PURETENSOR
INTELLIGENCE DISTILLED

The Reranker Is Not the Lever

A Stale-Versus-Current Eval, and Three Ideas That Measured Nothing

Paper: PT-R-2026-063 v1.0

Author: PureTensor Ltd

Date: 2 November 2026

Status: Published

Abstract

Our agent memory is a set of Markdown notes searched by keyword and dense-vector legs, fused by reciprocal rank fusion (a rule that merges ranked lists by position), then re-scored by a cross-encoder reranker (a model that reads the query and one passage together and returns a relevance score). A live probe suggested this reranker prefers a retired claim to its true negation. We built an evaluation of 198 questions about facts that changed, 169 of them paired so that both a stale and a current passage are live in the index, scored on the top 5 of up to 10 results.

The headline is a null, and its direction depends on the scoring rule. Against the same pipeline with the reranker off, 15 paired items stopped being stale-first and 9 became stale-first (exact McNemar $p = 0.31$), but only under the tool's default rule, which drops any item whose expected file is missing from the top 10 in either arm; 51 of 169 items were dropped. Using every item with the flag as the harness sets it, 17 improved and 20 worsened ($p = 0.74$), and the raw counts are 47 stale-first items without the reranker and 50 with it. The reranker's effect on stale answers is not distinguishable from zero here, and its sign is not stable across rules.

Three cheap ideas for helping the reranker measured nothing useful. Claim-polarity tags cannot change ranking by construction. A metadata header fixed 2 items and broke 4 after the tool drops items ($p = 0.69$). A counter-evidence companion row fired on 2 of 26 held-out items, changed no outcome and added 123 ms at the 95th percentile. A planned edit to the reranker's weights was gated on a feasibility count that, as reported, passed at exactly its threshold; a raw check shows the header arm contains four stale-first items absent at baseline, two of which the feasibility report counts among its eight.

The one significant cell in the study is that the reranker helps the true passage reach the top 5 (27 items gained against 12 lost, $p = 0.024$, recall-failed items scored as losses). That result is exploratory and uncorrected. Section 7 says which claims survive.

1. The Question, and Why It Looked Urgent

Notes about a changed fact tend to come in pairs: one says a service is retired, an older one still says the opposite. Retrieval returns both, and the order matters because agents read from the top.

Our retrieval stack has four stages. A keyword leg and a dense-vector leg each produce candidates from a Postgres store. Reciprocal rank fusion merges them, and up to 50 fused candidates are re-scored by the cross-encoder reranker (bge-reranker-v2-m3). A provenance and recency blend then re-sorts the list. The reranker is served from a GPU endpoint with a 512-token input limit and a CPU fallback, and by default reads only the first 800 characters of each passage. It was already live on every serving path.

The worry came from a single live probe in which the reranker appeared to score a stale affirmative passage far above the true negation. We do not use its scores, because no raw record was retained. Two readings were possible: a preference for passages whose polarity matches the query, or a vocabulary mismatch, since the affirmative passage shares more words with the query. The data below do not separate them.

Three questions follow. Does the reranker put the current claim first when a stale one is present? Can a cheap addition (tags, a metadata header, a companion row of opposing evidence) help? Is it worth editing the weights? A companion study owns the stage-by-stage recall trace and fixes, and we do not retell it.

2. The Evaluation

Each item is a real question about a changed fact, built from our own notes. It carries regular-expression patterns for the current claim (the truth) and the stale or opposite claim (the lure), the file holding the truth, and a polarity type. For a paired item, both chunks must be live in the index. This is a new, larger evaluation than the question set of our earlier memory audit, scored at top 5 rather than top 3, so the numbers should not be compared directly.

An item is lure-first (stale-first) if a lure passage ranks in the top 5 above the first truth passage, or if no truth passage ranks. Otherwise it is truth if a truth passage is in the top 5, then recall-failed if the expected file is absent from the top 10, else no-truth. So the stale-first flag is set even when the expected file is absent. Of the 50 stale-first items in the baseline arm, 24 have no expected file in the top 10.

We ran two versions of the eval, both on 2 November 2026 (UTC).

