Embodied-RAG: General Non-parametric Embodied Memory for Retrieval and Generation

Quanting Xie[1,*], **So Yeon Min**[1,*], **Tianyi Zhang**[1], **Kedi Xu**[1], **Aarav Bajaj**[1], **Ruslan Salakhutdinov**[1], **Matthew Johnson-Roberson**[1], **Yonatan Bisk**[1]
[1]Carnegie Mellon University, Pittsburgh, PA 15213, USA
quantinx@andrew.cmu.edu, soyeonm@andrew.cmu.edu

Abstract

There is no limit to how much a robot might explore and learn, but all of that knowledge needs to be searchable and actionable. Within language research, retrieval augmented generation (RAG) has become the workhouse of large-scale non-parametric knowledge, however existing techniques do not directly transfer to the embodied domain, which is multimodal, data is highly correlated, and perception requires abstraction. To address these challenges, we introduce Embodied-RAG, a framework that enhances the foundational model of an embodied agent with a non-parametric memory system capable of autonomously constructing hierarchical knowledge for both navigation and language generation. Embodied-RAG handles a full range of spatial and semantic resolutions across diverse environments and query types, whether for a specific object or a holistic description of ambiance. At its core, Embodied-RAG's memory is structured as a semantic forest, storing language descriptions at varying levels of detail. This hierarchical organization allows the system to efficiently generate context-sensitive outputs across different robotic platforms. We demonstrate that Embodied-RAG effectively bridges RAG to the robotics domain, successfully handling over 200 explanation and navigation queries across 19 environments, highlighting its promise for general-purpose non-parametric system for embodied agents.

1 Introduction

Humans excel as generalist embodied agents in part due to our ability to build, abstract, and reason over rich memories. In contrast, current embodied agents[Chaplot et al., 2020, Khanna et al., 2024, Krantz et al., 2022, Zhou et al., 2023] lack such versatile memory capabilities, limiting their ability to operate effectively in unbounded and complex real-world environments. While existing methods such as semantic mapping[Chaplot et al., 2020, Khanna et al., 2024] and scene graphs[Li et al., 2022, Rana et al., 2023] attempt to capture spatial and contextual relationships, they largely fall short of the dynamic and flexible memory, retrieval, and generative abilities exhibited by humans.

In the language domain, foundation models combined with non-parametric memory mechanisms have achieved near human-level performance across various tasks. Retrieval-Augmented Generation (RAG) [Asai et al., 2023, Chen et al., 2023, Lewis et al., 2021] has been widely adopted in the field of Natural Language Processing (NLP) as a non-parametric memory mechanism over large document corpora, enhancing the accuracy and relevance of responses generated by Large Language Models (LLMs). Similarly, the continuous stream of experiences gathered by embodied agents forms vast databases that exceed the context window limitations of LLMs.

* Equal contribution.

Preprint. Under review.

However, applying RAG to embodied scenarios presents unique challenges due to key differences between textual data and embodied experiences. First, while RAG relies on existing documents, building memory from embodied experiences is itself a core research challenge. Current methods, such as dense point clouds or scene graphs, fail to capture the full range of experiences beyond object-level attributes, without relying on human-engineered schemas or exceeding memory budgets. Second, unlike documents, embodied experiences have inherent correlated structure — semantically similar objects are often spatially correlated and hierarchically organized so embodied experiences should not be treated as independent samples. Finally, embodied observations vary in granularity and structure: outdoor scenes might be sparse, while indoor environments are cluttered, and repeated objects across frames can confuse LLMs, complicating retrieval.

We present **Embodied-RAG**, a system with two key components: Memory Construction and Retrieval and Generation. In Memory Construction, Embodied-RAG autonomously builds a topological map and a hierarchical semantic forest that organizes observations based on spatial correlations. This forest allows retrieval at different abstraction levels (explicit, implicit, global) by matching the query with corresponding memory resolution (local, intermediate, global). To mitigate perceptual hallucinations, the Retrieval and Generation process incorporates parallel tree traversals scored by a language model, using retrieved results for explanations or navigational actions via an LLM.

