# Tenet: Agent Memory as a Self-Consistent Belief State

Anas
Global AI Hackathon with Qwen Cloud (Track 1), 2026
https://github.com/Nas01010101/tenet

**Abstract**

Long-term memory for LLM agents is almost universally implemented as *retrieval over a growing log of past turns*, a document-retrieval abstraction. We argue this is the wrong abstraction for an agent that must model a changing world. We introduce **knowledge churn**, the repeated updating of a fact over a long interaction, and show that retrieval-augmented memory (RAG-memory) *silently degrades* under it: as the number of stale versions of a fact exceeds the retrieval budget $k$, the reader is handed conflicting values and answers incorrectly. On a controlled benchmark, a strong RAG-memory falls from 100% to 50% current-value accuracy as a fact is updated $2 \rightarrow 12$ times. We propose **Tenet**, which reframes memory as a **self-consistent, bi-temporal belief state**, a compact, supersession-aware set of facts about the user, rather than a document store. Tenet (i) distills raw turns into atomic, keyed facts; (ii) maintains a **bi-temporal** record so a changed fact *supersedes* its predecessor (retired to history, not overwritten); (iii) enforces **belief–evidence consistency** by retiring raw evidence that echoes a superseded belief; (iv) applies a **predictive-coding write policy**, surprise-gating, that stores only observations the model cannot already predict; and (v) closes the accuracy gap to raw retrieval with **belief-anchored evidence expansion**, spending spare context on query-relevant turns from the sessions the belief state already surfaced. Tenet holds 100% **current-value accuracy across all levels of the templated single-attribute churn primitive** (where a strong RAG-memory falls to 50%; on a harder *paraphrased* multi-attribute variant this claim is falsified and recovered by a read-time consistency fix, full preprint / `BENCHMARK.md` §9), matches strong RAG on retrieval recall (95–97.5%), and, with expansion, **matches its one-shot answer accuracy at equal-or-lower token budget** (57.5% vs 57.5%, gpt-4o reader) while retaining a high-efficiency point at **half the context** and the **best accuracy-per-token** of the systems we evaluate. Tenet thus traces an accuracy–efficiency *frontier* that meets RAG at its budget and beats it at every lower one. We release all code.

## 1 Introduction

An agent that talks to a user for months does not need a transcript; it needs a *model of the user* that stays true as the user changes. Yet the dominant memory design, retrieval-augmented generation over stored turns [1, 2], treats memory as a document index: embed every turn, retrieve the top-$k$ most similar, let the reader sort it out. This works for one-shot recall of a *static* fact. It fails, quietly, when facts **change**.

We formalize this as **knowledge churn**. A user who moves cities several times generates many "I moved to $X$" turns; all are similar to "where do I live?", so the top-$k$ fills with *stale* versions. Once the number of updates exceeds $k$, the latest value may not be retrieved at all, and even when it is, the reader must infer recency from contradictory statements. Accuracy collapses (§4.2).

The root problem is abstraction. A document store has no notion that "I live in Boston" and "I live in Seattle" are the *same fact* with a *changed value*. We argue memory should be a **belief**

**state**: a compact set of *current* beliefs, each with a temporal extent, updated by observation, kept internally consistent, and queryable across time, the stance predictive-coding accounts take toward perception [19], brought to agent memory.

**Contributions.**

1. We identify and name **knowledge churn**, a failure mode of retrieval memory, and give a controlled benchmark exhibiting it (RAG: 100%→50%; §4.2).
2. We present **Tenet**, a memory that is a self-consistent belief state: bi-temporal supersession, a **belief–evidence consistency** rule, and a **surprise-gated** (predictive-coding) write policy.
3. We evaluate on LongMemEval_S and controlled tests: Tenet is **churn-robust (**100%**)**, on par with RAG on recall (95%), **best on accuracy-per-token**, and, with belief-anchored evidence expansion, **at parity with strong RAG on one-shot accuracy at equal token budget**, closing a gap belief-only compression left open.
4. On the standardized **MemoryAgentBench FactConsolidation** benchmark, ingestion-time supersession with zero-LLM deterministic keys **scores 97.0 single-hop pooled, above even the published gpt-4o-tier result (94.8), and 45.8 multi-hop, 1.5× the published SOTA (30.2)**, using only a local 7B backbone, where the benchmark's 22 systems score ≤60 / ≤7 (2026-07-19 run, after an ingestion-keyer fix our own miss-file audit exposed; pre-fix 86.5/30.0, both runs in the evidence artifact).