Eval version	Items	Paired stale-vs-current	Controls	Held-out / context split	Run window
First	57	28	5	26 / 31	18:51 to 19:34
Enlarged (frozen)	198	169	5	95 / 103	22:02 to 22:48

Table 1. The two eval versions. The enlarged set added 141 items and was frozen by content hash before any scoring.

The paired items split into 81 held-out and 88 context by a hash of the item id; polarity types range from 59 paired superseded items down to 4 hedged. Each arm ran three repeats; the baseline arm's verdicts were identical across repeats for all 198 items, so the repeats add no information about variance.

The arms were baseline (reranker on, the companion study's passage-window change off), norerank (reranker off), tags, counter (counter-evidence companion on), header (baseline plus the metadata header) and default (production, with the window change on by default that day). Each arm ran in its own process with its environment pinned, and counted as valid only if every query in a reranker-on arm was actually reranked.

For statistics we used the exact two-sided McNemar test on per-item flips between two arms. The tool's default is the drop rule: an item recall-failed in either arm is removed. Its alternative, recall-as-loss, scores a recall-failed arm as a miss, which is sensible for truth in the top 5; for the

stale-first flag it counts a stale-first item with its file absent as not stale-first, favouring the arm with more recall failures. We therefore also report the flag unscored, as the harness sets it. No test is corrected for multiplicity.

One caution: the source notes were chosen for containing correction sentences, which makes the truth passages cue-rich. A real corpus is probably harder.

3. Does the Reranker Put Current Ahead of Stale?

The first eval, with 57 items, looked like a gain: on the held-out split the stale-first rate was 18.2% (2 of 11 items with lures) with the reranker and 33.3% (4 of 12) without. The denominators exclude recall-failed items, and on raw counts across all 57 items the reranker-on arm has fewer top-5 truths, 31 against 34. The exact tests were not significant (held-out stale-first 3 fixed and 1 broken, p = 0.63; all 57 items on the drop rule 5 against 2, p = 0.45).

We enlarged the set to settle it. Table 2 gives the arms.

Arm	Stale-first items (of 169 paired)	Harness stale-first rate	Truth in top 5 (of 198)	Expected file absent from top 10	Valid
norerank	47	26.2% (33 of 126)	93	47	yes
reranker on (baseline)	50	19.8% (26 of 131)	106	44	yes
header	51	21.2% (of 132)	107	43	no (one item not reranked)
header, 6 s budget rerun	52	22.0% (of 132)	107	43	no (same item)
default (production)	45	19.5% (of 128)	115	45	yes

Table 2. The enlarged eval by arm. The harness rate excludes recall-failed items from its denominator; the item count does not.

On the harness rate, the reranker lowers the stale-first rate from 26.2% to 19.8%. On raw item counts, 47 items are stale-first without it and 50 with it. Of the 47, 14 have the expected file absent from the top 10; of the 50, 24 do. The reranker run therefore has a lower rate and a higher count. Our hypothesis is that turning it on moves items into the absent-file group, and so out of the denominator, but the records do not test why.

Comparison (169 paired items unless stated)	Rule	Improved	Worsened	p
Reranker on vs off, stale-first	drop (n = 118; 51 dropped)	15	9	0.31
same	flag unscored, no dropping (n = 169)	17	20	0.74
same	recall-failed items counted as not stale-first (n = 169)	19	12	0.28
same, held-out only	drop (n = 59)	9	4	0.27
same, context only	drop (n = 59)	6	5	1.0
Reranker on vs off, truth in top 5 (198 items)	drop (n = 141)	18	10	0.18
same	recall-as-loss (n = 198)	27	12	0.024

Table 3. Reranker on against reranker off, paired exact McNemar tests under each rule.

No stale-first test is significant, and the direction depends on the rule: the drop rule and the recall-counted-as-not-stale rule favour the reranker, while the unscored flag, the only analysis

that uses all 169 items as the harness sets them, points slightly against it, matching the raw counts of 47 against 50. With 24 to 37 discordant items the test has little power either way.

The truth-in-top-5 row is the only cell that clears conventional significance, and only under recall-as-loss: 27 gained and 12 lost, p = 0.024. It is exploratory and uncorrected. It says the reranker helps the true passage reach the top 5, not that it orders current ahead of stale, which was the original worry.

Latency was p50 / p95 of 234 / 305 ms without the reranker and 271 / 347 ms with it.

