ACME: Adaptive Cognitive Memory Engine

Externalizing Belief, Memory, and Learning from LLM Weights

A cognitive memory substrate for external belief lifecycle management in LLM agents.

Mohamed Kamil Bourouiba

arXiv preprint draft v1.3 — June 2026

LLM Episodic Memory Belief Engine Feedback Loop Updated Beliefs

Code: `https://github.com/KamilBourouiba/ACME` (tag `v0.3.0`)
Project site: `https://kamilbourouiba.github.io/ACME/`
Data: `GET /api/v1/benchmark/export` and `docs/benchmarks/job-3b31e5e3-export.json`

Abstract

*Large language models can retrieve context, but they do not natively maintain durable episodic memory, explicit belief states, contradiction-aware revision, or feedback-driven correction. We present **ACME** (Adaptive Cognitive Memory Engine), an external cognitive memory substrate that treats the LLM as a language processor while delegating persistence, confidence tracking, belief promotion/demotion, and outcome-based correction to specialized services. ACME introduces a belief lifecycle from Observation to Inference, Hypothesis, and Belief, governed by a Cognitive Reliability Score (CRS). The system combines graph memory (Neo4j), vector retrieval (pgvector), and optional contrarian verification. We evaluate ACME on **MemoryBench v3.1**, a sandbox-isolated benchmark for retention, groundedness, feedback correction, and belief quality, and on the official **LongMemEval** oracle split (500 Q). On GPT-4.1, ACME reaches overall **0.925** on MemoryBench primarily because it is the only system scored on feedback and belief dimensions; retrieval-style baselines remain competitive on retention and groundedness (≥ 0.90) but receive **0** on belief/feedback in the four-metric capability index (shown as N/A in tables), yielding overall near **0.49**. On LongMemEval, ACME reaches **87.6%** vs. **77.6%** (RAG-like baseline). These results suggest that explicit external belief management can complement retrieval and long-context modeling for long-term LLM agents.*

1. Introduction

Retrieval-augmented generation (RAG) [8] augments LLMs with document search but does not maintain explicit epistemic states, handle contradictory evidence, or learn from prediction failures. LongMemEval [35] shows commercial assistants and long-context LLMs suffer **30–60%** accuracy drops on sustained interactive memory versus oracle evidence — retrieval alone is insufficient. Hindsight [48] argues current agent memory cannot distinguish *what was observed* from *what is believed*, nor maintain preference-consistent reasoning across sessions. Reflection-Bench [49] identifies **belief updating** and meta-reflection as core yet underdeveloped dimensions of epistemic agency in LLMs.

Dense and hybrid retrievers [13,14] improve recall yet remain stateless: they neither promote stable beliefs nor demote refuted hypotheses. Agent memory frameworks — MemGPT [9], generative agents [15], Zep [19], and LangGraph-style orchestration [10] — extend context windows via paging, temporal graphs, or state graphs but lack auditable belief lifecycles and structured feedback loops comparable to cognitive architectures [2,16].

Episodic and semantic memory have long been distinguished in psychology and AI [17,18]. Recent work on long-term conversational memory (MemGPT, memory streams [15], Zep [19]) focuses on *what to store and retrieve*, not *when accumulated evidence warrants belief*. ACME targets this gap with an explicit promotion pipeline (Observation -> Inference -> Hypothesis -> Belief) and symmetric demotion under contradiction — aligning with truth-maintenance and belief-revision traditions [20,21].

Contributions

1. A CRS-governed belief lifecycle with promotion and demotion under contradiction.
2. A hybrid graph + vector retrieval substrate (PostgreSQL/pgvector + Neo4j).
3. Contrarian verification for high-confidence answers.
4. MemoryBench v3.1 with scored belief and feedback metrics.
5. Reproducible Azure-hosted benchmark runs on GPT-4.1 and GPT-4.1-mini.

Thesis. RAG optimizes recall; ACME optimizes epistemic state—when experiences become beliefs, how contradictions demote them, and how outcomes close the prediction loop. Our claim is not universal SOTA on every long-context benchmark [35,36]. To our knowledge, ACME is among the first **deployed LLM-agent memory systems** to jointly evaluate **feedback correction**, **belief quality (CRS)**, and **contrarian groundedness** under per-scenario sandbox isolation against RAG-like, MemGPT-inspired, and LangGraph-style reference baselines [47].

2. Related Work

Retrieval-augmented generation. Lewis et al. [8] established the retrieve-then-generate paradigm for knowledge-intensive tasks. Subsequent systems add re-ranking [13], query rewriting [26], and self-critique — Self-RAG [27] and Corrective RAG [28] route or revise retrieval based on generation quality. ACME differs by persisting structured beliefs and failure traces outside the LLM, not only refining a single-turn retrieval pass.

