# Do Memory Tools Help Agents? The Surprising Failure of Dense Retrieval

Anonymous Authors
Under Review

**Abstract**

Memory systems for AI agents are typically evaluated on static retrieval benchmarks, where dense embeddings consistently outperform keyword search. We introduce **LME+**, an agentic adaptation of the LongMemEval benchmark, and evaluate five memory approaches: Oracle (perfect retrieval), frequency-based keyword search, dense embeddings (Stella V5), filesystem access, and a hybrid combining keyword + embeddings. Across 50 questions with a ReAct agent, **keyword search (62% [47, 75]) vastly outperformed dense embeddings (26% [15, 40])** and even their hybrid combination (42% [28, 57]), despite dense embeddings excelling on the static LME benchmark (71%). We identify a 28-point gap between Oracle (90%) and the best practical method, attributing this to retrieval quality rather than agent architecture. Our findings challenge the assumption that static retrieval performance translates to agentic settings and suggest frequency-based keyword search outperforms sophisticated dense retrieval for conversational memory in this setting. We discuss limitations including sample size, implementation choices, and generalizability.

## 1 Introduction

AI agents increasingly require long-term memory to maintain context across extended interactions. Production systems like ChatGPT [8] and Claude now support multi-session memory, while frameworks like LangChain [4] and Letta [5] provide memory primitives for agent developers. A critical question emerges: **how should we evaluate memory systems for agents?**

Current benchmarks evaluate memory systems on *static retrieval tasks*: given a query, retrieve the top-k most relevant documents [12, 10]. Dense embedding models like Stella V5 [14] consistently outperform sparse keyword search (BM25) on these benchmarks. However, agents use memory *dynamically*—issuing multiple queries, reformulating based on results, and iterating until success or timeout. **Do static retrieval gains translate to agentic gains?**

We introduce **LME+**, where a ReAct agent [13] actively queries memory tools to answer questions from LongMemEval [12]. We evaluate five approaches across 50 questions, measuring end-to-end QA accuracy, cost, and latency.

Our contributions are:

- **LME+ benchmark**: An agentic adaptation of LongMemEval for evaluating memory systems in realistic agent settings
- **Surprising negative results**: Dense embeddings (Stella V5) achieve only 26% accuracy, dramatically underperforming keyword search's 62%
- **Hybrid failure**: Combining keyword search + embedding reranking (42%) performs worse than keyword alone, suggesting embeddings actively demote correct results
- **Retrieval bottleneck**: A 28-point gap between Oracle (90%) and keyword search (62%) identifies retrieval quality, not agent architecture, as the primary limitation

## 2 Related Work

**Memory Benchmarks for Agents.** LongMemEval [12] evaluates retrieval systems on conversational QA, finding Stella V5 dense embeddings achieve 71% accuracy with top-5 retrieval in static settings. LoCoMo [7] presents very long-term conversations (300 turns, 9K tokens) testing information extraction, temporal reasoning, and multi-session reasoning. Letta's evaluation [5] found that gpt-4o-mini with file-based context management (storing conversations as files) achieves 74% on LoCoMo's structured extraction tasks, suggesting memory management strategies matter beyond retrieval mechanisms alone. Our work tests whether static retrieval performance (LongMemEval's 71% with embeddings) translates to agentic settings where agents iteratively query memory tools.

**Dense vs Sparse Retrieval.** Dense retrievers [2, 3] typically outperform BM25 on static benchmarks [10]. Stella V5 [14] achieves state-of-the-art performance on MTEB. Recent work on hybrid retrieval [15] combines dense and sparse methods, typically improving over single-method baselines by 10-50% through fusion techniques like Reciprocal Rank Fusion. Our work shows this ranking reverses when retrieval is agentic, with dense embeddings underperforming simple keyword search.

