

A Grounded Memory System For Smart Personal Assistants

Felix Ocker^{1,*}, Jörg Deigmöller¹, Pavel Smirnov¹ and Julian Eggert¹

¹Honda Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach am Main, Germany

Abstract

A wide variety of agentic AI applications – ranging from cognitive assistants for dementia patients to robotics – demand a robust memory system grounded in reality. In this paper, we propose such a memory system consisting of three components. First, we combine Vision Language Models for image captioning and entity disambiguation with Large Language Models for consistent information extraction during perception. Second, the extracted information is represented in a memory consisting of a knowledge graph enhanced by vector embeddings to efficiently manage relational information. Third, we combine semantic search and graph query generation for question answering via Retrieval Augmented Generation. We illustrate the system’s working and potential using a real-world example.

Keywords

Memory System, Ontology Construction, Retrieval Augmented Generation, GraphRAG, Grounding

1. Introduction

The emergence of Large Language Models (LLMs) has advanced conversational assistants beyond rule-based systems, enabling them to operate within a user’s perceptual and conceptual context. For this, assistants must integrate stored knowledge with ongoing interactions to ensure that responses remain relevant and grounded. Retrieval Augmented Generation (RAG) techniques combine LLMs with external knowledge bases and multimodal LLMs process diverse inputs, enabling richer, more context-aware interactions. However, the need for assistants with personal and situational support based on a large-scale memory also highlights critical challenges. First, to effectively deal with memories in the form of multimodal inputs, a robust conceptual framework is needed that acknowledges the role of space and time as fundamental dimensions of experience and memory. Inspired by Kantian notions, which describe space and time as fundamental structuring elements imposed by the mind [1], we recognize that memory systems must integrate these dimensions to maintain coherent, grounded knowledge. Second, standard RAG stores information as disconnected snippets, failing to capture the relational dependencies needed for complex queries [2]. Third, true situational awareness requires structured, concept-based retrieval and inference for more advanced reasoning and decision-making. To address these challenges, we propose a novel approach for grounded memory-based personal assistants. Our approach builds on a structured memory akin to human episodic and biographic memory, ensuring that information is pre-structured before inference rather than relying on on-the-fly conceptualization like standard RAG. Each of these components addresses a specific challenge, resulting in three pillars:

1. *Grounded Perception*: Structure multimodal inputs with spatial and temporal awareness, categorizing them into actions, agents, and objects.
2. *Memory Graph*: Overcome standard RAG limitations by using a richer knowledge representation in the form of an ontological framework for representing memories, i.e., structuring interconnected concepts and enhancing memory versatility through semantic embeddings.
3. *Agentic Retrieval*: Use graph querying and expansion together with semantic search for improving coherence and context-awareness for complex queries.

By combining these elements, our system enables assistants to deliver personalized, context-aware support with enhanced reasoning and decision-making.

ESWC’25: Workshop on LLM-Integrated Knowledge Graph Generation from Text (TEXT2KG), June 01–05, 2025, Portoroz, Slovenia

*Corresponding author.

✉ felix.ocker@honda-ri.de (F. Ocker)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Related Work

This section reviews related work for the three pillars of our memory system: grounded perception, memory graphs, and agentic retrieval. Grounded perception organizes multimodal data into actions, agents, and objects with temporal awareness, forming structured action patterns [3]. Memory graphs overcome standard RAG limitations by structuring knowledge to capture even implicit relations. Agentic retrieval enhances reasoning via graph-based inference instead of relying solely on embedding similarity.

2.1. Grounded Perception

Multimodal perception has advanced with LLMs, aiding AI applications like robotics and surveillance. For instance, robots critically depend on their visual understanding capabilities for navigation [4] and object localization tasks [5, 6, 7]. Recent work such as 3D Dynamic Scene Graphs (DSGs) [6] and TASKOGRAPHY [8] rely on creating structured models of the environment. However, perception in robotics usually does not focus on building a lifelong memory, but rather on creating a faithful representation of the current environment which could be recalled for specific tasks. For Embodied RAG [9], the authors build a structured semantic forest based on spatial proximity which can be used in combination with LLMs to support robotic navigation. Other examples of multimodal perception systems specializing in human activity recognition are systems for understanding long videos [10]. Such systems, e.g., VideoAgent [11] and AMEGO [12], focus on person-object annotations, primarily tracking hand-object interactions without explicit action labeling. Effective memory-based assistants need persistent representations of actions, agents, and objects with contextual tracking. Many multimodal perception systems offer contextualized understanding but lack structured long-term recall. Our approach integrates LLM-based perception with a structured graph-based memory to ensure interpretability and retrieval. With advances in LLMs, many specialized environment recognition and action detection approaches are being replaced by multimodal LLMs [13]. In the context of this paper, we rely on multimodal LLMs, specifically Vision Language Models (VLMs), for these tasks, since they generally provide more contextualized information for building a grounded memory system. While VLMs provide a flexible and context-aware understanding, they lack the structured memory needed for long-term, explainable recall by themselves. Our approach addresses this by integrating VLM-based perception into a structured, graph-based memory, ensuring that memories remain interpretable and retrievable.

