OmniMem 7.0.8 vs 7.0.12

7.0.12 passed with 8 of 8 gates held; **7.0.8** failed with 7 of 8. Of 14 measures, 8 improved, 0 got worse and 6 are unchanged. Neither run was enriched: no API key was passed to the server.

RUN STATUS	GATES HELD	S4 FORGETEVAL	MEASURES
FAILED → PASSED	7 → 8	71 → 221	8 up 0 down
7.0.8 → 7.0.12	of 8 gates	cases passing, of 400	6 unchanged

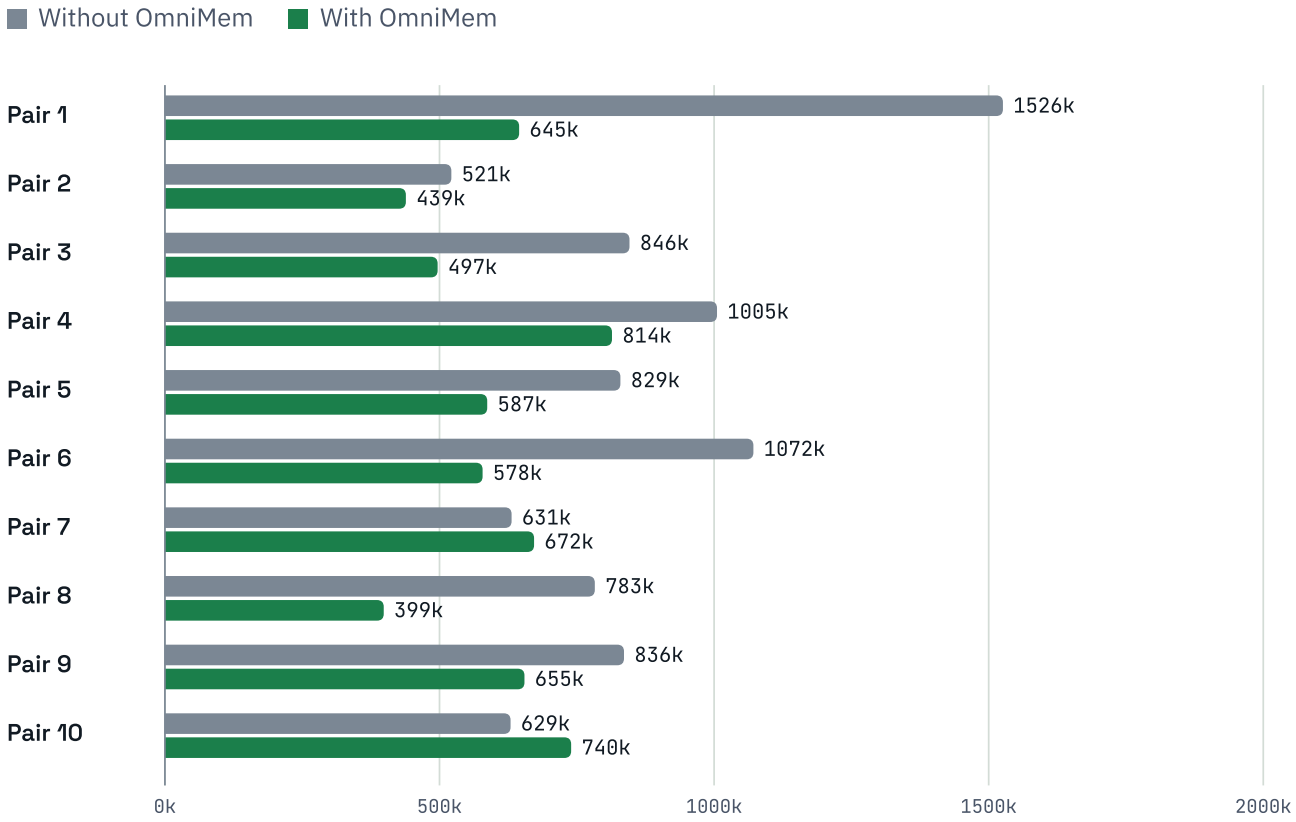
Most of the tests come from the Agent Memory Atlas (neoneye.github.io), an independent review of agent memory systems. The harness that runs them, and these results, are OmniMem's own. Section 06 says where each suite comes from.

HEADLINE · CLAUDE CODE WITH AND WITHOUT OMNIMEM

Fewer tokens, and the known dead end avoided

The same task, run by Claude Code (claude-haiku-4-5-20251001) twice in a row on a fresh copy of the code: once with no OmniMem configured, once with OmniMem's server and hook and the earlier failure in its store. The task's TODO recommends a crate that fails at runtime; only running the tests shows it. `cargo test` decides whether the task was done. Figures are for 7.0.12.

