International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V73I11P103

Volume 73 Issue 11, 17-25, November 2025
© 2025 Seventh Sense Research Group®

Original Article

Contextual Coherence in Conversational AI: Leveraging a Memory Agent

Vidya Vishal Wadkar

Distinguished Engineer and Individual Researcher, New Jersey, USA.

Corresponding Author : wadkarvidya84@gmail.com

Received: 17 September 2025 Revised: 25 October 2025 Accepted: 12 November 2025 Published: 28 November 2025

Abstract - *This study presents a Memory Agent Framework specifically designed for engaging conversational AI while addressing context coherence and memory scalability. The design is based on a Model Context Protocol (MCP) to synchronise many conversational agents over four memory layers inspired by cognitive functions (Short-Term, Episodic, Semantic, and Procedural). The Python-based asynchronous orchestration helps quickly retrieve memories, using FAISS/Pinecone vector storage and a Neo4j knowledge graph to dynamically reason a conversation (like a human would use memories). When the Memory Agent Framework was evaluated in three different modes (baseline, fine-tuned, and embedding-enhanced), it showed a real performance advantage through the Memory Agent Framework compared with baseline evaluation mode. In the Short-Term Memory, each perplexity level decreased from 327.18 to 294.47; in the Episodic Memory, it decreased from 348.49 to 313.64; and in Semantic Memory, it decreased from 344.22 to 309.79. However, the semantic coherence improved by 28.5% in contextual reliability, reaching 0.5499. The fine-tuned models achieve BLEU and ROUGE-L scores above 0.5, indicating improved grammatical correctness and relevance. These results suggest that the Memory-driven paradigm improves multi-turn conversation comprehension, contextual fragmentation, and episodic interaction continuity and understanding. Overall, the Memory Agent architecture supports scalable, context-rich conversation systems that promote coherence and adaptive reasoning in real-world communication.*

Keywords - *Multi-Agent Conversational AI, Memory Agent Framework, Model Context Protocol, Layered Memory Architecture, Contextual Coherence.*

1. Introduction

Large Language Models (LLMs) have fundamentally changed the reality of conversational AI interactions, which can now be as rich and fluent as never before [1]. They are still fighting long-term coherence, personalisation, and contextual carryover over extended dialogue sessions. The challenge has primarily been a function of finite context windows, typically leading to conversational data in prior interactions being lost, and thus creating a break in the natural flow of interaction. Particularly, the same challenges exist when it comes to multi-agent conversational ecosystems with many agents, retrievers, planners, tool executors, and dialogue handlers working together towards shared outcomes. Each agent typically exists in a stateless scenario, retaining only its local context without knowledge of the total engagement history. This execution of an implementation of distributed memory leads to fragmented memory artefacts, redundant memory, inconsistent personalisation, and unnecessary computational overhead. The absence of a unified, governance-aware, scalable memory architecture to ensure coherence among stateful conversational agents. The problem statement, first, is related to how memory can be externalised and orchestrated effectively with trust, auditability, and privacy maintained in multi-agent environments. To address these problems, this paper conceptualises a Memory Agent being developed through MCP to standardise memory access, update, and persistence mechanisms across agents [2]. This method enables conversational bots to behave as though they have constant context knowledge while remaining modular and stateless [3]. In order to provide contextual coherence and adaptive learning in multi-agent conversational AI, this approach lays memory as the fundamental layer. The suggested design unifies context management across several stateless conversational agents by integrating a centralised Memory Agent, as shown in Figure 1.

Figure 1 presents a Memory Agent framework that ensures contextual coherence in multi-agent conversational AI. User input flows through PatExpert to a Meta-Agent, which coordinates specialised Sub-Agents and Critique-Agents. Outputs are refined via feedback loops and stored in a Conversational Database, enabling memory-driven, context-aware dialogue across tasks like summarisation, classification, and prediction.

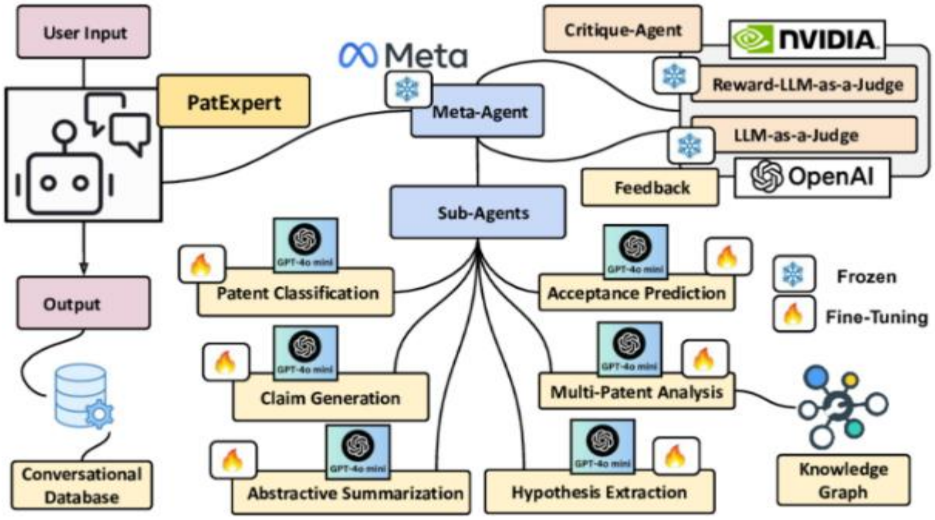

Fig. 1 Proposed Memory Agent Framework for Multi-Agent Conversational AI [4]