## 2 Related Work

**Retrieval memory.** Mem0 [1] distills salient facts at write time over a vector store; it attaches only a *creation* timestamp, and its own report prices the graph variant at ∼1.8× p95 latency for a ∼2% quality gain, evidence we weigh in choosing a light substrate. LongMemEval [2] is the standard long-horizon benchmark; its V2 adds a *latency-aware* metric, a field shift toward accuracy *per cost* that our per-token results target. **Temporal knowledge graphs.** Zep/Graphiti [3] maintain a bi-temporal graph with automatic invalidation, but pay heavy per-write extraction and require graph infrastructure; Tenet keeps the bi-temporal semantics without the graph. **The 2026 bi-temporal convergence.** Concurrent systems adopted bi-temporal supersession: MemStrata [7] applies deterministic $(s, r, o)$ supersession over a bi-temporal ledger with no LLM on the read path, and shows *similarity-threshold* supersession leaks stale values where deterministic keying does not, independently corroborating our keyed design; Engram [8] pairs a bi-temporal graph with a hybrid facts-plus-chunks read path (converging on our dual-pool finding); TOKI [9] formalizes contradiction resolution as a bitemporal operator algebra; and [6] sets the current MemoryAgentBench FactConsolidation SOTA with deterministic post-retrieval aggregation. Tenet differs in *where* consistency is enforced, at ingestion and at recall (belief–evidence consistency), and in coupling the belief state to surprise-gated writes and budget-bounded evidence expansion, evaluated on the knowledge-churn axis none of them report. **Memory in the Qwen ecosystem.** Alibaba's own 2026 memory line takes the opposite design point: QwenLong-L1.5 [11] reaches 76.4 on LongMemEval by RL-training a 30B reasoner that reads up to 128K tokens per query; AgeMem [12] RL-trains the memory policy itself (Qwen2.5-7B backbone); ActMem [13] (Alibaba-co-authored) reaches 75.6 with LLM-built causal graphs at ingestion. All three spend heavily, at training time, ingestion time, or read time. Tenet reaches 60.0 at the same reader tier with **zero-LLM ingestion and ∼2K read tokens per query** (∼79% of QwenLong-L1.5's score at under 2% of its read budget), and its structural wins survive on a 7B backbone. **Active memory navigation.** A concurrent 2026 line makes memory access itself agentic. NapMem [14] exposes a four-level memory pyramid as a structured action

space and trains a GRPO policy that descends it and stops once evidence is sufficient (trained 9B beats untrained 397B on memory benchmarks); QwenLong-L1.5 [11] plans a navigational path over memory chunks (LongMemEval +15.6); MemFlow [15] is training-free but LLM-mediated (router + LLM-Validator retry); MemReader [16] moves active navigation to the write side; AgeMem [12] and Mem-$\alpha$ [17] RL-train the policy itself; MemCog [18] reports a similar frame (affiliation unverified). All make the *read* path agentic; Tenet keeps reads LLM-free and gets adaptive depth from an embedding saturation gate (§3) instead of a learned stop. **OS-style / observational memory.** MemGPT/Letta [4] page memory between context and archival storage; observational schemes keep a stable summary. Both are largely append-oriented and do not treat supersession or forgetting as first-class. **What is missing.** No prior system combines bi-temporal supersession, *belief–evidence consistency*, *predictive-coding write-gating*, principled forgetting, and *belief-anchored evidence expansion* in a light, graph-free store, nor reports the knowledge-churn regime.

