TECHNICAL WHITEPAPER

How a Solo Dev Built the #1-Ranked Memory System for AI Agents

95.4% on LongMemEval. Local-first. Zero funding.
Here's what I learned.

The Problem

Your AI coding agent has amnesia. Every session starts from zero. Yesterday's architecture decisions, last week's debugging insights, the coding conventions you've established over months — gone when the session ends.

The industry has noticed. Over $180 million has flowed into AI memory startups: Mem0 ($24M, 44K stars), Mastra ($13M, YC-backed), Letta ($10M, $70M valuation), Emergence ($97M Series C), Supermemory ($3M, backed by Jeff Dean).

The problem is real, funded, and unsolved.

I built OMEGA to solve it for myself. It scores 95.4% on LongMemEval — first on the leaderboard, ahead of Mastra's 94.87%. It runs entirely on my laptop. No cloud, no API keys, no external databases. One SQLite file.

The Retrieval Pipeline

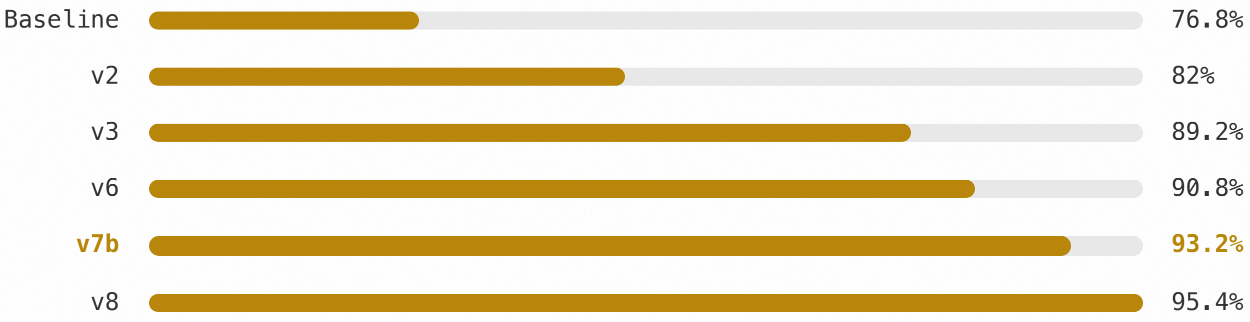
From 76.8% to 95.4%

```
1. Vector Search    bge-small-en-v1.5 (384-dim), sqlite-vec

|
  2. Full-Text       FTS5 with BM25 scoring
     Search
↓
  3. Blended Rank    70% vector + 30% text score
↓
  4. Type Weighting  decisions/lessons weighted 2x
↓
  5. Re-ranking      temporal, overlap, abstention floor

~50ms retrieval · All on SQLite · Zero external services
```

Everything runs on SQLite with two extensions: `sqlite-vec` for vector similarity and FTS5 for full-text search. Embeddings from bge-small-en-v1.5 via ONNX on CPU.

The abstention floor matters more than you'd expect. When no memory meets the minimum relevance threshold, the system returns nothing rather than surfacing low-quality matches. Hallucination from irrelevant context destroys trust faster than no result at all.

The Optimization Journey



Each iteration taught something transferable:

> ### Cross-encoders trained on web search hurt conversational memory
>
> Added an MS-MARCO cross-encoder expecting a big win. It dropped the score by 7 points. Web search relevance and memory relevance are different distributions.

> ### More context isn't always better
>
> K=35 retrieves more relevant memories but also more near-duplicates. The LLM deduplicates similar notes, causing under-counting. K=25 gives enough signal without the noise.

> ### Aggressive dedup instructions backfire

> “VERIFY each item and REMOVE duplicates before counting” caused 9 regressions. Simple “list all relevant items” prompts with soft dedup consistently outperform aggressive filtering.

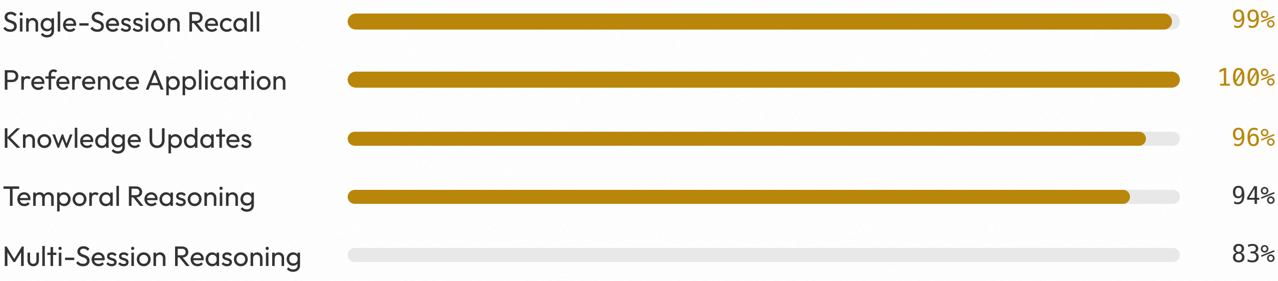
> **Query expansion is free accuracy**
>
> Extracting temporal dates (“last Tuesday” → “2026-02-11”) and entity names before retrieval adds ~2 points with zero downside.