1.1. Memory Matters in Conversational AI Conversational

Without a proper memory structure, AI cannot attain full intelligence or user delight. Engagement and trust are jeopardised by systems that forget previous interactions or are unable to remember user preferences [5]. Coherence, customisation, and task continuity are guaranteed by an efficient memory. Scalable AI systems require a layered memory model, which is based on cognitive psychology [6]. For coherence at the turn and session levels, Short-Term Memory (STM) preserves immediate context [7]. Episodic Memory enhances long-term personalisation by storing organised historical data about previous sessions [8]. Contextual truths and factual information are stored in semantic memory regardless of time, as shown in Figure 2 [9].

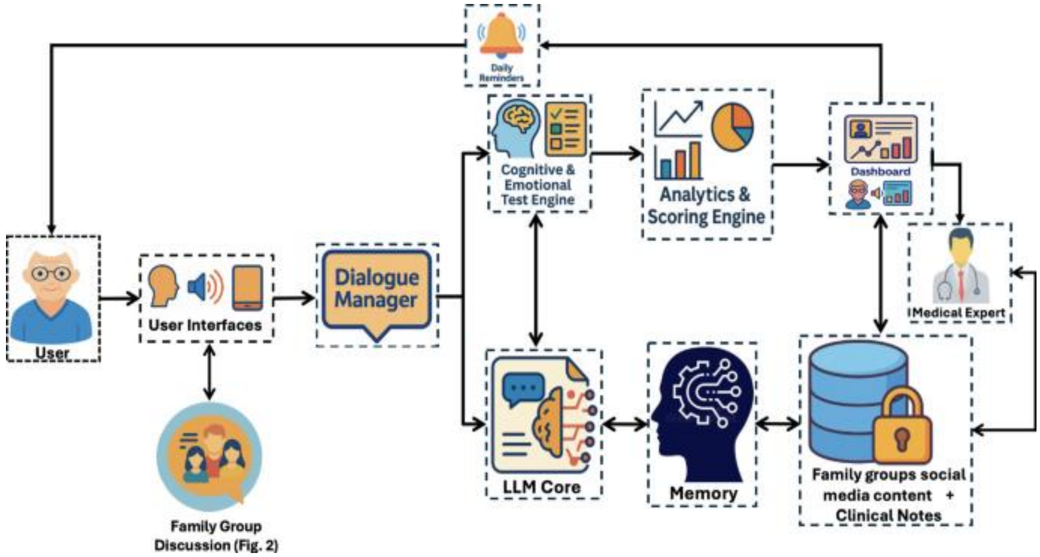
Daily Reminders
Cognitive & Emotional Test Engine
Analytics & Scoring Engine
Dashboard
Medical Expert
User
User Interfaces
Dialogue Manager
LLM Core
Memory
Family groups social media content + Clinical Notes
Family Group Discussion (Fig. 2)

Fig. 2 Cognitive-Inspired Layered Memory Model for Conversational AI [9]

Figure 2 outlines a memory-driven conversational framework for cognitive and emotional assessment. User interactions via dialogue and daily engagement flow into a test engine, which connects to analytics powered by LLMs and memory modules. These modules integrate clinical notes and social media content to generate expert-facing insights, ensuring context-aware, personalised evaluations.

Workflows and activity sequences are controlled by procedural memory to ensure consistent system behaviour [10]. Conversational systems either forget important information or overburden context windows with unrelated material in the absence of these hierarchical levels. Therefore, memory orchestration is essential to creating AI systems that are adaptable, scalable, and user-centric.

The novelty of this research lies in the cognitively inspired memory orchestration model that unifies short-term, episodic, semantic, and procedural memory under a single MCP-based architecture.

2. Literature Review

2.1. Context Management and Dialogue Continuity

The main goal of early conversational AI research was to increase dialogue continuity by expanding Large Language Models' (LLMs') effective context window [11]. To preserve pertinent conversation history, techniques including prompt-chaining, context summarisation, and knowledge graph-based augmentation were created. Although token-level context can be effectively extended by these methods, durable memory continuity spanning multi-turn or multi-session interactions is not addressed, and they are still session-dependent. Consequently, earlier knowledge is irrevocably lost once the context length is exceeded, which diminishes coherence and personalisation in lengthy interactions.

2.2. Initial Frameworks for Multi-Agents

Buffer memory architectures, first proposed by OpenAI's early conversational frameworks, preserve local context by storing brief user-agent interactions [12,13]. However, because these buffers lacked a temporal hierarchy and semantic depth, they produced surface-level recall that was irrelevant to the context.

Conversational agents were modularised into retrievers, planners, and executors by later innovations like Google's ReAct architecture and LangChain [14]. Despite boosting modularity and interpretability, these systems relied on localised and agent-specific memory, resulting in fragmented knowledge representation and duplicate retrieval processes. This decentralised method hindered global context awareness and system scalability.