2.2. Memory Graphs

A scalable memory is essential for assistants with personal support capabilities. Due to their benefits regarding the integration of heterogeneous data [14], knowledge graphs provide an excellent technological basis for such a memory [15]. For instance, TobuGraph [2] is an approach to transform pictures and conversations with a text-based chatbot into a memory graph. The authors demonstrate the limitations of the standard RAG approach for describing personal memories. In [16], the authors describe MemPal - a wearable video-based conversation device for assisting elderly with memory impairments. MemPal focuses on a use-case of finding lost objects and is evaluated on the effects of voice-enabled multimodal LLMs. These systems address two deficiencies of standard RAG approaches: 1) The problem of scaling them to large-scale multimodal real-world scenarios and 2) the deficiencies in terms of representing complex memories of interconnected world entities. To address the second deficiency, the authors of [17] describe a framework for capturing lifelong personal memories from images and videos by memorizing them via a natural language interface. The approach includes extracting a taxonomy of contextual information out of textual information obtained from videos and images, with contexts being described by time, location, people, visual elements of environment, activities and emotions. The extracted taxonomy is used for a special retrieval which augments semantic search. While this demonstrates that RAG-based approaches can be used to retrieve snippets of personal experiences, it lacks the power of relational memories as provided by underlying memory graphs. In this paper, we rely on a combination of RAG techniques with knowledge graphs for improving the retrieval capabilities of such systems.

2.3. Agentic Retrieval

GraphRAG is a retrieval-augmented generation that enriches conventional RAG pipelines with a graph-based representation of knowledge. In standard RAG systems, semantic search is used to retrieve relevant text snippets from a vector store, which are provided as context to an LLM for question answering. However, this chunk-oriented retrieval can miss deeper relationships and dependencies among pieces of information. GraphRAG addresses this limitation by building and utilizing knowledge structured in graphs, enabling more coherent reasoning over interconnected facts. One way of realizing GraphRAG is to conduct semantic search to find entry points in the graph and then expand the context for further relevant, but less explicit, information. There are several suitable algorithms for graph expansion, PageRank being one of them [18]. Another approach to GraphRAG is to translate natural language queries into graph queries, such as Cypher, for structured database access [19]. Another GraphRAG application is presented in [20], where a knowledge graph is built from textual data. Instead of retrieving isolated text snippets, the system retrieves relational subgraphs relevant to a user query, which are then passed to an LLM. By leveraging both textual and structured graph-based knowledge, this approach enables deeper reasoning over complex, interconnected facts, making it highly effective for answering intricate queries. By leveraging a combination of these techniques, our system ensures more explainable and context-aware responses, combining the flexibility of text-based search with the expressiveness of a graph-based memory.

3. Grounded Memory System Architecture

The memory system is based on a **schema** that revolves around textual notes, which are represented as nodes in a graph, cp. Section 3.1. Leveraging this schema, the memory system is designed to seamlessly capture, structure, and retrieve real-world observations through a three-phase process, see Figure 1.