We evaluated Embodied-RAG, with a novel benchmark with over 200 tasks requiring multimodal outputs and reasoning. Compared to Semantic Match and vanilla RAG, Embodied-RAG demonstrated superior performance: (1) more robust against object detection errors on explicit queries, leveraging spatial relevancy; (2) improved reasoning on implicit queries, with a 220% improvement over Semantic Match and 30% over RAG; (3) better global summarization and trend analysis, where Semantic Match and RAG showed poor results.

The key contributions and implications of this paper include:

- **Method** We introduce the system of Embodied-RAG. This method addresses problems of naively apply non-parametric memories like RAG to embodied setting.
- **Task** We introduce the general task of *Embodied-RAG benchmark*, formulating semantic navigation and question answering under a single paradigm (Figure 2).
- **Implications** Our results and discussion provide a basis for rethinking approaches to generalist robot agents based on non-parameteric memories.

2 Task

We introduce the Embodied-RAG benchmark, which contains queries from the cross-product of {explicit, implicit, global} questions with potential {navigational action, language} generation outputs.

A task consists of:

- **Query**: The content can be explicit (e.g. a particular object instance), implicit (e.g. looking for adequacy, instruction with more pragmatic understanding required), or global. The request might pertain to a location or general vibe.
- **Experience**: The experience is a sequence of egocentric visual perception and odometry, occurring in indoor, outdoor, or mixed environments.
- **Output**: The expected output can be both navigation actions with language descriptions (Fig 2 top, Fig. 1 c-1), or language explanations (Fig 2 bottom, Fig. 1 c-2).

Example tasks are shown in Fig. 2, with instances of explicit, implicit, and global queries in Fig. 1.

3 Method

3.1 Memory Construction

The memory construction process of Embodied-RAG consists of two parts: a topological map and a semantic forest.

Topological map We employ a topological graph composed of nodes with the following attributes:

- Position Information: Allocentric coordinates (x, y, z) and the yaw angle θ.

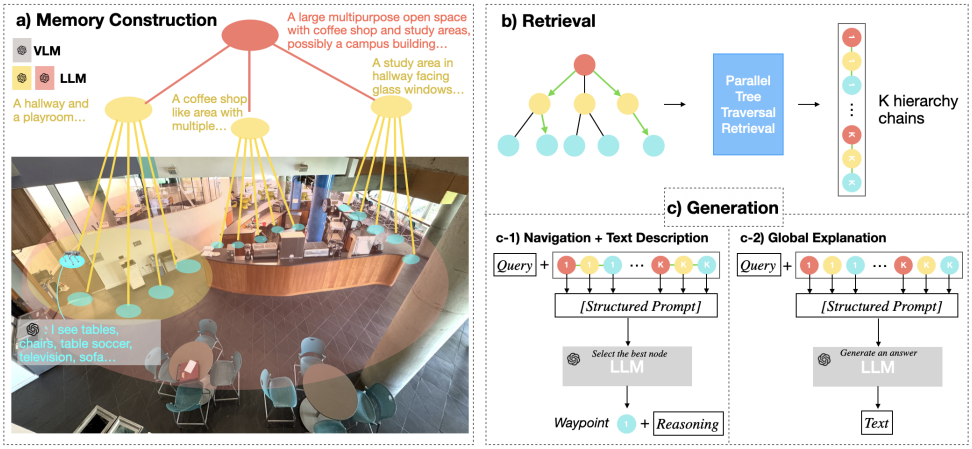

Figure 1: **Embodied-RAG method overview**. (a) Memory is constructed by hierarchically organizing the nodes of the topological map into a semantic forest. (b) The memory in (a) can be retrieved for a query, with parallelized tree traversals. (c) Navigation actions with text outputs, or global explanations can be generated for the query, with using the retrieval results as LLM contexts.

- Image Path: Each node contains a path to an associated ego-centric image.
- Captions: Generated by a vision-language model, these captions provide object-level natural language textual descriptions of the image.