TOKENS	COST	DEAD END WALKED INTO	TIME
-26.0%	-25.9%	8/10 → 0/10	-34.6%
median per run, fewer in 8 of 10 pairs	median per run, $2.16 → $1.51 in all	runs, without → with OmniMem	median wall time per run



Run	Median tokens	Tokens	Cost	Dead end	Failing tests	Pairs fewer
7.0.8	711k → 572k	-19.6%	-17.6%	9/10 → 0/10	9/10 → 0/10	7/10 (p 0.17)
7.0.12	833k → 616k	-26.0%	-25.9%	8/10 → 0/10	7/10 → 0/10	8/10 (p 0.05)

Each cell reads without OmniMem → with it. **Dead end**: runs where the agent's code reached for the failing crate. **Failing tests**: runs that sat through a red `cargo test`. On 10 pairs a one-sided sign test gives p = 0.05: too few pairs to rule out chance for the token saving, though the dead-end result doesn't depend on it. One scenario and one model, with a shallow dead end: the agent without memory finds its way out after a test run or two, so a dead end that takes longer to find would be worth more.

01 · VERDICT

What changed

Improved on 7.0.12

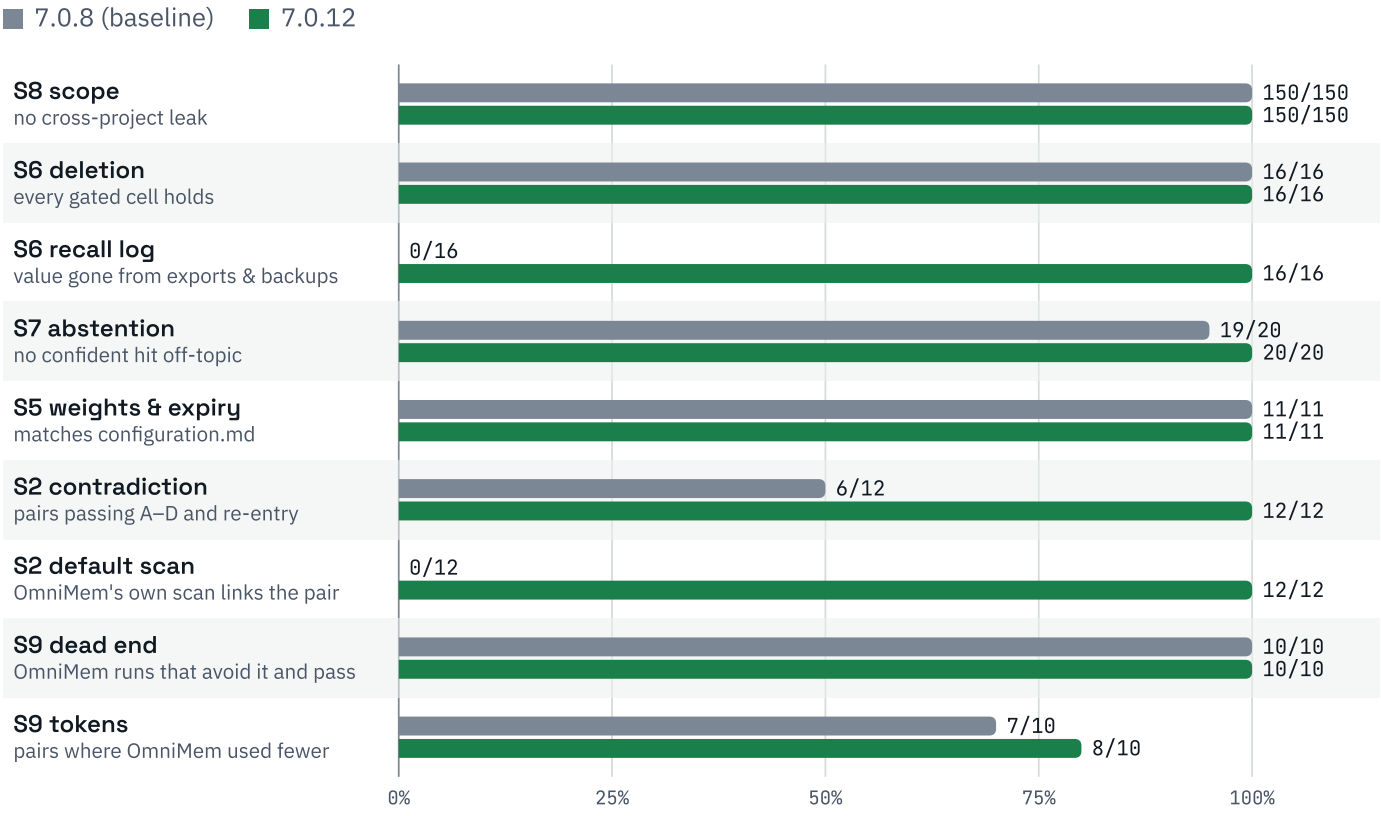
- **s7** gate now holds: off-topic questions return nothing above the weak band
- **S6 `recall log`** value gone from exports & backups: 0/16 → 16/16 (+100 points)
- **S7 `abstention`** no confident hit off-topic: 19/20 → 20/20 (+5 points)
- **S2 `contradiction`** pairs passing A–D and re-entry: 6/12 → 12/12 (+50 points)
- **S2 `default scan`** OmniMem's own scan links the pair: 0/12 → 12/12 (+100 points)
- **S9 `tokens`** pairs where OmniMem used fewer: 7/10 → 8/10 (+10 points)
- **S4 `Supersession`** replaced fact stops winning: 0/128 → 79/128 (+62 points)
- **S4 `Drift`** gradually changed fact: 2/64 → 61/64 (+92 points)
- **S4 `Purge`** deleted fact stays gone: 44/128 → 56/128 (+9 points)

