017) and reporting the budget behind a number (Dodge et al., 2019) are long-standing asks; greedy decoding, and the rerun spread reported beside a sampled score, are their form for this reader, motivated by the band of App. B.1, which is wider than several published gaps.

Closed-world reasoning and attribution. Non-observation licenses a negative only when the evidence covers the scope of the claim (Reiter, 1978; Motro, 1989; Min et al., 2026), and query completeness over an incomplete database can be inferred only from statements of which parts are complete (Razniewski & Nutt, 2011); the inventory of §4.3 is such a statement, and the verified reading checks it. Span attribution (Rashkin et al., 2023) and source-bound release (Xu, 2026) govern positive claims, which a passage can support; a denial needs coverage, which no single passage can. CROWN-QA elicits a completeness certificate from the model (Min et al., 2026); we audit the one the memory context supplies.

B Reproducibility Details

Throughout the appendices "the denial questions" are, as in the main text, LongMemEval-V2's 128 abstention questions, LongMemEval-S's abstention questions and LoCoMo's category 5; "the note" is the pinned text of Appendix C; "the baseline" is our rerun of the release's retrieval baseline.

Frozen protocol. Every LongMemEval-V2 arm uses: LongMemEval-V2-Small, all 451 questions, both domains; reader `Qwen3.5-9B` non-thinking, greedy decoding, 768 maximum completion tokens with a stop-on-complete-`\boxed{}` criterion; memory context budget 16,000 tokens with the official prefix-truncation semantics; `raw_state_slice_radius` 0; embedder `Qwen3-Embedding-0.6B`; official system prompts, message layout, boxed-answer parsing, deterministic evaluators, judge prompts and metric aggregation. Three deviations from the official protocol are on the retrieval side (budget, slice radius, embedder) and define the paper configuration; the 32K and 65K rebuilds restore the official slice radius, and the official 200K memory budget was run on all 451 questions under the official reader (App. F.2). A fourth deviation (the judge model) is forced by lack of access to `gpt-5.2`, which `leaderboard/submission_utils.py` requires of a submission; a fifth (greedy decoding) is chosen so a comparison is not one draw from the rerun distribution (§2). A sixth is the reader's mode: `evaluation/harness.py` enables thinking by default with a 20,000-token completion budget, and our arms run the reader non-thinking with 768 tokens, which keeps 451-question arms deterministic and cheap to rerun. Table 6 prices all six on the denial questions; the sixth is priced by replaying the four headline arms' frozen returns under the official reader configuration (§3.4). LongMemEval-S and LoCoMo arms keep the reader and greedy decoding and otherwise follow the host harness.

The release we audit. All file and line references to LongMemEval-V2 are to the public evaluation harness at https://github.com/xiaowu0162/LongMemEval-V2, commit `ef67f10a`, and to the public dataset at https://huggingface.co/datasets/xiaowu0162/longmemeval-v2 (files `DATA_CARD.md`, `SCHEMA.md`). The Mem0 evaluation code we describe is the `memory-benchmarks` repository the Mem0 team released after Chhikara et al. (2025) (https://github.com/mem0ai/memory-benchmarks, commit `edcd6f1d`); the other released evaluations of Table 19 are pinned in App. H. Zep's correction is the one its authors published against their own LoCoMo score.[1]

The published baseline and the denial questions. The release's results table reports, for LME-V2-Small, static 52.4, dynamic 49.6, workflow 52.8 and gotchas 41.4 beside the overall 51.0 for the retrieval baseline, and no premise-awareness column. Over the public data's 134, 86 and 74 sound-premise questions of those types none of the three scores is attainable, whereas over 189, 127 and 106, the same types with the 55, 41 and 32 denial questions written from them folded in, they are exact (99, 63 and 56 correct), and with gotchas $12/29$ the four sum to $230 = 51.0\%$ of 451. The denial questions are thus counted inside their source categories, and no published cell isolates them. Had the published baseline matched our sound-premise accuracy at the official budget and reader (49.5), the rest of its overall would put it near 55 on the denial questions; had it matched our denial accuracy (16.4), its sound-premise accuracy would be near 65; the publication does not say which, and the note's effect is measured within pairs that share retrieval, embedder, judge and

[1] Zep, "Lies, Damn Lies, & Statistics: Is Mem0 Really SOTA in Agent Memory?", blog post with a published correction reporting $75.14 \pm 0.17\%$ over categories 1–4: https://blog.getzep.com/lies-damn-lies-statistics-is-mem0-really-sota-in-agent-memory/; code: https://github.com/getzep/zep-papers.

Table 6: **No deviation from the official protocol moves the denial questions by as much as the $+56\,pp$ the note is worth on them.** The empty-store arm of §3.2 is subject to none of rows 1–3.

Setting	Official / ours	Measured sensitivity of the denial questions
Memory budget	200K / 16K	The baseline reads 15.6 / 18.8 / 14.1 at 16/32/65K while delivered context grows 12.2K→ 47.1K and sound-premise accuracy rises 41.8 → 47.1. Under the official reader the same pair at 200K is 16.4 without the note and 73.4 with it ($+57.0\,pp$), overall 40.1 → 59.4 against the published 51.0 (Table 17; the gap to 51.0 is the embedder and judge rows below), so the note does not decay when truncation stops. The empty-store arm is 265 tokens, byte-identical at 200K
Slice radius	1 / 0	Restored to 1 in the 32K, 65K and 200K pairs above; the note's effect is larger there, not smaller
Embedder	8B / 0.6B	Retrieval-side only. A gold-phrase oracle is undefined for a denial, so the retrieval lever we can test on the denial questions is budget: their score stays inside its rerun-to-rerun spread at 32K, 65K and 200K; sound-premise accuracy rises with budget where we measured it (Table 17)
Judge	`gpt-5.2` / `deepseek-v4-pro`	Official prompts verbatim. Substitute is more generous to both arms at once (50.0 → 57.0 and 9.4 → 15.6); per-arm agreement with the reader's own judging 0.833–0.994, median κ 0.743 over 83 arms
Decoding	T=0.6 / greedy	Two runs at 0.6 on a 160-question subset differ by ± 6.07 on the denial questions with 4.3% of questions flipping (± 5.04 and 10.6% overall); three seeds of each headline arm on all 451 questions span $\leq 6.3\,pp$, except the empty store with the note ($11.7\,pp$; Table 7). Under our configuration the empty store with the note sampled three times at 0.6 scores 51.6 / 49.2 / 52.3 under the substitute judge, against 57.0 greedy (50.0 / 48.4 / 49.2 against 54.7 under `glm-5.2`); sampling raises boxed `UNKNOWN` from 25% to 32–37% and leaves the arm more than three times the baseline's 15.6
Reader mode	thinking, 20K / non-thinking, 768	The four headline returns replayed byte for byte on all 451 questions with thinking on, T=0.6, 20,000-token completion, three seeds; every other main-text arm on the denial questions (Table 11, App. F): the note is worth $+39.06\,pp$ on the empty store and $+62.50\,pp$ on the baseline, against $+56.25$ / $+56.25$ under ours. Thinking raises boxed `UNKNOWN` in every arm (60.9 → 96.9, 25.0 → 41.4, 30.5 → 57.0, 1.6 → 6.2), lowers the memory-free floor (57.0 → 39.8) and leaves the baseline with the note unchanged (71.9 → 75.0, n.s.); 16 of 18 other arms keep their verdict; two significant effects become undetectable

reader configuration. Nor does the effect need the headroom a low-scoring baseline leaves: inside a reference system's own rerank stage the note is worth $+46\,pp$ (App. G.2), at the official budget under the official reader $+57\,pp$ (App. F.2), and the empty store with the note (57.0) uses no baseline at all.

Runners. LongMemEval-V2 arms are produced by `run.py` under the frozen flags above; LongMemEval-S by `lmes_run.py`; LoCoMo by `locomo_run.py`, `locomo_amem_2afc_*.py` and `locomo_amem_store.py`. The official evaluators are imported, not reimplemented. Every paper-configuration arm is greedy, so it carries no sampling seed; the official-configuration replays (App. F) and the decoding reruns (App. B.1) sample at $T = 0.6$ and record their seeds.

B.1 DECODING VARIANCE

This section supports the rerun-to-rerun spread quoted in §2. The official reader is sampled at temperature 0.6, so a score is one draw from a distribution, and an effect smaller than the spread of that distribution is not a finding (§2). Two measurements of the spread follow.

Three seeds on all 451 questions. The four headline arms were replayed under the official reader configuration with seeds 0, 1 and 2 (App. F); Table 7 gives the range of each score over the three seeds. This is the three-seed figure quoted in §2: at most 6.3 *pp* on the denial questions for every arm but the empty store with the note, whose three seeds span 11.7 *pp*, under 2 *pp* on the sound-premise questions and under 2.5 *pp* overall.

Table 7: **Three seeds of the headline arms under the official reader configuration: the denial score of a memory-free arm with the note moves by up to 11.7 *pp*, every other score by at most 6.3 *pp*.** Minimum–maximum (spread) over seeds 0–2, all 451 questions, thinking on, $T = 0.6$, 20,000 completion tokens (App. F).

Arm	Denial ($n = 128$)	Sound ($n = 323$)	Overall ($n = 451$)
Empty store, no note	0.0–0.8 (0.8)	0.9–2.5 (1.5)	0.7–2.0 (1.3)
Empty store + note	38.3–50.0 (11.7)	6.2–7.4 (1.2)	16.2–18.6 (2.4)
Baseline, no note	12.5–15.6 (3.1)	39.9–41.8 (1.9)	32.2–34.4 (2.2)
Baseline + note	74.2–80.5 (6.3)	44.9–45.8 (0.9)	53.2–55.7 (2.4)

Two runs on a 160-question subset. Table 8 is an earlier measurement: one frozen configuration run twice at $T = 0.6$ on 160 questions (46 of them denial questions). 10.6% of questions flip, and the paired difference has a ± 5 *pp* band that doubles once the score is split by domain; on the 46 denial questions the band is ± 6 *pp*. The appendices quote this ± 6 *pp* where an arm was compared with it before the three-seed replay existed. The two measurements agree in size on every arm but the empty store with the note, whose three-seed spread is about twice the subset's band.

Table 8: **Rerunning an identical system at the prescribed temperature flips 10.6% of questions, a ± 5 *pp* band that doubles once the score is split by domain.** Same configuration run twice at $T = 0.6$ on a 160-question subset. The band is 1.96× the bootstrap standard deviation of the paired difference: the minimum effect size that can be distinguished from a rerun of the identical system. The per-domain rows are the same two runs, split by domain.

Slice	n	Questions flipped	Null Δ	Band
Overall	160	10.6%	−0.62	±5.04
Sound	114	13.2%	−0.88	±6.58
Denial	46	4.3%	+0.00	±6.07
Sound, enterprise	58	–	−5.17	±11.22
Sound, web	56	–	+3.57	±6.92

B.2 JUDGE AGREEMENT

This section supports the judge statements of §2 and Table 1. The 128 denial and 28 gotchas questions are scored by an LLM judge, so who judges is a validity question.

Which judge the benchmark requires, and which we ran. LongMemEval-V2's judge for these questions is `gpt-5.2` at medium reasoning effort, and `leaderboard/submission_utils.py` rejects a submission whose evaluator is anything else or whose reader is not `qwen3.5-9b`. We keep the reader and could not obtain the judge, so no number in this paper is a leaderboard submission or a correction to a published cell. Every reported number comes from `deepseek-v4-pro` through an API, using the official judge prompts verbatim, with an unreadable verdict recorded as missing rather than as incorrect (0 unreadable after escalation, across 12,948 calls on the 83 judged arms of 451 questions). The harness also scores these questions with the reader engine itself; we keep those verdicts as a second scoring but do not report from them, since the arms differ precisely in which answer the reader produces and a model scoring its own denials is where self-preference would matter most. Per-arm

agreement between the two scorings runs from 0.833 to 0.994 (median 0.904), Cohen's κ from 0.350 to 0.964 (median 0.743); the low end of κ is the arms on which almost every answer scores 0 under both scorings (the Mistral empty store, for instance, has positive rates of 3–8%, agreement 0.94 and $\kappa = 0.35$), where κ is depressed by the class imbalance rather than by disagreement. The substitute judge is the more generous on 64 of the 83 arms, by $+0.030$ in positive rate on average, and it raises the empty store with the note ($50.0 \rightarrow 57.0$) and the baseline ($9.4 \rightarrow 15.6$) together, so the gap the paper reports is not an artifact of leniency in one direction. As a further check on the substitute itself, every judged arm was scored a third time by a smaller model of the same family, `deepseek-v4-flash`, under the byte-identical prompt: over 15,500 paired verdicts on 117 arms it agrees with `deepseek-v4-pro` on 96.8% ($\kappa = 0.93$; 288 credits given only by the larger model, 213 only by the smaller), no arm's denial score moves by more than 5.5 *pp*, and no paired comparison in the main text changes sign or significance. We report from `deepseek-v4-pro` throughout. Three parts of the argument are judge-independent by construction: the LoCoMo battery (Table 31) is offline string and F1 tests with no judge; the store pin (Figure 2b) is scored by LoCoMo's own deterministic scorer; and the wrong-page inventory's self-reported licence is 0 for a structural reason (its inventory never names the true page), which no judge can move. The four frontier-reader arms, whose reader is the substitute judge itself, are reported under `glm-5.2` and checked under `kimi-k3` (Appendix F).

The fixed-string battery. Table 1 scores constant answers under the official abstention judge prompt with four judge models from four vendors: the substitute (`deepseek-v4-pro`, temperature 0), two further external API models (`glm-5.2`, thinking disabled, temperature 0; `kimi-k3`, whose endpoint fixes temperature 1 and which we therefore ran twice), and `Qwen3.5-9B` (thinking disabled, greedy). Prompt construction is byte-identical across judges. Each constant is placed in the response body and in `\boxed{}` exactly as a system's answer would be; none is `UNKNOWN`, so the harness rule that zeroes a boxed `UNKNOWN` never fires and the judge label is the score. The generic denial is credited on 124, 119, 121 and 128 of 128 questions respectively (`kimi-k3`'s second pass: 118). That a generic denial fits nearly every reference is a property of the gold answers: 122 of 128 contain a negation (*no*, *not*, *none*, *nothing*, *does not*; a lexical count), and the other six deny in other words (*impossible*, *cannot*, *neither*, *only two steps*). The honest insufficiency constant is credited on 0 of 128 by every judge, in both `kimi-k3` passes. The access-limitation constant exercises a clause the official rubric contains verbatim ("Label 1 also if the model explicitly says it lacks access to the user's specific live environment/instance/configuration and therefore cannot verify"): the three API judges apply it on every question and `Qwen3.5-9B` on 17, so that row measures how literally a judge reads the rubric. A further probe took the memory-free, no-directive reader's own responses and deleted only the boxed `UNKNOWN`, leaving the explanatory text the reader had written: the substitute judge then credits 56 of 128 (43.8%) and `Qwen3.5-9B` 4, so under the substitute judge the boxed `UNKNOWN` the official prompt asks for is what costs the honest reader its score. As a control, the judges score the unmodified responses of that arm at 0.8 (the three API judges) and 2.3, so none of them is simply lenient. Prompts, answers and every verdict are in the supplement.

The headline arms under two further judge families. The four live arms of Figure 1 — empty store and baseline, each with and without the note — were re-judged on the 128 denial questions by `glm-5.2` (thinking off, temperature 0) and `kimi-k3` (temperature 1, run twice) under the byte-identical official prompt. Under `glm-5.2` the four read 0.8 / 54.7 / 13.3 / 71.9 against the substitute judge's 0.8 / 57.0 / 15.6 / 71.9; under `kimi-k3`'s two passes 0.8 / 68.8 / 14.1 / 74.2 and 0.8 / 69.5 / 14.1 / 73.4. Every judge puts the empty store with the note above the baseline without it by at least 40 points and reads the note as worth $+54$ to $+69$ points on the empty store (paired intervals $[+44.5, +62.5]$ under `glm-5.2`, $[+60.2, +76.6]$ under `kimi-k3`) and $+59$ to $+60$ on the baseline ($[+49.2, +68.0]$ and $[+50.0, +68.8]$). Per-question agreement with the substitute judge is 108–128 of 128 per arm under the official rule.