# 3 Method

Tenet stores two layers over one bi-temporal table: a **belief layer** of distilled, keyed facts, and an **evidence layer** of raw turns. Reads never call an LLM.

**Distillation.** Each turn is distilled into atomic facts, each with a stable semantic key $\kappa = \textit{subject}$: *attribute* (e.g. `user:residence`), a salience $s \in [0, 1]$, and an event time. The key makes supersession reliable: embedding similarity cannot separate a *restated* fact from a *value-changed* one (we measure a value change at cosine 0.988, *higher* than a plain rephrasing of the same fact at 0.79, so no similarity threshold separates them), but a shared key can.

**Bi-temporal supersession.** Every memory carries event time (`valid_at`, `invalid_at`) and transaction time (`created_at`, `expired_at`). Storing a fact at key $\kappa$ whose value differs from the current one *supersedes* it: the old fact's `invalid_at`/`expired_at` are set; it leaves the current set but stays in history. Current recall filters `expired_at` $= \varnothing$; recall($as\_of{=}t$) reconstructs the belief set at any past $t$ (time-travel).

**Belief–evidence consistency (key rule).** The evidence layer answers detail questions but is where stale values hide: "I moved to Boston" survives after the belief moves on. We retire from current recall any raw slice $e$ close to a *superseded* belief:

$$\text{exclude } e \iff \max_{f \in \mathcal{F}_{\text{expired}}} \cos(e, f) \;\geq\; \tau_{\text{stale}}, \qquad \tau_{\text{stale}} = 0.80.$$

This single rule turns supersession into end-to-end correctness: 55% → 100% (§4.3).

**Predictive-coding write policy.** Predictive coding stores *prediction error*, not everything. On write, a raw observation $e$ is discarded if the store already predicts it:

$$\text{store } e \iff \max_{e' \in \text{store}} \cos(e, e') \;<\; g_{\text{surprise}}, \qquad g_{\text{surprise}} = 0.97.$$

This bounds the store and drops redundant repetition with no accuracy loss (§4.3).

**Forgetting.** Rank = relevance × decay, with $d = 2^{-\Delta t/h}\,(1 + \beta \log(1{+}\text{uses}))\,(0.6 + 0.8s)$, half-life $h = 14$ d; a sweep archives low-$d$ unpinned memories. Retrieval is a *dual pool* (beliefs for consistency, evidence for verbatim detail), each guaranteed a budget share.

**Fact dynamics: learning how facts change.** The ledger doubles as training data for drift: superseded facts are observed lifetimes, current facts censored ones. Per key class a conjugate Gamma–exponential model gives closed-form survival $P(\text{valid} \mid \Delta t) = \left(\frac{\beta_0 + T}{\beta_0 + T + \Delta t}\right)^{\alpha_0 + n}$, "residence" learns a slow hazard, "mood" a fast one, per user, no LLM. Co-supersession statistics add *ripple*: a

correlated key's change discounts its neighbors' survival, so one observation updates beliefs about unobserved attributes. Surfaced as per-fact `confidence` + an `uncertain_facts()` re-verification list; deliberately annotation-only (rank-demotion measurably re-created the churn failure before we reverted it). A trained alternative, a 276k-param GRU temporal point process (Weibull hazard + next-key + contrastive next-value heads), beats the closed form decisively on planted non-memoryless structure (NLL 2.76 → 1.99, CI excludes 0; next-value 0.997 recall@5 vs 0.05 chance; 5 seeds) but is honestly mixed on sparse real chains; it ships as a 1 MB numpy artifact (106 $\mu$s/query), opt-in, defaults unchanged.

**Belief-anchored evidence expansion.** Compressing a session into a few beliefs wins churn and efficiency but can drop the detail a multi-hop question needs, even when the right session *is* retrieved (recall 95–100%). The top-$k$ result names both the belief state *and the sessions it came from* (each memory's `source`). Given spare budget $B$, Tenet fills it with query-relevant raw turns drawn *only from those already-surfaced sessions*, evidence anchored to the belief state, not the whole haystack, under the same stale-echo filter. Setting $B$ to the baseline RAG budget makes expansion never spend more context than flat retrieval; the count $m$ is a knob ($m$=0 → efficiency point; $m$ large under $B$ → parity point) that lets one system trace a frontier.

