Context/Memory Implementation & Performance: An Enhanced AI Agent with Long-Term Memory

Ziye Deng, Zheqi Liu, Zhengan Cheng, Tianzuo Liu, Hengjia Yu
UC San Diego

Abstract

We present an enhanced AI agent system with comprehensive context and memory capabilities. Our implementation features a hybrid extraction approach combining pattern-based and LLM-based methods, persistent vector storage using ChromaDB, and dynamic tool optimization based on user preferences. Experimental results demonstrate significant improvements over the baseline: 100% task completion on all benchmarks (up from 21.4% on long conversations), response quality improvements of up to 114.3%, and 85.20% accuracy on cross-session memory retrieval. The system provides effective personalization through automatic preference extraction and maintains coherence across extended multi-turn conversations.

1 Implementation

1.1 Context Extraction from Conversations and Tool Outputs

Design Decision: We employ a hybrid extraction approach combining pattern-based and LLM-based methods to balance real-time performance with extraction accuracy.

Pattern-Based Extraction Our pattern-based extraction system provides fast, rule-based extraction for common preference indicators. It enables real-time preference detection during conversations and classifies preferences into communication style, domain interests, response format, tool preferences, and interaction patterns. The system uses confidence scoring to prioritize high-quality extractions.

LLM-Based Extraction For deeper analysis, we leverage Claude's structured output capabilities for semantic analysis, capturing nuanced preferences that pattern matching might miss while providing structured preference data for storage and retrieval.

Key Technical Challenge: Balancing extraction speed (for real-time use) with accuracy (for quality personalization). The hybrid approach addresses this by using fast pattern matching for common cases and LLM extraction for complex scenarios.

Preference Management The preference management system includes:

- Intelligent merging of preferences across sessions with weighted confidence scores
- Temporal tracking to identify preference evolution over time
- Deduplication to prevent redundant storage

1.2 Storage System

Design Decision: ChromaDB serves as the vector database backend for persistent, scalable semantic search.

Why ChromaDB We selected ChromaDB for several reasons: it is a lightweight, embedded database suitable for local deployment with native support for vector similarity search. It provides efficient metadata filtering for user isolation and requires no external service dependencies.

Key Technical Challenges Solved

1. **User Isolation:** ChromaDB's flat metadata structure required careful design to ensure complete user data isolation using `user_id` metadata filters, preventing cross-user data leakage in multi-tenant scenarios.
2. **Metadata Management:** Complex nested metadata (session IDs, document types, timestamps) needed flattening for ChromaDB compatibility while maintaining query flexibility.
3. **Embedding Pipeline:** Integration with OpenAI embeddings (`text-embedding-3-small`) with batch processing to optimize API usage and reduce latency.
4. **Semantic Search:** Cosine similarity search with configurable similarity thresholds to balance recall (finding relevant memories) with precision (avoiding irrelevant results).

Storage Architecture The storage architecture follows a three-layer design: Agent Memory Tools (`search_memory`, `store_memory`) connect to the VectorStorageBackend (ChromaDB with persistent storage), which in turn interfaces with the EmbeddingService (OpenAI).

1.3 Tool Call Optimization Based on Context

Design Decision: Dynamic tool loading with user-specific memory tools and context-aware tool selection.

Key Innovation: Integration with LangMem enables per-user memory tool generation while maintaining complete namespace isolation. This allows the agent to have memory capabilities without exposing one user's data to another.

Context-Aware Optimization

- User preferences influence tool prioritization (e.g., language preferences trigger translator tool suggestions)

- Historical context retrieved through semantic search informs tool selection
- Memory tools automatically search user's past conversations for relevant information

Technical Challenge: Ensuring memory tools are dynamically created per-user while maintaining performance and isolation. LangMem's namespace-based isolation solves this by creating separate tool instances per user namespace.

1.4 Error Handling and Resource Management

Design Philosophy: Graceful degradation ensures the agent continues operating even if memory components fail, maintaining core functionality while logging errors for debugging.

Key Design Decisions

1. **Graceful Degradation:** If memory storage is unavailable, the agent falls back to stateless operation, ensuring reliability in production environments.
2. **Rate Limit Management:** Configurable token limits and request delays prevent API rate limit violations, critical for production deployment with Anthropic's API constraints.
3. **Resource Cleanup:** Proper connection management for ChromaDB ensures no resource leaks during long-running operations.
4. **Structured Logging:** Comprehensive logging with request tracking, user context, and performance metrics enables debugging and performance monitoring.