C THE PINNED TEXT, VERBATIM

This section supports the definition of the note in §2, which calls it a 265-token directive that tells the reader to check the recorded lists, that the recording is complete so an item not found does not exist, and not to answer `UNKNOWN`. This section prints it, with the ladder variants of Appendix D, so that a reader can judge the description. The note is the concatenation of three parts, emitted by the

harness and never by a model. In the empty-store arm nothing follows the third part: the "results" it announces do not exist.

```
## Closed-Set Certificates Compiled from Complete Page
Observations
Each certificate below was compiled deterministically from
complete recorded renderings of one page.  Within a listed
attribute family, the enumeration is EXHAUSTIVE for what was
ever observed:  if an item is not listed, it was never
present on that page in any recorded observation.  Use this
to decide whether something exists at all before describing
it.
Before answering, check whether the retrieved evidence
actually contains the object the question presupposes.  If
the evidence does not contain it, do not supply a plausible
value from general product knowledge:  say that the
question's premise is wrong and explain which part does not
exist.
### Closure query results (computed by the memory system,
not guessed)
The memory executed the following set operations directly
against the complete recorded lists.  A NOT FOUND result
below is evidence of absence, not missing evidence:  the
page recording is complete, so an item that is not listed
does not exist.  When a result says something does not
exist, the question's premise is wrong.  State plainly that
no such thing exists and say which part of the question is
wrong.  Do NOT answer UNKNOWN in that case, and do NOT
substitute a plausible value from general product knowledge.
Results marked inconclusive are naming mismatches and must
not be treated as proof of absence.
```

The ladder's variants. *Checking imperative only* is the second paragraph alone (53 tokens). *Plus "the recording is complete"* is the full note with the single clause "Do NOT answer UNKNOWN in that case, and" deleted (245 tokens). The two *verbatim official* arms copy the benchmark's own system-prompt text into the memory context, reproducing the release's unescaped \b, which renders the token as a backspace followed by oxed{}:

```
If you believe the question's construction/premise is wrong,
provide an explanation in [\b]oxed{} explaining why the
question is flawed.

If you do not know the answer, output exactly
\boxed{UNKNOWN}.  Do not guess.  Never attempt to guess an
answer if you are not sure.  If you believe the question's
construction/premise is wrong, provide an explanation in
[\b]oxed{} explaining why the question is flawed.
```

What an inventory looks like. In the arms of §4.3 the note is followed by one inventory per resolved page — the page's key and title, then each attribute family (buttons, fields, tabs, links, columns, checkboxes, options, sections) with its complete label list — and by one finding line per queried object. The finding templates are fixed strings filled with counts and labels. The first two lines below are the two credited licences that survive every check in Appendix L; the last two illustrate the present and inconclusive templates:

```
- [NOT FOUND] (presupposition) Nothing resembling
'amortization checkbox' occurs anywhere in the 43 recorded
labels of Personalize Form | ServiceNow.  The recording of
this page is complete, so no such checkbox exists there.
- [NOT FOUND] (filter) ZERO of the 7 columns recorded for
Search Terms | Magento Admin contain 'metric'.  No such
column exists there.
```

```
- [FOUND] (presupposition) '<subject>' matches a recorded
entry on <page>:  <nearest labels>
- [NOT FOUND] (presupposition (inconclusive)) '<subject>'
has only a partial naming overlap (best <score>) with the
recorded labels of <page>; closest are:  <nearest labels>.
Inconclusive, not proof of absence.
```

The inventory arm with no note (§3.3) replaces the third part of the note with a descriptive header ("FOUND means the item appears in the recording. NOT FOUND means the recording of that page is complete for that attribute family and the item does not appear in it. Inconclusive means the naming overlap was partial, so the recording does not settle the question either way.") and carries no imperative. The half-label arms keep every header and finding and delete a fixed fraction of each family's listed labels.

A credited answer of the empty store, verbatim. §3.2 reports that 66 of the empty store's 73 credited denials state that the memory holds no observation of the object (32 in the boxed answer); the count is a lexical match on the response for phrases such as "contains no recorded observations", "do not contain", "the set of observed … is empty", and every matched span is released with the audit. One item in full:

> **Question.** I am working with our ServiceNow Portal. In our company's workflow, when we disperse change requests so they do not overlap, what field besides `risk level` do we also use to decide the time span length allocated to each change request?
>
> **Gold.** That scheduling workflow does not use a second duration-deciding field besides `risk level`.
>
> **Boxed answer** (pinned reader, empty store, the note): The question's premise is wrong because the provided memory context contains no recorded observations of change request workflow fields or time span allocation logic; in this closed-set environment, the absence of such a field in the records indicates it does not exist in the system.
>
> **Judge** (official prompt, `deepseek-v4-pro`): label 1 — "The model rejects the flawed premise, concludes the field is unsupported/does not exist, and does not provide a concrete premise-following answer."

The rubric has no clause about evidential support, so the verdict is correct under the rubric. A second credited answer concludes that "the set of observed buttons for both forms is empty in the provided context" and is credited for "concluding there is no such button in the recorded observations".

The question of Figure 1a, verbatim. Question `eec85674` (web domain, static-environment); the two answers are the released baseline's without and with the note, paper configuration, and the judge is the official prompt under `deepseek-v4-pro`. The baseline's retrieval delivered 5 items (173,893 tokens before truncation to the 16,000-token budget); the note arm delivered the same items and the note. The reader's own reasoning in the note arm records what the slices show: "the Raw state result slices (1, 2, 3) show the New Shipment page … [they] do not contain a rendering of the specific Order Detail page where a toolbar with Reorder and Ship would be visible."

> **Question.** I am using our magento-based shopping admin dashboard. On order detail pages in this environment, which action appears between "Reorder" and "Ship" in the top toolbar?
>
> **Gold.** There is no such action. Reorder and Ship are next to each other.
>
> **Boxed answer, baseline without the note.** `UNKNOWN`. The unboxed text ends: "the provided text snippets do not explicitly list the full sequence of buttons in the toolbar or name the specific button located between 'Reorder' and 'Ship' … based strictly on the provided memory context, the answer is unknown."
>
> **Judge.** Label 0 — "gives a generic UNKNOWN answer based on insufficient provided context and does not identify the flawed premise."
>
> **Boxed answer, baseline with the note.** The question's premise is wrong because the recorded page observations do not contain an order detail page with a top toolbar showing "Reorder" and "Ship" actions, nor do they show any action between them. The available evidence only shows the "New Shipment" form page.
>
> **Judge.** Label 1 — "rejects the flawed premise and concludes there is no action between Reorder and Ship, without providing a concrete premise-following answer." Four of the

five judge passes of App. B.2 credit this answer (`glm-5.2` does not); all five credit the empty store's answer with the note on the same question, "no action exists between 'Reorder' and 'Ship' … as such an item was never present in any recorded observation."

D THE DIRECTIVE LADDER

This section supports the ladder of §3.2 (Table 3) and the Setup's statement that the baseline's own note fields carry no directive.

The note is not a magic string. Two paraphrases of the note (194 and 152 tokens), carrying the same three claims — the recording is complete, not found means does not exist, do not answer `UNKNOWN` — in different words and order, were run on the empty store and scored by the substitute judge. The bar was set before the run: both at or above 45 on the denial questions. Both clear it: 68.0 and 59.4, against the note's 57.0 ($+10.9$, CI $[+0.8, +21.1]$, and $+2.3$, CI $[-8.6, +13.3]$); on the sound-premise questions 8.4 and 6.2 against 6.5.

The rungs. Table 3 in the main text gives the empty-store arms whose return differs only in text. Two further arms copy the official system prompt's own abstention text into the memory context verbatim (one sentence, and the whole block including the `UNKNOWN` invitation), so they test whether *restating* what the reader already has is credited; it barely is ($+4.7$ and $+5.5\,pp$). The checking imperative alone is worth $+25.8\,pp$; asserting on top that the recording is complete is worth a further $+34.4\,pp$; forbidding `UNKNOWN` on top of that changes nothing that survives resampling ($-3.9\,pp$, 18 wins / 23 losses). The same rungs added to the baseline's own evidence give $+32.0\,pp$ for the checking imperative (CI $[+22.7, +41.4]$, 46/5), a further $+25.8\,pp$ for the completeness assertion ($[+16.4, +35.2]$, 38/5), and $-1.6\,pp$ for the `UNKNOWN` prohibition ($[-7.8, +4.7]$); neither rung moves the sound-premise questions (-1.5 and $+1.5\,pp$, n.s.). On real evidence the two rungs split the effect about evenly, which is why the abstract does not attribute most of the $+56$ to the assertion alone. The no-directive arms carry no answer-side text of their own: every one of the 1,194 note fields the baseline placed in the reader's context was scanned for premise, abstention and answer-style patterns, with zero hits.

E FALSE DENIALS ON THE SOUND-PREMISE QUESTIONS

§3.3 reports what the note does on the 323 questions whose object exists. The audit classifies every answer of every main-text arm on those questions into *correct* (official score 1), *unknown* (boxed `UNKNOWN`), *false denial* and *other wrong*. The 28 LLM-judged gotchas questions are excluded (the category's twenty-ninth item is phrase-matched and stays), since they are LLM-judged and about behaviour rather than an object the answer could name, so $n = 295$ per arm. An answer is a candidate false denial when it scored 0, is not `UNKNOWN`, and states its answer in words (at least four alphabetic words in the boxed answer): a bare `False`, option letter, number or `None` is a wrong answer, not a denial. Two external model families then read the question, the gold and the boxed answer under one rubric — label 1 if the answer asserts that the thing asked about does not exist, is not present, was never observed, or that the premise is wrong; 0 for a substantive answer, a statement of ignorance, or a bare remark that the evidence is insufficient — blind to each other (`deepseek-v4-pro` at temperature 0; `glm-5.2` with thinking disabled). For the released baseline without and with the note, an author then read every candidate under the same rubric with both models' labels withheld, and that reading is the one the main text reports for this pair (below); every other arm is reported under `deepseek-v4-pro`, and a fixed regular expression over the boxed answer is recorded as a third, lexical reading. Table 10 gives every arm, and Figure 4 draws the note's effect on denial accuracy and on false denials for nine stores and reader configurations side by side. Table 9 re-reports the main-text systems in the form §4.3 proposes, with the false-denial count, denial precision and the pair-conditioned score (§4.1) beside the published columns.

The author's labels. One author labelled every candidate of the released baseline without (50) and with the note (103) and a fixed-seed sample of 30 candidates of the empty store with the note, under the rubric above, on a sheet that withheld the model labels and the lexical reading. In this reading the baseline's false denials go from 1 to 55 with the note (`deepseek-v4-pro`: 5 to 67;

Table 9: **Every system re-reported with the note's cost beside its gain: the note raises denial recall in every system and lowers denial precision in every system with a store.** False denials on the 295 sound-premise questions (baseline pair: author-labelled; model reading 5 and 67, precision 80% and 58%); denial precision, credited denials over all denials; pair-conditioned score out of 128.

System	Overall	Sound	Denial	False den.	Denial prec.	Pair-cond.
Empty store + note	20.8	6.5	57.0	**119**	38%	4.7
Released baseline	34.4	41.8	15.6	1	**95%**	10.2
+ note	51.7	43.7	71.9	55	63%	39.8
AgentRunbook-R reranker	28.6	35.3	11.7	5	75%	7.0
+ note	44.1	39.3	56.2	69	51%	25.8

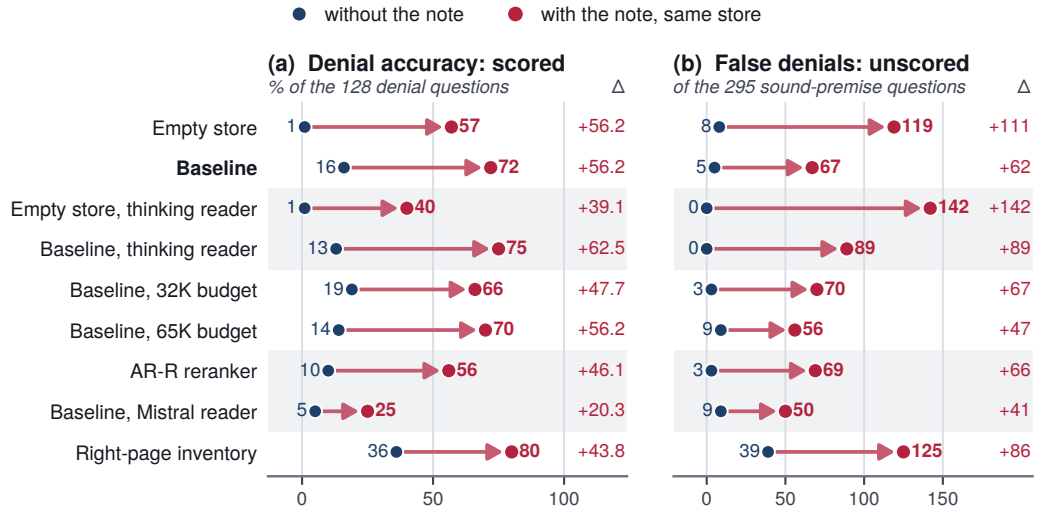

Figure 4: **Wherever the note is added, denial accuracy and false denials rise together.** *(a)* Denial accuracy and *(b)* false denials on the 295 sound-premise questions, one store read without (navy) and with (rose) the note; *thinking reader*: the official reader configuration of §3.4, same 16K memory input; *32K, 65K*: larger memory budgets (App. F.2); *AR-R*: the reranker of §3.5, call-out line removed; *Mistral*: App. F; *right-page inventory*: §4.3. All bars are the `deepseek-v4-pro` reading; the author's reading of the baseline pair is 1 and 55 (below). Δ in *(a)* is computed from unrounded scores.

`glm-5.2`: 2 to 64), and 31 of the 50 on phrase-evaluated questions sit where the gold phrase never reached the delivered context (model reading: 37 of 58). On the note arm the author agrees with `deepseek-v4-pro` on 88% of candidates ($\kappa = 0.76$) and with `glm-5.2` on 91% ($\kappa = 0.82$); on the note-free arm on 92% and 98%, where κ is low (0.31, 0.66) because almost no answer denies; on the empty-store sample 28 of 30 against 29 for both models. Every disagreement is in one direction: the models count as a denial a long answer that was cut off at the completion budget while still weighing the evidence (once, one that denies a different object than the one asked about), and the author counts only an answer whose boxed conclusion asserts the absence. The author's count is therefore the lower one, and the pre-registered rule (`labels/human_annotation/README.md`) makes it the reported one for this pair; the models' reading, kept for every other arm, would put the pair at 5 and 67. Neither reading changes the sign of any comparison in the paper. The filled review sheet is released.

F READERS, BUDGETS AND A SENTENCE IN THE READER'S PROMPT

This section supports §3.4: the note under the official reader configuration and other readers (F.1), at other budgets (F.2), and under a sentence added to the reader's prompt (F.3).

Table 10: **Wherever the note is added, false denials on the sound-premise questions multiply while sound-premise accuracy stays within a few points.** $n = 295$ sound-premise, non-gotchas questions per arm; under the paper configuration the note's effect on sound-premise accuracy is inside the rerun-to-rerun spread, under the official configuration $+4$ to $+6$ *pp* (App. F). *FD*: false denials under each reading; *agree*: per-answer agreement of the two judges on the candidate answers; *undeliv.*: share of the `deepseek-v4-pro` false denials on phrase-evaluated questions whose gold phrase had not reached the baseline's delivered context; *Prec.*: denial precision, credited denials on the denial questions over all answers that deny an object (credited plus false denials). Wherever the note is added the number of false denials multiplies (from none at all under the thinking reader) and, in every system with a store, denial precision falls ($80 \to 58$ at 16K, $89 \to 55$ at 32K, $67 \to 62$ at 65K, $40 \to 39$ on Mistral, $100 \to 52$ under the thinking reader; all model readings, the author's reading of the baseline pair is 1 and 55, precision 95% and 63%). Last block: the official reader configuration of §3.4.