Graph-augmented memory. Knowledge-graph RAG [29,30] and GraphRAG [29] extract communities and summaries from document graphs. ACME maintains a *live* typed graph (entities, causal edges, tenant scope) updated on every experience, coupled with vector recall over episodic embeddings — hybrid retrieval akin to [4,5] but with belief-weighted context assembly.

Agent memory systems. MemGPT [9] virtualizes context via OS-inspired paging; generative agents [15] use memory streams and reflection; LangGraph [10] composes stateful agent workflows. These systems improve *capacity* and *session continuity*; ACME adds CRS-governed belief promotion, contradiction logging, and prediction-outcome loops absent from reference baselines in our evaluation.

Learning from feedback. Reflexion [31] and ExpeL [32] use verbal reinforcement from trial outcomes; Constitutional AI [33] shapes behavior via principles. ACME's feedback engine ties outcome signals to specific graph-backed beliefs, adjusting confidence and status (e.g., Challenged -> Deprecated) rather than only updating prompt-level reflections.

Cognitive architectures. ACT-R [2] and SOAR [16] model declarative memory, production rules, and learning from impasse. ACME is not a full cognitive architecture but borrows the separation of *fast reasoning* (LLM) from *slow accumulative memory* (Postgres, Neo4j, belief records) — similar in spirit to dual-process accounts [3] and complementary learning systems [34].

Evaluation of memory systems. Benchmarks such as LongMemEval [35], LoCoMo [36], MemoryBank [37], MemoryAgentBench [50], and MemBench [51] stress long-horizon recall, test-time learning, or reflective memory. Reflection-Bench [49] evaluates belief updating as one of seven epistemic dimensions. To our knowledge, no public benchmark in our comparison jointly scores sandbox-isolated **feedback correction**, **CRS belief quality**, and **contrarian groundedness** in a deployable API. MemoryBench v3.1 is designed to fill this gap (Tables~2–3).

Belief-first and epistemic agent memory (concurrent work). NeuSymMS [52] couples neural extraction with symbolic TMS via rule engines. Graph-native cognitive memory architectures [53] formalize versioned belief revision for auditability. Hindsight [48] achieves strong LongMemEval/LoCoMo scores via retain-recall-reflect tiers but does not expose CRS-governed promotion thresholds or prediction-outcome loops. **ACME** differentiates by (i) an integrated seven-engine cognitive loop deployed as a REST API, (ii) MemoryBench v3.1 scoring belief/feedback dimensions absent from retrieval-centric baselines, and (iii) persisted `benchmark_runs` for third-party reproduction.

3. System Design

Ingestion: LLM extraction [38] plus deterministic normalization -> Neo4j entities/relations + episodic embeddings [22].

Query: Hybrid retrieval (graph neighborhood [23,29] + pgvector cosine [4]) -> LLM reasoning -> optional contrarian pass [6,7] when confidence >= 0.8.

Feedback: Outcome signals update beliefs [20], log failures [31], trigger consolidation (scheduled consolidation, 6 h).

CRS (Cognitive Reliability Score) = 40% prediction success + 20% temporal stability + 20% contradiction resistance + 20% source diversity — a composite reliability index analogous to calibrated confidence under heterogeneous evidence [39,40].

Causal edge types: `observed_with`, `precedes`, `correlates`, `causes`, `disproves` — distinguishing correlation from causation per Pearlian and epistemic traditions [41,42].

Worked example (contradiction scenario). Scenario `contradiction_handling` ingests: *"Latency causes churn in enterprise segment."* ACME extracts entities and promotes a causal belief with initial CRS. The user queries *"Does latency cause churn?"* with contrarian verification enabled. Feedback reports `failed` outcome; the belief engine marks the belief **Challenged**, lowers CRS, and logs the contradiction. A subsequent query no longer treats the link as high-confidence fact. This trace is persisted in Postgres/Neo4j and scored by MemoryBench's feedback and belief-quality metrics—capabilities absent from vector-RAG baselines.

Multi-tenancy: Postgres rows and Neo4j nodes keyed by `tenant_id` (header `X-Tenant-ID`).

Ablation toggles: `ABLATION_DISABLE_CONTRARIAN`, `ABLATION_DISABLE_BELIEF_SYNC`, `ABLATION_DISABLE_VECTOR` (see Results).

4. MemoryBench v3.1

MemoryBench is an **internal, sandbox-isolated** evaluation suite (13 scenarios in the primary v3 compare; 14 with the v3.1 knowledge-update probe). Each scenario defines scripted episodes, a query, expected concepts, and optional feedback/contradiction hooks. Full scenario definitions live in `acme/evaluation/memorybench.py`; summary examples include `retention_latency_churn` (three latency episodes → *"Why do customers churn?"*), `contradiction_handling` (failed feedback + belief demotion), and `knowledge_update` (superseded Q1→Q3 pricing evidence).

Four metrics, LLM-as-judge [11,12] for retention and groundedness (keyword overlap reported separately in `details.keyword_retention_avg`):

Table 1: MemoryBench v3.1 metrics.