2 Experimental Results and Performance Analysis

2.1 Benchmark Results

We evaluated the enhanced agent on three benchmark datasets: **short**, **medium**, and **long** conversations, plus the **LoCoMo** cross-session memory benchmark.

Evaluation Framework

Benchmark Datasets

- **Short Benchmark** (6 cases): Basic tool usage validation, single-turn interactions, answer correctness validation
- **Medium Benchmark** (4 cases): Multi-turn conversations (2-4 turns), context retention testing, preference extraction validation
- **Long Benchmark** (4 cases): Extended conversations (9-11 turns), complex task completion, memory integration testing
- **LoCoMo Benchmark**: Cross-session memory evaluation, 19 sessions, 419 conversation turns, 196 QA questions across 5 categories

Evaluation Metrics

- **Accuracy Metrics:** Answer correctness (semantic matching), per-turn and per-case analysis
- **Performance Metrics:** Latency per turn, memory storage statistics, token usage and rate limiting
- **Quality Metrics:** Preference extraction success rate, context retention across turns, personalization effectiveness

Short Benchmark Results

Table **??** presents the short benchmark results covering 6 test cases for basic tool usage.

Table 1: Short Benchmark Results

Metric	Result
Total Cases	6
Answer Correct	6/6 (100%)
Response Quality	4.8/5.0
Tool Call Efficiency	1.00
Average Latency	∼14.0 seconds

Key Observations The system achieved **perfect accuracy** on all test cases with **perfect tool call efficiency** (1.00). All expected tools were called with correct frequency. The agent successfully handled various task types including calculator, weather, translator, file search, and tool combinations, demonstrating context awareness in multi-turn conversations.

Medium Benchmark Results

Table **??** shows the medium benchmark results for 4 test cases with multi-turn conversations requiring context retention.

Table 2: Medium Benchmark Results

Metric	Result
Total Cases	4
Answer Correct	4/4 (100%)
Response Quality	4.775/5.0
Tool Call Efficiency	1.00
Average Latency	∼12.0 seconds

Key Observations The system achieved **perfect accuracy** with excellent response quality (4.775/5.0) on multi-turn conversations. Successful context retention across 2-4 conversation turns was observed, along with effective preference extraction and personalization.

Key Improvements Demonstrated

1. **Perfect Context Retention:** Agent maintains conversation context across multiple turns (2-4 turns) with 100% accuracy

2. **Effective Preference Personalization:** User preferences extracted and applied in 2/4 cases, including `living_place`, `hobbies`, and `preferred_translation_language`
3. **High Response Quality:** 4.775/5.0 average quality rating demonstrates excellent response relevance and coherence
4. **Multi-Turn Coherence:** Successfully handles complex multi-turn tasks including reading comprehension, trip planning, arithmetic, and translation

Long Benchmark Results

Table **??** presents the long benchmark results for 4 complex test cases with 9-11 conversation turns.

Table 3: Long Benchmark Results

Metric	Result
Total Cases	4
Answer Correct	4/4 (100%)
Response Quality	4.65/5.0
Tool Call Efficiency	0.714

Key Observations The system achieved **perfect accuracy** on all test cases with **good tool call efficiency** (0.714). Most expected tools were called correctly, with some cases using alternative tools. Excellent response quality (4.65/5.0) was observed on extended conversations, with successful context maintenance across 9-11 conversation turns.

Key Improvements

1. **Long Conversation Handling:** Maintains coherence across 9-11 turns
2. **Memory Integration:** Successfully stores and retrieves information using `store_memory` tool
3. **Preference Persistence:** Extracted preferences influence responses throughout long conversations

LoCoMo Cross-Session Memory Benchmark

Table **??** shows the cross-session memory evaluation results.

Table 4: LoCoMo Benchmark Results

Metric	Result
Personas Run	1
Total Chunks Stored	144
Total Characters Stored	67,162
QA Accuracy	167/196 (85.20%)
Response Quality	4.29/5.0
Verified Memories	100

Table **??** presents the category-wise breakdown of results.