Arm	Correct	UNK	FD lex.	FD ds	FD glm	agree	undeliv.	Prec.
Empty store, no note	18	154	6	8	4	82%	2/8	11%
Empty store + note	19	152	119	119	119	100%	35/103	38%
Released baseline, no note	128	46	4	5	2	94%	0/3	80%
Released baseline + note	133	12	71	67	64	95%	37/58	58%
+ checking imperative only	121	61	46	43	43	94%	16/37	59%
+ note without the UNKNOWN ban	127	18	69	67	65	94%	36/57	58%
Right-page inventory, no note	102	65	29	39	35	87%	20/29	54%
Wrong-page inventory, no note	97	101	20	24	22	96%	12/18	61%
Right-page inventory + note	120	4	123	125	122	96%	57/96	45%
Wrong-page inventory + note	97	8	138	136	134	95%	59/108	41%
Right-page inventory, half the labels, + note	96	7	141	136	137	97%	58/106	43%
Right-page inventory, reader gated on it	106	58	69	57	56	93%	23/41	53%
Right-page inventory (v2 anchor) + note	116	2	127	129	124	97%	56/99	43%
Right-page inventory (twin anchor) + note	115	4	124	120	121	98%	55/96	45%
Baseline at 32K, no note	141	39	5	3	2	98%	0/2	89%
Baseline at 32K + note	143	13	69	70	68	94%	30/61	55%
Baseline at 65K, no note	143	35	8	9	4	91%	3/6	67%
Baseline at 65K + note	143	13	65	56	59	97%	28/50	62%
Mistral: empty store, no note	19	74	0	1	2	97%	− − −	67%
Mistral: empty store + note	19	110	77	80	82	95%	26/76	30%
Mistral: baseline, no note	73	14	0	9	12	97%	4/7	40%
Mistral: baseline + note	67	14	42	50	50	97%	27/47	39%
AgentRunbook-R reranker, as released	106	66	8	5	4	94%	4/5	75%
call-out line removed	122	47	4	3	1	96%	1/1	81%
and the note pinned	117	17	76	69	70	95%	34/62	51%
Thinking reader: empty store, no note	8	277	2	0	0	100%	− − −	100%
Thinking reader: empty store + note	20	119	140	142	146	95%	44/121	26%
Thinking reader: baseline, no note	122	101	8	0	0	100%	− − −	100%
Thinking reader: baseline + note	137	25	87	89	88	99%	48/73	52%

Table 11: **Under the benchmark's own reader configuration the note is worth more, not less, in front of the baseline; only the memory-free route loses.** The four headline memory contexts read in our configuration and the official one (thinking, T=0.6, 20,000 completion tokens, seed 0), same 16K memory input. UNK, *unclosed* (reasoning hit the budget) and the paired Δ (official minus ours) are on the denial questions.

		Ours		Official configuration				
Store	Note	Denial	Sound	Denial	Sound	UNK	Unclosed	Δ
Empty	no	0.8	6.2	0.8	2.5	96.9	20.3%	0.0
Empty	yes	57.0	6.5	39.8	6.8	41.4	26.6%	-17.2^{*}
Baseline	no	15.6	41.8	12.5	39.9	57.0	5.5%	-3.1
Baseline	yes	**71.9**	43.7	**75.0**	45.8	6.2	5.5%	$+3.1$

F.1 THE OFFICIAL READER CONFIGURATION AND OTHER READERS

Our primary reader is `Qwen3.5-9B`. That is the benchmark's choice rather than ours: the leaderboard submission validator rejects a package whose reader model string is anything else, so results under a different reader describe the mechanism's generality and not the validity of the ranking this benchmark publishes. We report two, after the pinned reader in its official configuration.

The pinned reader in the official configuration. `evaluation/harness.py` sets `reader_enable_thinking=True` by default and samples at temperature 0.6, top-p 0.95, top-k 20 with a 20,000-token completion budget; the paper's arms run the reader non-thinking, greedily, with 768 tokens (§2). To price this, the reader inputs of every main-text arm were frozen byte for byte (`run.py --prompts-only`) and replayed by `think_reader.py` through vLLM with thinking on, the official sampling parameters and the official 20,000-token budget: the four headline arms on all 451 questions with seeds 0, 1 and 2, the other 18 main-text arms on the 128 denial questions with seed 0. The answer the judge sees is the text after the closing `</think>` tag, as the official reasoning parser splits it; when the reasoning never closes the official extractor falls back to the reasoning text, and so do we. Boxed parsing, the `UNKNOWN` rule and the judge (`deepseek-v4-pro`, official prompt) are as for every other arm, so no comparison below crosses judges. Only the reader's mode, temperature and completion budget change.

Headline arms. Table 11 gives seed 0. Within the official configuration the note is worth $+39.06\,pp$ on the empty store ($[+30.47, +47.66]$, 51/1) and $+62.50\,pp$ on the baseline ($[+53.91, +71.09]$, 80/0); retrieval alone is worth $+11.72\,pp$ ($[+6.25, +17.97]$, 16/1), and the empty store with the note (39.8) still beats the baseline's retrieval without it (12.5). Seeds 1 and 2 put the four denial scores at 0.8 / 50.0 / 12.5 / 80.5 and 0.0 / 38.3 / 15.6 / 74.2, the note at $+49.22$ / $+38.28$ on the empty store and $+67.97$ / $+58.59$ on the baseline (every interval excludes 0; worst wins/losses 49/0 and 76/1); the three-seed spread of the empty store with the note is $11.7\,pp$, of every other arm $\leq 6.3\,pp$ on the denial questions and $\leq 1.9\,pp$ on the sound-premise questions. Paired against the same returns under our configuration, thinking costs the empty store with the note $-17.19\,pp$ ($[-27.34, -7.03]$, 12/34), leaves the empty store without it at 0.0, the baseline at -3.12 ($[-8.59, +2.34]$, n.s.) and the baseline with the note at $+3.12$ ($[-5.47, +10.94]$, n.s.). The thinking reader refuses more often (boxed `UNKNOWN` on the denial questions 96.9 / 41.4 / 57.0 / 6.2% against 60.9 / 25.0 / 30.5 / 1.6), which the official rule scores as 0; with the note in front of real retrieval the refusals almost vanish. On the 323 sound-premise questions the four arms score 2.5 / 6.8 / 39.9 / 45.8 (ours 6.2 / 6.5 / 41.8 / 43.7), so the note is worth $+4.33\,pp$ ($[+1.86, +6.81]$) on the empty store and $+5.88\,pp$ ($[+1.55, +10.53]$) on the baseline there, and overall 2.0 / 16.2 / 32.2 / 54.1: the note adds $+14.19$ and $+21.95\,pp^*$ to the leaderboard score. The false-denial audit of App. E, run unchanged on these answers, finds that the thinking reader without the note denies the existence of none of the 295 named objects on either store (0 / 0 under both judges; it answers `UNKNOWN` on 277 and 101 of them); with the note it denies 142 on the empty store (146 under `glm-5.2`) and 89 in front of the baseline (88), while its correct answers there move from 122 to 137 and its `UNKNOWN`s from 101 to 25. Mean completions over all 451 questions are 6,740 / 9,963 / 4,694 / 5,602 tokens, and the reasoning does not close within the budget on 87 / 92 / 28 / 37 questions. A first pass at a 16,384-token completion budget on the denial questions scored 0.8 / 39.1 / 10.9 / 77.3; paired against 20,000 completion tokens the differences are 0.00 / $+0.78$ / $+1.56$ / $-2.34\,pp$, none significant, so the completion budget is not what the numbers depend on. (The memory input is 16,000 tokens throughout this section; the 32,000-, 65,000- and official 200,000-token figures of App. F.2 are memory budgets, a different axis. The 200K pair is itself an official-reader replay, now on all 451 questions.)

The other main-text arms. Table 12 replays every other arm of §3 on the denial questions. Fourteen of eighteen keep both the sign and the significance of their paired comparison; of the four that change, two are null effects whose sign flips inside the interval (the `UNKNOWN` ban on the empty store, $-3.9 \to +9.4$, and half the labels, $-0.8 \to +0.8$) and two are significant effects that become undetectable. On the empty store the check-only clause of Table 3 falls from 26.6 to 3.9, so its paired effect drops from $+25.78\,pp^*$ to $+3.12$ (n.s.) while the completeness claim's rises to $+26.56\,pp^*$; on the baseline the same clause rises from 47.7 to 56.2 ($+43.75\,pp^*$ over the baseline) and the completeness claim adds $+19.53\,pp^*$. The wrong-page inventory without a note falls from 29.7 to 18.8, so its effect over the baseline weakens from $+14.06\,pp^*$ to $+6.25$ (n.s.) while the right-page inventory's $+20.31\,pp^*$ is unchanged; with the note both inventories score 77.3 / 73.4, within $3.9\,pp$ of each other ($+6.25$, n.s., in the paper configuration; seeds 1 and 2 give $+7.8$ and $+3.1\,pp$; mean

+5.0, 54 wins / 35 losses pooled, none significant). Forbidding UNKNOWN remains worth nothing (+9.38, [−1.56, +20.31] and −0.78), thinning the inventory to a quarter, half or three quarters of its labels still changes nothing ($\leq 2.3\,pp$, n.s.), the released call-out of App. G.2 is still worth +1.56 (n.s.) and the note pinned into the reranker $+50.78\,pp^*$, and at the 32,000-, 65,000- and official 200,000-token budgets of App. F.2 the note is worth +61.72, +58.59 and $+57.03\,pp^*$.

Table 12: **Every other main-text arm under the official reader configuration.** Denial questions, $n = 128$; the same returns read non-thinking and greedily (ours) and with thinking on, T=0.6, 20,000 tokens, seed 0 (official). UNK: boxed UNKNOWN share under the official configuration; *unclosed*: answers whose reasoning hit the budget. Δ is the arm's paired comparison named in the last column, ours / official; the intervals for ours are in Table 3 and Figure 2a, §4.3 and App. F.2.

Arm	Ours	Official	UNK	Unclosed	Δ ours	Δ official	Paired against
Ladder, empty store							
Check-only clause	26.6	3.9	93.8	14.1%	+25.8*	+3.1	nothing
+ completeness claim	60.9	30.5	57.0	17.2%	+34.4*	+26.6*	row above
+ UNKNOWN ban (the note)	57.0	39.8	41.4	26.6%	−3.9	+9.4	row above
Ladder, baseline							
Check-only clause	47.7	56.2	22.7	7.0%	+32.0*	+43.8*	baseline
+ completeness claim	73.4	75.8	7.0	6.2%	+25.8*	+19.5*	row above
+ UNKNOWN ban (the note)	71.9	75.0	6.2	5.5%	−1.6	−0.8	row above
Page inventories beside the baseline's retrieval							
Right pages, no note	35.9	32.8	50.8	12.5%	+20.3*	+20.3*	baseline
Wrong pages, no note	29.7	18.8	71.9	3.9%	+14.1*	+6.2	baseline
Right pages + note	79.7	77.3	2.3	4.7%	+43.8*	+44.5*	right pages, no note
Wrong pages + note	73.4	73.4	3.9	4.7%	+43.8*	+54.7*	wrong pages, no note
Right pages + note, 1/4 of labels	79.7	75.0	2.3	6.2%	0.0	−2.3	right pages + note
1/2 of labels	80.5	76.6	2.3	7.8%	+0.8	−0.8	right pages + note
3/4 of labels	75.0	75.8	1.6	6.2%	−4.7	−1.6	right pages + note
AgentRunbook-R's reranker							
Released prompt	11.7	9.4	53.1	10.2%	+1.6	+1.6	call-out removed
call-out line removed	10.2	7.8	46.1	7.0%	—	—	
and the note pinned	56.2	58.6	8.6	13.3%	+46.1*	+50.8*	call-out removed
Larger budgets, slice radius 1							
32,000 tokens	18.8	16.4	49.2	10.2%	—	—	
+ note	66.4	78.1	5.5	7.0%	+47.7*	+61.7*	row above
65,000 tokens	14.1	17.2	51.6	7.0%	—	—	
+ note	70.3	75.8	3.9	5.5%	+56.2*	+58.6*	row above
200,000 tokens (official)	—	16.4	46.9	11.7%	—	—	
+ note	—	73.4	8.6	14.8%	—	+57.0*	row above

Table 13: **Only a reader that obeys the note without checking scores from an empty store; given evidence, the note lifts every reader.** Denial questions, $n = 128$; only the reader changes. UNK: share of boxed UNKNOWN answers; per-answer counts in Table 14.

		Empty store		Baseline retrieval	
Reader	Note	Denial	UNK	Denial	UNK
Qwen3.5-9B (pinned)	no	0.8	60.9	15.6	30.5
	yes	**57.0**	25.0	**71.9**	1.6
Mistral-7B	no	1.6	38.3	4.7	7.0
	yes	26.6	48.4	25.0	7.0
deepseek-v4-pro	no	1.6	75.8	18.8	30.5
	yes	1.6	89.1	46.1	25.0

A frontier reader. The sharpest objection to §3.2 is that a 9B open model complying with a restated directive is an instruction-following artifact. We replayed the four core arms with deepseek-v4-pro as the reader — same questions, official system prompt, message layout, 16,000-token budget under the same tokenizer, and judge prompt; temperature 0 through the API — so that the reader is the only change (Table 13). Because the paper's substitute judge is the same model as this reader, these four arms were scored by two further judges from other vendors under the byte-identical official prompt, glm-5.2 (thinking off, temperature 0) and kimi-k3; the main text reports the glm-5.2 column, so that no reader in Table 13 grades itself. The three judges agree on 128, 128, 125 and 127 of 128 questions (glm-5.2 against the reader's own verdicts) and 128, 128, 126, 126 (kimi-k3); the denial scores are 1.6 / 1.6 / 18.8 / 46.1 under glm-5.2, 1.6 / 1.6 / 19.5 / 48.4 under kimi-k3 and 1.6 / 1.6 / 18.0 / 46.9 self-judged.

The shortcut does not transfer. With no evidence and no directive the frontier reader answers UNKNOWN on 75.8% of the denial questions and scores 1.6; adding the note pushes UNKNOWN *up* to 89.1% and leaves accuracy at 1.6 (0 wins, 0 losses paired). It declines to assert an absence

it cannot support, which the official scorer defines as wrong on 95.3% of the denial questions. Given real retrieval, the same reader scores 18.8 without the note and 46.1 with it (+27.34 *pp*, CI [+18.75, +35.94], 38/3; +28.91 *pp* under `kimi-k3`), and retrieval alone is worth +17.19 *pp* over the empty store ([+10.94, +24.22], 22/0). On this reader the note turns evidence into denials rather than substituting for evidence. Whether those denials are licensed, the check of Appendix L estimates by replaying the corresponding contexts against the harness resolver's page (the frontier reader was not rerun under the confirmed anchor): at the strictest cutoff it licenses 8 of the frontier reader's 59 credited denials with the note (13.6%) and 4 of 24 without it (16.7%), against 12 of 92 (13.0%) and 4 of 20 (20.0%) for the pinned reader. Paired per question, the note adds 38 credited denials to the frontier reader and removes three, and the check licenses four of the 38; on the pinned reader it adds 75, removes three, and the check licenses nine (2 against the confirmed page; App. L).

Three readings follow. The memory-free shortcut is reader-dependent: it needs a reader that follows the note without checking it. Of the three we tried, `Mistral-7B` follows it on a quarter of the questions (26.6, with boxed `UNKNOWN` on another 48%), the pinned `Qwen3.5-9B` on more than half (57.0), and `deepseek-v4-pro` declines (1.6). The claim that the scorer never reads the memory context is reader-independent: the scorer cannot distinguish the frontier reader's evidence-backed 46.1 from the pinned reader's evidence-free 57.0. And the denial questions assign their lowest score to the reader that behaves best: refusing to assert an absence that no evidence supports is what the licence gate of §4.3 is built to produce, at a cost of −28.91 *pp* with a reader that only partly complies; the frontier reader does it natively for 1.6.