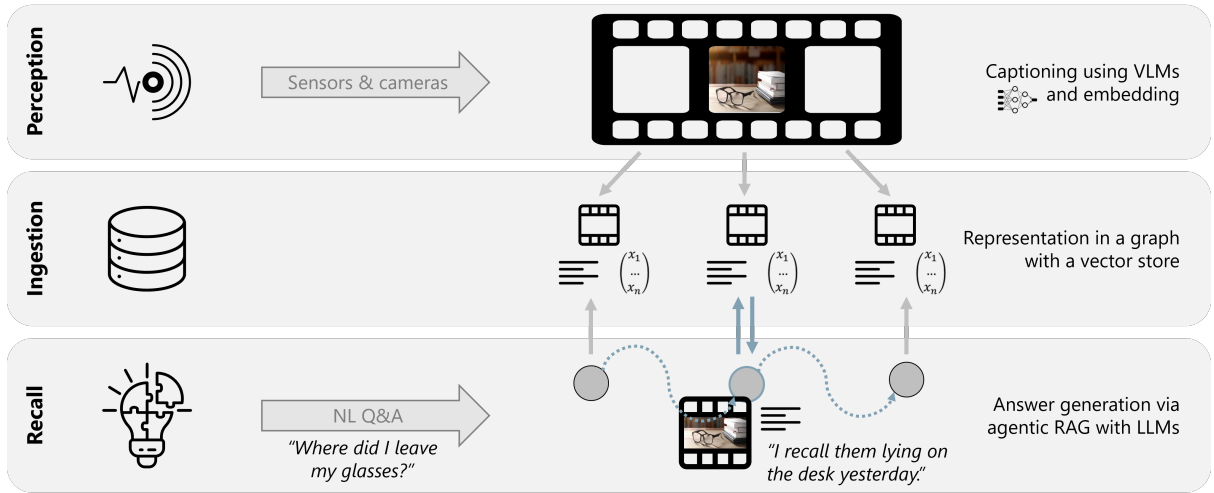


Figure 1: Memory system architecture overview.

In the **perception** phase, cp. Section 3.2, cameras observe the environment, allowing a VLM to generate descriptive captions for detected events. While this can be extended for further modalities such as audio, we focus on visual inputs in the context of this paper. During the **ingestion** phase, cp. Section 3.3, these images and captions undergo a structured analysis before being stored in a persistent knowledge graph. Unlike unstructured memory systems, this graph-based representation explicitly encodes who performed which action on which object, when, and where. Finally, in the **recall** phase, cp. Section 3.4, the system retrieves stored information for question answering, event verification, and intelligent recommendations. Throughout the following, we rely on a video showing an individual in a home setting as a running example to exemplify the concepts introduced.

3.1. Representing Memory Notes

The knowledge base builds on a schema for representing so-called memory notes, cp. Figure 2. A *MemoryNote* can be used to describe a time period of arbitrary length and it can be generated from arbitrary sources, e.g., manually crafted for diary entries or generated automatically to describe a single frame in a video. Each memory note is characterized by its note content, which is a natural language string, and an optional list of data files from which it has been created. To create a structured representation, every memory note is also represented as a node in a knowledge graph. For our application, we introduce *Image* nodes, which are specialized memory notes that have an image caption as note content and that refer to an image as a data file. To create a structured representation, each *MemoryNote* is linked to the entities it mentions, categorized as *Agents* ("Who performed the action?"), *Objects* ("What was acted upon?"), and *Actions* ("What was done?"). Images are temporally ordered using *has-previous* links, cp. Figure 2, and agents, objects, and actions are connected to the images they occur in via *has-element* links.

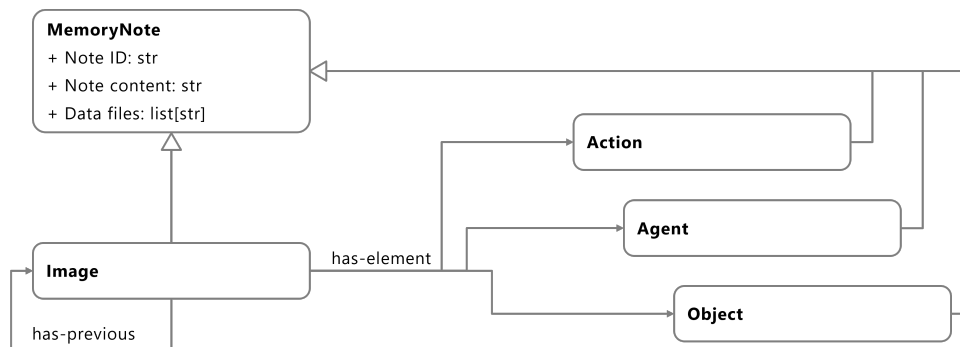





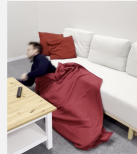

Figure 2: Schema for representing images and action patterns.