2.3. Retrieval-Augmented Generation (RAG) Approaches

By incorporating vector-based document retrieval into LLM inference pipelines, recent advancements in Retrieval-Augmented Generation (RAG) enhanced factual accuracy and knowledge grounding [15]. RAG systems improved reaction dependability by effectively referencing outside sources [16]. However, because retrieval is still unaffected by previous conversational context, RAG models still lack consistency and personalisation [17, 18]. Additionally, factual retrieval, rather than procedural or episodic continuity, which are crucial for adaptive and customised dialogue systems, is the main focus of RAG-based architectures.

Although current models have improved context extension and knowledge retention, none of them offer a single memory orchestration layer that can control information across several agents, according to the studied literature. A governance-aware, scalable, and cognitively inspired memory management framework that allows conversational agents to cooperatively exchange, evolve, and maintain context is still desperately needed. This gap is the basis for the current study, which unifies short-term, episodic, semantic, and procedural memory in multi-agent ecosystems by introducing a Memory Agent Framework based on the Model Context Protocol (MCP).

3. Methodology

This procedure employs Memory Agent to establish contextual coherence in conversational AI systems. Upon user interaction, the Conversational Agent functions as a planner, retriever, and executor of tools to understand questions and determine the next step.

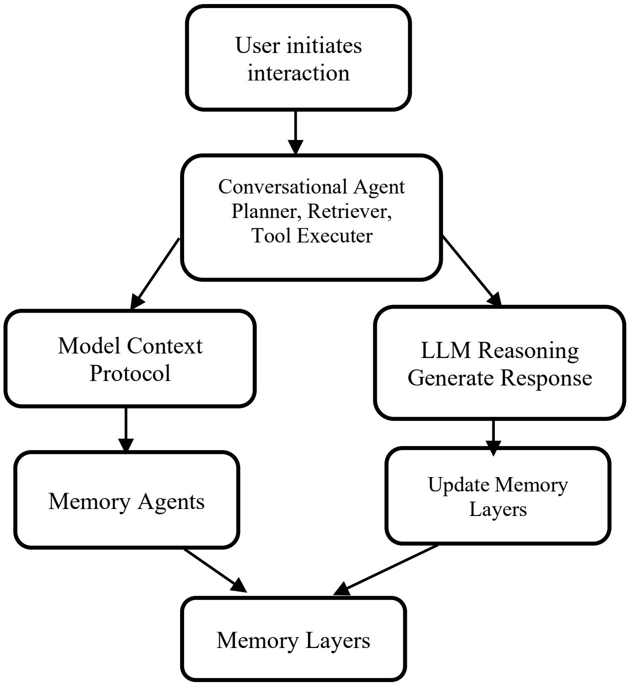
User initiates interaction
Conversational Agent Planner, Retriever, Tool Executer
Model Context Protocol
LLM Reasoning Generate Response
Memory Agents
Update Memory Layers
Memory Layers

Fig. 3 Methodology Framework

The Model Context Protocol (MCP) allows for the efficient communication of components to ensure context continuity. Using the context retrieved and the current user input, the LLM reasoning module provides logical and context-sensitive responses. Memory Agents manage the retrieval and storage of previous interactions, retaining long-term and short-term contexts. During the conversation, Memory Layers are updated hierarchically. The use of layers provides a way to ensure logical progression, prevent context fragmentation, and increase involvement of users in a way that enables contextual coherence throughout multiple turn conversations.

3.1. Overview of the Architecture

Four main parts make up the architecture of the suggested system. Conversational Agents (Dialogue Handler, Tool Executor, Planner, and Retriever). The central controller for unified context management is called a memory agent. Memory Layers: Procedural, Semantic, Episodic, and Short-Term Memory. A standardised interface for memory operations and communication is the Model Context Protocol (MCP). Through the MCP interface, each conversational agent exchanges information with the Memory Agent, sending new dialogue data or asking for contextual information. In order to guarantee that every component has access to a single, consistent knowledge source, this approach externalises memory from individual agents. In multi-agent contexts, the architecture ensures reduced latency and improved scalability by supporting both synchronous and asynchronous memory updates.

3.2. Design of Memory Agents

These coordinates synchronise information among agents. Four separate memory subsystems that are modelled after human cognitive architecture are included in it:

Table 1. Memory Types and Storage in the Memory Agent Framework

Memory Type	Functionality	Storage Type
Short-Term Memory (STM)	Retains immediate conversational context and recent utterances.	In-memory buffer
Episodic Memory	Stores structured representations of past dialogues and interactions.	Time-indexed database
Semantic Memory	Maintains factual knowledge and user-specific data independent of time.	Vector database
Procedural Memory	Contains predefined workflows, reasoning paths, and execution templates.	Knowledge graph

3.3. The Function of Model Context Protocol (MCP)

Consistent contextual reasoning across interactions is made possible by the Model Context Protocol (MCP), which creates a standardised framework for communication between conversational agents and the memory layer. MCP functions in three main stages and specifies APIs for synchronisation, updates, and context retrieval. An agent asks the Memory Agent for pertinent contextual data during the Context Request Phase.