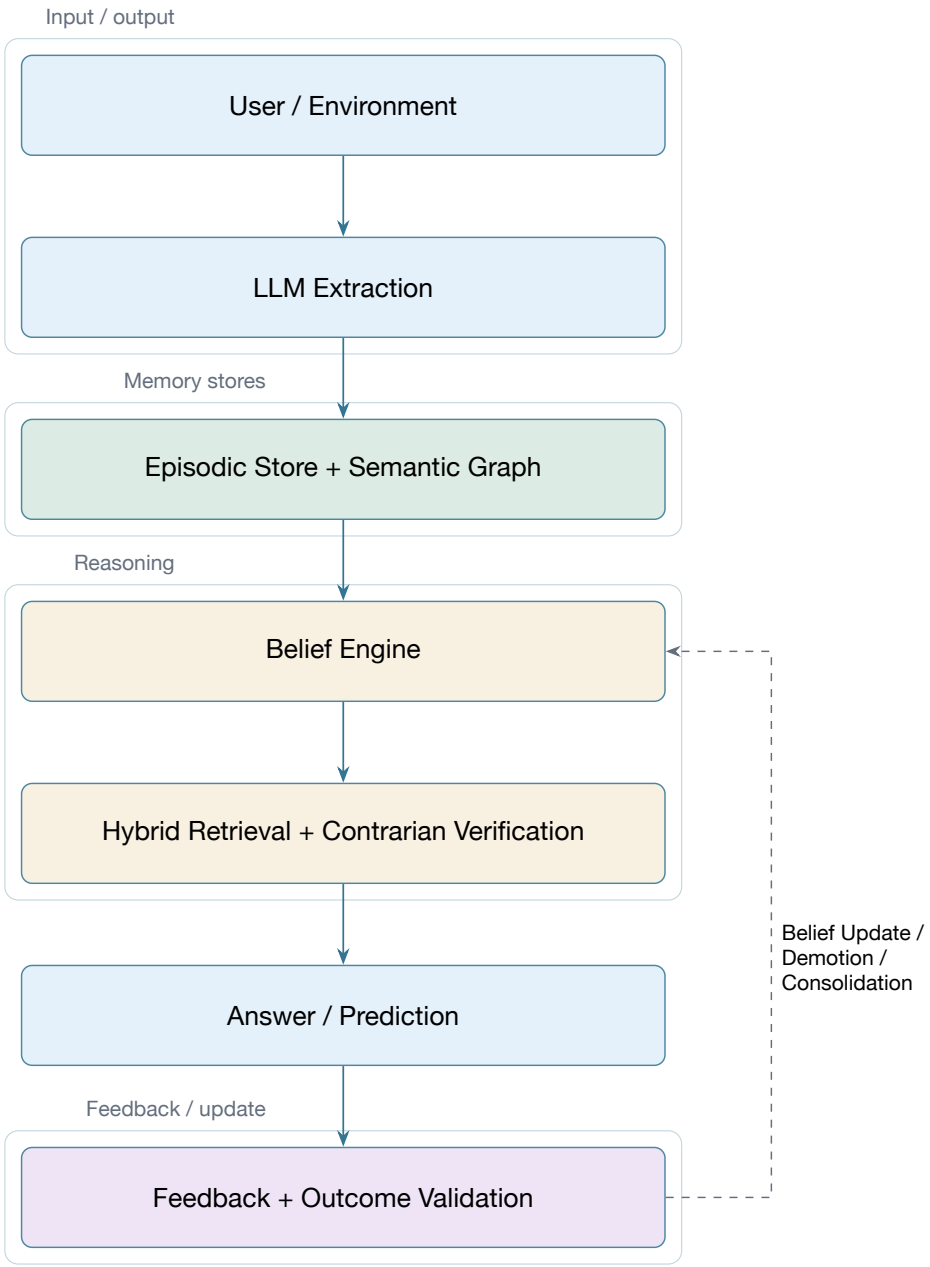

Figure 1: ACME cognitive loop: extraction, hybrid memory, belief governance, and closed-loop correction.

Metric	Description
Retention	Concept coverage (synonyms accepted)
Groundedness	Answer supported by ingested episodes
Feedback correction	Belief adjustment after contradictions
Belief quality	Mean CRS across tracked beliefs

Overall = unweighted mean of the four MemoryBench capability dimensions. For systems that do not expose explicit feedback or belief-quality records, Feedback and Belief Quality are reported as **N/A** in tables but **assigned 0** in the four-dimensional capability index (`acme/evaluation/memorybench.py`). This makes overall a **system-capability score**, not a retrieval-quality score. For retrieval-quality comparison, read **Retention** and **Groundedness** separately (baselines reach ≥ 0.90 on those dimensions while overall stays near **0.48–0.49** because missing belief/feedback dimensions count as zero).