3.2. Perception

The perception phase captures raw video input images, and generates descriptive captions using gpt-4o's [21] vision capabilities, thus laying the foundation for a structured representation of events.

Role: You are proficient in creating scene descriptions. Your job is to analyze a sequence of images captured by a mobile camera carried by a person.

Task: Summarize the sequence as a whole in a single description in the style of a technical documentary. Do NOT refer to individual images in your description. Focus on changes in consecutive images. Changes include the appearance or disappearance of objects, persons, or actions. Describe all objects in front of the camera by their type and take special care to not miss any object. Use adjectives to describe persons and objects in a way that they are distinctive and well described by their appearance, state, expression, or emotion. Limit the caption to two paragraphs.

Keep the original description of the last image and add unique labels to all agents, objects, and actions mentioned using square brackets. Also include the type, i.e., "Agent", "Object" or "Action" in the label after a colon. Return an adapted caption for the last image. If you are provided with two images reuse the labels from the first image whenever applicable for the same objects, agents, and actions.

Here is an example response: A person [person_1:Agent] is pouring [pour_1:Action] a bottle [bottle_1:Object]. Another person [person_2:Agent] is holding [hold_1:Action] a glass [glass_1:Object] until the glass [glass_1:Object] is filled up half. There are two further bottles visible [bottle_2:Object], [bottle_3:Object].

Assess whether there may be an emergency situation in the image sequence. Focus on dangerous and confusing situations that indicate that someone needs help. If there is likely to be an emergency situation in the image sequence return a single sentence describing the emergency, otherwise return an empty string.

Figure 3: Three-step VLM prompt applied to an image sequence. Each processing window consists of multiple consecutive frames, analyzed together. The first and last frame overlap with adjacent windows to ensure continuity. Captions for the sequence are appended to the last frame.

Unlike traditional video memory systems that passively store raw visual inputs, our system actively identifies and links key entities in the environment. This includes detecting agents, objects, and their spatial relationships, forming a structured representation. To achieve this, the system processes sequences of n consecutive frames from the video stream using the prompt shown in Figure 3. Each sequence is analyzed as a single unit, where the first and last frame overlap with adjacent sequences, ensuring temporal continuity. To find a balance between efficiency and accuracy, we caption only each n -th frame. This strategy maintains temporal coherence, reduces redundant descriptions, and results in more accurate, context-aware scene summaries. Each described instance in the captions is indexed with a unique label in the format $[label_x:Type]$ to ensure consistent tracking across frames.

3.3. Knowledge Graph and Vector Store Population

The information captured during the perception phase, cp. Section 3.2, is stored in a hybrid knowledge base combining a knowledge graph consisting of structured relationships and a vector store, i.e., a text-oriented representation allowing for semantic search. The ingestion process consists of four steps. First, all entities identified during the perception phase are extracted from the image captions. Together with the entity names, we extract the entity types. Second, we create embedding vectors for the image captions using an embedding model, resulting in high-dimensional numeric representations of the text. If necessary due to context window limitations, the captions are split up into several parts. Third, we create nodes in the knowledge graph for all images and connect them sequentially. To these we add the respective attributes, such as the captions and the paths for the image files, and the embedding vectors for the image captions. Fourth, we add nodes for all actions, agents, and objects identified and connect them to all the image nodes in which they appear, turning the sequence of images into a connected graph. The knowledge graph, cp. Figure 4, maintains temporal order via sequentially connected image nodes, while objects, agents, and actions structure events. Consistent entity labels ensure continuity and enable context-aware retrieval.