**Saturation-gated navigation.** Associative multi-hop recall re-scores the store at a caller-fixed depth $N$; `navigate()` instead deepens iteratively, reusing belief-anchored expansion at each hop, and adopts hop $d$+1 only while its new evidence stays informative:

$$\text{adopt hop } d{+}1 \iff \max_{e \in new_{d+1}} score(e) \geq \tau_{\text{gain}} = 0.15,$$

stopping (budget $\leq 4$ hops) once no hop clears the gate. The gate is embedding-only, no LLM, no training, the training-free counterpart to NapMem's learned stop and MemFlow's LLM-Validator retry, at ∼ms latency. Validated as a *mechanism* on a deterministic synthetic store (bridge fact reached via an associative hop; early-stop at hop 2 on a saturated query vs. hop 3 on a multi-hop one). A subsequent seeded A/B at the `qwen3.7-plus` tier (FC 6k cells, $n$=20) measured *no benefit*: identical accuracy to default recall, −5 to −10pp vs. a fixed-4-hop arm, all within Wilson CIs, a strong reader is robust to the larger fixed pool, so adaptive early-stopping buys no precision. We report navigation as a mechanism, not a measured gain.

# 4 Experiments

**Protocol.** LongMemEval_S (500 questions, ∼115k-token histories). We use a `gpt-4o` **reader** (the same reader Mem0 and Zep report against), a local embedder (`bge-small`), and a cheap `gpt-4o-mini` distiller; the shipped system runs the same code on Qwen Cloud (`text-embedding-v4`, `qwen3.7-plus`) by a config flip. Numbers are compared only to baselines run under identical settings. Baselines: **RAG** (top-$k$ raw turns) and **full-context** (entire history).

## 4.1 Recall, and the accuracy–efficiency frontier (n=40, gpt-4o reader)

| System | mode | recall@10 | QA acc | reader tok | **acc/1k tok** |
|---|---|---|---|---|---|
| full-context | n/a | n/a | 65%* | ∼124k | 0.5* |
| RAG | top-$k$ turns | 95% | 57.5% | 2,101 | 27.4 |
| **Tenet** | efficiency ($m$=0) | **97.5%** | 52.5% | **1,067** | **49.2** |
| **Tenet** | parity (expansion) | **97.5%** | **57.5%** | 2,083 | 27.6 |

Tenet is a *frontier*, not a point. At its **efficiency** point it answers at half the context (1,067 tok) for the best accuracy-per-token of any system (49.2, 1.6× RAG) at recall parity. Turning up belief-

anchored expansion under RAG's own budget brings raw QA accuracy to **parity with a strong RAG (**57.5%=57.5%**) at fewer tokens**, closing the one-shot gap belief-only compression left open. Tenet thus meets RAG's accuracy at its budget and beats it at every lower one; the one category still behind is *multi-session* synthesis (42.9 vs 57.1, up from 28.6), where a question needs several evidence sessions but only some are surfaced. *Full-context, at a weaker reader, spends 100× the tokens for no gain over RAG. Reader-robust: on a cheaper `gpt-4o-mini` reader the parity point edges ahead (Tenet 60.0 vs RAG 55.0); per-token dominance holds across the `gpt-4o-mini` and `gpt-4o` readers we ran (≈1.6×).

### 4.2 Knowledge churn (headline)

One fact updated $N$ times amid distractors, $k = 6$, 12 principals/point:

| updates $N$ | 2 | 4 | 6 | 8 | 10 | 12 |
|---|---|---|---|---|---|---|
| RAG | 100 | 100 | 100 | 67 | 58 | **50** |
| **Tenet** | 100 | 100 | 100 | **100** | **100** | **100** |

RAG degrades monotonically once $N > k$ ($-50$ pp); Tenet is flat at 100%. Supersession keeps exactly one current value regardless of churn, the property a *belief state* has and a document index cannot. **The curves are identical under a `gpt-4o-mini` and a `gpt-4o` reader**: the failure is structural (once $N > k$ the latest value is not reliably retrieved), so a stronger reader cannot rescue RAG.