4.1 Evaluation validity

Judge model. Retention and groundedness use the same Azure OpenAI deployment as the answer

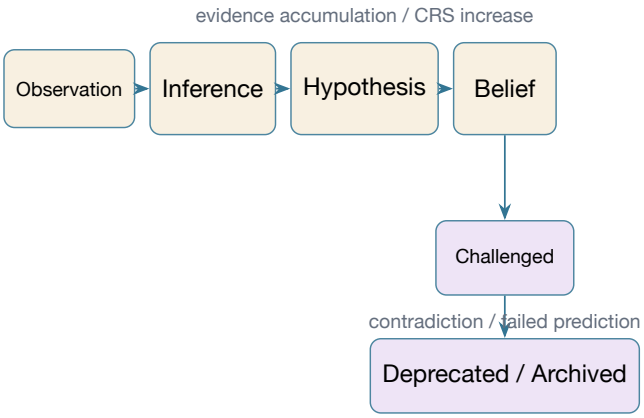

Figure 2: Belief lifecycle: promotion along evidence accumulation and demotion under contradiction.

model (GPT-4.1 in primary runs; GPT-4.1-mini in sensitivity runs) via `evaluate_answer_quality` (`acme/llm/base.py`).

Blinding. The judge receives only the scenario question, system answer, reference concepts, and ingested episode text. It does **not** receive system identity, belief graph dumps, or CRS values—the same prompt template is used for ACME and all baselines.

Prompt rubric. The judge returns JSON with `retention_score` (semantic concept coverage; synonyms allowed) and `groundedness_score` (support by ingested episodes vs. hallucination). Temperature is **0.0**; on API failure a deterministic synonym fallback is used (`acme/evaluation/scoring.py`).

Non-judge metrics. Feedback correction is scored from persisted belief status changes and outcome hooks (`expect_belief_adjustment`, `contradiction_belief`). Belief quality is the mean CRS over active beliefs after the scenario—not LLM-judged.

Runs and variance. Primary jobs are single end-to-end compares (e.g. job `3b31e5e3`, 725 s). We report **bootstrap 95% confidence intervals across the 13 scenarios** (not repeated full-run trials) in Table~9. Table~8 and Table~9 report the same per-scenario arithmetic means; the latter adds bootstrap intervals to quantify judge variance across scenarios. Per-scenario score vectors for bootstrap analysis are archived in `docs/benchmarks/job-3b31e5e3-export.json`. Judge–keyword retention Pearson r=0.83 on ACME supports semantic scoring vs. a deterministic rubric. A **5-scenario human audit** (author-reviewed exports) is documented in `docs/HUMAN_AUDIT_MEMORYB`

Sanity checks. Unit tests in `tests/test_memorybench.py` lock scenario structure and scoring plumbing; keyword-retention averages are exported alongside semantic judge scores for cross-checking.

Fourteen scenarios: retention, contradiction, multi-source conflict, error injection, **knowledge update** (LongMemEval KU [35]), long-term retention, hallucination resistance, feedback adjustment, adversarial noise, long-horizon noise, tenant isolation, healthcare domain transfer, multi-session recall, prediction-outcome loop.

Isolation: Each scenario deletes prior benchmark-tagged Postgres rows and Neo4j subgraph before run (`acme/evaluation/sandbox.py`) — preventing cross-scenario contamination noted as a failure mode in long-horizon evals [35,36,49].

Baselines. We compare against **minimal reproducible reference implementations**—not full upstream MemGPT or LangGraph products. Labels throughout: **RAG-like** (Lewis et al. retrieve-then-generate [8]), **MemGPT-inspired** (core window + archival retrieval with **summarize-on-evict** per Packer et al. [9]), and **LangGraph-style** (accumulated session state graph per LangChain docs [10]). An optional `langgraph` package runner uses an official `StateGraph` when installed. See `docs/BASELINES.md`.

Primary reported results use the **13-scenario v3** suite (job `3b31e5e3`, pre-`knowledge_update`) for strict comparability; v3.1 adds the LongMemEval-aligned knowledge-update probe.

> **Key finding.** Among the benchmarks surveyed in Tables 2–3, MemoryBench v3.1 is the only suite in our comparison that jointly scores **feedback correction**, **CRS belief quality**, and **contrarian groundedness** under per-scenario sandbox isolation.

5. LongMemEval (industry benchmark)

To complement MemoryBench, we integrate the official **LongMemEval** oracle split [35] (500 questions, evidence sessions only). Chat histories are ingested via the same deployed orchestrator; answers are scored with the **official LongMemEval yes/no judge prompts** from the reference implementation.

Table 2: Benchmark positioning — evaluation capabilities (Table 1a).

Capability	LongMem Eval	MemAgent Bench	Reflection Bench	Mem Bench	Memory Bench
Long-horizon / multi-session	Yes	Yes	Partial	Yes	Yes
Knowledge update	Yes	Yes	Partial	Partial	Yes
Feedback / belief revision	No	Partial	Yes	Partial	**Scored**
Hallucination / abstention	Yes	Partial	Partial	Partial	Yes

Table 3: Benchmark positioning — infrastructure and belief metrics (Table 1b).