Table 5: LoCoMo Category Breakdown

Category	Total	Correct	Accuracy
Category 1	31	22	71.0%
Category 2	37	33	89.2%
Category 3	11	10	90.9%
Category 4	70	62	88.6%
Category 5	47	40	85.1%

Analysis The system achieved **high overall accuracy** (85.20%) across all question categories with **good response quality** (4.29/5.0). Excellent performance was observed in specific categories (Categories 2-5: 85-91% accuracy). Category 1 performance (71% accuracy for identity/personal information questions) indicates that category-specific optimization for identity/personal information extraction would improve performance.

Note on Temporal Questions: The benchmark does not include exact timestamps of sessions when storing memories or querying the model. Therefore, the model is only expected to output **relative time** information (e.g., “earlier”, “in a previous session”) rather than absolute timestamps.

2.2 Performance Improvements Over Baseline

Better Generation Quality

Table **??** shows the task completion rate comparison between phases.

Table 6: Task Completion Rate Comparison

Benchmark	Phase 1	Phase 2	Improvement
Short	94.3%	100%	+5.7%
Medium	66.7%	100%	+33.3%
Long	21.4%	100%	+78.6%

Table **??** presents the response quality rating comparison.

Table 7: Response Quality Ratings (5-point scale)

Benchmark	Phase 1	Phase 2	Improvement
Short	4.45/5.0	4.8/5.0	+7.9%
Medium	3.85/5.0	4.775/5.0	+24.0%
Long	2.17/5.0	4.65/5.0	+114.3%

Key Observations

- **Short:** Small improvement from already high baseline (94.3% → 100%)
- **Medium:** Major improvement (+33.3%), demonstrating better context retention in multi-turn conversations

- **Long:** Dramatic improvement (+78.6%), showing the critical value of memory system for extended conversations

Better Personalization Based on User Preferences

Preference Extraction Success The system successfully extracted preferences in multiple test cases:

- `living_place`: San Diego
- `hobbies`: Diving
- `preferred_translation_language`: French, Spanish
- `teaching_tone`: Casual/friendly

Personalization Examples

1. **Location-based personalization:** The agent responded "Since I know you live in San Diego, let me compare the water temperatures for you..." providing relevant comparisons between San Diego and Hawaii water temperatures.
2. **Language preference:** Consistent translation language maintained across conversations with teaching style adapted to user's learning preferences.

Response Time/Latency

Average Latency by Benchmark

- **Short:** ∼14.0 seconds per turn
- **Medium:** ∼12.0 seconds per turn
- **Long:** ∼17.0 seconds per turn (complex multi-tool operations)

Optimization Opportunities

- Batch embedding generation reduces per-request overhead
- Memory search is asynchronous and non-blocking
- Tool execution parallelization where possible
- Embedding caching to reduce repeated API calls

2.3 Side-by-Side Comparisons with Baseline (Phase 1)

Phase 1 Baseline Characteristics

Phase 1 Implementation

- **Architecture:** Basic ReAct pattern with LangGraph
- **Memory:** No long-term memory, only in-memory conversation history
- **Context:** Limited to current conversation window
- **Personalization:** No user preference extraction or storage
- **Tools:** Static tool set (calculator, weather, translator, web_reader, file_system_search)
- **Session Management:** Basic thread-based isolation, no cross-session continuity

Key Limitations

1. No persistent storage—all context lost after session ends
2. No user preference learning—generic responses for all users
3. Limited context window—cannot maintain coherence in long conversations
4. No cross-session memory—cannot recall information from previous sessions

Comprehensive Feature Comparison

Table **??** presents a comprehensive feature comparison between phases.

Performance Comparison

Comparison 1: Context Retention **Phase 1 (Baseline):**

- No long-term memory
- Limited context window (only current session)
- Cannot reference information from previous sessions
- Context lost when conversation ends

Phase 2 (Enhanced):

- Maintains context across 9-11 turns (long benchmark)
- Persistent storage in ChromaDB
- Recalls user preferences from previous sessions
- References earlier conversation points across sessions

Quantitative Improvement:

- Long conversations: Phase 1 struggled with coherence beyond 3-4 turns; Phase 2 maintains 100% accuracy across 9-11 turns
- Cross-session: Phase 1 had 0% capability; Phase 2 achieves 85.20% accuracy on LoCoMo benchmark