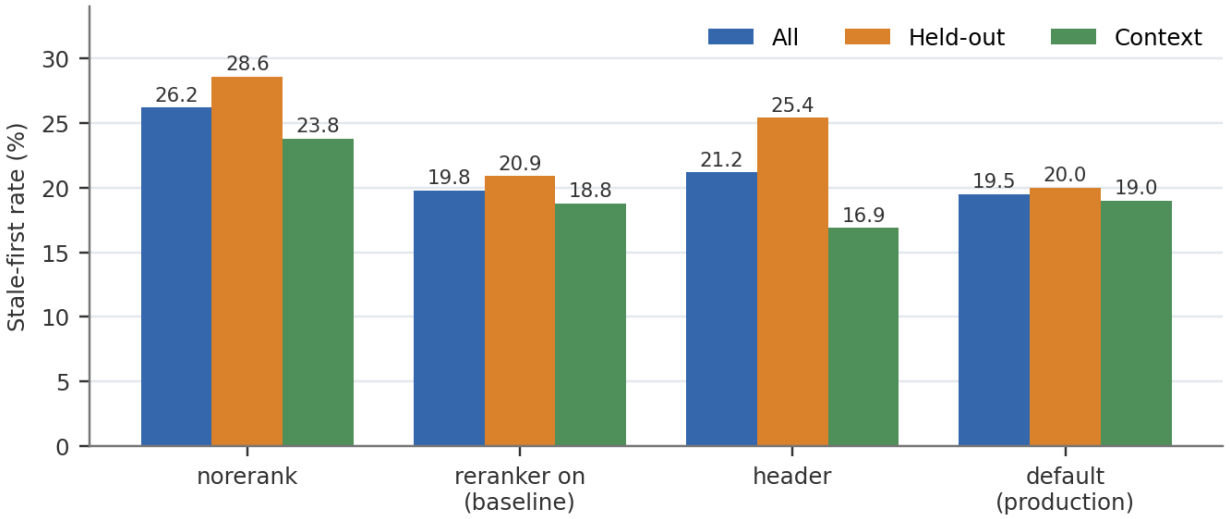


Figure 1. Harness stale-first rate by arm on the enlarged eval, overall and by split. Lower is better; values are in Table 4. The header arm is invalid (one item not reranked).

Arm	All	Held-out	Context
norerank	26.2%	28.6%	23.8%
reranker on (baseline)	19.8%	20.9%	18.8%
header	21.2%	25.4%	16.9%
default (production)	19.5%	20.0%	19.0%

Table 4. Harness stale-first rate by arm and split, recall-failed items excluded from each denominator.

4. Where the Failures Sit

If the reranker is not clearly the lever, where do the stale-first failures come from? The run records give a partial answer. Of the 50 paired stale-first items in the baseline arm, 24 have the expected file absent from the top 10 results and 26 have it present. In 33 of the 50, no truth passage appears in the returned top 10. That is a statement about the final list. The run records do not say whether the truth was in the pool of up to 50 candidates the reranker rescores, so we cannot say how many of the 33 were out of the reranker's reach; a reranker that scored the truth higher might have lifted it into the top 10.

The item set also carries a hand flag, "answer text begins beyond character 800", which marks passages where the reranker's default window could not have seen the answer. It is set on 13 of the 50 stale-first items. The companion study changed that window behaviour, and the production default arm reflects it: 45 items are stale-first, 20 with the expected file absent, and relative to the baseline 10 items were fixed and 5 became stale-first.

Arm	Stale-first items	With expected file absent from top 10	With expected file present
norerank	47	14	33
reranker on (baseline)	50	24	26
default (production)	45	20	25

Table 5. Stale-first items split by whether the expected file appears anywhere in the top 10.

Many stale-first outcomes therefore look like recall problems wearing a ranking symptom: the truth does not arrive, the lure is the best thing available, and the item is scored stale-first. With the reranker on, absent-file stale-first items rise from 14 to 24 while present-file ones fall from 33 to 26.