The Memory Agent creates a single context vector during the Memory Retrieval Phase by combining information from the episodic, procedural, short-term memory (STM), and semantic layers. To maintain historical continuity and facilitate auditability, new agent outputs are ultimately recorded as structured changes during the Context Update Phase. MCP ensures excellent interoperability and efficiency of operations through the use of asynchronous messaging and lightweight JSON schemas. The protocol architecture enables coherent multi-turn conversational flows across multiple concurrent agents, allowing for effortless scaling in distributed or cloud-based environments.

3.4. Operational Workflow and Data Flow

A conversational agent and the user initiate a conversation, and the agent calls on MCP to retrieve the prior context. The Memory Agent retrieves and assembles the context bundle. The conversational agent uses LLM reasoning to deliver the output. The conversational agent logs the response and its associated metadata back to the relevant memory layers for later retrieval. The feedback-based memory development enables the system to continuously learn and improve its contextual knowledge over sessions. This results in increased coherence and tailoring.

3.5. Strategy for Implementation

The system architecture employs a modular memory-agent system to maintain contextual coherence in conversational AI and to enable multi-turn conversations. The internal core is a Python-based agent framework that facilitates asynchronous I/O (using FastAPI or LangChain for orchestration). Moreover, semantic memory is managed using a vector storage technology (FAISS or Pinecone) to maintain and retrieve contextually relevant embeddings to support dynamism within conversations. Procedural linkages and knowledge about workflows are recorded as a Neo4j knowledge graph, which facilitates reasoning about the sequence of conversations as they occurred, while structured episodic interactions are maintained in a relational database (PostgreSQL). Docker is used to containerise each module, enabling flexible deployment in cloud or distributed systems. Real-time memory retrieval and horizontal scalability are guaranteed by this approach, which together improve contextual continuity throughout concurrent agent interactions.

4. Results and Discussion

Here, it provides a comprehensive analysis of the proposed Memory Agent architecture for conversational AI. The work revolves around three experimental setups: (i) a baseline language model that has already been trained (Mode 1), (ii) a variant that has been fine-tuned to fit the dataset (Mode 2), and (iii) a retrieval system that uses embeddings to ensure semantic coherence (Mode 3). It evaluates the efficiency of various modes across three levels of memory: Short-Term Memory (STM), Episodic Memory (EM), and semantic memory, using a battery of quantitative metrics, including perplexity, BLEU, ROUGE-L, and semantic coherence.

In this section, the effects of memory-layer design and model adaptation on contextual continuity, multi-turn coherence, and conversation quality are shown via tables and graphs. This provides them with a sense of the framework's efficacy for conversational AI that takes context into account.

Table 2. Performance of Memory Layers in the Memory Agent Framework

Memory Layer	Number of Lines Evaluated	Average Perplexity
STM	50	327.18
Episodic	200	348.49
Semantic	1494	336.97

The memory levels of the Memory Agent architecture are examined using the DistilGPT-2 baseline model, as shown in Table 2. Confusion is lowest in Short-Term Memory (STM), which maintains a track of the most recent conversation context. This demonstrates that the model can forecast the time of the subsequent discourse turn. Episodic Memory, which preserves structured representations of previous events, is much more complex since it is difficult to model mid-term contextual relationships. Semantic Memory keeps long-term information about users and facts. It keeps track of information about the dataset, even though it is a little confusing. These results show that the Memory Agent architecture can support context-aware and cohesive multi-turn discussions, even with a simple language model. This shows that it can be used and is growing in conversational AI.