Comparison 2: Personalization **Phase 1 (Baseline):**

- Generic responses for all users
- No user-specific adaptation
- No preference learning

Phase 2 (Enhanced):

- Extracts and stores user preferences
- Adapts responses based on stored preferences
- Maintains consistent personalization across sessions
- Context-aware tool selection

Comparison 3: Cross-Session Memory **Phase 1 (Baseline):**

- No cross-session memory capability (0% accuracy)
- Each session starts from scratch

Phase 2 (Enhanced):

- Stores conversation summaries in ChromaDB
- Retrieves relevant memories using semantic search
- 85.20% overall accuracy on LoCoMo benchmark
- 85-91% accuracy in specific question categories

Architecture Comparison

Phase 1 Architecture:

User Request → Agent (ReAct) → Tools → Response

Phase 2 Architecture:

User Request → Agent (ReAct) → Context Extraction
→ Preference Storage
↓

Table 8: Feature Comparison Between Phase 1 and Phase 2

Feature	Phase 1 (Baseline)	Phase 2 (Enhanced)	Improvement
Long-term Memory	None	ChromaDB vector storage	New capability
Context Extraction	None	Pattern + LLM-based extraction	New capability
Preference Learning	None	Automatic extraction and storage	New capability
Cross-Session Recall	Not possible	85.20% accuracy (LoCoMo)	New capability
Personalization	Generic responses	User-specific adaptation	Significant improvement
Context Retention	Limited to session	Multi-turn + cross-session	Major improvement
Tool Optimization	Static tool set	Context-aware tool selection	Enhanced
Storage Backend	None	Persistent ChromaDB	New capability

Tools ← Memory Search ← ChromaDB ← Embeddings
↓
Response (Personalized)

Key Architectural Improvements

1. **Memory Layer:** Added persistent vector storage (ChromaDB)
2. **Extraction Layer:** Added context and preference extraction
3. **Personalization Layer:** Added user-specific adaptation
4. **Tool Enhancement:** Dynamic tool loading based on user context

2.4 Value Demonstration

The context/memory system provides clear value:

1. **Improved User Experience:**
    - Users don't need to repeat information
    - Responses are personalized and contextually relevant
    - Long conversations maintain coherence
2. **Better Task Completion:**
    - Higher accuracy on multi-turn tasks
    - Successful tool selection based on context
    - Complex planning tasks completed successfully
3. **Scalability:**
    - Vector storage enables efficient semantic search
    - User-specific namespaces prevent data leakage
    - Persistent storage survives server restarts

2.5 Statistical Summary

Table **??** presents the comprehensive performance summary.

Note: LoCoMo benchmark answers do not require tool calls, and due to token limits we do not test its latency.

3 Novelty and Technical Depth

3.1 Novel Contributions

Hybrid Context Extraction System

Novelty: The system combines **pattern-based extraction** with **LLM-based structured extraction** for comprehensive preference detection.

- **Pattern-Based** (`ContextExtractor`): Fast, rule-based extraction for common preference indicators
- **LLM-Based** (`extract_preferences`): Deep semantic analysis using Claude's structured output for nuanced preferences

This hybrid approach balances speed and accuracy, allowing real-time extraction while maintaining high-quality preference detection.

User-Specific Memory Namespaces with LangMem Integration

Novelty: Integration of LangMem with ChromaDB backend, providing:

- **Dynamic tool generation** per user
- **Namespace isolation** for multi-user scenarios
- **Unified API** for both agent tools and REST endpoints

The system creates user-specific memory tools (`search_memory`, `store_memory`) dynamically, ensuring complete isolation while sharing the same storage backend.

Preference-Aware Tool Selection

Novelty: Tool selection is influenced by extracted preferences:

- Language preferences → Translator tool prioritization
- Communication style → Response format adaptation
- Domain interests → Relevant tool suggestions

This goes beyond simple tool availability to context-aware tool optimization.