The nodes form a topological map (blue nodes in Fig. 1), eliminating the need for specific control parameters like velocity and yaw, which often vary across different drive systems. This abstraction enables compatibility with any local planner, regardless of the robot's embodiment. Furthermore, the topological structure is far more memory-efficient than traditional metric maps [Chaplot et al., 2020, Min et al., 2021, Shafiullah et al., 2022], allowing for efficient scaling in both large outdoor and complex indoor environments. Our experiments show that this approach successfully navigates kilometer-scale simulated environments.

Semantic Forest We use a separate tree structure, referred to as a semantic forest, to capture meaning at various spatial resolutions. The nodes of this forest are those of the topological map, with the non-leaf nodes capturing larger spaces at a thinner density of semantic specificity. First, we create the forest through hierarchical clustering. Since spatially approximate leaf nodes exhibit semantic correlations, we employ an agglomerative clustering mechanism [Sneath and Sokal, 1973] to group nodes based on their physical positions assigning the mean position of the leaves.

This iterative process continues until a root node is formed, stopping when no further relevance is found based on a threshold set by the algorithm. Once we have a complete forest with one or more root nodes, each non-leaf node receives a language description. We achieve this by prompting a large language model (LLM, e.g., GPT-4) to generate a abstraction that encompasses the descriptions of its direct child nodes (see website for the prompting). This process is conducted bottom-up, starting from the leaf nodes and moving up to the parent nodes. We parallelized this process across all nodes at the same hierarchy.

3.2 Retrieval

We run the following *process*, which takes a single tree as input and outputs a single leaf node. Starting by visiting the root node, we run BFS with LLM selection; we ask *LLM_Selector* to choose the best child node of the currently visited node based on compatibility with the given query. For example, if the query is "find me a place that is bright and quiet but has some presence of people," we prompt the LLM to select the best description among the children of the currently visited node. We then visit the selected best child node and iterate this process until we reach a leaf node. Once we obtain k leaf nodes ($\frac{k}{N}$ nodes from each tree) by running this process $\frac{k}{N}$ times for each of the N trees, we obtain the "chain" from the selected node to the root node. The $\frac{k}{N}$ processes are parallelized across the N trees. The set of these best k chains is the retrieval output, containing semantics at all scales for any specific location that corresponds to the leaf scale. Embodied-RAG unifies the retrieval process to handle explicit, implicit, and global queries, producing both explanations and navigational actions as outputs. Note, these hierarchies and corresponding trees allow for querying

automatically created semantic regions, something particularly useful for outdoor navigation where walls and structures cannot be used to determine function.

3.3 Generation

We pass the retrieved k best chains as part of a context, for the LLM to generate navigation and text description (Fig. 2 top) or global explanations (Fig. 2 bottom). Given the query and the k chains, we prompt the LLM to "select" a waypoint with a reasoning, or to "explain" (prompt in our project website).

Navigation We select a waypoint (a leaf node of the semantic forest) and use a planner to generate navigational actions—sequences of (torque, velocity) pairs— to reach the waypoint. To select this waypoint, we ask the LLM to choose the best single leaf node, togehter with textual reasoning, using the query and the chain as input. Again, including the entire chain as input ensures that a waypoint can be generated for implicit navigation tasks as well.

Text Answers As depicted in Figure 1 (c), we concatenate the k chains as part of the prompt to the LLM. We ask it to generate an answer to the query based on the k retrieved chains. The spatial scale of attention in each node of the chain facilitate the LLM to generate responses at any semantic scale (explicit, implicit, general) based on the retrieved result.

4 Results

Table 1: Comparison of Methods on different Embodied-RAG Benchmarks.

Env.	Explicit			Implicit			Global		
	Embodied-RAG	RAG	Sem.	Embodied-RAG	RAG	Sem.	Embodied-RAG	RAG	Sem.
Small	0.955	0.955	0.955	1.000	0.818	0.364	4.88	3.67	-
Large	0.977	0.947	0.895	0.914	0.695	0.426	4.86	2.43	-
Total	**0.969**	0.949	0.877	**0.926**	0.706	0.410	**4.87**	2.68	-

We present **quantitative** results in Table 1, demonstrating the effectiveness of Embodied-RAG across explicit, implicit, and global retrieval tasks. We also classify environments as small or large based on the number of mapped nodes. Embodied-RAG consistently outperforms RAG and Semantic Match across all tasks and environments. While all methods perform well on explicit queries, Embodied-RAG provides a slight advantage due to its hierarchical structure. For implicit queries, RAG and Semantic Match performance drops significantly, especially in large environments, while Embodied-RAG remains robust. On global questions, Embodied-RAG excels, while Semantic Match, lacking summarization and reasoning, cannot be applied.