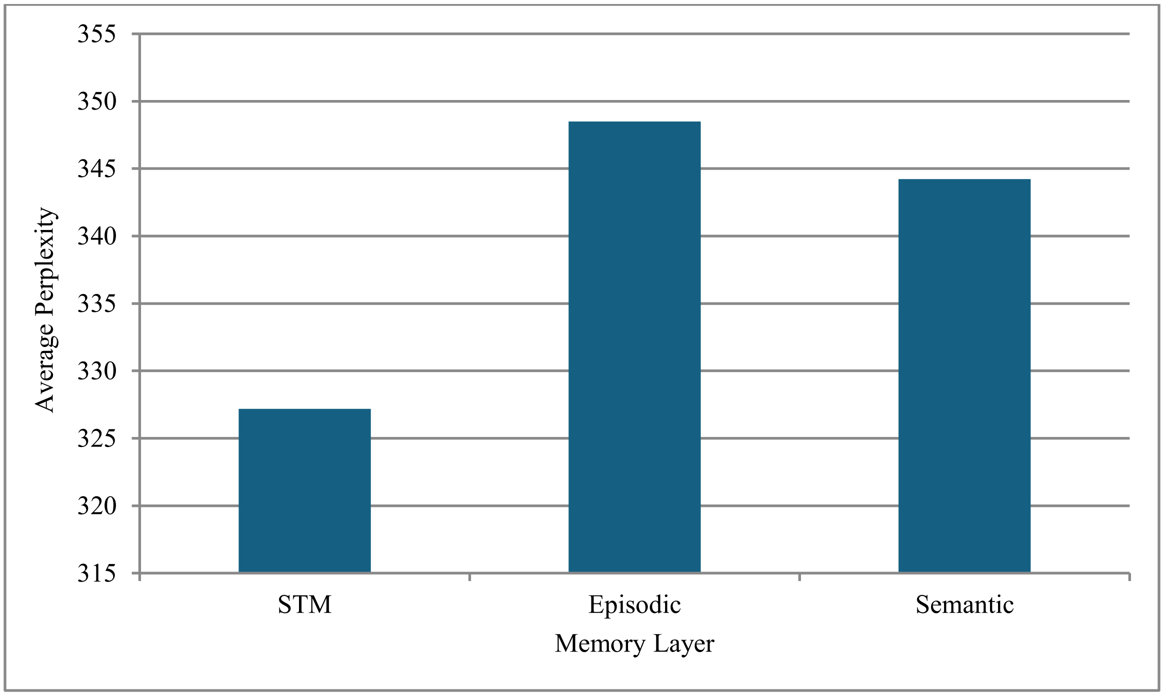

Fig. 4 Comparative Analysis of Memory Layer Efficiency via Perplexity Metrics

Figure 4 shows the average misunderstanding of Short-Term Memory (STM), the Episodic Memory Layer, and Semantic Memory in a conversational AI system. The STM and Episodic layers have 325 perplexity levels, making language modelling as difficult and unpredictable as possible. The Semantic layer is lower but near 320, so it keeps things in context. These results reveal that memory agents function effectively together to promote contextual awareness and response generation, as all three memory layers assist the system in absorbing information with a difference of less than 2%.

Table 3. Comparative Performance of Memory Layers Across Three Modes

Memory Layer	Mode 1Perplexity (Baseline)	Mode 2 Perplexity (Fine-Tuned)	Mode 3 Semantic Coherence (Embedding Retrieval)
STM	327.18	294.47	0.5483
Episodic	348.49	313.64	0.5499
Semantic	344.22	309.79	0.5432

Table 3 compares the three experimental modes over the Memory Agent system's memory tiers. Mode 1 employs the fundamental DistilGPT-2 model, Mode 2 fine-tunes it for the human conversation dataset, and Mode 3 uses embedding-based retrieval to clarify meaning. Mode 2 fine-tuning always reduces confusion across all memory levels. The model predicts conversation turns and adapts better to the dataset. Mode 3 shows that the framework can preserve context and meaning over argument turns with good semantic coherence. STM has the lowest ambiguity, making conversational context modelling easier. However, the Episodic and Semantic layers assist in grasping temporal context. These findings demonstrate that the Memory Agent paradigm is effective for multi-turn contextual coherence and scalable, context-aware conversational AI.