### 4.3 Ablations

**Belief–evidence consistency.** Removing the rule drops Tenet to 55% current-value accuracy with a 45% stale-leak; adding it restores 100% / 0% leak, the single most important mechanism. **Surprise-gating** discards 15% of observations as redundant with no accuracy change, yielding a bounded store where RAG grows unboundedly.

### 4.4 Standardized conflict resolution: MemoryAgentBench FactConsolidation

On the MemoryAgentBench conflict-resolution benchmark [5] (SubEM metric and official reader prompt verbatim; all 800 questions; Wilson 95% CIs), ingestion-time supersession with **fully deterministic, zero-LLM keys** and a deliberately weak **local 7B backbone** scores, pooled over four haystack lengths (6K–262K):

| pooled | naive-RAG (same reader) | **Tenet** | published SOTA (mini / gpt-4o) [6] |
|---|---|---|---|
| FC-SH ($n$=400) | 47.8 | **97.0** [94.8, 98.3] | 78.0 / 94.8 |
| FC-MH ($n$=400) | 4.5 | **45.8** [40.9, 50.6] | 30.2 / 51.5 |

Single-hop **exceeds the published mini-tier SOTA (78.0; our CI excludes it by 17 points) and the gpt-4o-tier pooled point estimate (94.8) on a weaker backbone**; every system in the original benchmark table scores ≤60 (Zep 7, Mem0 18, MemGPT 28). Multi-hop is **1.5× the published mini-tier SOTA** (CI excludes it), where the original table's best is ≤7, still below the gpt-4o tier's 51.5, reported honestly. Accuracy barely degrades with haystack length (SH ≥96 from 6K→262K): the store is conflict-resolved at ingestion, the property assembly-time aggregation must re-derive at every read. Multi-hop still degrades with length (55→35); reported honestly. *Provenance: 2026-07-19 run after an ingestion-keyer fix found by auditing our own miss files (the zero-LLM key only collided two-word values, so stale facts silently survived ingestion); still*

*deterministic and zero-LLM, RAG control reproduced exactly, pre-fix run (86.5/30.0) preserved in* `docs/factcon_results.json`.

To remove the backbone confound entirely, we reimplemented four published memory mechanisms as arms of the *same* harness (same 7B reader, embedder, SubEM, prompt; 6K+32K cells, $n$=200/axis): CAR [Freshness 2026] 87.5 SH / 33.0 MH, Mem0-style consolidation 81.0 / 12.0, HippoRAG-v2-style OpenIE+PPR graph 66.0 / 9.0, MemAgent-style overwrite memory 44.0 / 16.0 ($n$=25). **Tenet leads every arm on both axes (90.0 / 36.0 on the same cells)**, ingestion-time supersession matches or beats the best assembly-time aggregation while doing the consistency work once at write time. Beyond reimplementations, one released framework run fully *black-box* through its own ingest-and-retrieve pipeline (ReMe, reme-ai 0.4.1.1; identical Qwen reader/judge for all arms): on LongMemEval_S full haystacks ($n$=100) **Tenet 67.0% [57.3, 75.4] vs ReMe 34.0% [25.5, 43.7]** (McNemar $p \approx 2 \times 10^{-6}$; matched RAG 64.0, blind 0.0), Tenet ahead on every question type.