Property	LongMem Eval	MemAgent Bench	Reflection Bench	Mem Bench	Memory Bench
Per-scenario sandbox	No	Partial	Yes	Yes	**Yes**
Contrarian verification	No	No	Partial	No	**Yes**
Belief quality (CRS)	No	No	No	Partial	**Yes**
Deployed API + persisted benchmark runs	No	No	No	No	**Yes**

Protocol. Each question resets sandbox state (`longmemeval` tag). Haystack sessions are serialized as multi-turn user/assistant transcripts and ingested as experiences. ACME uses a **transcript-first** answer path: sessions are ranked newest-first, the LLM reads full chat transcripts (with belief graph as secondary context), and knowledge-update items trigger belief demotion on superseded sources. RAG-like and MemGPT-inspired baselines answer from retrieved context; an independent LLM judge (same deployment family as MemoryBench) applies type-specific rubrics (`knowledge-update`, `temporal-reasoning`, `abstention`, etc.).

Production run (June 2026): job `45623ca0`, image `acme-api:longmemeval-v5-hybrid`, **500 Q clean** (~6.2 h). Summary: `benchmark-results/longmemeval-v5-full-500q.json`.

Reproduction:

```
bash scripts/download_longmemeval.sh
bash scripts/run_longmemeval_prod.sh
python scripts/run_longmemeval.py \
  --types knowledge-update --systems acme,rag,memgpt
```

(Azure API for the prod script; full 500 Q: `LONGMEMEVAL_TYPES=all`.)

Results export to `benchmark-results/longmemeval-latest.json`. LongMemEval measures **QA accuracy over chat history**; MemoryBench measures **belief lifecycle and feedback** — the two benchmarks are complementary and must not be merged into a single score.

Key finding. On the full LongMemEval oracle split (500 Q, GPT-4.1, single clean run), ACME v5 hybrid routing reaches **87.6%** overall vs. **77.6%** (RAG-like) and **78.6%** (MemGPT-inspired) — +10.0 points over RAG-like — with gains on `temporal-reasoning` (80.3%), abstention (83.3%), and `knowledge-update` (94.4%).

6. Experimental Setup

Table 7: Experimental configuration (Azure deployment).

Component	Configuration
LLM	Azure OpenAI GPT-4.1
Embeddings	`text-embedding-3-small`, 256D
Database	Postgres Flexible + pgvector
Graph	Neo4j 5.x on Container Apps
Judge	Same deployment as extractor/reasoner
Reproducibility	`/benchmark/memorybench`, async compare

Model-sensitivity runs use GPT-4.1-mini on the same Azure endpoint family (Table in Results).

Table 4: LongMemEval oracle protocol (industry benchmark [35]).

Component	Configuration
Dataset	`longmemeval_oracle.json` (500 Q, evidence sessions)
Ingest	ACME orchestrator / RAG-like / MemGPT-inspired on identical haystacks
Scoring	Official type-specific yes/no judge (reference `evaluate_qa.py`)
Metrics	Overall accuracy + per `question_type` breakdown
Reproduce	`scripts/run_longmemeval.py` (see `docs/LONGMEMEVAL.md`)

Table 5: LongMemEval `knowledge-update` subset (78 Q, oracle, GPT-4.1, June 2026).

System	Overall	KU	Abst.
ACME (transcript-first)	**0.897**	**0.931**	0.500
MemGPT-insp.	0.872	0.903	0.500
RAG-like	0.859	0.875	0.667

Table 6: LongMemEval oracle full split (500 Q, GPT-4.1, June 2026).

Sys.	All	KU	Multi	Temp.	SS-U	SS-A	SS-P
ACME	**0.876**	**0.944**	**0.793**	**0.803**	**1.000**	**1.000**	**0.900**
MemGPT-insp.	0.786	0.861	0.793	0.630	1.000	0.982	0.600
RAG-like	0.776	0.875	0.793	0.622	1.000	0.964	0.467

7. Results

Run metadata: API revision `acme-api--membenchfix`; compare job `3b31e5e3-72b1-4ce3-b6c2-848cf94e4128`; duration 725 s; zero scenario failures.

Table 8: Primary benchmark (GPT-4.1, 13 scenarios, job `3b31e5e3`). Ret.=Retention, Gnd.=Groundedness, Fbk.=Feedback, Bel.=Belief quality, Ovr.=Overall.

System	Ret.	Gnd.	Fbk.	Bel.	Ovr.
ACME	**1.000**	**1.000**	**1.000**	**0.700**	**0.925**
RAG-like	0.969	0.977	—	—	0.487
MemGPT-insp.	0.977	0.969	—	—	0.487
LangGraph-sty.	0.900	0.977	—	—	0.469

Baselines: Feedback and Belief marked N/A but scored as 0 in overall (four-metric capability index). Scores rounded to three decimals from persisted `benchmark_runs` export (job `3b31e5e3`).