3.2 Technical Depth

Vector Storage Architecture

Implementation Depth

Table 9: Phase 2 Performance Summary

Benchmark	Cases	Answer Accuracy	Response Quality	Tool Call Efficiency	Avg Latency
Short	6	100%	4.8/5.0	1.00	14.0s
Medium	4	100%	4.775/5.0	1.00	12.0s
Long	4	100%	4.65/5.0	0.714	17.0s
LoCoMo	1	85.20%	4.29/5.0	N/A	N/A

- **ChromaDB Integration:** Full implementation of vector storage interface
- **Embedding Pipeline:** OpenAI embeddings with batch processing
- **Metadata Management:** Complex metadata flattening for ChromaDB compatibility
- **Query Optimization:** Efficient similarity search with configurable thresholds

Technical Challenges Solved

- Metadata flattening for ChromaDB's flat metadata structure
- User isolation using metadata filters
- Batch embedding generation for performance
- Error handling and connection management

Asynchronous Architecture

Implementation Depth

- Full async/await support throughout the stack
- Non-blocking memory operations
- Concurrent tool execution where possible
- Proper resource cleanup and error handling

Code Quality

- Type hints throughout
- Comprehensive error handling
- Structured logging with request tracking
- Modular design with clear interfaces

Memory Management System

Implementation Depth

- **Short-term memory:** In-memory conversation history
- **Long-term memory:** Persistent vector storage
- **Preference storage:** Structured preference extraction and storage
- **Session management:** Session-based document organization

Memory Lifecycle

1. Messages added to short-term memory
2. Preferences extracted periodically
3. Session summaries generated
4. Long-term storage in ChromaDB with embeddings
5. Semantic search for context retrieval

4 Conclusion

The enhanced agent with context/memory implementation demonstrates significant improvements over the baseline:

1. **High Accuracy:** 100% on all benchmarks (short, medium, and long), representing improvements of +5.7%, +33.3%, and +78.6% over Phase 1 baseline
2. **High Response Quality:** Excellent quality ratings (4.65-4.8/5.0) across short, medium, and long benchmarks, representing improvements of +7.9%, +24.0%, and +114.3% over Phase 1 baseline. LoCoMo benchmark achieved 4.29/5.0 quality rating.
3. **Effective Personalization:** Successful preference extraction and application in 50% of medium benchmark cases (2/4 cases)
4. **Cross-Session Memory:** Functional long-term memory with 85.20% overall accuracy (85-91% in Categories 2-5)
5. **Robust Architecture:** Modular design with proper error handling and resource management
6. **Significant Improvements over Phase 1:** New capabilities in memory, personalization, and cross-session recall

Key Achievements

- Comprehensive context extraction (pattern + LLM-based)
- Persistent vector storage with ChromaDB
- User-specific memory namespaces
- Preference-aware tool selection
- Cross-session memory retrieval

Future Work

- Category 1 (identity/personal) accuracy improvement
- Latency optimization through parallelization
- Advanced preference merging with temporal decay
- Context compression for very long conversations

The system provides a solid foundation for production deployment with measurable improvements in user experience, task completion, and personalization.

A Experimental Results Summary

A.1 Short Benchmark Detailed Results

Table **??** presents the detailed results for each short benchmark test case.

Table 10: Short Benchmark Detailed Results

Test ID	Tools Used	Correct	Latency (s)
test_001	calculator		12.2
test_002	get_weather		15.9
test_003	translator		13.7
test_004	file_system_search		18.7
test_005	web_reader		10.4
test_006	weather, translator		17.1

A.2 Medium Benchmark Detailed Results

Table **??** shows the medium benchmark detailed results.

Table 11: Medium Benchmark Detailed Results

Test ID	Turns	Correct	Preferences
test_006	4	4/4	None
test_007	3	3/3	living_place, hobbies
test_008	2	2/2	None
test_009	3	3/3	translation_lang

A.3 Long Benchmark Detailed Results

Table **??** presents the long benchmark detailed results.

Table 12: Long Benchmark Detailed Results

Test ID	Turns	Correct	Memory Used
test_030	11	11/11	Context retention
test_031	4	4/4	store_memory
test_032	11	11/11	Preferences
test_033	9	9/9	Context retention

A.4 LoCoMo Memory Benchmark Results

Storage Statistics

- Sessions: 19
- Total turns: 419
- Chunks stored: 144
- Characters stored: 67,162

QA Performance

- Total questions: 196
- Correct answers: 167
- Overall accuracy: 85.20%
- Category 1 accuracy: 71.0%
- Category 2-5 accuracy: 85-91%