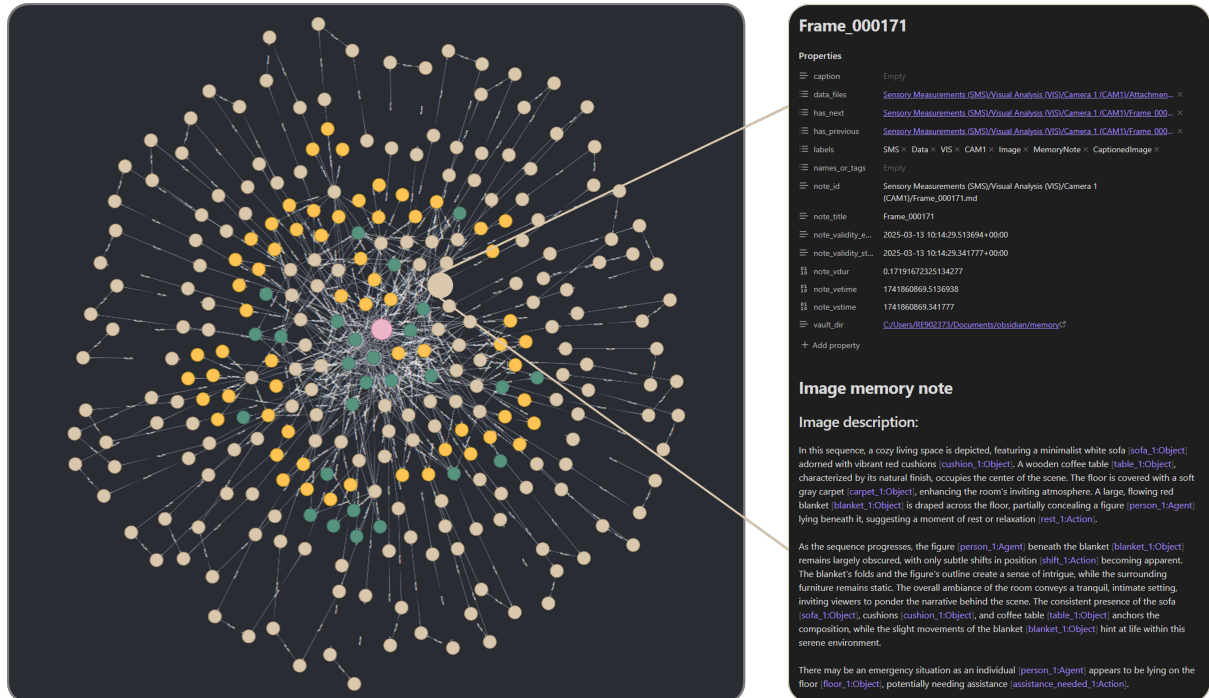


Figure 4: Knowledge graph (left) and image note (right). Sand-colored nodes represent sequential video frames, green nodes denote detected object instances, the pink node corresponds to the agent, and yellow nodes represent actions. Entities are linked across frames and via action patterns.

3.4. Agentic Retrieval for Question Answering

Combining a graph-based structured representation with natural language text notes allows the use of several retrieval techniques. First, we use the memory system as a standard RAG system for *semantic search* as it is based on natural language notes with embedding vectors associated to them. By embedding the user query with the same embedding model, we create an embedding vector which we compare to the embeddings of all the notes, retrieving the semantically most relevant notes. To increase the relevance of the context provided to the LLM, we optionally use a reranker to check the retrieved notes and filter out less relevant ones. Providing the retrieved notes as context, we let the LLM answer the original question. This type of retrieval is especially efficient for questions which are likely to have responses that are semantically close. Second, we leverage the structure of the memory system for *graph expansion*. Relevant information may be included in notes which are not found when relying purely on semantic search, but that are linked to the notes found in the graph. This is usually the case for relevant background information, e.g., personal preferences not showing up in individual notes, but represented in a note for an agent. Here, we start with regular semantic search for identifying an initial set of relevant notes. Then we expand the search results using an expansion algorithm. Specifically, we use PageRank [22], but other algorithms, e.g., random walks, also work. Finally, the expanded search results are used as context for LLM-based question answering. This second type of retrieval is beneficial whenever there is implicit background information, which cannot easily be found via semantic search, but which becomes apparent when analyzing the surrounding environment of the relevant nodes in the graph. Third, we rely on the graph representation for structured information retrieval in the form of *text2cypher*. For this, we let the LLM translate the user input into a Cypher query, which is run against the graph database. The result in the form of a table is interpreted by the LLM, which formulates a natural language response. This type of retrieval is ideal for answering structure-oriented queries, for instance questions that require counting entities. To leverage the benefits of all three retrieval techniques, we combine them in an agentic retrieval system. For this, we wrap the three retrieval functionalities into tools that we provide to an LLM-based agent, who can access them as needed for answering user questions. The agent is prompted to select the most suitable tool, or several if necessary, to retrieve information from the memory system and eventually formulate an appropriate answer.