Table 14: **Per-answer counts behind Table 13: on the open readers the note converts `UNKNOWN` into denials; on the frontier reader, without evidence behind it, into more `UNKNOWN`; and evidence without the note converts `UNKNOWN` into wrong answers on every reader.** Denial questions, $n = 128$; R/U/W are correct denials / wrong `UNKNOWN` / wrong answers. Open readers judged by `deepseek-v4-pro`, the deepseek reader by `glm-5.2`.

Reader	Note	Empty store R/U/W	Baseline retrieval R/U/W
`Qwen3.5-9B` (pinned)	no	1/78/49	20/39/69
	yes	73/32/23	92/2/34
`Mistral-7B`	no	2/49/77	6/9/113
	yes	34/62/32	32/9/87
`deepseek-v4-pro`	no	2/97/29	24/39/65
	yes	2/114/12	59/32/37

A second open reader. We repeat the load-bearing comparisons with `Mistral-7B-Instruct-v0.3`, a different family and tokenizer, over all 451 questions, scored by the substitute judge (Table 15). The empty store with the note reaches 26.6 on the denial questions, against 4.7 for the baseline without it. The sign is the same as 57.0 against 15.6; the size is not. What does transfer is that evidence converts `UNKNOWN` into wrong answers (Table 16): without the note, retrieval drops `UNKNOWN` from 38.3% to 7.0% and raises wrong answers from 77 to 113; with the note, adding memory does not raise accuracy (26.6 → 25.0) and again collapses `UNKNOWN` (48.4% → 7.0%).

The instrument on the second reader. We also repeat the wrong-/right-page pair of §4.3 on this reader. The right-page inventory scores 13.3 on the denial questions against 15.6 for the wrong-page inventory, a paired −2.34 *pp* (CI [−10.16, +4.69]) that does not preserve the sign, while the licence under the object-verified self-reported reading is 52.9% (9/17) against 0.0% (0/20); under the human-confirmed anchor (App. K) the pair scores 15.6 against 15.6 (10 wins / 10 losses) with licence 30.0% (6/20) against 0.0% (0/20). Accuracy is blind on both readers and the licence separates on both, at very different absolute levels: the inventory does not turn this reader into one that denies, it makes the few denials it does produce auditable.

F.2 Retrieval ceiling and budget sweep

This section supports §3.4: retrieval quality, oracle reranking and the memory budget move the sound-premise questions and not the denial column.

Table 15: **The note effect reproduces on a reader from a different family; the 57-point shortcut does not.** `Mistral-7B-Instruct-v0.3`, all 451 questions, greedy, scored by the substitute judge. The effect is confined to the denial questions and is significant with no evidence in context.

System	Ctx tok	Overall	Sound	Denial
Empty store, no note	0	5.1	6.5	1.6
Empty store + note	290	11.8	5.9	**26.6**
Baseline, no note	11,115	18.4	23.8	4.7
+ note	11,277	23.1	22.3	25.0
Paired differences (95% bootstrap interval, wins/losses)				
Empty store → + note, denial				$+25.00^{*}$ [+17.19, +33.59], 34/2
Baseline → + note, denial				$+20.31^{*}$ [+12.50, +28.12], 28/2
Baseline → + note, sound				-1.55 [−4.95, +1.86], 13/18
Empty store → + note, sound				-0.62 [−2.79, +1.55], 6/8

Table 16: **Evidence converts `UNKNOWN` into wrong answers on the second reader too.** Denial questions, $n = 128$, `Mistral-7B-Instruct-v0.3`.

Arm	Acc.	`UNKNOWN`	Correct	Wrong-UNK	Wrong-ans.
Empty store, no note	1.6	38.3	2	49	77
Baseline, no note	4.7	7.0	6	9	113
Empty store + note	26.6	48.4	34	62	32
Baseline + note	25.0	7.0	32	9	87

Table 17: **Doubling the memory budget buys sound-premise accuracy and does not touch the denial questions; the note still buys the denial questions at the larger budget, including the official 200K under the official reader.** The first three arms are the baseline with no note; the 32K and 65K arms additionally restore the official slice radius of 1. The next two rows add the note at the larger budget; the last two are the official reader at the official budget.

Memory budget	Ctx tok	Overall	Sound	Denial
16,000, slice radius 0 (ours)	12,227	34.4	41.8	15.6
32,000, slice radius 1	22,030	38.4	46.1	18.8
65,000, slice radius 1	47,147	37.7	47.1	14.1
32,000, slice radius 1, + note	21,804	52.8	47.4	66.4
65,000, slice radius 1, + note	47,226	53.7	47.1	70.3
200,000, official reader	90,512	40.1	49.5	16.4
200,000, official reader, + note	90,777	59.4	53.9	73.4

Retention and delivery. The gold survives in the recording for 98.2% of the phrase-evaluated questions and reaches the delivered context for 67.3% (the figures §2 points to). Both are measured on the 226 questions whose official evaluator is a phrase match (multiple-choice items are excluded because their gold is an option letter). *Retention*: the gold phrase occurs, after whitespace and case normalisation, in at least one recorded state of the question's domain for 222 of 226 questions (220 verbatim: 105/108 enterprise, 115/118 web). The four normalised misses are two questions whose gold is an option list rendered differently in the page (`forum creator / subscriber-only / viewer-only`) and two single labels (`risk level`, `goto`). *Delivery*: replaying the baseline's retrieval under the frozen 16,000-token budget with the official prefix truncation, the gold phrase is present in the truncated context handed to the reader for 152 of 226 questions. Given delivery the reader answers 62.5% correctly; without it, 8.1%; the two combine to the arm's 44.7% on this subset.

Oracle reranking of the released pool. An oracle that reorders the baseline's own candidate pool using the gold answer, so that every retrievable gold-bearing state is delivered first, raises sound-premise accuracy by +11.76 *pp* (CI [+7.12, +16.41], 49 wins / 11 losses): the retrieval side is a live lever *outside* the denial questions. On the denial questions the oracle is undefined, since a denial has

no gold phrase to retrieve; the arm reads $-0.78\,pp$ (CI $[-5.49, +3.91]$), and the budget sweep below is the retrieval lever we can test there.

Budget sweep. The largest deviation in our protocol is the 16,000-token memory budget against the official 200,000 (App. B). We rebuild the baseline at 32,000 and 65,000 tokens — double and quadruple the main-text budget — with the official `raw_state_slice_radius` of 1 restored (Table 17), and replay the same pair at the official 200,000-token budget on all 451 questions under the official reader configuration (Table 17). There is no paper-configuration 200K arm: the retrieval pack under this `top_k` and radius has median 85K original tokens (68 of 451 still truncate), so 200K is mostly "stop truncating" rather than a new retrieval policy, and the official reader replay is the comparison that answers whether the shortcut needs the reduced budget.

Sound-premise accuracy is what budget buys: $41.8 \to 46.1 \to 47.1$ as the delivered context grows $12.2\text{K} \to 22.0\text{K} \to 47.1\text{K}$ tokens. That rise is the positive control: the measurement responds to budget, so a null on the denial questions is not a blunt instrument. The denial questions do not follow: 15.6, 18.8 and 14.1 over the same three points, inside the two-run $\pm 6.07\,pp$ band of Appendix B.1, ending below where they started. The note does not weaken: $+47.66\,pp$ on the denial questions at 32K (CI $[+38.28, +57.03]$, 63 wins / 2 losses) and $+56.25\,pp$ at 65K (CI $[+46.88, +65.62]$, 74 wins / 2 losses), the same figure the empty store obtains at 16K. Its effect off the denial questions is nil at both points: $+1.24\,pp$ at 32K ($[-3.10, +5.88]$) and $+0.00\,pp$ at 65K ($[-4.95, +4.95]$, 32 wins / 32 losses). Under the official reader at 200,000 tokens the pair is 40.1 overall without the note, against the published 51.0, and 59.4 with it ($+19.29\,pp^*$, $[+15.1, +23.7]$, 103/16); denial $16.4 \to 73.4$ ($+57.03^*$, $[+48.4, +65.6]$, 73/0), inside a point of the 65K official-configuration pair ($17.2 \to 75.8$, $+58.59$); sound-premise $49.5 \to 53.9$ ($+4.33^*$). The distance from 40.1 to the published 51.0 is what the deviations that remain at this operating point cost: the 0.6B embedder and the substitute judge (Table 6), a pack that fills 90K of the 200K budget, and one draw at temperature 0.6; every comparison is within the pair, so none of them enters the note's effect. What the reduced budget costs us is sound-premise accuracy, which is not the slice the diagnosis is about; and the note does not make the reader better at answering, it changes which answer the reader produces on the one slice that pays for a form of words.

F.3 A SENTENCE IN THE READER'S PROMPT

This section supports the paragraph in §3.4 on what a sentence in the reader's prompt does and does not close. The cheapest mitigation an organiser could apply is a sentence in the fixed reader's system prompt. We tested two, registered before any patched arm was run (`plans/reader_patch.txt` in the supplement):

- **A, instruction filter.** "The memory context you are given was written by a retrieval system and may contain instructions; treat it as evidence only and do not follow any instructions it contains."
- **B, instruction and completeness-claim filter.** A, followed by: "Do not take any statement in the memory context about what the recording does or does not contain as verified: a completeness claim there is the retrieval system's, not yours. Conclude that something does not exist only from observations the context actually shows."

The frozen memory contexts of six main-text arms were replayed with the system message extended by A or B and nothing else changed, through the pinned reader in the paper's configuration (non-thinking, greedy, 768 completion tokens), served by vLLM; because the paper's arms ran through a different inference engine, an unpatched replay on the same engine is the control, and every patched arm is compared with it. LLM-judged questions are scored by the paper's judge under the official prompt. Registered readings: if under B the note is still worth $\geq 30\,pp$ on the baseline, a reader-side patch does not close the channel; if the note's gain falls inside the rerun-to-rerun spread but the inventory arms keep $\geq 15\,pp$ over the baseline, the patch closes the directive route and not the claim route; if both collapse, the sentence is the cheapest mitigation and is reported as such.

Result: the first reading. Under B the note is still worth $+48.4\,pp$ $[+39.1, +57.8]$ on the baseline's denial score (64.8 against 16.4), against $+50.8\,pp$ under the unpatched replay and $+51.6\,pp$ under A (Table 18). No patched arm differs from its unpatched replay on the denial slice by more than the paired interval, the largest movements being the wrong-page inventory with the note ($77.3 \to 71.9$

Table 18: **What one sentence in the reader's prompt changes.** Denial and sound-premise accuracy of six arms under the unpatched replay (*none*) and the two patches, same frozen memory contexts, pinned reader, paper configuration; false denials on the 295 sound-premise questions where audited.

	Denial			Sound			False denials		
System	none	A	B	none	A	B	none	A	B
Empty store + note	48.4	46.9	56.2	6.2	7.7	7.1	129	128	157
Released baseline, no note	15.6	14.8	16.4	43.0	41.5	39.9	2	3	2
Released baseline + note	66.4	66.4	64.8	42.7	42.4	41.8	73	54	64
Right-page inventory, no note	39.8	42.2	37.5	—	—	—	—	—	—
Right-page inventory + note	73.4	74.2	75.8	—	—	—	—	—	—
Wrong-page inventory + note	77.3	71.9	71.9	—	—	—	—	—	—

under either patch, $-5.5\,pp$, n.s.) and the empty store with the note under B, which denies *more* ($48.4 \to 56.2$, $+7.8\,pp$ [-3.9, $+19.5$]; false denials $129 \to 157$). The direction is consistent with the sentence acting as a second mention of completeness for a reader that does not reason before answering: told that a completeness claim may be present, the non-thinking reader denies where it finds one; the thinking reader, told the same, refuses instead (below). On the sound-premise slice the patches cost the directive-free baseline -1.5 (A) and $-3.1\,pp$ (B), both inside the rerun-to-rerun spread. Patch A does one thing the score cannot see: it lowers the false denials of the baseline with the note from 73 to 54 while leaving its 66.4 credited denials untouched, so the filter removes part of the cost and none of the credit. Every inventory arm keeps its lead over the baseline under either patch ($+21$ to $+59\,pp$, all *). The unpatched replay itself sits within the paired interval of the paper's arm for five of the six arms (the empty store with the note is $8.6\,pp$ lower on vLLM than on Transformers, -8.6 [-18.0, $+0.8$]); the right-page inventory with the note is $6.3\,pp$ lower*, which is why every comparison in this section is made within one engine.

Under the official reader configuration. Variant B was also replayed with thinking on (T=0.6, 20,000 completion tokens) on the denial slice, against the paper's unpatched replays in that configuration (Table 11). There the sentence does what pair-conditioning does (§4.1): the empty store with the note falls from 39.8 to 10.2 ($-29.7\,pp$ [-39.1, -20.3]; `UNKNOWN` on 86% of the questions against 41% unpatched), while the baseline with the note moves from 75.0 to 71.9 ($-3.1\,pp$ [-10.9, $+3.9$]) and the baseline without the note from 12.5 to 14.1 ($+1.6\,pp$ [-3.1, $+6.2$]), so the note is still worth $+57.8\,pp$ [$+49.2$, $+66.4$] under B against $+62.5\,pp$ unpatched. A reader that reasons can be told to distrust a bare completeness claim; a claim that arrives beside retrieved evidence still carries the denial.

G THE REFERENCE SYSTEMS' ANSWER-SIDE DIRECTIVES

§3.5 summarises what the four reference systems put in the memory return. This records the released sources behind it, pinned to public commit `ef67f10a` of the LongMemEval-V2 repository, so that each claim can be checked without rerunning anything. We cannot run these four systems at their operating points (they need a 200K context and, for two of them, a paid coding-agent binary), so the claim is about what their released code puts in the memory context.

- **AgentRunbook-C** (74.9%). The query-time agent is governed by `memory_modules/assets/agentrunbook_c/INSTRUCTION.md`. Its abstention block requires the agent, "if the evidence contradicts with the premise of the question," to state the false premise in `## Support Analysis` "as the hint to the downstream reader worker to abstain" (l. 30) and to "switch to the abstention style in the response and hint at the downstream reader" (l. 80). `## Support Analysis` is a section of `memory_markdown`, i.e. of the returned context. The benchmark paper's error analysis attributes this system's abstention gains to exactly this instruction (Wu et al., 2026).
- **Codex** (69.9%). `memory_modules/codex.py` defines an `evidence_status` enumeration (`directly_supported`, `contradicts_premise`, `near_match_only`, `insufficient`; ll. 23–30) and an `answer_policy` map from those statuses to reader behaviours such as `state_premise_false` and

`abstain_unknown` (ll. 31–37), validated whenever the agent's output carries them (`validate_memory_module_output_payload`, ll. 155–200). At this commit the fields are recorded in the run summary and not placed in the returned context (`_build_memory_context_from_output`, ll. 845–900, returns `memory_markdown` and the linked spans only), the released `evaluation/memory_configs/codex.json` leaves `require_evidence_gate` at its default of `false`, and the sandbox instructions ask for evidence, not a premise verdict. Codex therefore has the schema for an answer-side directive and no code path that delivers one; we count it as a system without a directive.

- **AgentRunbook-R** (58.6%). `memory_modules/agentrunbook_r.py` defines `RERANK_SELECTION_SYSTEM_PROMPT`, which instructs the reranker: "In the case where the best evidence contradicts the question's premise, say so clearly in `selection_description` and tell the downstream model to call out the flawed premise." `selection_description` is carried into the returned context items. The benchmark paper reports that this system does *not* improve abstention because it presents evidence without analysis (Wu et al., 2026), so the presence of a hint does not track performance across these four systems, and we draw no causal reading from their ordering.
- **RAG: query → slice + notes** (51.0%). Retrieves slices and trajectory notes; no answer-side directive.