No regressions

No measure is lower on 7.0.12 than on 7.0.8, and no gate that held on 7.0.8 fails on 7.0.12.

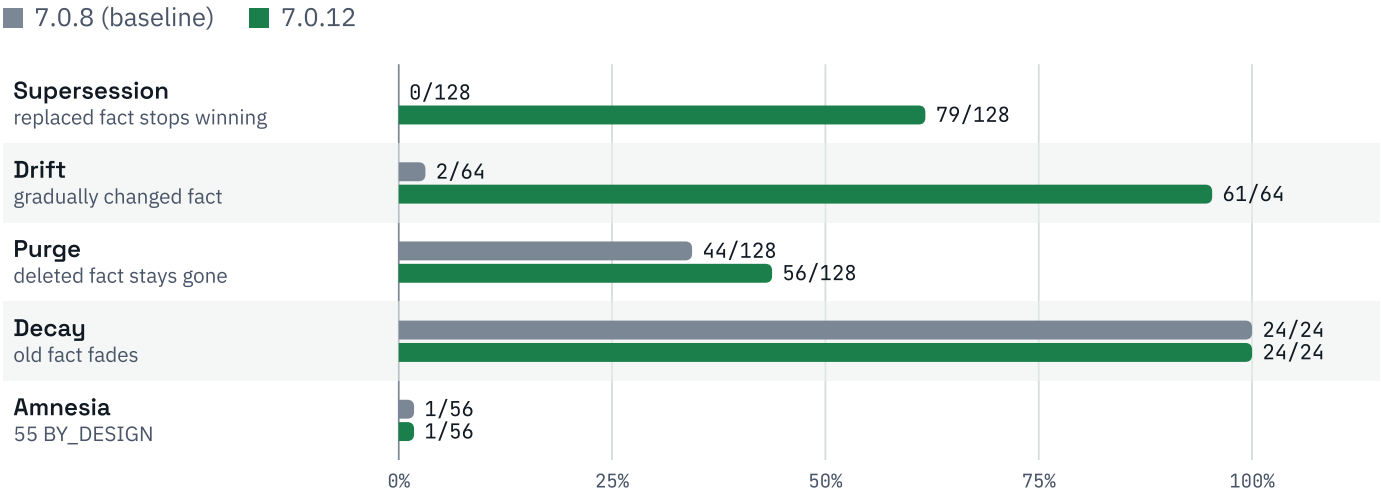
02 · SUITES

Gated checks and findings



03 · FORGETTING

S4: Lethe ForgetEval, by family



ForgetEval's own checks, run through an OmniMem adapter. Reported per family, not gated.

04 · GATES

Every gate

Suite	Gate	7.0.8	7.0.12
s8	zero cross-project leaks	held	held
s6	every gated cell in the §4.3 table holds	held	held
s7	off-topic questions return nothing above the weak band	failed	held
s5	expired articles are archived by one maintenance pass	held	held
s2	zero STALE after resolution, maintenance and re-entry	held	held
s9	OmniMem arm passes the suite honestly in every run	held	held
s9	no OmniMem run walks into the dead end	held	held
s9	OmniMem arm uses fewer tokens (median)	held	held