4. Implementation Details

We implemented a prototype using Neo4j as the graph database and vector store. Videos were recorded using an iPhone 15, split into frames at 3 Hz, and captions were generated for every fifth frame using gpt-4o for vision and text-embedding-3-small for storing embeddings in the vector store. New entities (images, actions, agents, objects) are inserted into the graph using dynamically generated Cypher queries. Memory notes are also stored in Obsidian, enabling easy browsing. A minimalist gradio frontend allows users to interact with the system via a chat interface, cp. Figure 5, that provides links to the sources used for transparency. Table 1 presents representative questions and system responses, demonstrating retrieval capabilities across different query types.

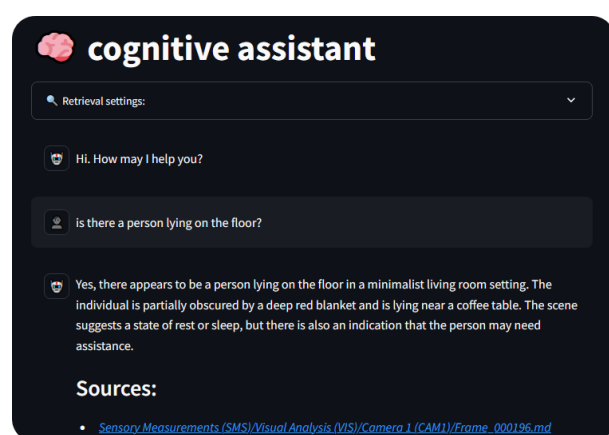


Figure 5: Chat frontend for interacting with the memory system.

Category	Question	System answer
Counting in frame	How many images are there on the wall?	There are five photographs pinned to the wall, adding a personal touch to the minimalist living space.
Counting across frames	How many people are there?	There is 1 person present.
	How many images are there in memory?	There are 329 images in memory.
Existence	Is there a person lying on the floor?	Yes, there is a person lying on the floor.
	Is there something dangerous happening?	An individual appears to be lying on the floor, potentially needing assistance or indicating a medical emergency.
Emotional	What is the person's emotional state?	The person's emotional state appears to be relaxed and contemplative, as they are depicted reclining on a sofa, enveloped in a vibrant red blanket.
Descriptive	Please describe the setting of the scene.	The setting of the scene is a minimalist living space characterized by a light-colored wall and a soft gray carpet. Central to the scene is a modern white sofa adorned with vibrant red accent pillows, creating a cozy and inviting atmosphere. A wooden coffee table with a natural finish is positioned in front of the sofa, and it features a remote control, suggesting a casual and comfortable environment.
	What is the man wearing?	The man is wearing a dark navy sweater and gray pants. He is also depicted with short, neatly styled hair and glasses, which add to his studious demeanor.

Table 1

Representative examples for questions and system answers (all of which are correct).

5. Summary and Outlook

This paper presents a grounded memory system that integrates the strengths of a knowledge graph and a vector store for agentic RAG with an LLM as an intuitive natural language interface. The system leverages a minimalist schema and operates through three key phases: perception, ingestion, and retrieval. The system has potential applications ranging from robotics to assistive technologies, such as support systems for dementia patients. Our approach provides a foundation for structured memory-based retrieval and serves as a starting point for future research in long-term knowledge representation and context-aware reasoning. The integration of conceptual nodes provides additional flexibility, allowing retrieval to be guided by semantic relationships rather than purely temporal order. This structured approach enables conversational assistants to reason over past events, improving long-term memory consistency compared to standard RAG techniques.

Future work should focus on scaling up the system and conducting large-scale evaluations in real-world scenarios. Expanding to longer multimodal sequences will allow the system to capture broader temporal dependencies and leverage its effectiveness in retrieving and reasoning over complex event histories. While we expect challenges in long-term entity disambiguation – ensuring that agents, objects, and actions are consistently recognized across different scenes and timeframes – moving beyond individual observations enables high-level behavior pattern identification by recognizing repetitive actions, activity trends, and structured sequences of human interactions. Additionally, we will further advance RAG techniques, such as recursive summarization and query rewriting, to enhance contextual understanding and improve response accuracy. This can be supported by advancing agency in the retriever and providing further retrieval tools, e.g., via frameworks focusing on leveraging large sets of tools for LLMs [23]. Further improvements will also involve extending the system's multimodal capabilities beyond vision, incorporating audio and spatial information. In doing so, we aim to move toward a comprehensive memory system capable of supporting autonomous agents in complex environments.