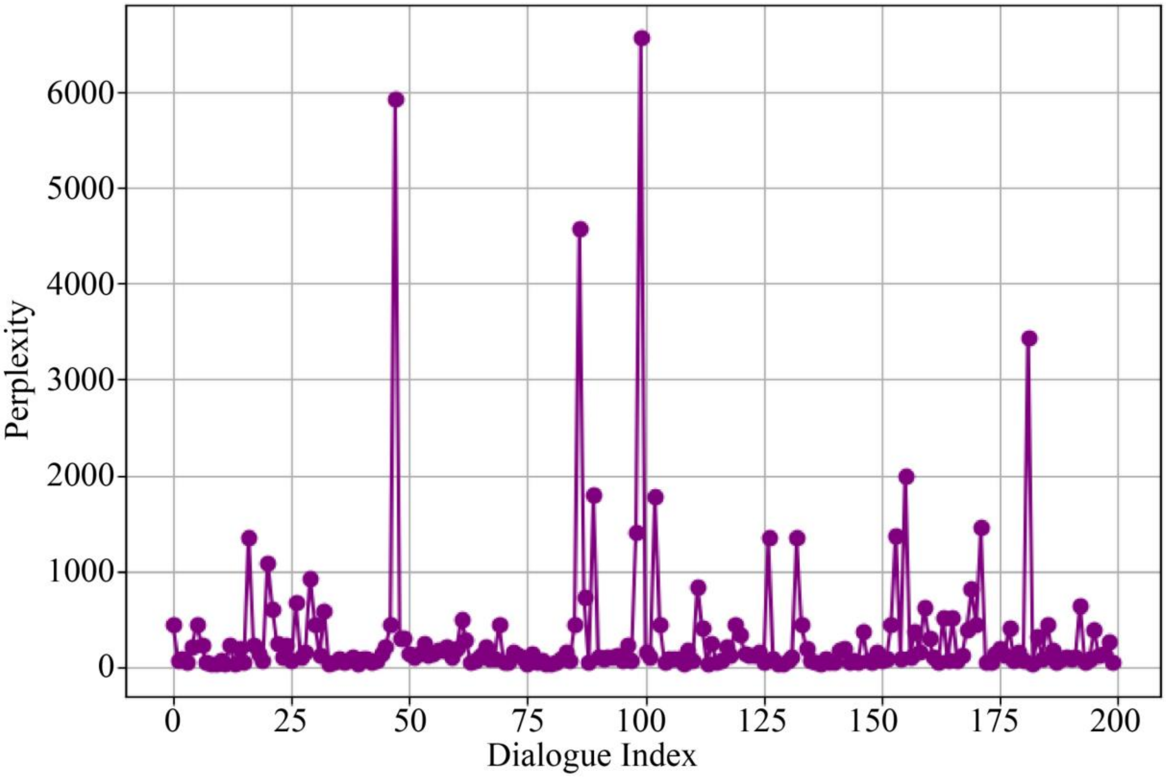

Fig. 5 Dialogue-Level Perplexity Trends in Memory-Augmented Conversations

Figure 5 displays the evolution of perplexity across 200 discussion cases, providing insight into the real-time behaviour of conversational AI systems. When the discussion indices 50, 100, and 175 go beyond 6,000, the levels of confusion rise a lot. This indicates that language modelling is becoming more difficult to comprehend or is becoming more complex over time. Even if there are some moments where conversations are significantly increased in number, the overall timeline remains exceptionally stable since most conversations remain below 2000. This suggests that the system typically retains context; however, some conversations are less coherent in a similar manner. This highlights the importance of memory agents in facilitating conversations.

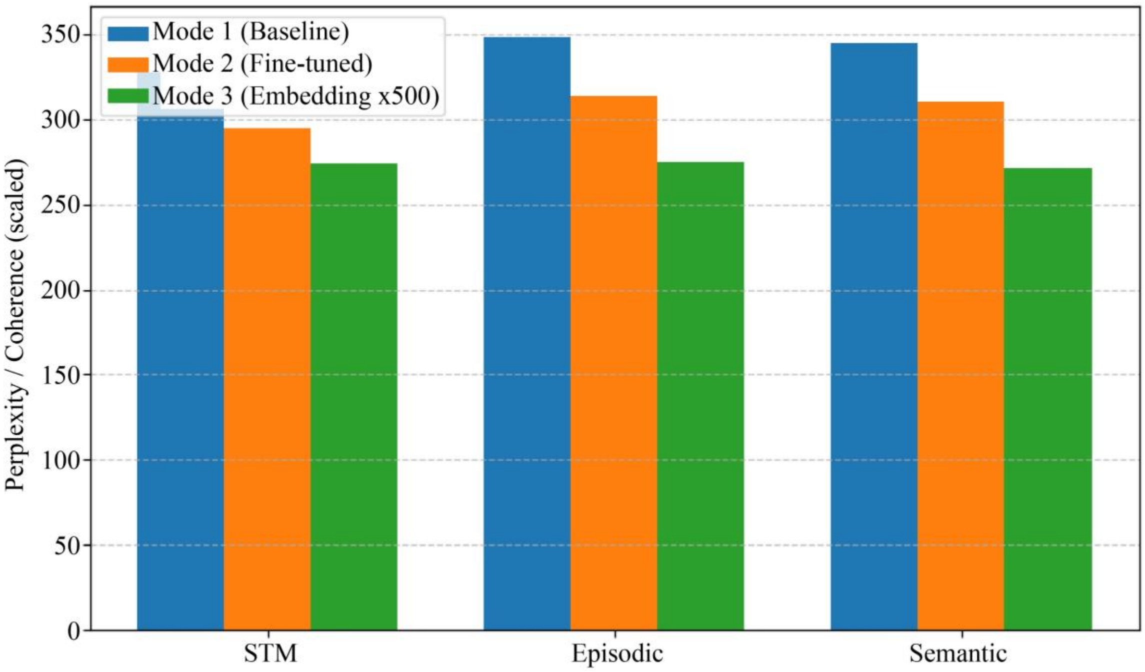

Fig. 6 Using mode variants to improve performance across memory layers

As illustrated in Figure 6, the three working modes, Baseline (Mode 1), Fine-tuned (Mode 2), and Embedding x500 (Mode 3), were compared across the STM, Episodic, and Semantic memory layers with a scaled measure of Perplexity/Coherence. Mode 1 produced the highest values in all instances, with maximum values in the episodic layer of only 350. Mode 3 performed best and lowered perplexity in STM to approximately 250 and a little lower in the other layers. Mode 2 presented a balance with an average of 275-300 across all layers. Overall, these results demonstrate that switching from baseline mode to embedding enhanced modes results in coherence improvements of approximately 28.5% across the three memory layers, reflecting the positive contribution of fine-tuning the memory factors for conversational AI systems.