### 4.5 Accurate retrieval at zero ingestion cost: MemoryAgentBench AR

On MAB's other core competency (∼2,000 questions, 197K–534K-token contexts; official per-benchmark metrics: SubEM for RULER-QA, the official LLM-judge for LongMemEval(S*), choice accuracy for EventQA; matched gpt-4o-mini reader), Tenet with **zero-LLM ingestion** (date-aware structured chunks + embeddings only) averages **59.3**, second only to HippoRAG-v2 [10] (65.1), which runs LLM OpenIE over every context token at ingestion, and 20+ points above Mem0 (32.6), Zep (37.5) and MemGPT. Per sub-benchmark: EventQA **70.7** ($n$=1,500; Wilson CI [68.3, 72.9] excludes HippoRAG-v2's 67.6), RULER SH-QA 75.0 (parity with 76), LME(S*) 46.3 vs 50.7, and RULER MH-QA 45.0 vs 66, the honest loss: Personalized-PageRank graph traversal is genuinely stronger at multi-hop chaining over narrative text.

On MAB's third competency, **Test-Time Learning** (5 ICL classification cells, ∼6–8K ingested demonstrations each, official prompt + substring-EM scoring, $n$=500), Tenet pools **77.2** [73.3, 80.7], above BM25 (75.4) and every published memory system (MemGPT 67.6, Zep 62.8, HippoRAG-v2 61.4, Mem0 32.4) despite using a *weaker* local 7B reader against their gpt-4o-mini, and ∼5 points under its own long-context backbone ceiling where Mem0 loses 50 to its own. TTL is retrieval-bound, no conflict to resolve, so Tenet's supersession machinery is inert and the matched-RAG control ties exactly, by design; the result is a breadth check, not a win claim. Together with §4.5, Tenet leads or ties the published memory field on three of MAB's four competencies while being the only published memory framework in this comparison whose ingestion never calls an LLM.

### 4.6 Case study: web-agent trajectory memory (LongMemEval-V2)

Adapting Tenet's ingestion to LME-V2's web-agent trajectory haystacks [Wu 2026] (DOM states, actions; up to 115M tokens), three LLM-free changes, structure-aware chunking, query cleaning, and Qwen3-Embedding+BM25 hybrid retrieval (RRF), lifted gold-evidence recall from a naive port's ∼12% to **59.7%** @48K budget (63.3% @72K; corpus ceiling ∼92%) at ∼0.25 s query latency. End-to-end accuracy is reader-gated, not retrieval-gated, and the gate is the *reasoning mode*, not precision: 4-bit local 45.1% [39.5, 50.8], 8-bit 40.3%, full-precision hosted *without* extended thinking 40.7% (over-abstains), all near the same $P(\text{correct} \mid \text{gold}) \approx 0.5$–$0.6$ extraction ceiling; the leaderboard-default extended thinking is what converts retrieval headroom, and lies outside our compute budget. The substrate transfers; the binding constraint is measurable and external to the memory.

### 4.7 Benchmark breadth: wins, ties, and honest losses

We evaluate Tenet on the standard agent-memory suite and report every outcome, losses included. On **LongMemEval_S** against Alibaba's own ReMe framework (same reader and judge) Tenet scores 67.0 vs ReMe's 34.0 (McNemar $p \approx 2 \times 10^{-6}$); a live head-to-head against the `mem0ai` package on staleness-sensitive queries gives Tenet 100% current-value accuracy at 0% stale-leak vs mem0's 73.3%/26.7% ($p = 0.0078$). On **MemoryAgentBench Test-Time-Learning** Tenet ties naive-RAG (77.2; retrieval-bound, as both readers see the same in-context pool). Two honest losses: on **LoCoMo-10** Tenet trails RAG (33.8 vs 38.8, $p = 0.031$, $n$=500), and on **PersonaMem-v2** the two are statistically indistinguishable (50.5 vs 50.9, $p = 0.92$; a blind control scores 34.4, confirming both actually use the memory). **Systems:** recall latency is flat at $\sim$11 ms from $10^3$ to $10^5$ resident facts (embedder-bound, zero LLM calls), and ingestion throughput stays near-flat ($\sim$7,400 facts/s at $10^5$) after the resident-index fix. Cross-reader robustness replicates the frontier result in the same direction on 6/6 reader×length cells across three frontier readers. Full tables and reproduction commands are in `docs/BENCHMARK.md`.