05 · NUMBERS

Every measure

Measure	What passing means	7.0.8	7.0.12	
S8 scope	no cross-project leak	150/150	150/150	=
S6 deletion	every gated cell holds	16/16	16/16	=
S6 recall log	value gone from exports & backups	0/16	16/16	▲
S7 abstention	no confident hit off-topic	19/20	20/20	▲
S5 weights & expiry	matches configuration.md	11/11	11/11	=
S2 contradiction	pairs passing A–D and re-entry	6/12	12/12	▲
S2 default scan	OmniMem's own scan links the pair	0/12	12/12	▲
S9 dead end	OmniMem runs that avoid it and pass	10/10	10/10	=
S9 tokens	pairs where OmniMem used fewer	7/10	8/10	▲
Supersession	replaced fact stops winning	0/128	79/128	▲
Drift	gradually changed fact	2/64	61/64	▲
Purge	deleted fact stays gone	44/128	56/128	▲
Decay	old fact fades	24/24	24/24	=
Amnesia	55 BY_DESIGN	1/56	1/56	=

- **`S6 recall log`** The recall log (log:recall:*) keeps query text; a deleted value must not survive in it.
- **`S2 contradiction`** UNLINKED pairs count as not passing: OmniMem couldn't be told which statement holds.
- **`S9 tokens`** Each pair is one run without memory and one with it, back to back on the same task.
- **`Amnesia`** A release maps to deprioritise, which keeps the memory at 0.2× its score; ForgetEval counts that as a fail, so the harness labels it BY_DESIGN.

06 · METHOD

What these tests are based on

Every suite except S9 comes from the Agent Memory Atlas by Simon Strandgaard (neoneye.github.io), which reads each memory system's code and defines how to test it. Its benchmarks page lists the tests it recommends. The Atlas didn't run these tests and doesn't endorse these numbers: the harness and its results are OmniMem's own. S9 is OmniMem's own benchmark of whether memory stops an agent repeating a known failure. The harness, its suites and every run's raw records are at code.squarecows.com/ric/omnimem-test-harness.

Suite	Test	Based on
s8	Scope and isolation	The Atlas's scope_enforced mark
s6	Deletion durability	The Atlas's 13-step deletion sequence
s7	Recall-level abstention	The Atlas's negative-control rule
s5	Ranking weights and expiry	The Atlas's notes on decay and membench, adapted to what OmniMem's configuration claims
s2	Contradiction Test	The Atlas's own Contradiction Test specification
s4	ForgetEval	ForgetEval in Lethe, from the Atlas's benchmark list, run through an OmniMem adapter
s9	Dead-end interception	OmniMem's own dead-end benchmark (run_deadend_bench.py), not from the Atlas

07 · NOTES

What 7.0.12 reported

- **s2** Partial supersession, bounded validity, equal weight and criterion E need OmniMem to decide, so they wait for the --enrichment configuration (phase 5).
- **s4** 186 supersede calls found no heuristic link, so OmniMem couldn't be told which fact holds; those cases depend on recall ranking alone.
- **s4** OmniMem's own query mode (archive/forget by query, needs a score above 0.85) matched nothing 118 times; the adapter then acted on the top recall hit by key.
- **s6** Arm F: 8 of 8 forgotten values came back after being fed in again (NO_TOMBSTONE: forget removes the row, and only suppress_topic stops a value).
- **s6** Steps 11-13: a backup taken before the forget still held the value in 16 of 16 cases (restore_from_file restores what a backup holds).
- **s6** The briefing never showed a suppressed value, so its EXPECTED_GAP may be closed.
- **s9** Median tokens -26.0%, median cost -25.9%, with OmniMem against without; fewer tokens in 8 of 10 pairs (one-sided sign test p = 0.055).
- **s9** Walked into the dead end: control 8/10, OmniMem 0/10. Agent model claude-haiku-4-5-20251001.
- **s9** Memory tool calls in the OmniMem arm: 0. The hook does the work; the agent rarely calls OmniMem itself, so this measures the hook, not recall.
- **s9** One scenario, one model, a shallow dead end: a dead end that takes longer to find is worth more.

08 · PROVENANCE

About these runs

	Baseline	Candidate
Instance	test-7.0.8-pairs10	test-7.0.12
OmniMem	7.0.8	7.0.12
Binary	release	release
sha256	bb3fa477e99f55ca	ac388fbf9adead47
Harness	de1e67133f	8bdccde36d
Enrichment	off	off
Started	2026-11-09T12:39:33+0000	2026-11-10T09:26:33+0000
Run folder	test-7.0.8-pairs10/2026-11-09/123933	test-7.0.12/2026-11-10/092633

---

omnimem-test-harness · generated by `omh compare` on 10 November 2026 from the two run folders above. Every number on this page is read from their gates.json, summary and record files.