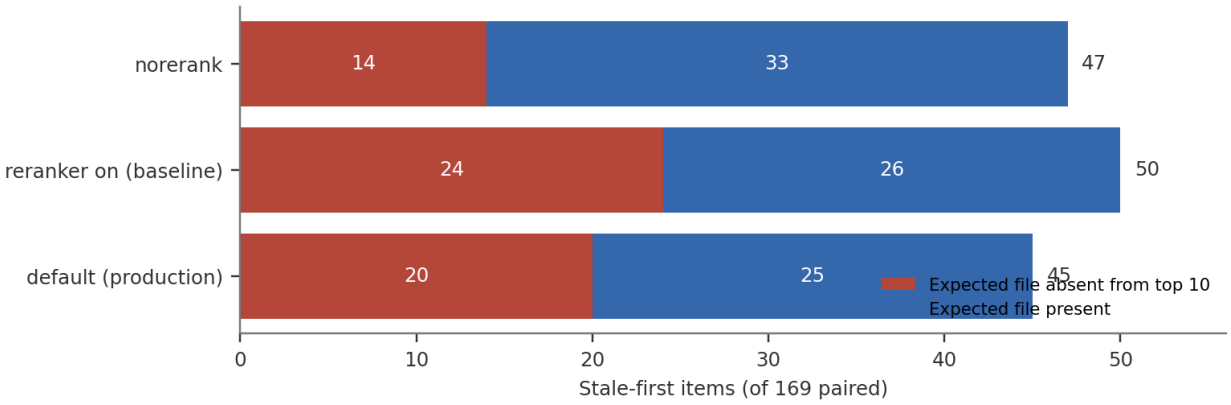


Figure 2. Stale-first items by arm, split by whether the expected file appears anywhere in the top 10. Values are in Table 5.

5. Three Ideas That Measured Nothing

Claim-polarity tags

The first idea was to annotate each chunk with the kind of claim it makes. A deterministic lexicon tagger, with no model involved, marks chunks as negated, prohibited, conditional, hedged, retired or superseded. The tagger's documentation reports, for 100 blind double-annotated chunks, precision of 0.93 for negated, 0.86 for retired, 0.75 for prohibited, 0.62 for superseded and 0.44 for hedged, and recall for conditional of only 0.46. We did not re-measure these. A further weakness: 71% of real chunks contain some negation, so a negated flag barely discriminates.

Tags never change ranking by design: the tags arm had 0 discordant items out of 57 against the baseline, the only result such an arm could produce. We ship tags as annotation only.

A metadata header

The second idea was to prepend a line of metadata to each passage before the reranker sees it: source, status, updated date and superseded-by.

On all 169 paired items the header arm fixed 3 stale-first items and broke 4. After the tool drops items recall-failed in either arm, that is 2 fixed and 4 broken ($p = 0.69$). On truth in the top 5 the header gained 3 and lost 2 ($p = 1.0$). The arm is also formally invalid: one item's header pushed its input past the 512-token limit, the GPU endpoint errored, and the harness counted the item as not reranked. In a rerun with a 6 s search budget that item is reranked and becomes

stale-first, a fifth break (2 fixed, 5 broken, p = 0.45).

A plausible reason is that note metadata points both ways: a retirement notice carries a retired status and is the current truth, while a stale page often carries an active status and an old date. We did not test this. We left the header off by default.

A counter-evidence companion

The third idea attaches a companion row of opposite evidence to a result when the top hit looks like a stale claim. In the first eval's three-repeat run, the companion fired on 2 of 26 held-out items (7.7%; 7.0% over all 57). No item changed outcome against the baseline (0 discordant of 57), and the harness's directional metric flagged none of the companions as a hit. The 95th-percentile latency rose from 300 ms to 423 ms on the held-out split (+123 ms).

The stance check is lexical, so it selects rows that share the query's vocabulary, which are rarely rows that contradict the claim. The flag stays off.

Idea	What it can change	Result on the eval	Latency	Decision
Polarity tags	Annotation only	0 discordant of 57	not measured	ship as annotation
Metadata header	Passage text	2 fixed, 4 broken, p = 0.69	p95 349 ms vs 347 ms	off, not adopted
Counter-evidence	Adds a row	0 discordant of 57; fired on 2 of 26 held-out	p95 +123 ms	off, not adopted

Table 6. The three interventions. Header latency is over all 198 items against the baseline arm.