## 5 Limitations

**Multi-session synthesis** is the one category where RAG still leads (42.9 vs 57.1). Belief-anchored expansion lifted it from 28.6 but does not close it: these questions need evidence from *several* sessions, and expansion only deepens the sessions the top-$k$ already surfaced, if a required session is not among them, its detail is still missing. Session-diverse retrieval is the natural next step; elsewhere Tenet is at or above RAG. **The frontier is a knob, not free lunch**: parity accuracy costs RAG-equal tokens, and the 1.6× per-token win is at the efficiency point, which trades $\sim$5 pp of raw accuracy. **Evaluation**: $n$=40, off-Qwen (gpt-4o / gpt-4o-mini readers, local embedder), one seed; reader stochasticity is $\approx\pm$5–7 pp, so the one-shot result is reported as *parity*, not a win. The shipped system uses Qwen Cloud.

## 6 Conclusion

Treating agent memory as a **self-consistent belief state** rather than a document index makes it robust to the way real knowledge behaves: it changes. Tenet stays correct under knowledge churn where retrieval memory collapses, matches a strong RAG's one-shot accuracy at equal token budget while offering the best accuracy-per-token we tested, with a light graph-free substrate and no LLM in the read path, and gets time-travel and principled forgetting for free. We hope *knowledge churn* becomes a standard axis for evaluating agent memory.

## References


[1] Chhikara et al. *Mem0: Production-Ready AI Agents with Scalable Long-Term Memory.* arXiv:2504.19413, 2025.

[2] Wu et al. *LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory.* arXiv:2410.10813, 2024.

[3] Rasmussen et al. *Zep: A Temporal Knowledge Graph Architecture for Agent Memory.* arXiv:2501.13956, 2025.

[4] Packer et al. *MemGPT: Towards LLMs as Operating Systems.* arXiv:2310.08560, 2023.

[5] Hu, Wang, McAuley. *MemoryAgentBench: Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions.* arXiv:2507.05257, 2025.

[6] *Don't Ask the LLM to Track Freshness: A Deterministic Recipe for Memory Conflict Resolution.* arXiv:2606.01435, 2026.

[7] *Temporal Validity in Retrieval Memory: Eliminating Stale-Fact Errors for AI Agents over Evolving Knowledge.* arXiv:2606.26511, 2026.

[8] *Less Context, More Accuracy: A Bi-Temporal Memory Engine for LLM Agents.* arXiv:2606.09900, 2026.

[9] *TOKI: A Bitemporal Operator Algebra for Contradiction Resolution in LLM-Agent Persistent Memory.* arXiv:2606.06240, 2026.

[10] Gutiérrez et al. *From RAG to Memory: Non-Parametric Continual Learning for Large Language Models.* arXiv:2502.14802, 2025.

[11] Shen et al. *QwenLong-L1.5: Post-Training Recipe for Long-Context Reasoning and Memory Management.* arXiv:2512.12967, 2026.

[12] Yu et al. *Agentic Memory: Learning Unified Long-Term and Short-Term Memory Management for LLM Agents.* arXiv:2601.01885, ACL 2026.

[13] Zhang et al. *ActMem: Bridging the Gap Between Memory Retrieval and Reasoning in LLM Agents.* arXiv:2603.00026, 2026.

[14] *NapMem: Memory as a Structured Action Space for Long-Horizon Agents.* arXiv:2607.05794, 2026. Qwen Large Model Application Team, Alibaba.

[15] *MemFlow: Training-Free Intent-Routed Memory Orchestration with LLM Validation.* arXiv:2605.03312, 2026.

[16] *MemReader: Active Extraction for Agent Memory Construction.* arXiv:2604.07877, 2026.

[17] *Mem-$\alpha$: Reinforcement-Learned Memory Construction for LLM Agents.* OpenReview, 2026.

[18] *MemCog: Cognitively-Inspired Active Navigation of Agent Memory.* arXiv:2605.28046, 2026. (Affiliation unverified.)

[19] Friston. *The free-energy principle: a unified brain theory?* Nat. Rev. Neurosci., 2010.