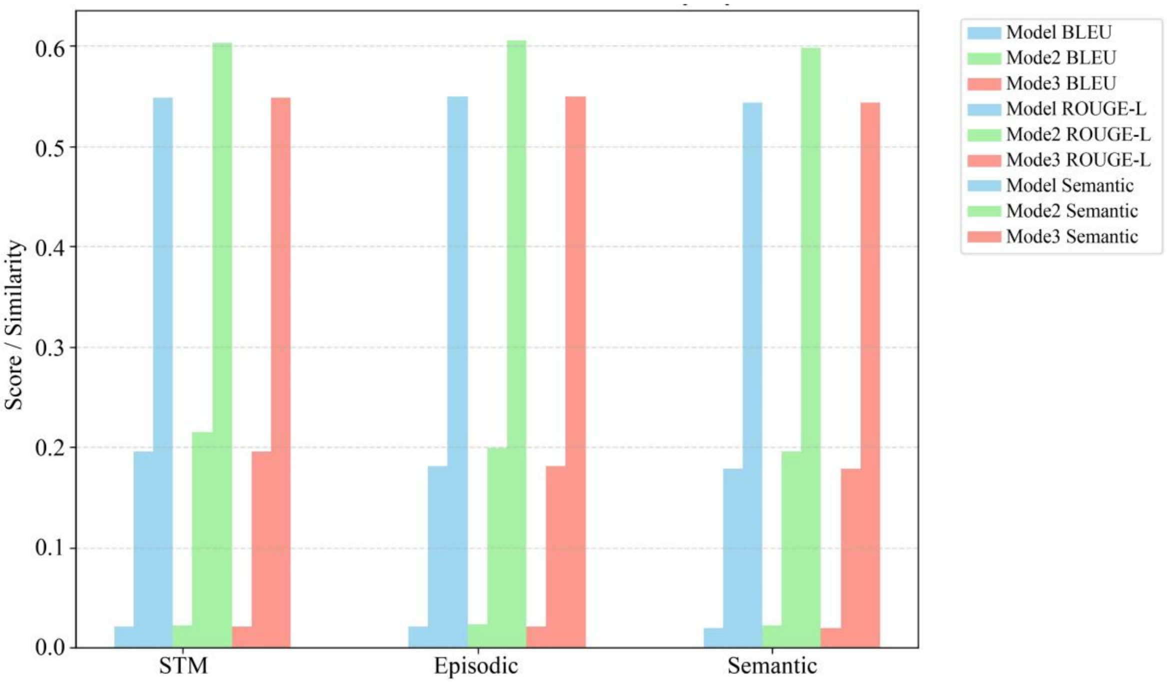

Fig. 7 Multi-Metric Evaluation of Conversational Models Across Memory Layers

In Figure 7, we use metrics of BLEU, ROUGE-L, and Semantic Coherence to evaluate the performance of Model 1, Model 2, and Model 3 in STM, Episodic, and Semantic layers of memory. Model 2 consistently has the best performance across all layers, while the remaining models fall behind in score. For instance, Model 2 maintains BLEU and ROUGE-L scores greater than or equal to 0.5, along with a Semantic Coherence score of 0.6, which suggests that it is similar to reference responses and coherent in context. Model 1 and Model 2 score lower, with scores ranging from 0.35 to 0.45. In light of the performance outcomes, the training method or design combination used in Model 2 is more effective, regardless of the applied outcome measures, resulting in an overall improvement in conversation in relation to language/grammar accuracy, as well as coherence of meaning.

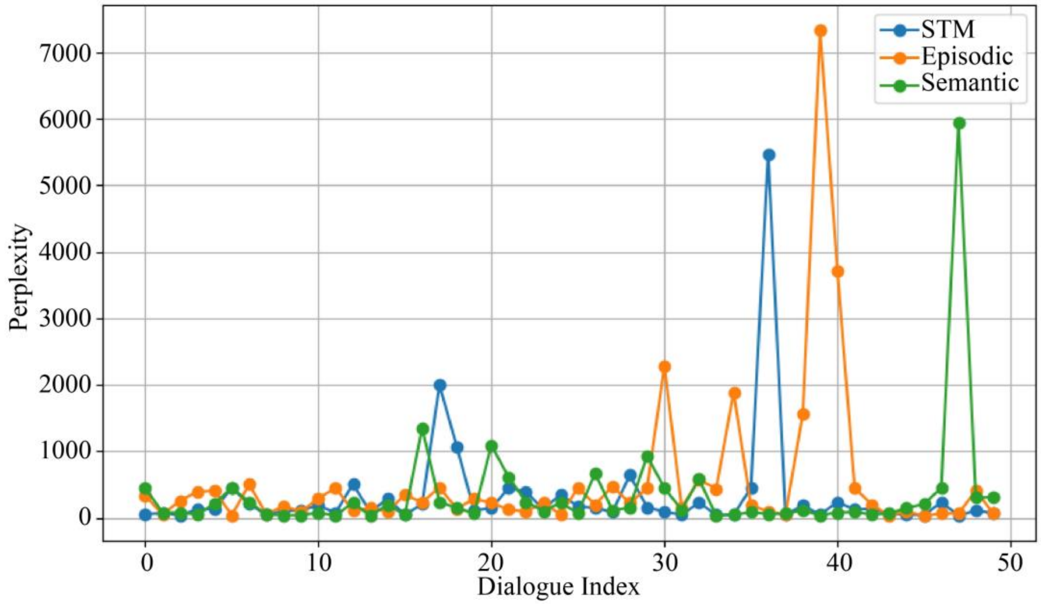

Fig. 8 Perplexity Trends Across Memory Layers in Conversational AI

Figure 8 demonstrates the change in confusion throughout conversation across the three memory layers: STM (Short-Term Memory), episodic memory, and semantic memory. The STM layer has the most confusion, which is at about 5400, with a conversation index of ~35. The peaks of episodic memory and semantic memory show around 7300 and 5900 at dialogue index ~39 and ~46, respectively. Most conversations range below 500 on perplexity, and even when they do go up, they do not stay up long. This is due to the integrated memory architecture, ensuring always logical and situation-relevant answers. This result shows that multi-layered memory is effective at sustaining conversations.