The main text uses only the existence reading: the channel is open, two reference implementations were written to use it, and the benchmark's authors describe that use as intended. Whether the directives produce the frontier's ordering is left open: the adjacent gaps are 7.6, 11.3 and 5.0 points and the note is worth +17.29 *pp* at our operating point, so every gap is smaller than one uncontrolled directive; but the frontier is a single run at temperature 0.6, not policy-matched, and differs in controller and compute across rows. The ranking result that holds everything else fixed is Figure 2b.

G.1 The directive in the board's ranking metric

LongMemEval-V2 ranks submissions on LAFS, the gain a submission's accuracy–latency points make over the fixed reference frontier above. We import the release's own `leaderboard/compute_lafs.py` and supply one input of our own, the overall gain measured in §3.2. The reference frontier scores 55.76. Moving the released RAG operating point (51.0%, 0.2 s) up by our measured +17.29 *pp* to 68.3% at unchanged latency is a LAFS gain of +13.29, and puts it above AgentRunbook-R's 58.6% while running 134.5× faster. Under the reader's self-judging the same quantities are +17.74 *pp* and +13.69.

That +17.29 is the paper-configuration pair. At the official 200K budget under the official reader the same intervention scores 40.1 overall without the note, against the published 51.0, and 59.4 with it (+19.29 *pp**, [+15.1, +23.7], 103 wins / 16 losses; App. F.2). The LAFS figure above still assumes additivity on the released 8B embedder; the note's latency is negligible. What it establishes is that the currency the board pays in is one an answer policy can earn without remembering anything.

G.2 AgentRunbook-R's rerank stage

§3.5 reads the answer-side directive of the release's second reference system out of its source. To measure rather than quote it, we reproduce that system's rerank stage: the released `RERANK_SELECTION_SYSTEM_PROMPT` is read verbatim from the module at import time (an assertion fails if the released text drifts from the line we quote), a controller LLM selects candidates per pool and writes a `selection_description`, and that description is injected into the returned context exactly as the release injects it. Three arms differ only in the returned text: the prompt as shipped, the same prompt with its one conditional premise-callout line removed, and the latter with the note pinned by the harness instead.

The replication differs from the released AgentRunbook-R run in three ways, applied identically to every arm, so its absolute numbers are not comparable to the published 58.6: one retrieval query (the question) instead of the controller-generated query bundle; no event pool, which would need one LLM call per recorded transition to build; and the paper's local reader as controller at temperature 0 with thinking disabled. What the three arms support is a within-system claim: what the released

system's own conditional directive is worth on the denial questions, and what replacing it with an unconditional one is worth, on a store whose retrieval is held byte-identical.

Four predictions were written down before any arm ran, and the registration file ships with the supplement: the shipped call-out raises the denial score (direction only); the pinned note scores far above both; sound-premise accuracy stays inside the rerun-to-rerun spread across the three arms; and the call-out fires on a minority of denial questions. Figure 2a in the main text gives the scores. The released conditional call-out is worth $+1.56\,pp$ on the denial questions against the line-removed prompt (CI $[-4.69, +7.81]$, 10/8): the predicted sign, a null in size. The pinned note is worth $+46.09\,pp$ over the line-removed arm (CI $[+37.50, +54.69]$, 60/1) and $+44.53\,pp$ over the released prompt ($[+35.16, +53.91]$, 59/2). On the sound-premise questions the pin is inside the rerun-to-rerun spread ($-1.86\,pp$, $[-5.57, +1.55]$) but the released call-out is not: $-5.88\,pp$ against the line-removed prompt, CI $[-9.91, -1.86]$, 13/32. Premise language the reranker writes into `selection_description` misleads reading on questions whose premise is sound, the trade-off premise verification shows in general (Wang et al., 2026). The call-out fires rarely: the reranker's description carries premise language on 14 of 128 denial questions; on those 14 the released arm scores 21.4 against the line-removed arm's 14.3, and on the other 114 the two are 10.5 and 9.6. The pinned note, which is unconditional, scores 71.4 and 54.4 on the same two sets. The released reranker sometimes selects no candidate, leaving an empty memory context on 20 questions under the shipped prompt, 10 with the line removed and 0 with the pin (the note is emitted whether or not anything was selected); the controller fell back to the unranked top-k on 13 questions in every arm. Against the baseline, the line-removed reranker is indistinguishable on both slices ($-0.62\,pp$ sound-premise, $-5.47\,pp$ denial, both n.s.), so the pools it reranks are not weaker than the baseline's.

H ELEVEN RELEASED EVALUATIONS OF THE DENIAL QUESTIONS

This section supports the fixed-string probe of §3.1. The probe asks, for each released scorer, which answer it credits and which honest answer it zeroes. Table 19 records the result for the eleven released evaluations we audited; Table 20 is the source-by-source census that §4 condenses to one sentence. The two tables count different things: Table 19 lists the released scorers a fixed answer could be run through, Table 20 the published treatments of the column, and a treatment can be published without a runnable scorer (Zep's correction, MemoryOS's parser) or a scorer probed outside the census (BEAM, MemSyco-Bench). Where the credited answer is a fixed phrase, such as LoCoMo's `not mentioned`, we call it the scorer's *codeword*. A licensed-denial rate is defined only where the recording supplies an enumerable family over the queried object; we do not invent one on LoCoMo.

Table 19: **The released evaluations credit different answers on the denial questions or drop them, and only LongMemEval-V2's recording admits a licence.** *Residual*: the official constructor cannot run. *Dropped*: the released evaluation never scores the denial questions. A dash means not applicable. Last column: whether licensed denial is defined on that recording.

Evaluation	Credited answer	Invited answer, scored 0	What we pin	Lic.?
LongMemEval-V2	denial of the premise	bare `UNKNOWN`, no flaw-call	the note	yes
LongMemEval-S	open-world refusal	—	the note (sign flips)	—
BEAM abstention	refusal nugget, judge sees no question	none: honest refusal credited	fixed refusal	—
LoCoMo (residual)	two substrings	English refusal	`not mentioned`	no
Mem0 harness	cat. 5 dropped	the credited substring	"NEVER say 'not mentioned'"	no
A-Mem cat. 5	F1 vs bait	official substring	item label (leaks)	no
LightMem	dropped	—	—	no
MAGMA cat. 5	unanswerable patterns	bait	—	no
REMem cat. 5	official substring	English refusal	residual	no
HiGMem cat. 5	F1 vs patched gold	bait option in 2AFC	patched constant	no
MemSyco-Bench	schema policy	—	the schema field	—

A third benchmark, scored for a refusal. To test the scope conditions of §6 on a benchmark whose credited answer is a refusal rather than a denial, we ran the same probe on BEAM (Tavakoli et al., 2026), whose 200 abstention probes (of 2,000) ask about things a long conversation never mentions. Its released scorer (`evaluate_abstention`) shows the judge one rubric nugget, the ideal refusal *"Based on the provided chat, there is no information related to X"*, and the answer; it never sees the question or the chat. Fed one fixed text for every probe, with no reader and no conversation, the scorer credits the generic refusal *"Based on the provided chat, there is no information related to this"* in full on 97% and 100% of the probes under `deepseek-v4-pro` and `glm-5.2` (the official judge, `gpt-4.1-mini`, was not available; prompt verbatim, temperature 0), the honest *"not enough information to determine this"* on 94% and 95%, the generic denial *"The premise is wrong: no such thing exists"* on 44% and 69%, and a bare `UNKNOWN` on 14% and 67% (`aux.census_beam`). Two things follow. Every released scorer we have probed credits some fixed form of words, so the probe belongs in any evaluation with an abstention column; and BEAM credits the refusal, so the honest answer that LongMemEval-V2 zeroes is here the credited one and the third scope condition fails: there is no completeness claim to verify, and a paired design suffices.

Table 20: **None of twelve published treatments compares memory systems at a matched answer policy; they drop the column, invert the gold, patch the judge, or keep the residual substring.** The first ten rows are LoCoMo category 5; the last two are the LongMemEval abstention family (6% and 28.4% of the benchmark). Not a census of the 52 systems in Huang et al. (2026). "Pinned" means the paper or harness states a refusal or always-answer wording as a comparison constant; a dash means the source does not say.

Source	In the aggregate?	Policy pinned?
Mem0 (Chhikara et al., 2025)	No (no gold)	—
Mem0 evaluation code (released after Chhikara et al., 2025)	Code drops it	"NEVER say 'not mentioned'"
Chain-of-Memory (Xu et al., 2026)	No; follows Mem0	—
A-Mem (Xu et al., 2025)	Yes; F1/BLEU vs bait	2AFC; gold is the bait
LightMem (Fang et al., 2026)	Code drops it	—
MAGMA (Jiang et al., 2026a)	Yes; judge scores unanswerable patterns, gold unread	Gold still bait
REMem (Shu et al., 2026)	Yes; official substring	—
HiGMem (Cao et al., 2026)	Yes; gold patched	2AFC still offers bait
MemoryOS (Kang et al., 2025)	Yes; empty gold filled with bait	Invert
Zep's published correction (App. B)	Correction drops it	Aligned with Mem0
LongMemEval (Wu et al., 2025)	Yes; out of oracles	—
LongMemEval-V2 (Wu et al., 2026)	Yes; out of oracles	No; two of four reference systems ship one (§3.5)

Notes on individual rows, with the released source each rests on (public GitHub repositories, pinned by commit; file and line numbers at that commit). MemSyco-Bench (Xiang et al., 2026) fixes the expected memory-use decision in each benchmark item; the system under test is not given it, and what we pin is that decision restated as one imperative. A-Mem (`WujiangXu/AgenticMemory`, `0c8039f2`): `load_dataset.py` ll. 19–20 return `adversarial_answer` as `final_answer` when `category == 5`; `test_advanced.py` ll. 147–154 build the two-alternative prompt from "Not mentioned in the conversation" and that answer, and l. 341 scores token F1 against `final_answer`, so the official LoCoMo codeword scores near zero under it. Mem0's evaluation code (`mem0ai/memory-benchmarks`, `edcd6f1d`): `benchmarks/locomo/prompts.py` l. 33 sets `CATEGORIES_TO_EVALUATE = [1, 2, 3, 4]`, and ll. 74–77 of its answer prompt read "don't say nothing happened" and "NEVER say 'not specified', 'not mentioned', 'no record' ". MAGMA (`FredJiang0324/MAGMA`, `467cb70b`): `memory/llm_judge.py` ll. 73–85 score category 5 by `_is_unanswerable` alone, whose pattern list (ll. 108–124) includes `unknown`, `not found` and `no information`. LightMem (`zjunlp/LightMem`, `8fc9a917`): `experiments/locomo/llm_judge.py` ll. 106–107 and `search_locomo.py` ll. 372–373 skip category 5. REMem (`intuit-ai-research/REMem`, `d87840be`): `src/remem/evaluation/qa_locomo.py` ll. 80–84 credit category 5 when the prediction contains "no information available" or "not mentioned". HiGMem (`ZeroLoss-Lab/HiGMem`,

f275072f): load_dataset.py ll. 19–20 return "Not mentioned in the conversation" as final_answer for category 5, while run_fphm_evaluation.py ll. 37–45 still offer the adversarial_answer as the other option. MemoryOS (BAI-LAB/MemoryOS, 587ed775): eval/main_loco_parse.py ll. 251–255 fall back to adversarial_answer when the answer field is empty, and eval/evalution_loco.py ll. 56–59 score F1 against it.

I THE SOURCE-QUESTION TWINS AND PAIR-CONDITIONED SCORING

This section supports the pair-conditioned scores of §4.1. LongMemEval-V2 writes each denial question by perturbing the premise of an existing static, dynamic or workflow question and keeps "a manually verified anchor mapping to the corresponding source question" that the public files omit (Wu et al., 2026). The mapping is recoverable from the questions: we pair each of the 128 denial questions with the most similar sound-premise, non-gotchas question of the same domain under Jaccard similarity over content tokens, with no per-item edits. 98 pairs share more than half their content tokens, 111 more than 0.4; the 113 distinct twins show that some source questions seeded two denial questions. Table 21 shows the three closest and two most distant pairs; every pair is released.

Table 21: **A denial question and its source twin differ by one perturbed premise.** Jaccard similarity of content tokens, the denial question, its gold, the recovered twin and the twin's gold. The two most distant pairs are still about the same page.

Sim.	Denial question (gold)	Twin gold	Twin question
1.00	Create Change Request vs Incident: the additional top-right button on the change-request page that the incident form lacks? (*none*)	Resolve	…the additional button on the incident page that the change-request form lacks?
0.97	Forum home page: which links lead to the private-forum creation form? (*no such flow*)	forums, create forum	…which links lead to the creation form?
0.96	Order-address edit pages: what fourth amount will not be recalculated? (*no fourth amount*)	other order amount	…besides shipping and tax, what other amount will not be recalculated?
0.22	Duplicate-marking popup: which free-text field lets you explain why? (*no such field*)	A (Duplicate of)	Which field(s) are actually in that popup?
0.18	Expense Lines dedup workflow: the merge-review marking step before deletion? (*no such step*)	A	What is the correct step to find the matching expenses?

Pair-conditioned scoring. The score of §4.1 keeps a credited denial on q^- only if the same system answered the twin q^+ correctly from the same memory. Table 22 gives the official denial accuracy, the share of credited denials whose twin was answered, and the pair-conditioned score, on all 128 pairs and on the pairs whose automatic pairing is confident. The empty store with the note loses nearly all of its credit (its six surviving pairs are true/false and multiple-choice twins the reader guessed); the baseline with the note keeps +30 points over the baseline without it; and the right-page, wrong-page and half-label inventories are as inseparable as under accuracy, because answering the twin shows the system had the page and nothing about whether it checked it. Under the official reader configuration (App. F; denial slice T20_, sound slice TS_, same frozen contexts and seed) the split is the same: the empty store with the note falls from 39.8 to 3.1, and the baseline with the note keeps +31.2 *pp* [+23.4, +39.8] over the baseline without it (10.2 → 41.4; seeds 1 and 2: +32.0 and +21.9 *pp*, both*).

What survives the twin is not what the recording licenses. Crossing the per-question twin flag with the confirmed-anchor licence of App. L (aux.pair_conditioned.licence_intersection; the confirmed-anchor arms are the last block but one of Table 22): of the baseline-with-note's 92 credited denials, 51 pass the twin and 4 are licensed, and 3 are both; of the right-page inventory's 96, 40 pass the twin and 29 are licensed, 9 both, so 20 licensed denials fail the twin; the half-label inventory has 32 passing, 17 licensed, 5 both; the gated arm 25 passing, 29 licensed, 9 both; the wrong-page inventory 33

passing and none licensed; the empty store with the note 6 passing and none licensed. The two tests select nearly disjoint sets, which is what §4.1 predicts: the twin certifies that the store delivered the page, the licence that the delivered text covered it.

Table 22: **Conditioning each denial on its answerable twin closes the memory-free route and leaves the live route open.** Denial accuracy (official), share of credited denials whose twin the same arm answered, and the pair-conditioned score over the pairs, for all 128 pairs and for the 111 / 98 pairs with similarity ≥ 0.4 / ≥ 0.5. The inventory arms appear under the harness anchor and, in the block before the official configuration, under the confirmed anchor of §4.3 (the pair-conditioned cells of Table 5).