We conducted a **qualitative** comparison between Embodied-RAG and baseline models.

Implicit Query: Where can I buy drinks? Embodied-RAG correctly identifies a food service area, while the baselines return irrelevant answers like a refrigerator or water fountain. These results reflect a mismatch between the user's intent (to buy) and the retrieved objects. Embodied-RAG performs multi-step reasoning and retrieves more suitable locations, such as counters or vending machines, matching the user's intent.

Global Query: As illustrated in Figure 2, Embodied-RAG accurately describes the environment as a suburban neighborhood with a park, using its hierarchical structure to provide a cohesive view. In contrast, RAG retrieves isolated nodes, resulting in a fragmented and redundant interpretation. Embodied-RAG integrates elements like trees and shrubs into the broader park context, offering a more human-like understanding.

5 Conclusion

We present Embodied-RAG, a system that captures spatial memory at any resolution and generates responses for both navigation and explanation requests. We also introduce the Embodied-RAG benchmark, unifying semantic navigation and question answering. Our results show that Embodied-RAG handles implicit and global queries, as well as ambiguous human requests, demonstrating its

potential for integrating large non-parametric memories into robotics models. We look forward to future extensions involving manipulation and dynamic environments, enabling robotics tasks beyond current memory-constrained methods.

References

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. Acl 2023 tutorial: Retrieval-based language models and applications. *ACL 2023*, 2023.

Chaplot, Russ R, et al. Object goal navigation using goal-oriented semantic exploration. *NeurIPS*, 33, 2020.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation, 2023.

Khanna, Roozbeh, et al. Goat-bench: A benchmark for multi-modal lifelong navigation. *arXiv:2404.06609*, 2024.

Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances. *CVPR*, 2022.

Patrick Lewis, Douwe Kiela, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

Li, Fuchun, et al. Embodied semantic scene graph generation. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *CoRL*. PMLR, 2022.

So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods. *ICLR*, 2021.

Krishan Rana et al. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. In *CoRL*, 2023.

Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv: Arxiv-2210.05663*, 2022.

Peter HA Sneath and Robert R Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W.H. Freeman, 1973.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv:2305.16986*, 2023.