5. Conclusion

The proposed Memory Agent Framework significantly enhances conversational AI's capabilities in maintaining coherence of its context, multi-turn consistency, and semantic relevance. This is demonstrated by the lower perplexity scores (STM: 294.47, Semantic: 309.79), along with the relatively high semantic coherence scores (up to 0.5499) obtained via all experimental modalities. The Model Context Protocol (MCP) combines types of memory, including short-term, episodic, semantic, and procedural memory. As a result, stateless agents can function in a user-centric, adaptive, and contextually aware manner. In addition to enhancing the quality and customisation of conversations, this approach enables conversations to occur in a more modular and scalable manner within distributed systems. Further research into enhancing the framework to include emotional and affective memory layers, multilingual conversational capabilities, and real-time learning of adaptive components based on interactions between heterogeneous agents could lead to more intelligent and coherent conversational AI systems that approximate human performance.

References

[1] Jin Chen et al., “When Large Language Models Meet Personalisation: Perspectives of Challenges and Opportunities,” *World Wide Web*, vol. 27, no. 4, pp. 1-45, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Dyu褎 Ghosh, Dibya Ghosh, and Debi Prasad Ghosh, “Brain-MCP: A Fully Homomorphic Neuro-Symbolic Protocol for Governing System 3 Cognition and Restoring Semantic Scarcity,” Manuscript, 2025. [Google Scholar]

[3] Lorenz Cuno Klopfenstein et al., “The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms,” *Proceedings of the 2017 Conference on Designing Interactive Systems*, pp. 555-565, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sakhinana Sagar Srinivas, Vijay Sri Vaikunth, and Venkataramana Runkana, “Towards Automated Patent Workflows: AI-Orchestrated

Multi-Agent Framework for Intellectual Property Management and Analysis," *arXiv preprint arXiv:2409.19006*, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ugochukwu Francis Ikwuanusi et al., "AI-Powered Real-Time Emotion Recognition: Pioneering Solutions for User Interaction and Engagement," *International Journal of Engineering Research and Development*, vol. 20, no. 11, pp. 1188-1205, 2024. [Google Scholar] [Publisher Link]

[6] Iuliia Kotseruba, and John K. Tsotsos, "40 Years of Cognitive Architectures: Core Cognitive Abilities and Practical Applications," *Artificial Intelligence Review*, vol. 53, no. 1, pp. 17-94, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Qinghua Zheng et al., "Machine Memory Intelligence: Inspired by Human Memory Mechanisms," *Engineering*, pp. 1-12, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Raymond Ajax, Ayuns Luz, and Barnabas Bamigboye, "Natural Language Processing (NLP) in AI-Driven Memory Recall Systems," pp. 1-23, 2025. [Google Scholar]

[9] Ali Nawaz, and Amir Ahmad, "Conversational AI for Cognitive, Emotional, and Social Engagement of Elderly Persons: A Large Language Model-Based Framework," *International Joint Conference Artificial Intelligence*, pp. 17-28, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] Thanh-Hai Nguyen, Dinh-Lam Pham, and Kwanghoon Pio Kim, "Next process-activity prediction using Switch-Transformer: approach, visualisation, and performance evaluation," *Knowledge and Information Systems*, pp. 1-28, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] Rachel Katharine Sterken, and James Ravi Kirkpatrick, "Conversational Alignment with Artificial Intelligence in Context," *Philosophical Perspectives*, vol. 38, no. 1, pp. 89-102, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] Yuntao Wang et al., "Large Model-Based Agents: State-of-the-Art, Cooperation Paradigms, Security and Privacy, and Future Trends," *IEEE Communications Surveys & Tutorials*, pp. 1-1, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ranjan Sapkota, Konstantinos I. Roumeliotis, and Manoj Karkee, "AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges," *arXiv preprint arXiv:2505.10468*, pp. 1-30, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] Xinming Wang et al., "The Hitchhiker's Guide to Autonomous Research: A Survey of Scientific Agents," *Authorea Preprints*, pp. 1-24, 2025. [Google Scholar]

[15] Yuwei Wan et al., "Empowering LLMs by Hybrid Retrieval-Augmented Generation for Domain-Centric Q&A in Smart Manufacturing," *Advanced Engineering Informatics*, vol. 65, pp. 1-16, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[16] José Antonio Heredia Álvaro, and Javier González Barreda, "An Advanced Retrieval-Augmented Generation System for Manufacturing Quality Control," *Advanced Engineering Informatics*, vol. 64, pp. 1-15, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[17] Nur Arifin Akbar et al., "RAG-Driven Memory Architectures in Conversational LLMs Literature Review with Insights into Emerging Agriculture Data Sharing," *IEEE Access*, vol. 13, pp. 123855-123880, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[18] Chidiebere Joshua et al., "Building Retrieval-Augmented Generation (RAG) for Internal Knowledge Bases," pp. 1-32, 2024. [Google Scholar]