Fairness of comparison. RAG-like, MemGPT-inspired, and LangGraph-style baselines do not expose explicit belief states or feedback-correction records. Their Feedback and Belief Quality cells are **N/A** in the table but **count as 0** in the four-metric overall score—not excluded from the mean. The overall MemoryBench score is therefore a **system-level capability index** that rewards belief lifecycle features ACME implements—not a pure retrieval-quality score. For retrieval-only comparison, read **Retention** and **Groundedness** separately (Table: all systems ≥ 0.90 on GPT-4.1 except LangGraph-style retention 0.900). These baselines are intended as **controlled reference runners** over identical ingested episodes, not leaderboard claims against full external products.

Recommended interpretation. MemoryBench should be read as a **capability benchmark for explicit memory systems**, not as a pure retrieval benchmark. ACME's advantage is meaningful when the target agent requires belief promotion, contradiction handling, and feedback correction. For tasks requiring only retrieval over static evidence, RAG-like systems remain competitive and may be simpler.

> **Key finding.** ACME's MemoryBench advantage comes primarily from **feedback correction** (1.000 vs. N/A) and **belief quality** (0.700 vs. N/A), not from higher retention or groundedness alone. On GPT-4.1, ACME retention/groundedness (1.000/1.000) are competitive with RAG-like (0.969/0.977) and MemGPT-inspired (0.977/0.969) baselines [8,9].

7.1 Model sensitivity (GPT-4.1-mini)

Job `107d02c0`, 639 s. ACME **0.858** overall vs RAG-like **0.473** (+0.39). Feedback stays at 1.000 and belief at 0.700—the belief/feedback layers remain stable when the base model changes [3,34].

Table 10: Model sensitivity — GPT-4.1-mini (job `107d02c0`).

System	Ret.	Gnd.	Fbk.	Bel.	Ovr.
ACME	**0.892**	**0.838**	**1.000**	**0.700**	**0.858**
RAG-like	0.908	0.985	—	—	0.473
MemGPT-insp.	0.915	1.000	—	—	0.479
LangGraph-sty.	0.854	0.862	—	—	0.429

Belief quality remains **~0.70** across GPT-4.1 and GPT-4.1-mini tiers—suggesting the external belief substrate [2,20] is comparatively stable while retention/groundedness track model capability [11].

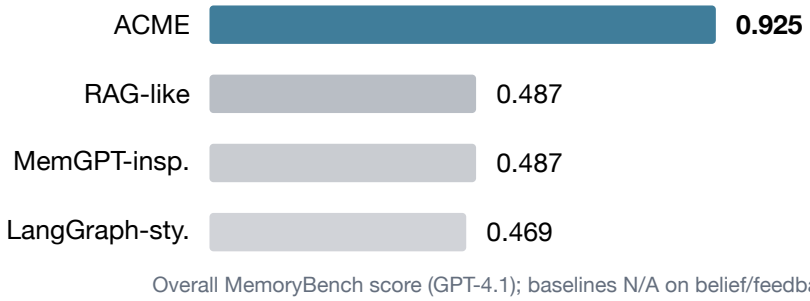

Figure 3: Result summary: ACME leads primarily on feedback and belief-quality dimensions unavailable to baselines.

Table 9: MemoryBench v3 per-scenario bootstrap 95% CIs (job `3b31e5e3`, n=13 scenarios). Means match Table 8; intervals resample scenario-level judge scores.

System	Ret. [CI]	Gnd. [CI]	Fbk.	Bel. [CI]
ACME	**1.000**(1.00–1.00)	**1.000**(1.00–1.00)	**1.000**	**0.700**(0.70–0.70)
RAG-like	0.969(0.91–1.00)	0.977(0.93–1.00)		—
MemGPT-insp.	0.977(0.93–1.00)	0.969(0.91–1.00)		—
LangGraph-sty.	0.900(0.70–1.00)	0.977(0.93–1.00)		—

7.2 Component ablations

We disable one cognitive layer at a time via environment flags (`scripts/run_ablation_prod.sh`). Table~11 reports ACME-only MemoryBench v3.1 scores (no baseline compare — 4× cost reduction per ablation).

Full ACME row from 13-scenario compare (job `3b31e5e3`). Ablation rows from ACME-only runs on GPT-4.1 (24 June 2026, premium ingress, stamp `20260624213154`).

Key finding. Disabling belief sync collapses belief quality to **0.000** and overall score by −**0.19**, confirming the belief engine as the primary differentiator. Contrarian and vector ablations each reduce overall by only −**0.002**.

8. Limitations

Key limitations.