Where I Am on the Leaderboard

#	System	Score	Model
1	OMEGA	95.4%	GPT-4.1
2	Mastra OM	94.87%	gpt-5-mini
3	Mastra OM	93.27%	gemini-3-pro
4	Hindsight	91.4%	gemini-3-pro
5	Emergence	86.0%	Internal
6	Supermemory	85.2%	gemini-3-pro

The gap to #1 is 1.67 points — 8 questions. Scores are self-reported and model-dependent.

Category Breakdown



The Academic Connection

MAGMA (Jiang et al., January 2026) independently validated the idea that orthogonal decomposition improves memory retrieval. They decompose memory into four semantic subspaces — semantic, temporal, causal, and entity graphs.

OMEGA applies the same principle at the system architecture level. The name stands for **O**rthogonal **M**ulti-Agent **E**ngine for **G**eneralized **A**gents. Five independent modules that compose without coupling:

Module	Tools	Scope
Core Memory	26	Store, query, traverse, checkpoint, resume, compact
Coordination	28	File claims, branch guards, task DAG, peer messaging, deadlock detection
Router	10	Multi-LLM routing, intent classification
Entity	8	Entity registry, relationship graphs
Knowledge	5	Document ingestion (PDF, web, markdown)

What Memory-Only Systems Don't Build

Multi-agent coordination. Mem0 doesn't do it. Mastra doesn't do it. Supermemory doesn't do it.

When you run multiple AI coding sessions on the same repo, they will step on each other. Edit the same file simultaneously. Push to the same branch. Make contradictory architecture decisions.

File Claims
Two agents editing the same file

Branch Guards
Two agents pushing to the same branch

Task DAG
Duplicated work; dependency violations

Peer Messaging
Agents working without awareness of each other

Intent Broadcasting
Overlapping work plans

Deadlock Detection
Circular waits across claims

All backed by the same SQLite database. No message broker, no distributed consensus, no external service.

Zero-Configuration Capture

```
Claude Code event → fast_hook.py → (~5ms UDS) → hook_server.py → memory
+ coordination
```

Seven hooks, 12 handlers. Session start/end, user prompts, edits, reads, git pushes. Fail-open — if the daemon is down, work continues unblocked.

Intelligent Forgetting

Memory systems that never forget become noise generators. OMEGA implements structured forgetting:

- **TTL:** Session summaries expire after 1 day. Checkpoints after 7 days. Lessons and decisions are permanent.
- **Three-layer dedup:** Exact hash, semantic similarity (≥ 0.85), and per-type Jaccard similarity.
- **Memory evolution:** Similar content gets appended to existing memories rather than creating duplicates.
- **Compaction:** Clusters related memories, creates summaries, marks originals as superseded.
- **Audit trail:** Every deletion is logged with its reason. Deterministic, auditable, reversible.

Local-First, Actually

Everything runs on your machine. SQLite for storage. ONNX on CPU for embeddings. FTS5 for text search. macOS Keychain for encrypted profiles. ~31MB startup, ~337MB after first query.

No cloud account needed. Optional Supabase sync if you choose. Your architecture decisions, debugging history, and credential patterns never leave your machine.

Try It

```
$ pip install omega-memory
```

Open source. Apache 2.0. One command. Works with any MCP client.

See Comparisons

References

[1] Wang, Y., et al. "LongMemEval: Benchmarking Long-Term Memory in AI Assistants." *ICLR 2025* Link

[2] Jiang, D., et al. "MAGMA: A Multi-Graph based Agentic Memory Architecture for AI Agents." *arXiv, January 2026* Link

[3] Mastra. "Observational Memory: A New Architecture for Long-Term Agent Memory." *mastra.ai, February 2026*

[4] Letta. "Context Repositories: Git-Based Memory for Coding Agents." *letta.com, February 2026*

[5] Anthropic. "Donating the Model Context Protocol and Establishing the Agentic AI Foundation." *December 2025*