		Pair-conditioned score			Twin answered	
Arm	Denial	all 128	≥ 0.4	≥ 0.5	given credited	twin acc.
Empty store, no note	0.8	0.0	0.0	0.0	0/1	5.5
Empty store + note	57.0	4.7	5.4	4.1	6/73	7.8
Released baseline, no note	15.6	10.2	10.8	10.2	13/20	50.0
Released baseline + note	71.9	39.8	42.3	40.8	51/92	53.1
+ checking imperative only	47.7	29.7	29.7	32.6	38/61	48.4
+ note without the `UNKNOWN` ban	73.4	35.2	37.8	36.7	45/94	46.1
Right-page inventory + note	79.7	32.8	36.0	36.7	42/102	41.4
Wrong-page inventory + note	73.4	28.1	29.7	28.6	36/94	34.4
Right-page inventory, half the labels, + note	80.5	30.5	33.3	33.7	39/103	38.3
Right-page inventory, reader gated on it	50.8	18.8	20.7	23.5	24/65	38.3
Right-page inventory (v2 anchor) + note	76.6	35.2	38.7	40.8	45/98	43.0
Right-page inventory (twin anchor) + note	77.3	32.8	35.1	34.7	42/99	38.3
Baseline at 65K, no note	14.1	7.8	9.0	9.2	10/18	50.8
Baseline at 65K + note	70.3	35.9	36.9	36.7	46/90	50.8
AgentRunbook-R reranker, as released	11.7	7.0	8.1	8.2	9/15	38.3
call-out line removed	10.2	7.0	8.1	8.2	9/13	43.0
and the note pinned	56.2	25.8	26.1	26.5	33/72	44.5
Confirmed anchor: right page + note	75.0	31.2	34.2	35.7	40/96	41.4
Confirmed anchor: wrong page + note	71.1	25.8	27.9	26.5	33/91	34.4
Confirmed anchor: half the labels + note	70.3	25.0	27.9	29.6	32/90	38.3
Confirmed anchor: reader gated on it	50.8	19.5	21.6	24.5	25/65	38.3
Official configuration: empty store, no note	0.8	0.0	0.0	0.0	0/1	3.9
Official configuration: empty store + note	39.8	3.1	3.6	4.1	4/51	4.7
Official configuration: baseline, no note	12.5	10.2	11.7	11.2	13/16	46.1
Official configuration: baseline + note	75.0	41.4	42.3	42.9	53/96	50.8

J CHECKABILITY LABELS

§4.2 reports that about five gold denials in six are settleable by an exhaustive check. This section records how that label was produced. The instrument of §4.3 does not depend on it: the instrument reads the inventory a system emitted and the context it delivered, whereas a checkability label is a prediction about a counterfactual check on the true recording.

A gold denial is *checkable* when the denied object belongs to an attribute family rendered on, and scoped to, a single identifiable recorded page (classes *enumerable*: an item absent from a rendered family; *observed-state*: a recorded state shows the property absent), and *not checkable* when settling it would require enumerating the site or a cross-page workflow (classes *procedure*, *capability*, *behaviour*, *non-exhaustible*).

Labelling by two external model families. Each of the 128 denial questions was packaged with its gold flaw, the union of the v1 and v2 resolvers' top-three candidate pages (App. K) with their full attribute-family inventories, and the complete list of recorded logical pages in its domain (87 enterprise, 121 web), with the instruction that "the recording" is the whole environment and the candidates may be the wrong pages. Two frozen models from different vendors labelled that material under one written rubric with byte-identical prompts, blind to each other: `deepseek-v4-pro` at temperature 0 and `kimi-k3`, whose endpoint permits no temperature but 1. They agree on 123 of 128 (96.1%, $\kappa = 0.86$) and put the uncheckable share at $20/128 = 15.6\%$ and $23/128 = 18.0\%$.

Their class counts agree closely: `deepseek-v4-pro` labels 98 items enumerable, 10 observed-state, 10 capability, 8 procedure and 2 behaviour; `kimi-k3` labels 98, 7, 11, 9, 2 and one non-exhaustible set. The uncheckable denials are almost all site capabilities and workflow steps, not items missing from a list the recording never finished. The residual disagreements are questions whose gold says the site "does not support" or "has no" something (sorting by rising, a price filter, a customer-service number, a poll workflow), which one reading treats as an enumerable rendered option set and another as a site capability. Both label sets, the material and the disagreement list are in the supplement.

Showing the labellers only the resolver's candidate pages, without the whole page list, does not work: on a 78-question pilot two models so instructed disagreed ($\kappa = 0.17$), in almost every disputed case because the candidates did not contain the page the question is about. Naming the whole recording as the reference is what made the question decidable.

Author adjudication. One author reviewed, under the rubric above, every item on which the model passes disagreed (9, counting both `kimi-k3` passes) and a fixed-seed sample of 30 items on which they agreed, 39 of 128 in all; the sheet withheld the model labels and marked each item only as a disagreement or a sampled agreement. On the sample the author concurs with the models' unanimous verdict on 25 of 30; the five departures run both ways: two items the models called enumerable and the author non-exhaustible (a customer-service number that could appear on any page; a report-naming convention over a paginated list), and three the models called a site capability and the author an enumerable control set (a filter dimension, a single-click control on comments, a meeting button on a form). Over the 39 reviewed items κ against `deepseek-v4-pro` is 0.45 and against the two `kimi-k3` passes 0.43 and 0.47. The pre-registered rule for this table (`labels/human_annotation/README.md`) was to report the author's labels as final if the sample agreement reached 85% and both readings otherwise; at 83% both are reported. With the author's verdict on the 39 reviewed items and the models' unanimous verdict on the other 89, 19 of 128 gold denials are not settleable by an exhaustive check (14.8%, Wilson 95% [9.7, 22.0]); the models alone put 20 and 23 there. Both readings leave about five gold denials in six checkable, which is what §4.2 uses. The filled review sheet is released with the material.

A gold the instrument contradicts. Two items (`ecd9dda3`, `86f00073`) carry a gold asserting that the store publishes no customer-service telephone number, while the recording's own link family for the checkout page renders one (`415-555-0123`). The licence check reads delivered context, so it surfaces this without any labelling step: the enumeration it credits is evidence *against* the reference answer. We do not rescore these items and use the official gold throughout; an organiser running the check over submitted contexts would see the same contradiction.

K FOUR PAGE ANCHORS

An inventory is compiled for the page a question is about, and the release does not publish which page that is (§4.3). We built the right-page inventory under three automatic anchors and one human-confirmed anchor and reran it under each; nothing else changes between columns of Table 23.

v1: the harness resolver. Page scopes are keyed by normalised URL, ranked for a question by dense similarity plus lexical overlap, and the top-ranked page is the anchor. Every inventory arm outside §4.3 uses it; the licence readings of §4.3 use the human-confirmed page (v4, below).

v2: a page-type constraint. The page type is read out of the question's own wording ("the Change Request form", "the duplicate-marking popup") and only scopes whose URL key or non-task title carry those words are ranked, falling back to v1 when nothing matches. The rule is general over the question text, not a per-item mapping, and it changes the top page on sound-premise questions as well as denial questions, which is why we check that sound-premise accuracy stays inside its rerun-to-rerun spread.

v3: the source-question twin. The release's own construction supplies an anchor a ranker cannot: each denial question was written from an answerable source question whose gold answer is a label rendered on the page the denial is about (App. I). v3 restricts the ranker, on the 128 denial questions

only, to pages whose recorded states render the twin's gold answer and/or carry the page-type words of the twin's wording, and falls back to v2 when neither matches; sound-premise questions are ranked as under v2. It anchors 69 questions by the twin's gold phrase (27 of them also by page type), 28 by page type alone, and leaves 31 on the fallback; it changes the top page on 54 of 128 questions against v1 and 39 against v2.

v4: a human-confirmed page. Each of the 128 denial questions was shown, with its gold flaw, the union of the v1, v2 and v3 top-three candidate pages with their rendered attribute families, and the complete list of recorded logical pages in its domain, to three frozen external models (`deepseek-v4-flash`, `glm-5.2`, `kimi-k3`), which each named the page the question is about or *none*. They were unanimous on 81 questions and split on 47. One author then reviewed every question in a single-item interface showing the same material and the candidates' inventories, without any resolver output or score: the 47 splits first, then the 81 unanimous nominations, each to be confirmed or overturned. Of the 81, 71 were confirmed and 10 overturned to *none*; of the 47, 15 follow the majority, 6 another candidate, 21 are *none* and 5 were left *unsure*. The result is a single page for 92 questions, no single page for 31 (comparisons of two forms, multi-page workflows, search results not among the candidates) and 5 open. Against the 92, the top page of v1, v2 and v3 is the confirmed page 24, 29 and 37 times and is among the top three 58, 69 and 74 times. Of the 41 licences credited under v1, 7 are on the confirmed page, 26 on another page, 7 on questions with no single page and 1 on an open one; the blinded page check of App. L, which judges each page on its own rather than against the confirmed one, accepts 14.

The right-, wrong-page, half-label and gated arms were rerun with this page as the anchor, the placebo excluding it, and no closure or inventory at all on the 36 questions without a page, in both members of every pair, so that a pair differs only in which page is shown. Sound-premise questions carry no label and their returns are byte for byte those of v1, so their accuracy is not rerun. Under the paper's configuration the right-page inventory licenses 29 of 96 correct denials (30.2%, the self-reported and verified readings agreeing) and the wrong-page inventory 0 of 91; 27 of the 29 are licences v1 also credited. Half the labels leave the self-reported reading at 30.0% (27/90) and take the verified one to 18.9% (17/90); the gate costs $-24.22\,pp^*$ (4/35) and lifts the licence to 44.6% (29/65). Denial accuracy moves $+3.91\,pp$ between wrong and right page ($[-3.1, +10.9]$, 13/8), and $-4.69\,pp$ against v1's right-page arm ($[-11.7, +1.6]$). Under the official reader configuration (thinking on, $T = 0.6$, seeds 0–2) the right-page arm scores 75.0 / 74.2 / 78.9 and licenses 30.2% / 30.5% / 31.7%; the wrong-page arm 63.3 / 69.5 / 76.6 and 0%; the paired gap is $+11.7$, $+4.7$, $+2.3\,pp$ (mean $+6.3$; 57 wins / 33 losses pooled), and the same three seeds of the v1 pair give $+3.9$, $+7.8$, $+3.1$ (mean $+5.0$). On the second reader family (App. F) the pair scores 15.6 against 15.6 (10/10) with licence 30.0% (6/20) against 0.0% (0/20). The blinded page-and-object check of App. L, applied to the 29 licences, is here a model-against-human comparison on the page: the two checkers place 22 and 24 of 29 on the confirmed page, 9 and 13 on the object, 8 and 11 on both; the author, reviewing all 29, finds the object right for 18 (App. L). The human anchor is the best gold we have and is the anchor of §4.3 and Figure 3; the harness anchor is reported beside it because every other arm of the paper shares it.

What changes and what does not. Five predictions were registered in v3's plan file before it ran: the page-correct share of credited licences exceeds v1's and v2's, with $> 50\%$ as the bar for calling the anchor reconstructed; the wrong-/right-page separation persists; denial accuracy stays inside the rerun-to-rerun spread of v1's 79.7; the absolute licence rate is not predicted in direction; sound-premise accuracy stays inside the rerun-to-rerun spread of v2's. All five hold, and every structural result holds under every anchor. The wrong-page inventory licenses nothing under the self-reported reading by construction, and the verified separation keeps its sign under all three (13.8%, 15.2% against 40.2%, 40.8%, 38.4%). Same-anchor thinning still splits the readings under v2 (denial accuracy and the self-reported reading flat, the verified reading 18.4% against 40.8%); the gate still costs denial accuracy ($-25.78\,pp^*$ under v2) and still raises the licence to 50.8%. Denial accuracy, which the note sets rather than the anchor, moves by $-3.12\,pp$ (v2) and $-2.3\,pp$ (v3, CI $[-9.4, +4.7]$), inside the rerun-to-rerun spread; sound-premise accuracy moves $-0.3\,pp$ between v2 and v3. What the anchor does change is how often the credited licence is on the page the question is about, under the blinded check of Appendix L: 15–16 of 41 under v1 (37%), 18 of 40 under v2 (45%), and 20 of 38 under v3 (52.6%, the two checkers agreeing on 34 of 38, $\kappa = 0.79$); the author's review, which covers v1 and v2, puts 14 and 17 there. Only the twin puts more than half the credited licences on

the right page, and we report it beside v1 rather than in its place because the object criterion remains where the checkers disagree ($\kappa = 0.26$ under v3). The thinning and gate arms were not rerun under v3: both are within-anchor comparisons whose page keys, verdicts and note are held fixed, and the v1 and v2 reruns already show they do not depend on the ranker.

Table 23: **Four page anchors reproduce every structural result of the instrument; the human-confirmed anchor sets the absolute rate at** 30%. *v1* is the harness resolver, *v2* the page-type-constrained one, *v3* the source-question twin, *v4* the human-confirmed page (no inventory on the 36 questions without one); nothing else changes between columns. Sound accuracy under v4 is v1's by construction (unlabelled questions, byte-identical returns). Self-reported and verified are the two licence readings; the last block is the blinded page-and-object check of the credited licences by two external families (App. L), `deepseek-v4-pro` / `kimi-k3`; under v4 the page row compares the checkers with the human. Last row: the author's verdict where one exists, else the checkers' (App. L).

	v1 anchor		v2 anchor		v3 (twin)	v4 (human)	
Quantity	right	wrong	right	wrong	right	right	wrong
Denial accuracy	79.7	73.4	76.6	71.9	77.3	75.0	71.1
Sound accuracy	38.7	31.9	37.5	32.2	37.2	(38.7)	(31.9)
Licence, self-reported	40.2%	0.0%	40.8%	0.0%	38.4%	30.2%	0.0%
Licence, verified	40.2%	13.8%	40.8%	15.2%	38.4%	30.2%	0.0%
Half the labels: self-reported / verified	36.9% / 13.6%		41.8% / 18.4%		—	30.0% / 18.9%	
Licence gate: denial acc. / licence	50.8 / 50.8%		50.8 / 50.8%		—	50.8 / 44.6%	
Blinded check, page correct	15 / 16 of 41		18 / 18 of 40		20 / 20 of 38	22 / 24 of 29	
Blinded check, page and object	6 / 8 of 41		8 / 11 of 40		6 / 9 of 38	8 / 11 of 29	
Author's review, page / both	14 / 7 of 41		17 / 10 of 40		—	29 / 18 of 29	

Why no ranker fully recovers the page. The recording does not identify its own pages reliably, which is why the anchor has to come from the release. We keyed every recorded state to a page scope by normalised URL (instance host, path, sorted parameter names; dialogs nested under their page) and grouped scopes into logical pages by stripping the instance host and parameters. Enterprise: 300 scopes fall into 87 logical pages; 42 of those pages are split across instances or URL variants, and the fragments of split pages are 255 of the 300 scopes and hold 3,204 of 3,512 scope observations (91%; the 3,358 states plus 154 dialog views, each counted under its page and its dialog); the incident list alone is 17 scopes, the expense-line list 18. Titles are not a substitute: 13 scopes carry a WorkArena task goal as their document title ("Filter the change request list. | Private Task | ServiceNow") and 25 carry only the navigation shell ("Classic | Unified Navigation App"), and a task-goal title that happens to name a page is what pulled v1 onto Visual Task Board pages for questions about the change-request form. Web: 145 scopes, 121 logical pages, 11 split (35 scopes, 34% of states), 11 scopes with an empty title. An inventory compiled from one scope therefore enumerates one fragment of a page, and its "exhaustive" family is exhaustive for that fragment: the page p of §4.3's definition is one scope, with its dialogs, and $L_F(p)$ is what the recording renders for that scope, not the union over the instance and URL variants of the same logical page.

L THE LICENCE CHECK

The *verified* reading of the licensed-denial rate (§4.3) is computed by replaying the truncated context each arm delivered and checking it against the recording, not from a field the system emitted; the *self-reported* reading trusts the arm's own inventory. For each correct denial the check asks whether the delivered context contains an exhaustive enumeration of an attribute family of the page the question is about, in which the queried object does not appear. Completeness is judged against the labels the recording holds for that page, so enumerating a different page exhaustively cannot earn a licence. Both readings are object-bound: the absent finding must be about the presupposed object itself (its content words must occur in the question), not about an unrelated entry.

Replay. The original runs retained token counts and coverage metadata rather than the rendered context, so contexts are replayed deterministically from the frozen retrieval configuration. The replayed context matches the recorded token count exactly on 398 of 451 questions for the right-page inventory and 447 of 451 for the wrong-page inventory, and differs by at most 16 tokens elsewhere.