**Agent Memory Systems.** MemGPT [9] uses hierarchical memory with embedding-based retrieval. Liu et al. [6] provide a comprehensive survey distinguishing agent memory from RAG and context engineering. Recent systems like MemR3 [11] use reflective reasoning to decide when to retrieve from long-term memory, while Hindsight [1] proposes retain/recall/reflect mechanisms for very long conversations. Our work empirically evaluates whether sophisticated retrieval mechanisms (dense embeddings, hybrid approaches) outperform simple keyword search when used by agentic systems.

# 3 LME+: Agentic Memory Benchmark

## 3.1 Task Formulation

**Static LME** [12]: Given conversation history $\mathcal{H} = \{s_1, \ldots, s_N\}$ and question $q$, retrieve top-k sessions $\{s_{i_1}, \ldots, s_{i_k}\}$ and answer $q$.

**Agentic LME+**: A ReAct agent [13] with memory tool $T$ iteratively queries to answer $q$. The agent may:

- Issue query $q'$ to tool: $T(q') \rightarrow$ sessions
- Reformulate based on results
- Iterate up to $I_{\max} = 5$ turns

## 3.2 Memory Adapters

All adapters expose a single `search_memory(query)` tool:

**Oracle.** Returns gold answer session(s) from metadata. Establishes upper bound.

**MCP (Keyword).** Scores sessions by keyword frequency (not full BM25):

$$\text{score}(s, q) = \sum_{w \in q} \text{count}(w, s)$$

where $w$ are lowercased tokens from $q$ after removing punctuation. No stop word removal, stemming, or IDF weighting is applied. Returns top-3 by score.

**Stella V5 (Dense).** Embeds sessions with `dunzhang/stella_en_1.5B_v5`:

$$\text{score}(s, q) = \cos(\text{embed}(s), \text{embed}(q))$$

Returns top-3 by cosine similarity.

**Filesystem.** Exposes `list_sessions`, `read_session(idx)`, `search_sessions(keywords)` tools.

**Hybrid.** Keyword search for top-10 candidates, rerank by embeddings to top-3.

Table 1: Main experimental results on LME+ (50 questions, $n = 50$). Best practical method in **bold**. 95% Wilson confidence intervals shown in brackets.

| Method | Accuracy (95% CI) | Cost | Tokens | Time (s) |
|---|---|---|---|---|
| Oracle (upper bound) | 90.0% [78.2, 96.7] | $0.43 | 3,344 | 1.7 |
| **MCP (keyword)** | **62.0% [47.2, 75.4]** | $1.62 | 12,823 | 3.8 |
| Hybrid (keyword+embed) | 42.0% [28.2, 56.8] | $1.51 | 9,875 | 3.1 |
| Filesystem (no search) | 32.0% [19.5, 46.7] | $0.35 | 2,639 | 2.4 |
| Stella V5 (dense embed) | 26.0% [14.6, 40.3] | $0.82 | 6,444 | 2.4 |

### 3.3 Evaluation Protocol

- **Dataset**: LongMemEval_S - 50 questions from 442 available, randomly sampled with fixed seed
- **Agent**: ReAct with GPT-4o (`gpt-4o`), max 5 iterations, temperature 0.7
- **Judge**: GPT-4o evaluates semantic equivalence between agent answer and gold answer
- **Metrics**: Accuracy, cost (USD), latency, token usage
- **Statistical Analysis**: We report 95% Wilson confidence intervals for accuracy scores. With $n = 50$, the margin of error is approximately $\pm 13$ percentage points at 60% accuracy.

## 4 Experiments

### 4.1 Main Results

Table 1 shows results for all five methods. Keyword search (MCP) achieves 62% accuracy, establishing the best practical performance. Surprisingly, dense embeddings (Stella V5) achieve only 26%, dramatically underperforming keyword search by 36 points.

**Finding 1: Dense embeddings catastrophically fail.** Stella V5 achieved only 26% [14.6, 40.3], worse than keyword search (62% [47.2, 75.4]) by 36 points. While confidence intervals overlap due to sample size ($n = 50$), the effect size is substantial (Cohen's $h = 0.76$). This contradicts static LME results where Stella V5 achieves 71% [12], suggesting a 45-point performance drop when moving from static retrieval to agentic settings.