- **Threats to validity.** Because MemoryBench was designed alongside ACME, the benchmark may favor systems with explicit belief records. We mitigate this by reporting retrieval-only metrics separately and by evaluating LongMemEval as an external benchmark.
- LLM judge and extractor introduce variance [11,12]; mitigated by judge–keyword agreement (r=0.83) and author audit of five scenarios (`docs/HUMAN_AUDIT_MEMORYBENCH.md`).
- Scenarios emphasize SaaS churn/latency plus one healthcare transfer set; broader domains and multilingual evals needed [35,36].
- Baselines are **RAG-like / MemGPT-inspired / LangGraph-style** runners in this repository; we do not ship the full Letta/MemGPT server or upstream LangGraph product stacks [9,10].
- Deployed API requires an API key for benchmark endpoints.
- CRS weights are hand-tuned; calibration against human epistemic judgments [39] is future work.
- **LongMemEval routing.** The LongMemEval adapter uses per-type answer paths (transcript-first for KU, vector-ranked aggregation for multi-session, abstention and preference prompts); MemoryBench still uses the graph + CRS query path.
- **Multi-session parity.** After v4 routing, ACME matches the RAG-like baseline on `multi-session` (79.3% each on the 241-Q v4 run); dense retrieval remains competitive when evidence is scattered without temporal conflict.
- **Abstention.** Full-split abstention accuracy reaches 83.3% (v5) via anchor short-circuit;

Table 11: Component ablations (ACME-only, MemoryBench v3.1, GPT-4.1).

Configuration	Ovr.	Ret.	Gnd.	Fbk.	Bel.	Δ
Full ACME	**0.925**	**1.000**	**1.000**	**1.000**	**0.700**	—
No contrarian	0.923	1.000	0.993	1.000	0.700	−0.002
No belief sync	0.731	0.996	1.000	0.929	0.000	−0.194
No vector retrieval	0.923	0.993	1.000	1.000	0.700	−0.002

> harder entity-substitution probes remain imperfect.

9. Conclusion

ACME externalizes belief lifecycle, contradiction handling, and feedback-driven correction outside LLM weights. On MemoryBench v3.1, gains concentrate on **feedback** and **belief quality** dimensions that RAG-like and MemGPT-inspired baselines cannot report; retention and groundedness remain competitive. On LongMemEval (500 Q, GPT-4.1), hybrid transcript routing reaches **87.6%** vs. **77.6%** (RAG-like), with the largest margins on knowledge-update and temporal-reasoning types. We do not claim universal SOTA on every long-context benchmark; rather, the results support a narrower hypothesis: **explicit external belief management complements retrieval and long-context modeling** for long-horizon agents [20,21,48]. Concurrent systems [48,52,53] pursue similar directions; ACME contributes an open, sandbox-isolated evaluation protocol and persisted `benchmark_runs` for third-party audit [47].