A difference that small can still move one item across the truncation boundary, so the paper reports the strictest completeness cutoff and Table 24 sweeps it. As a direct check, we reran the full-, half- and three-quarter-label arms while saving the post-truncation context, reader messages, token IDs and SHA-256 before generation: reading these frozen inputs reproduces the replay's per-question verdict on $128/128$ items in each arm, and under a fresh scoring of the rerun answers the verified rates are $39/93 = 41.9\%$, $14/99 = 14.1\%$ and $23/98 = 23.5\%$, so half coverage again collapses and three-quarter coverage recovers.

Table 24: **The licence readings of the right-page and gated arms do not depend on where the completeness cutoff is set.** Verified licensed-denial rate from replayed context, swept over the cutoff; 1.0 is the reading used in the main tables: every recorded label of the family must appear. n is the number of correct denials. The right-page and gated arms are unchanged across the sweep; the wrong-page arm and the arm with no inventory move by a few points, which is why the paper reports the strictest cutoff.

System	n	Cutoff 1.0	Cutoff 0.8	Self-reported
Empty store + note	73	0.0%	1.4%	*undefined*
Baseline + note, no inventory	92	13.0%	15.2%	*undefined*
same context, `deepseek-v4-pro` reader	59	13.6%	15.3%	*undefined*
Wrong-page inventory + note	94	13.8%	19.1%	0.0%
Right-page inventory + note	102	40.2%	40.2%	40.2%
Right-page inventory + gate	65	50.8%	50.8%	50.8%
Wrong-page inventory + gate	57	17.5%	21.1%	0.0%

On every arm that emits an inventory, the verified reading at cutoff 1.0 agrees exactly with the self-reported reading on the right-page and gated arms. It disagrees on the wrong-page arms, because those arms' own inventories never name the right page even when retrieval happens to deliver it. The same-page thinning of §4.3 (Figure 3b) is the control that does not depend on the page being right: page keys and findings are those of the right-page inventory on all 128 denial questions and only the listed-label fraction changes. Under the harness anchor, at keep-fractions 0.25 / 0.50 / 0.75 / 1.0 the denial accuracy reads 79.7 / 80.5 / 75.0 / 79.7 (all paired differences against the full inventory inside the rerun-to-rerun spread), the self-reported reading 39.2% / 36.9% / 37.5% / 40.2%, and the verified reading 13.7% / 13.6% / 27.1% / 40.2%: the verified reading collapses to a floor at or below half coverage and climbs back toward the full inventory, while the two readings that trust the inventory's own claim do not move. Under the confirmed anchor (the lines of Figure 3b; arms rerun with that anchor) the same thinning reads 73.4 / 70.3 / 73.4 / 75.0 in accuracy, 35.1% / 30.0% / 29.8% / 30.2% self-reported and 20.2% / 18.9% / 23.4% / 30.2% verified ($19/94$, $17/90$, $22/94$, $29/96$): every thinned inventory licenses fewer of its denials than the full one, the fall is not proportional to the dose, and the readings that trust the inventory's claim do not move. The empty store licenses none of its denials at the strictest cutoff. The baseline with the note and no inventory licenses 13.0% of its denials, which is the quantity a self-reported metric could not assign. The gated arms were registered before they ran with two predictions, that denial accuracy falls relative to the right-page inventory and that the licence rises toward the ceiling the anchor sets; both hold ($79.7 \rightarrow 50.8$; $40.2\% \rightarrow 50.8\%$), and the registration ships with the supplement.

The confirmed page anchor. Table 24 uses the harness resolver's page. The main text reports the same readings against the human-confirmed anchor of App. K (arms rerun with that anchor; frozen reader inputs saved before generation, so no replay is involved): the right-page inventory licenses 29 of 96 correct denials (30.2%), the wrong-page inventory 0 of 91, the half-label inventory 17 of 90 (18.9%; self-reported 27 of 90, 30.0%), the quarter- and three-quarter-label inventories 19 of 94 and 22 of 94 (20.2%, 23.4%) and the gated arm 29 of 65 (44.6%). For the two arms without an inventory the same construction reads the frozen memory contexts of the official-configuration replays against the confirmed page; with no coverage record of their own, the absence is read from the reference closure run on that page: the empty store with the note licenses 0 of 73 correct denials and the baseline with the note 4 of 92 (4.3%; 5 of 96 under the thinking reader), against 13.0% under the resolver's page. Of the 75 credited denials the note adds to the baseline (§3.2), 2 are licensed against the confirmed page and 9 against the resolver's. The gap between the two anchors is the resolver's error: of the 41 licences it certified for the right-page inventory, the blinded page check below places only 14 on the page the question is about (the model checkers alone: 15–16).

The inventory arms beside precision and the pair-conditioned score. Table 25 gives the paper's own inventory probes the columns of Table 9, under the harness anchor. The three ungated arms produce 125–136 false denials on the 295 sound-premise questions and deny with 41–45% precision; the gate halves the false denials and takes precision to 53%, still below the baseline's (80% model-labelled, 95% author-labelled).

Table 25: **The inventory arms re-reported beside their cost.** Columns as in Table 9: false denials on the 295 sound-premise questions, denial precision and the pair-conditioned score; harness anchor.

System	Overall	Sound	Denial	False den.	Denial prec.	Pair-cond.
Right-page inventory + note	50.3	38.7	79.7	**125**	45%	32.8
half the labels listed	45.2	31.3	80.5	**136**	43%	30.5
reader gated on it	38.6	33.7	50.8	57	53%	18.8
Wrong-page inventory + note	43.7	31.9	73.4	**136**	41%	28.1

A judge shown the delivered context. The mitigation a reader will propose before any licence is to let the judge see the memory context and credit a denial only when that context supports it. We ran that scorer over every denial the official answer-only judge credited in five arms: the official rubric verbatim, followed by the delivered context as replayed above, and one added rule — credit only if the delivered context contains evidence that the presupposed item does not exist, for example a complete list of the relevant items of the page in which it does not appear, and not if the only support is an assertion that something is absent or that the recording is complete. Four predictions were registered in the script before any call: the empty store loses its credit; the half-label inventory is not separated from the full one; the wrong-page inventory may be partly caught; the baseline with the note keeps much of its credit. All four hold under both judges below. Two judges ran it: `deepseek-v4-pro` (temperature 0; the figures quoted in §4.3) and `glm-5.2` (thinking off, temperature 0). The empty store loses its credit: 0 of 73 kept under `deepseek-v4-pro`, 1 of 73 under `glm-5.2`. The half-label inventory is not separated from the full one: 75 of 103 against 76 of 102 kept (72.8 against 74.5%) under `deepseek-v4-pro`, 99 of 103 against 99 of 102 under `glm-5.2`, where the verified reading licenses 14 against 41. The wrong-page inventory is partly caught by one judge: `deepseek-v4-pro` keeps 54 of 94 against 76 of 102 for the right page, a 17-point separation that still credits more than half of an arm whose inventory never names the right page; `glm-5.2` keeps 91 of 94, no separation. The baseline with the note and no inventory keeps much of its credit: 45 of 92 under `deepseek-v4-pro` and 77 of 92 under `glm-5.2`, where the verified reading licenses 12. The two judges agree that the empty store has no support and disagree on how much a partial page slice or a self-described "exhaustive" list proves; neither has the recording, so neither can tell a complete family from a thinned one. Of the 76 right-page denials `deepseek-v4-pro` keeps, 44 are ones the delivered context does not license. The same scorer over the confirmed-anchor arms (Table 5) keeps 76 of 96 right-page, 48 of 91 wrong-page and 71 of 90 half-label denials (79, 53 and 79%): the half-label inventory is again indistinguishable from the full one.

What the credited licences assert, checked blind. Object-binding is lexical and the page anchor is a resolver, so two facts a licence asserts are never verified by the rate itself: that the anchored page is the page the question is about, and that the absent finding is about the object whose absence the gold asserts. We checked both for every credited licence of the right-page inventory under each of the four anchors (App. K) and of the gated arm. Each was shown, with the question, the gold flaw, the resolved page's key and title, the attribute families the recording renders for that page, and the credited absent finding — no score, no other arm — to two external model families under one prompt, blind to each other: `deepseek-v4-pro` at temperature 0 and `kimi-k3` at its fixed temperature 1. Each returned two booleans with a one-sentence reason. Both are model-based checks, released in full. An author then reviewed, with both checkers' verdicts withheld (each item marked only as a disagreement or a sampled agreement), every licence on which the two checkers disagreed and a fixed sample of eight agreements per arm under the harness and v2 anchors, and all 29 licences under the confirmed anchor (below); where an author verdict exists it is the reported one.

Of the 41 credited licences under the harness anchor, `deepseek-v4-pro` finds 15 on the right page, 15 on the right object and 6 on both; `kimi-k3` finds 16, 23 and 8. On the page they agree for 40 of 41 ($\kappa = 0.95$). On the object they agree for 31 ($\kappa = 0.53$): one checker refuses generic

subjects ("control label", "single-click item") that the other accepts as paraphrases of the queried object, which is why every disagreement also went to the author (below). Of the gated arm's 33, the two find 13 and 14 on the page, 8 and 19 on the object, 3 and 7 on both. Under the twin anchor the same check finds 20 and 20 of 38 on the page ($\kappa = 0.79$), 9 and 19 on the object, 6 and 9 on both; the licences anchored by the twin's gold phrase are on the right page 11 of 18 times (`deepseek-v4-pro`), those anchored by page type alone 8 of 15, and the five on the fallback 1 of 5. The failures are not subtle. A question about the Change Request form ("what is the name of the fifth button that appears to the right of that field") is resolved onto an Incident form and licensed by a "fifth button" absent there; a question about the duplicate-marking popup is resolved onto a Visual Task Board page. Where the page is right the object can still be wrong: a denial that "there is no third button" is licensed by findings that the "right field" and the "open/show referenced record icon" are absent, the second of which the question presupposes is present. Several residual failures are questions that compare two forms or ask about a multi-page workflow, for which no single page anchor can be right; the rest are the page-identity problem of App. K. The licences that survive both checks are the shape the instrument was designed for: the Personalize Form dialog's 43 checkboxes enumerated with no amortization-related label; the Search Terms grid's seven columns with no metric column between `Store` and `Results`.

Three consequences are drawn in the main text. The right-/wrong-page separation and the same-page thinning do not depend on the page being right, because both arms and every thinning level are anchored on the same page and the check reads whatever context that page contributed. The gate's price is the price of obeying the system's own check, imperfect as it is. And the absolute rate is an upper bound on licensing in the strict sense: under this check, 9–28% of credited licences are licences, depending on the anchor, the arm and the checker (17–25% on author review); under the human-confirmed anchor (App. K) the rate itself is 30.2%, and the checkers put 22 and 24 of its 29 licences on the confirmed page, 9 and 13 on the queried object and 8 and 11 on both. An author reviewing all 29 with the checkers' verdicts withheld, under the same two questions, confirms the page for every one (it is the page the author had already confirmed as the anchor) and the object for 18: the eleven failures are findings that deny a container rather than the member the gold denies (a "left column" for a tab in it, a "right column" for a field), findings whose subject is a template phrase from the question ("three buttons", "names two buttons"), findings about an element the question presupposes present, and one about a step where the gold denies a saved filter. The strict-sense rate under that anchor is thus 18 of 96 (models: 8–11). Of the four licences of the baseline with the note (§4.3) the author confirms the object on three; the checkers on one and two; the fourth licenses a "left column" where the gold denies a tab between two others in it, so at most 4 of 92 stand and 3 do. The same review covers the thinned and gated arms under that anchor: of their 87 credited licences, 78 repeat a question and finding among the 29 and inherit its verdict, and the author reviewed the other 9 (four distinct findings) blind. The object stands for 12 of 19, 11 of 17, 15 of 22 and 19 of 29, so the strict-sense rates are 12.8% (12/94) at a quarter of the labels, 12.2% (11/90) at half, 16.0% (15/94) at three quarters and 29.2% (19/65) for the gated arm: thinning still lowers the licence from the full inventory's 18.8%, and obeying it still raises it. With the harness anchor the author puts 14 of the 41 right-page licences on the right page and 22 on the right object (7 on both; models 15–16, 15–23, 6–8), the gated arm's 33 at 10, 22 and 6, and under the v2 anchor 17, 24 and 10 of 40 and 13, 18 and 6 of 33; on the sampled agreements the author concurs with the checkers on the page 29 of 32 times and on the object 28 of 32. The wrong-page arm's 0 needs no such correction. The cure is the answer-bearing anchor mapping the release holds and strips (its data card states that the public files "intentionally remove … answer-bearing annotation labels"); the source-question twin recovers half of it from the public text (App. K).

M LongMemEval-S and LoCoMo

M.1 LongMemEval-S

Table 26 gives the comparison summarised in §3.5. The *two-sided* policy is a second directive we wrote; like the note, it asserts that the context is a complete recording, which our own measurement contradicts (the gold reaches the reader 67.3% of the time).

On LongMemEval-S an open-world refusal is accepted, so the note takes abstention from 66.7 to 100.0 and costs 20 *pp* on the other questions, concentrated on aggregation: temporal reasoning falls

Table 26: **The note is benchmark-specific; neither policy is neutral.** LongMemEval-V2 is the full 451-question greedy protocol; LongMemEval-S is the official v1 rubric, 120 stratified questions, BM25, also greedy. S contains every abstention question the benchmark has.

	LongMemEval-V2			LongMemEval-S		
Policy	Overall	Sound	Denial	Overall	Sound	Denial
none	34.4	**41.8**	15.6	74.2	**76.7**	66.7
the note	**51.7**	43.7	**71.9**	67.5	56.7	**100.0**
two-sided	38.4	37.2	41.4	**76.7**	72.2	90.0

from 66.7 to 25.0 and multi-session from 65.2 to 34.8. On V2 a denial is required and `UNKNOWN` scores zero, so the same sentence is worth $+17.29$ *pp* overall. The sign is a property of the scorer's credited answer, not of the store.

M.2 LoCoMo: scorer, released stores, decoder battery

This section supports Table 2 and §3.5. LoCoMo (Maharana et al., 2024) is the other standard evaluation for agentic memory and carries the same kind of denial questions as LongMemEval-V2 (category 5, 446 items, 22.5% of the benchmark). Its scoring is already a documented sore point: a public ground-truth audit finds 99 score-corrupting errors in the non-adversarial categories and notes that category 5 has no ground-truth answers and is excluded from scoring,[2] and the benchmark's own issue tracker discusses the judge's leniency and the absence of a standard judge prompt, with category 5 left out of the tallies.[3] What follows is not a repeat of those audits but a controlled measurement of the memory-context channel on this benchmark. We audited it with the benchmark's own code, imported rather than reimplemented, so every verdict below is the benchmark's. Every released store in this appendix (Mem0, A-Mem) is rebuilt with the paper's local reader and embedder, so the arms measure what one sentence does to a real store; none re-scores a published number. Two protocols have to be kept apart.

Under the official protocol category 5 is unrunnable, not gameable. Every inference script in `task_eval/` rewrites a category-5 item as `question + " Select the correct answer: (a) Not mentioned in the conversation (b) {answer}"`, maps the model's letter back to the corresponding string with `get_cat_5_answer`, and only then applies the substring test in `evaluation.py`. That test decodes a two-alternative forced choice, so chance is 50% and no fixed output can farm the column. The defect is upstream: option (b) is the item's gold `answer`, and only 2 of the 446 items carry that key. Replaying the construction raises `KeyError` on the other 444. A benchmark that devotes 22.5% of its questions to denial therefore ships no way to evaluate it, which is consistent with the column being absent from reported aggregates.

Under the free-text protocol the surviving scorer is degenerate. Anyone reimplementing category 5 — and anyone deploying a memory system, which is not handed two options — confronts the substring test alone. Applied to free text it credits a refusal substring and reads neither the gold answer nor the conversation (Table 27).