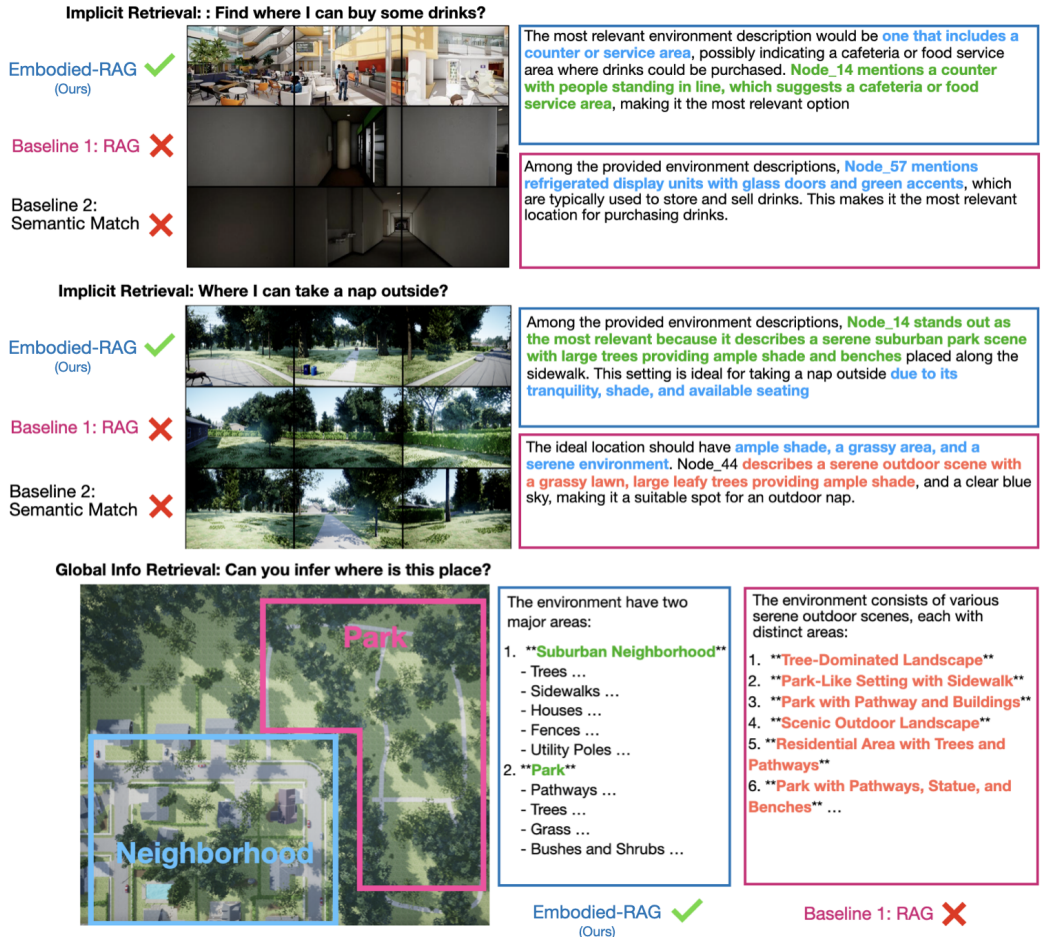

Figure 2: Example reasoning of Embodied-RAG and RAG for generation tasks are highlighted in blue and pink boxes, respectively.

A Computational Efficiency

Both memory construction and retrieval have a computational complexity of $O(\log N)$, where N represents the number of nodes in the environment. This choice allows us to efficiently scale to larger environments, as the time complexity only increases logarithmically with the number of nodes. Additionally, when performing the k retrievals, we execute them in parallel to minimize the overall time cost. In our real-life experiments, the time costs are demonstrated in the supplementary video, which is 8x fast-forwarded. On average, a single retrieval takes around 20 seconds in most of our environments, and the travel time depends on the speed of the specific embodiment in use.

B Ablation

We investigate the impact of $k \in \{1, \text{GPT4 Token Limit}\}$ on Embodied-RAG and RAG in Figure 3. A total of 15 experiments were conducted for each k in each environment. We observe that with larger k, both RAG and Embodied-RAG show improved performance, but this improvement plateaus at higher values. RAG still fails to capture the larger holistic resolution with just more object-level nodes and cannot adequately solve the implicit/general queries, further justifying our hierarchy and tree selection approach.

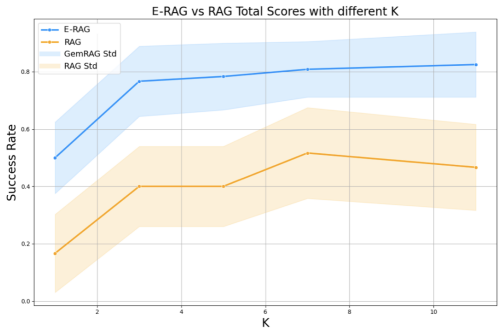

Figure 3: Effect of total number of K searches or K retrievals

C Limitations and Future work

We primarily focused on semantic forests rather than a topological map. Therefore, we may not be robust in obstacle avoidance involving dynamic objects and people. Furthermore, Embodied-RAG currently struggles with requests that require precise counting of objects at a small scale (e.g., "How many chairs are there around the red table?"). This limitation arises because the agglomerative clustering of the semantic forest does not consider multi-view consistency. Future work could incorporate multi-view consistency in the hierarchies of the semantic forest with a learned or pre-trained mechanism to cluster with positional information (e.g. utilizing a LLM).