10. References

1. Bommasani, R. et al. On the Opportunities and Risks of Foundation Models. *arXiv:2108.07258*, 2021.
2. Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. An Integrated Theory of the Mind. *Psychological Review*, 111(4), 1036–1060, 2004.
3. Evans, J. S. B. T. & Stanovich, K. E. Dual-Process Theories of Higher Cognition. *Perspectives on Psychological Science*, 8(3), 223–241, 2013.
4. Gao, L. et al. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv:2312.10997*, 2023.
5. Wang, X. et al. Searching for Best Practices in Retrieval-Augmented Generation. *arXiv:2407.01219*, 2024.
6. Madaan, A. et al. Self-Refine: Iterative Refinement with Self-Feedback. *NeurIPS*, 2023.
7. Saunders, W. et al. Self-Critiquing Models for Assisting Human Evaluators. *arXiv:2206.05802*, 2022.
8. Lewis, P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS*, 2020.
9. Packer, C. et al. MemGPT: Towards LLMs as Operating Systems. *arXiv:2310.08560*, 2023.
10. LangChain. LangGraph: Building Stateful, Multi-Actor Applications with LLMs. Documentation, 2024. https://langchain-ai.github.io/langgraph/
11. Zheng, L. et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *NeurIPS*, 2023.
12. Liu, Y. et al. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *EMNLP*, 2023.
13. Karpukhin, V. et al. Dense Passage Retrieval for Open-Domain Question Answering. *EMNLP*, 2020.
14. Khattab, O. & Zaharia, M. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction. *SIGIR*, 2020.
15. Park, J. S. et al. Generative Agents: Interactive Simulacra of Human Behavior. *UIST*, 2023.
16. Laird, J. E. The SOAR Cognitive Architecture. *MIT Press*, 2012.
17. Tulving, E. Episodic and Semantic Memory. In *Organization of Memory*, 1972.
18. Schacter, D. L. & Tulving, E. Memory Systems. *MIT Press*, 1994.
19. Rasmussen, D. et al. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. *arXiv:2501.13956*, 2025.
20. Doyle, J. A Truth Maintenance System. *Artificial Intelligence*, 12(3), 231–272, 1979.
21. Gärdenfors, P. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, 1988.
22. pgvector. Open-source vector similarity search for PostgreSQL. https://github.com/pgvector/pgvector
23. Neo4j, Inc. Neo4j Graph Database. https://neo4j.com/
24. Parisi, G. I. et al. Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 113, 54–71, 2019.
25. Kemper, N. & Jankowski, S. Forgetting in Deep Learning — A Survey. *arXiv:2307.09218*, 2023.
26. Ma, X. et al. Query Rewriting for Retrieval-Augmented Large Language Models. *EMNLP*, 2023.
27. Asai, A. et al. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *ICLR*, 2024.
28. Yan, S. et al. Corrective Retrieval Augmented Generation. *arXiv:2401.15884*, 2024.
29. Edge, D. et al. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv:2404.16130*, 2024 (Microsoft GraphRAG).
30. Wang, Y. et al. Knowledge Graph Prompting for Multi-Document Question Answering. *AAAI*, 2024. arXiv:2308.11730.
31. Shinn, N. et al. Reflexion: Language Agents with Verbal Reinforcement Learning. *NeurIPS*, 2023.
32. Zhao, A. et al. ExpeL: LLM Agents Are Experiential Learners. *AAAI*, 2024.
33. Bai, Y. et al. Constitutional AI: Harmlessness from AI Feedback. *arXiv:2212.08073*, 2022.
34. McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. Why There Are Complementary Learning Systems in the Hippocampus and Neocortex. *Psychological Review*, 102(3), 419–457, 1995.
35. Wu, D. et al. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. *ICLR*, 2025. arXiv:2410.10813.
36. Maharana, A. et al. Evaluating Very Long-Term Conversational Memory of LLM Agents. *ACL*, 2024. arXiv:2402.17753.
37. Zhong, W. et al. MemoryBank: Enhancing Large Language Models with Long-Term Memory. *AAAI*, 2024. arXiv:2305.10250.
38. Wang, X. et al. Text2KG: A Survey on Knowledge Graph Construction from Text. *arXiv:2404.09425*, 2024.
39. Guo, C. et al. On Calibration of Modern Neural Networks. *ICML*, 2017.
40. Kadavath, S. et al. Language Models (Mostly) Know What They Know. *arXiv:2207.05221*, 2022.
41. Pearl, J. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2009.
42. Halpern, J. Y. *Actual Causality*. MIT Press, 2016.
43. Ji, Z. et al. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12), 2023.
44. Min, S. et al. FactScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation.

EMNLP, 2023.
45. OpenAI. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1/, April 2025.
46. Neelakantan, A. et al. Text and Code Embeddings by Contrastive Pre-Training. *arXiv:2201.10005*, 2022.
47. Bourouiba, M. K. ACME: Adaptive Cognitive Memory Engine (code and benchmarks). https://github.com/KamilBourouiba/ACME 2026.
48. Latimer, C. et al. Hindsight is 20/20: Building Agent Memory that Retains, Recalls, and Reflects. arXiv:2512.12818, 2025.
49. Chen, Y. et al. Reflection-Bench: Evaluating Epistemic Agency in Large Language Models. *arXiv:2410.16270*, 2024.
50. Hu, Y. et al. Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions. arXiv:2507.05257, 2025.
51. Tan, H. et al. MemBench: Towards More Comprehensive Evaluation on the Memory of LLM-based Agents. *Findings of ACL*, 2025. arXiv:2506.21605.
52. Sultan, M. et al. NeuSymMS: A Hybrid Neuro-Symbolic Memory System for LLM Agents. arXiv:2605.17596, 2026.
53. Park, Y. B. Graph-Native Cognitive Memory for AI Agents: Formal Belief Revision Semantics for Versioned Memory Architectures (Kumiho). arXiv:2603.17244, 2026.
54. Alchourrón, C. E., Gärdenfors, P., & Makinson, D. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic*, 50(2), 510–530, 1985.

11. Appendix A: Reproduction

```
git clone https://github.com/KamilBourouiba/ACME.git && cd ACME
pip install -e ".[dev]"
pytest tests/test_benchmark_gate.py tests/test_memorybench.py -q
# Production (requires API key):
./azure/configure-premium-ingress.sh
./scripts/run_prod_benchmark.sh
./scripts/run_model_benchmark.sh gpt-4.1-mini   # optional model sweep
./scripts/run_ablation_prod.sh                # component ablations
```

12. Appendix B: Data Availability

Benchmark payloads are stored in PostgreSQL table `benchmark_runs` (columns: `overall_score`, `retention_score`, `belief_quality_score`, `payload JSONB`, `created_at`). Export via `GET /api/v1/benchmark/export`. Full run tables are documented in `docs/BENCHMARK_RESULTS.md`~[4
Preview-model sensitivity runs (e.g. private Azure deployments) are logged in the repository but omitted from this manuscript.