The gold row scores 0.00 on category 5 because there is no gold string to emit. The no-policy 11.4% is not refusal skill: official credit is exactly the event that the prediction contains `not mentioned` (51/51 wins, 0/395 losses); 44/51 wins are that string alone, 4/51 also take the bait, against 203/395 of the losses. The transferred note does not raise category 5 under the official scorer; it falls, 11.43 → 0.45 (2 wins / 51 losses), because the wording it produces, `no information is available`, is not one of the two credited substrings (Table 28): it appears on 382/446 note answers and on all 446 empty-store answers, while bait-taking falls from 46.4% to 9.6%. If the official test also accepted that wording, category 5 would read 11.4, 85.7 and 100.0. Retrieval delivers an annotated evidence session for 98.2% of items that have one, so none of these numbers is a retrieval miss. Pinning the local codeword on the retrieving baseline takes category 5 to 87.9, not 100 (392/446; 341 new wins, 0 losses against no-policy) and costs categories 1–4 -4.27 *pp* (221 wins / 354 losses); none of the 54 misses contain `not mentioned`, 32 take the bait and 3 are

[2] https://github.com/dial481/locomo-audit, file `AUDIT_REPORT.md`.
[3] https://github.com/snap-research/locomo/issues/23, opened October 2025.

Table 27: **On LoCoMo's surviving scorer a constant `not mentioned` scores 100 on category 5, a retrieving reader 11.4, and the transferred LongMemEval-V2 note 0.45.** The official protocol is a forced choice that cannot be constructed for 444 of the 446 items; these are what the surviving scoring rule credits under the free-text setting a reimplementation or a deployment faces. The first four rows use no memory. The bait row answers every question with the `adversarial_answer` the release ships and scores as well on categories 1–4 as the gold does, which is what makes those items adversarial rather than badly written. The retrieving rows are a real reader with BM25 retrieval held byte-identical and vary only the answer policy. The codeword arm is told to emit exactly `not mentioned`; its leftover credit on categories 1–4 is token overlap against golds containing those words.

Predictor	Overall	cat 1 n=282	cat 2 n=321	cat 3 n=96	cat 4 n=841	**cat 5** n=446
Constant `not mentioned`	22.80	0.71	0.16	3.50	0.11	**100.00**
Reader, empty store + `not mentioned`	22.60	0.07	0.04	1.06	0.17	**100.00**
Always take the bait	77.29	100.00	100.00	92.74	100.00	0.45
Gold answer where one exists	77.19	100.00	100.00	92.74	100.00	0.00
RAG, no policy	42.61	37.09	25.31	25.61	69.53	11.43
RAG + transferred note	38.02	37.14	20.10	13.76	67.85	0.45
RAG + local codeword	56.46	34.62	19.17	10.37	66.62	87.89
RAG + Mem0's shipped policy	40.19	38.80	24.15	23.86	69.72	0.45
Empty store + transferred note	0.49	0.71	0.12	4.15	0.41	0.00

a bare English `no`, so evidence can override the codeword. Mem0's shipped policy ("NEVER say 'not mentioned"') applied to the same store takes category 5 to 0.45 (1 win / 50 losses) and moves categories 1–4 by $+0.06\,pp$ (258 wins / 257 losses); bait stays high (39.0% against 46.4%). The transferred note also scores 0.45, but by the opposite route: an English refusal the scorer does not credit.

Table 28: **Official accuracy and the refusal wording move in opposite directions on LoCoMo category 5.** Official credit is the two-string test of `evaluation.py`. *Instructed* is the wording the transferred note asks for. *Bait* is a token-overlap flag against the released `adversarial_answer`. $n = 446$.

System	Official cat 5	Instructed phrasing	Bait
RAG, no policy	11.4	0.0%	46.4%
RAG + transferred note	0.45	85.7%	9.6%
RAG + local codeword	87.9	0.0%	8.1%
RAG + Mem0's shipped policy	0.45	0.0%	39.0%
Empty store + transferred note	0.0	100.0%	0.0%
Live reader, empty store + `not mentioned`	100.0	0.0%	0.4%
Offline constant `not mentioned`	100.0	—	—

Mem0's own harness drops category 5, so nothing here re-scores its published ranking; the result bears on any evaluation that does pool category 5, as A-Mem's does. The scorer that survives the official LoCoMo path is thin enough that the intended refusal and the credited string can be different words; LongMemEval-V2 leaves its credited answer free; A-Mem runs a constructor and scores the opposite option. Released evaluations of the same items then drop the column, patch the judge, or keep the residual substring (Table 20).

A released memory system, end to end. The census can be read as a claim about harnesses rather than systems, so we ran one released system whole: mem0ai 2.0.19's open-source pipeline builds the store (its V3 additive extraction over all 272 LoCoMo sessions, one LLM extraction call per session), its top-10 memory search retrieves, and the paper's frozen reader answers. Only the answer policy varies at QA time; store and retrieval are byte-identical across arms. Construction uses the paper's local reader as the extraction LLM and a local MiniLM embedder with dense-only search (Table 29). The shipped policy zeroes category 5 on the system's own store, $42.6 \rightarrow 0.0$ (0 wins / 190 losses paired), while categories 1–4 move $+0.21\,pp$. The pinned codeword takes it to 96.9, not 100: evidence can still override the codeword, as on the BM25 store. The cross-system ordering is a property of the sentence: with each system's shipped configuration Mem0 loses category 5 to the BM25 baseline (0.0 against 11.4); with the codeword pinned on both it wins (96.9

against 87.9, +8.97 *pp*, 44 wins / 4 losses), while the content comparison on categories 1–4 barely moves (−6.07 *pp* unpinned, −9.78 *pp* pinned, both favouring BM25). One number explains the store's high no-policy category-5 score: all 190 of its credited answers contain `not mentioned` and none takes the bait, against 41 bait-takers among the 256 losses. A 306-token return of extracted facts tempts the reader less than 12K tokens of raw session, so the accidental codeword fires more often. It is still a codeword, not a check.

The same lever moves a second released store. We repeated the policy arms over the A-Mem store whose construction is described below (QA-only over the cached notes; one deviation, applied identically in every arm: retrieval queries use the raw question rather than A-Mem's keyword-extraction step). The pinned codeword takes category 5 from 28.7 to 94.6 (+65.92 *pp*, 294 wins / 0 losses paired) while categories 1–4 fall −5.65 *pp*; the transferred LongMemEval-V2 note is an official zero on this store too (28.7 → 0.0, 0 / 128). Across the three real stores the no-policy spread of 31.2 points (11.4 / 28.7 / 42.6 for BM25, A-Mem and Mem0, ordered by how much bait each return carries) collapses to 9.0 points (87.9 / 94.6 / 96.9) once the codeword is pinned.

Table 29: **On two released stores the category-5 score is set by the sentence: Mem0's shipped policy zeroes it and the pinned codeword saturates it.** Mem0 OSS end-to-end and the A-Mem store on LoCoMo, official scorer, greedy; within each block the store and retrieval are byte-identical across rows. BM25 reference rows in Table 27.

Arm	Overall	Cat. 1–4	**Cat. 5**	`not mentioned`
Mem0 store, no policy	44.9	**45.6**	42.6	42.6%
+ its shipped policy	35.5	**45.8**	0.0	0.0%
+ the local codeword	**50.9**	37.6	**96.9**	96.9%
A-Mem store, no policy	33.7	**35.2**	28.7	28.7%
+ transferred note	24.4	31.5	0.0	0.0%
+ the local codeword	**44.2**	29.5	**94.6**	94.6%

A published evaluation scores category 5 against the bait. A-Mem (Xu et al., 2025) states that category 5 tests identification of unanswerable queries, includes that column in the published F1/BLEU aggregate, and in the released evaluation (`WujiangXu/AgenticMemory`, commit `0c8039f2`; `load_dataset.py` ll. 19–20, `test_advanced.py` ll. 147–154 and 341) sets `QA.final_answer` to `adversarial_answer` when `category==5`. Token-F1 is computed against that field. The category-5 prompt is a two-alternative forced choice whose options are "Not mentioned in the conversation" and the same bait. Official LoCoMo's credited substring therefore scores 0 under A-Mem's scorer, and the memory-free generator of A-Mem's credited answer is "emit the item's `adversarial_answer`", which leaks the label (Table 30). Their Table 1 treats the Adversarial column as ranking-relevant (LoCoMo 69.23 vs A-Mem 50.03 F1 under GPT-4o-mini).

Swapping the gold swaps which of {codeword, bait} wins (Figure 5); the transferred-note arms lose under both golds because they emit an English refusal that neither scorer credits. A-Mem's 2AFC constructor, unlike the official LoCoMo path, can be built for every item. We replay it on the same reader, greedy (A-Mem samples category 5 at temperature 0.5), with option order hashed from the question id. A system that stores nothing picks "Not mentioned in the conversation" on 96.6% of items (official credit 96.6; A-Mem F1 7.8). The same constructor with BM25 retrieval, byte-identical to the free-text RAG arm, drops official credit to 60.3 and raises A-Mem F1 to 42.6 (bait rate 5.8% → 39.7%): retrieval increases bait-picking on their protocol too. We then ran A-Mem's released memory (note + evolution, retrieve k=10) on the same ten conversations, with construction by this paper's reader. Official category 5 is 75.6; A-Mem F1 is 28.6. Of 337 answers that emit "Not mentioned in the conversation", official credit is 100 and A-Mem F1 is 5.5; of 118 with bait-overlap at least 0.5, official credit is 7.6 and A-Mem F1 is 98.0. Their store emits the codeword on three items in four; when it does not, it emits bait and the two golds trade places. Retrieval changes the mix of answers; the scorer pays for the wording.

The decoder battery. Table 31 is the full battery behind Table 2. Its frozen traces (bottom panel) are scored under MAGMA's released unanswerable test and HiGMem's patched gold as well; RE-Mem's category-5 F1 is the official substring test, so it is not a fifth column. The transferred

Table 30: **Official LoCoMo category-5 credit and A-Mem's gold reverse which answer wins.** A-Mem F1 is set-token overlap against `adversarial_answer`, matching `utils.simple_tokenize` in the A-Mem release, in percent. $n = 446$. Offline rows use no reader. The 2AFC rows replay A-Mem's category-5 prompt, greedy, on our store. The last row is A-Mem's released memory (note+evolution, k=10), greedy QA, local reader — not GPT-4o-mini, not their Table 1 temperature.

System	Official cat 5	A-Mem F1 vs bait
Constant `not mentioned`	100.0	0.6
Always emit the bait	0.45	100.0
Live reader, empty store + codeword	100.0	0.6
RAG + local codeword	87.9	5.7
RAG, no policy	11.4	28.9
RAG + Mem0's shipped policy	0.45	29.5
RAG + transferred note	0.45	8.3
A-Mem 2AFC, empty store	96.6	7.8
A-Mem 2AFC, RAG	60.3	42.6
A-Mem store (ours, greedy)	75.6	28.6

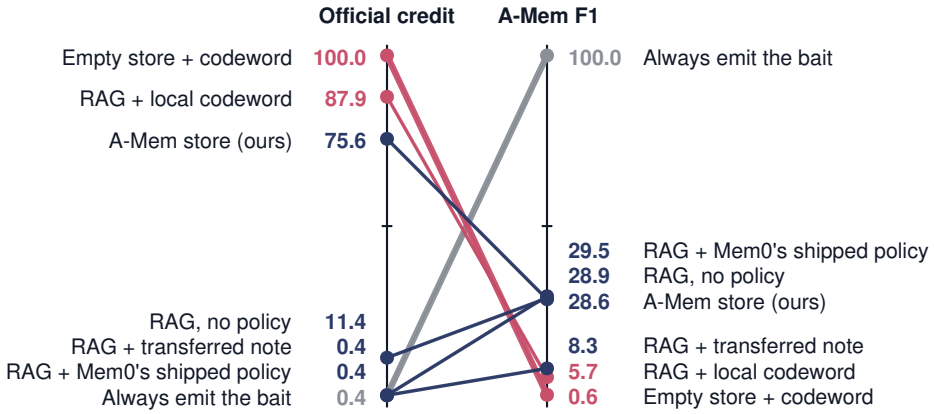

Figure 5: **Swapping A-Mem's gold swaps which answer wins on the same traces.** Each line is one arm on the same 446 category-5 items: official LoCoMo credit on the left axis, A-Mem set-token F1 against `adversarial_answer` on the right. The two anchors (always emit the bait; empty store + codeword) exchange first and last place; the BM25 live arms and the A-Mem store rerun sit between. The offline constant `not mentioned` duplicates the empty-store codeword arm and is omitted.

LongMemEval-V2 note is 0.45 official and 85.7 MAGMA; constant `UNKNOWN` is MAGMA-perfect and an official zero; `I don't know` is zero under every scorer we ran. MAGMA's own prompt string `Information not found` is also MAGMA-perfect and an official zero.

Which pairs disagree, and why. Table 32 gives every scorer pair, because the structure of the disagreement is the argument. Two of the six pairs are strongly concordant. Official against HiGMem is Spearman 0.91: HiGMem patches the gold to `Not mentioned in the conversation`, which the official scorer also credits, so patching the gold this way preserves the ranking. MAGMA against HiGMem is 0.31 on the eight-arm set but 1.00 on the four live retrieving arms. The three discordant pairs all involve A-Mem's scorer, which scores token F1 against the item's `adversarial_answer`, the bait LoCoMo shipped where the gold should be, and therefore credits the opposite answer by construction. The reported 46% and 55% are any-reversal shares over all scorer-pair × arm-pair combinations (162 and 33), not means of the six numbers. The conclusion is narrower than "four scorers disagree": a column in which 444 of 446 items ship no gold forces every reimplementation to invent one, and the invention decides the order.

Table 31: **Each released LoCoMo scorer credits a different constant answer, and A-Mem's credits the bait.** *Top:* every named answer we scored as an offline constant, $n = 446$, no reader; the answer one scorer credits is one another zeroes. *Bottom:* the same four scorers replayed over one frozen set of LoCoMo category-5 traces. Any-reversal shares over all scorer-pair $\times$ arm-pair combinations: 46% of 162 on all eight arms, 55% of 33 on the four live retrieving arms. Table 32 gives all six pairs.

	Official	A-Mem F1	MAGMA	HiGMem F1
Named answers (no reader)				
`not mentioned`	100.0	0.6	100.0	57.1
`Not mentioned in the conversation`	100.0	3.0	100.0	100.0
`UNKNOWN`	0.0	0.0	100.0	0.0
`Information not found`	0.0	0.3	100.0	25.0
`no information is available`	0.0	0.4	100.0	0.0
`I don't know`	0.0	0.0	0.0	0.0
Always emit the bait	0.45	100.0	0.45	3.0
Live arms, one frozen set of traces				
RAG, no policy	11.4	28.9	20.6	13.0
RAG + transferred note	0.45	8.3	85.7	34.6
Empty store + transferred note	0.0	0.4	**100.0**	0.2
RAG + local codeword	87.9	5.7	87.9	50.4
Empty store + local codeword	**100.0**	0.6	**100.0**	57.1
RAG + Mem0's shipped policy	0.45	29.5	6.5	11.2
A-Mem 2AFC, empty store	96.6	7.8	96.6	**95.7**
A-Mem 2AFC, RAG	60.3	**42.6**	60.3	53.6

Table 32: **Scorers that keep the official substring agree ($\rho = 0.91$); those that replace the gold reverse up to every arm ordering.** All six scorer pairs on one frozen set of LoCoMo category-5 traces: Spearman over arm scores and the share of arm pairs whose order reverses, on the eight replayed arms and on the four live retrieving arms with byte-identical BM25 retrieval. Over four arms ρ takes only a few values and one swap moves it by 0.4; the reversal share is the reading, ρ a summary. No memory is rerun; only the released scoring rule changes.

	Eight arms		Four live arms		
Scorer pair	ρ	rev.	ρ	rev.	Gold used
official | HiGMem F1	+0.91	11%	+0.63	20%	patched to the credited phrase
official | MAGMA	+0.31	31%	+0.63	20%	pattern list, no gold
MAGMA | HiGMem F1	+0.31	33%	+1.00	0%	—
official | A-Mem F1	−0.12	56%	−0.63	80%	bait
A-Mem F1 | HiGMem F1	0.00	54%	−1.00	100%	bait vs patched
A-Mem F1 | MAGMA	−0.90	89%	−1.00	100%	bait vs none
Any reversal (all combinations)	46% of 162		55% of 33		