6. A Gate for Editing the Weights, and a Raw Check of It

The more ambitious idea was to edit the reranker's weights so that current facts outrank stale ones. A feasibility step first counted stale-first failures fixable in principle, where stale and current passages differ by a visible transition cue a one-passage-at-a-time cross-encoder could use. The gate required at least 8 such failures in the baseline or header arm. Two raters labelled them by hand. One rater was the study's own agent, and no human adjudicated.

We could not re-examine the feasibility materials, so the rater counts are second-hand. As reported, the gate passed at exactly its threshold in the header arm, and by one in the production arm after the window change.

What we can check from the run records bears on how much that pass means. The header arm's stale-first set contains four items that were not stale-first at baseline. According to the feasibility report, two of them are among the failures it counts as fixable in the header arm. If so, those two exist only because the header was added, and without them the header arm would have fallen below the threshold. That inference rests on the report's labels, which we could not check. A gate that passes at its threshold, partly on failures the intervention created, is a weak licence for editing the weights.

The edit itself was not run.

7. What These Numbers Will Not Carry

The 169 paired items are not independent: several share source notes, so exact McNemar p-values are somewhat optimistic. A null at this n shows only that any effect is too small to detect here. There is one eval, one corpus (our own notes), one reranker model, and one

configuration per arm. Truth and lure patterns were written by model labellers and checked against live chunks by the labelling process; no human adjudicated them. One eval item is known to be flawed and stays in the frozen set.

The handling of recall-failed items changes the answer: truth in the top 5 gives p = 0.18 under the drop rule and 0.024 under recall-as-loss, and stale-first changes direction between rules.

Five claims survive. The reranker's effect on stale-first is not distinguishable from zero here, and its direction depends on the scoring rule (15 against 9 on the drop rule, 17 against 20 unscored; raw counts 47 off against 50 on). The metadata header does not help. Tags cannot change ranking by construction. The counter-evidence companion flipped no outcome and cost about 120 ms at the 95th percentile. And a large share of stale-first failures are recall-shaped: about half have the expected file absent from the top 10, and 33 of 50 have no truth passage in the final top 10, though we cannot say how many of those were in the reranker's pool.

Three things do not survive. We withdraw the live-probe scores that prompted the study, because no raw artefact exists. We do not report the manual count of genuine companion rows: the summary said none of 6 was genuine, while the run records show 7 companion rows over 4 items, and we cannot reconcile them. And we do not claim the reranker fails on polarity as a mechanism. Nothing here separates polarity from lexical overlap.

8. What We Changed, and What Is Still Open

We kept the reranker and treat it as a recall aid rather than an ordering authority. Tags ship as annotation only. The header and the counter-evidence companion are written, default off and not adopted. The larger lever appeared to be recall, which the companion study takes up.

Open questions: whether the stale preference is a polarity or a vocabulary-overlap effect (a matched probe pair with controlled word overlap would settle it cheaply); whether a weight edit can fix the visible cases without breaking passing items; how the failure split moves on an eval not mined from correction sentences; and whether a stance judge would make counter-evidence rows genuine.

9. Methods and Artefacts

The harness replays each question through the memory server's default search path with a search budget of 1.5 s (6 s in one header rerun), k = 5, a limit of 10 results, three repeats and per-arm environment pinning. A reranker-on arm below 100% reranked, or a norerank arm that shows any reranker score, is marked invalid. The statistic is the exact two-sided McNemar test on boolean per-item flips, in the variants named above. Tag precision comes from the tagger's own documentation, based on a 100-chunk blind sample.

One four-GPU compute node ran the harness against the production database. No new benchmark was run for this write-up. The arm rates, counts, flip tables, latencies and split counts were recomputed from the retained per-item records and matched the tool's own comparison output. Exceptions taken second-hand: the tagger precision figures, the 71% negation share, and the rater counts. Retained raw material is the per-item records for every arm and repeat (outcome, ranks, top-5 files and flags), the item set with its patterns and verification notes, and the harness.

PureTensor operates a sovereign AI infrastructure fleet (on-premises GPU compute, storage, and serving) run day-to-day by autonomous agents under human direction. This paper is PT-R-2026-063. All measurements were taken on 2 November 2026, and the per-item records behind the figures are retained.