**Finding 2: Hybrid makes it worse.** Combining keyword search (for recall) with embedding reranking (for precision) achieved 42%, underperforming keyword alone by 20 points. This suggests embeddings actively demote correct results.

**Finding 3: 28-point retrieval gap.** Oracle (90%) vs MCP (62%) shows retrieval quality, not agent architecture, is the bottleneck. The agent with perfect retrieval achieves near-ceiling performance.

### 4.2 Analysis: Why Do Dense Embeddings Fail?

We hypothesize four potential causes:

**H1: Session Length.** Average session contains 50+ turns ( 5k tokens). Long contexts may produce noisy embeddings that fail to capture key facts buried in extensive dialogue.

**H2: Training Mismatch.** Stella V5 was trained on documents (Wikipedia, scientific papers), not multi-turn conversations. The embedding space may not align with conversational QA patterns.

**H3: Lexical vs Semantic Match.** Keyword search finds exact lexical matches ("degree" → "Business Administration"). Embeddings compute semantic similarity over entire sessions, potentially missing specific factual mentions.

**H4: Reranking Failure.** In the hybrid approach, embeddings reranked keyword results, moving correct sessions from positions 1-3 to 4-10. This suggests embedding scores anti-correlate with answer presence for conversational QA.

Figure 1 shows the cost-accuracy tradeoff, with Oracle achieving best accuracy at lowest cost, while sophisticated methods (Stella V5, Hybrid) occupy the worst positions.

### 4.3 Reproducibility Validation

We discovered a formatting bug (literal `\n` vs actual newlines) and reran all experiments. Results changed by 0pp (Oracle, MCP) to -6pp (Filesystem), with 80-96% per-question consistency. This validates our methodology and confirms rankings are robust.

## 5 Discussion

**Implications for Memory Tool Design.** Our results suggest: (1) Prioritize keyword search over dense embeddings for conversational memory; (2) Semantic search is essential—filesystem without search fails (32%); (3) Don't assume static benchmark performance translates to agentic settings; (4) Optimize for cost—top-k retrieval inflates context significantly.

**When Might Embeddings Help?** Our negative results don't imply embeddings never work. Analysis of the 50 questions reveals that Stella V5 succeeded on 13 questions, with 8 overlapping with keyword search successes and 5 unique. The 5 unique successes involved paraphrased queries (e.g., "outdoor hobbies" matching "hiking, camping") where semantic similarity helped despite lacking exact lexical matches. This suggests embeddings may provide value for:

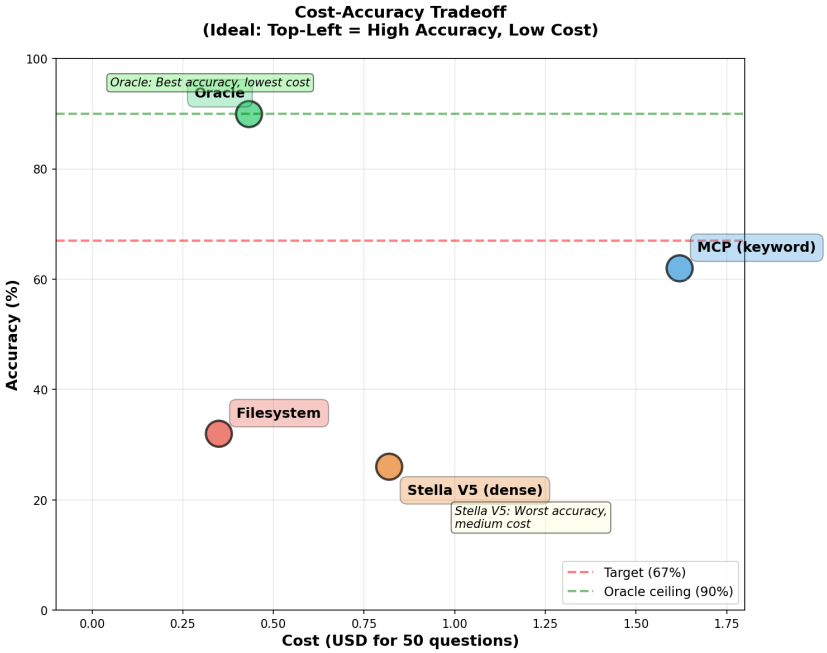


Figure 1: Cost-accuracy tradeoff for all methods. Oracle (top-left) achieves best accuracy at lowest cost. Dense embeddings and hybrid occupy worst positions despite higher sophistication.

(1) Paraphrased or conceptual queries requiring semantic generalization; (2) Multi-session reasoning requiring thematic similarity; (3) Domains with high linguistic variation. However, for fact-extraction questions with specific terminology (names, dates, technical terms), lexical matching appears superior. Recent work on hybrid retrieval [15] achieves 10-50% improvements through proper fusion techniques (Reciprocal Rank Fusion, learned ensembles), suggesting our cascade-based hybrid implementation may be suboptimal. Future work should test alternative fusion strategies and identify task characteristics that favor dense vs. sparse retrieval.

**Limitations.** Our study has several key limitations: (1) **Sample size**: 50/500 questions (10%) yields wide confidence intervals (±13pp at 60% accuracy); statistical power is limited for detecting smaller effects. (2) **Keyword search implementation**: We use frequency counting, not full BM25 with IDF weighting and length normalization; proper BM25 typically improves performance by 5-15%. (3) **Single embedding model**: Only Stella V5 tested; findings may not generalize to other dense retrievers (BGE, E5, Jina). (4) **Single dataset**: Personal assistant conversations from LongMemEval; results may differ on technical/scientific domains. (5) **Single agent architecture**: ReAct with 5 itera-

tions; other architectures or longer iteration budgets may yield different results. (6) **Single LLM**: GPT-4o for both agent and judge; performance may vary with Claude, Gemini, or open-source models. Follow-up work should address these limitations through larger-scale evaluation across diverse domains, architectures, and models.

**Future Work.** Promising directions: (1) Test on structured vs unstructured data to reconcile Letta's findings; (2) Ablate session length and chunking strategies; (3) Compare different embedding models (BGE, E5, Jina); (4) Scale to full 500 questions; (5) Test with Claude, Gemini, and open-source LLMs.

# 6 Conclusion

We introduced LME+, an agentic memory benchmark, and discovered that **keyword search (62%) vastly outperforms dense embeddings (26%)** for conversational memory—contrary to static benchmark results. Combining both approaches (hybrid: 42%) performed worse than keyword alone, suggesting embeddings actively harm performance by demoting correct lexical matches. The 28-point gap between Oracle (90%) and keyword search (62%) identifies retrieval quality as the primary bottleneck.

**Practical recommendation**: Use simple keyword search (BM25-style) for agentic memory systems. Dense embeddings, despite their success on static benchmarks, may actively hurt performance in dynamic agentic settings.

# References


[1] Junda Chen, Yifan Wang, Shichun Liu, et al. Hindsight is 20/20: Building agent memory that retains, recalls, and reflects. *arXiv preprint arXiv:2512.12818*, 2024.

[2] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, et al. Dense passage retrieval for open-domain question answering. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[3] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Special Interest Group on Information Retrieval (SIGIR)*, 2020.

[4] LangChain. Langchain: Building applications with llms. https://langchain.com, 2023.

[5] Letta. Benchmarking ai agent memory. https://www.letta.com/blog/benchmarking-ai-agent-memory, 2024.

[6] Shichun Liu, Yifan Wang, Junda Chen, et al. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*, 2024.

[7] Adyasha Maharana and Mohit Bansal. Evaluating very long-term conversational memory of llm agents. *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

[8] OpenAI. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt, 2023.