Declaration on Generative AI

During the preparation of this work, the authors used generative AI for grammar and spelling checks. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] I. Kant, Critique of pure reason. 1781, Modern Classical Philosophers, Cambridge, MA: Houghton Mifflin (1908) 370–456.
- [2] S. Kashmira, J. L. Dantanarayana, J. Brodsky, A. Mahendra, Y. Kang, K. Flautner, L. Tang, J. Mars, A graph-based approach for conversational AI-driven personal memory capture and retrieval in a real-world application, arXiv:2412.05447 (2024).
- [3] J. Eggert, J. Deigmöller, L. Fischer, A. Richter, Action representation for intelligent agents using Memory Nets, in: IC3K, 2020.
- [4] B. Al-Tawil, T. Hempel, A. Abdelrahman, A. Al-Hamadi, A review of visual slam for robotics: Evolution, properties, and future applications, Frontiers in Robotics and AI 11 (2024) 1347985.
- [5] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, S. Savarese, 3D scene graph: A structure for unified semantics, 3D space, and camera, in: ICCV, 2019.
- [6] U.-H. Kim, J.-M. Park, T.-J. Song, J.-H. Kim, 3D scene graph: A sparse and semantic representation of physical environments for intelligent agents, IEEE transactions on cybernetics 50 (2019) 4921–4933.
- [7] A. Rosinol, A. Gupta, M. Abate, J. Shi, L. Carlone, 3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans, in: RSS, 2020.
- [8] C. Agia, K. M. Jatavallabhula, M. Khodeir, O. Miksik, V. Vineet, M. Mukadam, L. Paull, F. Shkurti, Taskography: Evaluating robot task planning over large 3D scene graphs, in: CoRL, 2022.
- [9] Q. Xie, S. Y. Min, T. Zhang, K. Xu, A. Bajaj, R. Salakhutdinov, M. Johnson-Roberson, Y. Bisk, Embodied-RAG: General non-parametric embodied memory for retrieval and generation, arXiv:2409.18313 (2024).
- [10] Y. Wang, Y. Yang, M. Ren, LifelongMemory: Leveraging LLMs for answering queries in long-form egocentric videos, arXiv:2312.05269 (2023).
- [11] Y. Fan, X. Ma, R. Wu, Y. Du, J. Li, Z. Gao, Q. Li, VideoAgent: A memory-augmented multimodal agent for video understanding, in: ECCV, 2024.
- [12] G. Goletto, T. Nagarajan, G. Averta, D. Damen, Amego: Active memory from long egocentric videos, in: ECCV, 2024.
- [13] Y. Li, Z. Lai, W. Bao, Z. Tan, A. Dao, K. Sui, J. Shen, D. Liu, H. Liu, Y. Kong, Visual large language models for generalized and specialized applications, arXiv:2501.02765 (2025).
- [14] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gao, R. Navigli, S. Neumaier, et al., Knowledge graphs, ACM Computing Surveys 54 (2021) 1–37.
- [15] J. Eggert, F. Ocker, Graph based memory extension for large language models, 2025. US Patent App. 18/898,607.
- [16] N. Maniar, S. W. Chan, W. Zulfikar, S. Ren, C. Xu, P. Maes, MemPal: Leveraging multimodal AI and LLMs for voice-activated object retrieval in homes of older adults, in: IUI, 2025.
- [17] J. N. Li, Z. J. Zhang, J. Ma, Omniquery: Contextually augmenting captured multimodal memory to enable personal question answering, arXiv:2409.08250 (2024).
- [18] B. J. Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, Y. Su, HippoRAG: Neurobiologically inspired long-term memory for large language models, in: NeurIPS, 2024.
- [19] M. G. Ozsoy, L. Messallem, J. Besga, G. Minneci, Text2cypher: Bridging natural language and graph databases, arXiv:2412.10064 (2024).
- [20] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, J. Larson, From local to global: A graph RAG approach to query-focused summarization, arXiv:2404.16130 (2024).

- [21] J. Achiam, et al., Gpt-4 technical report, arXiv:2303.08774 (2024).
- [22] S. Bin, K. L. Page, The anatomy of a large-scale hypertextual web search engine, in: Computer Networks, 1998.
- [23] F. Ocker, D. Tanneberg, J. Eggert, M. Gienger, Tulip agent—enabling LLM-based agents to solve tasks using large tool libraries, arXiv preprint arXiv:2407.21778 (2024).