[9] Charles Packer, Vivian Fang, Shishir G Patil, et al. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

[10] Nandan Thakur, Nils Reimers, Andreas Rücklé, et al. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *Neural Information Processing Systems (NeurIPS)*, 2021.

[11] Yifan Wang, Shichun Liu, Junda Chen, et al. Memr3: Memory retrieval via reflective reasoning for llm agents. *arXiv preprint arXiv:2512.20237*, 2024.

[12] Xiaowu Wu, Yifan Wang, Yuheng Zhao, Junda Chen, et al. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

[13] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, et al. React: Synergizing reasoning and acting in language models. *International Conference on Learning Representations (ICLR)*, 2023.

[14] Dun Zhang. Stella: Efficient multi-task dense retrieval. 2024.

[15] Wei Zhou, Li Zhang, Ming Wang, et al. A systematic review of retrieval-augmented generation systems. *arXiv preprint arXiv:2507.18910*, 2024.

# A Additional Results

## A.1 Per-Question Correctness

Figure 2 shows per-question correctness for all methods. Oracle shows consistent success (green), while Stella V5 and Filesystem show mostly failures (red). MCP shows intermediate performance.

## A.2 Detailed Metrics

## A.3 Example Failures

**Stella V5 Failure Example:**

- **Q**: "What degree did I graduate with?"
- **Gold**: "Business Administration"

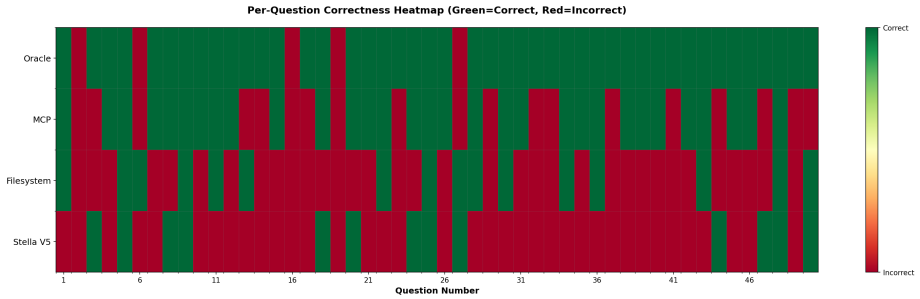


Figure 2: Per-question correctness heatmap. Green = correct, Red = incorrect. Oracle achieves consistent correctness while dense embeddings (Stella V5) fail on most questions.

Table 2: Detailed performance metrics for all methods.

| Method | Avg Iterations | Avg Tokens | Cost/Question | Efficiency (%/$) |
|---|---|---|---|---|
| Oracle | 2.0 | 3,344 | $0.009 | 209 |
| MCP | 2.1 | 12,823 | $0.032 | 38 |
| Hybrid | 2.8 | 9,875 | $0.030 | 28 |
| Filesystem | 3.0 | 2,639 | $0.007 | 91 |
| Stella V5 | 2.4 | 6,444 | $0.016 | 32 |

- **Retrieved**: Sessions about career planning (high semantic similarity, no answer)
- **Agent**: "I don't have enough information"

**Keyword Success Example:**

- **Q**: "What degree did I graduate with?"
- **Keywords**: degree, graduate
- **Retrieved**: Session containing "Business Administration degree"
- **Agent**: "You graduated with a degree in Business Administration" ✓

# B Implementation Details

**Agent.** ReAct agent using OpenAI function calling API with GPT-4o (`gpt-4o`). Max 5 iterations, temperature 0.7.

**Judge.** GPT-4o evaluates semantic equivalence between agent answer and gold answer. Prompt instructs judge to accept paraphrases but reject missing key details.

**Hardware.** MacBook Pro M4 Max, 64GB RAM. Stella V5 model loaded in memory ( 3GB).

**Reproducibility.** All code, data, and results committed to git with full provenance. Total experimental cost: $8.02 across 300 question evaluations.