

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

Quanting Xie¹, So Yeon Min¹, Pengliang Ji¹, Yue Yang², Tianyi Zhang¹, Kedi Xu¹, Aarav Bajaj¹, Ruslan Salakhutdinov¹, Matthew Johnson-Roberson¹, and Yonatan Bisk¹

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 250 explanation and navigation queries across kilometer-level environments, highlighting its promise for general-purpose non-parametric system for embodied agents.

Index Terms—Autonomous Agents, RAG, Embodied memory, Language-guided Robotics

I. INTRODUCTION

It is difficult for the human mind to determine what information should be remembered from our perceptually rich lived experiences. Where details are necessary, we revisit an experience or build explicit external representations like maps to capture the intricacies. This process begs questions of what the right semantic level and context should be indexed. Robots are now in the same but opposite position. While dense SLAM and metric maps can be constructed, they become intractable to scale, and they do not track with larger semantic categories we find most useful in human memory – we discard almost all low-level information as redundant and easy to rediscover.

Within Natural Language Processing (NLP), Retrieval-Augmented Generation (RAG) [1]–[3] integrates non-parametric memory into Large Language Models (LLMs),

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112490375 and partially supported by funding from Lockheed Martin Corporation.

¹Quanting Xie, So Yeon Min, Pengliang Ji, Tianyi Zhang, Kedi Xu, Aarav Bajaj, Ruslan Salakhutdinov, Matthew Johnson-Roberson, and Yonatan Bisk are with Carnegie Mellon University, Pittsburgh, PA 15213, USA

²Yue Yang with Apochs, Inc, CA 94560, USA

Corresponding author: quantinx@andrew.cmu.edu

<https://quanting-xie.github.io/Embodied-RAG-web/>

Category	Metrics Maps	Semantic Metric Maps	RAG	Embodied-RAG
Retrieval				
$P(A Q)$	×	✓	✓	✓
$P(A L)$	✓	✓	×	✓
$P(A L, Q)$	×	✓	×	✓
$P(A S, Q)$	×	×	✓	✓
$P(A S, L, Q)$	×	×	×	✓
Generation				
Text	×	×	✓	✓
Waypoint	✓	✓	×	✓
Path	✓	✓	×	✓

TABLE I: Comprehensive comparison of Metrics Maps, Semantic Metric Maps, RAG, and Embodied-RAG frameworks in terms of retrieval and generation capabilities. Here, Q represents the query, L denotes the embodied agent’s position, and S refers to other sensor data.

enabling the use of large text corpora as private knowledge bases to enhance a model’s memory, relevance, and factual grounding of model outputs, particularly in scenarios requiring access to up-to-date or domain-specific knowledge. We ask if such insights can be leveraged to endow robots with better scaling semantic memory, and what new technologies need to be invented to handle embodied experiences.

Applying RAG to robotics presents unique challenges due to key differences between textual data and embodied experiences. First, embodied experiences are multimodal – How do we make such data queryable for a RAG system? Unlike Internet documents, which are distinct and well-structured text, embodied data often consist of tuples of time, sensor observations, and robot poses, $E_t = (\tau_t, s_t, \mathbf{p}_t)$. These multidimensional data need to be efficiently coupled and stored. Further, current representations of embodied experiences, such as dense point-cloud maps, fail to abstract the relevant semantics needed for a natural language query. Although 3D scene graphs [4] are interpretable, they rely on human-engineered schemas that do no scale to diverse outdoor environments.

Second, Naive RAG [1] lacks the cross-document structural awareness needed for building spatially informed knowledge graphs, and structured graphical RAG methods [5], [6] are too inefficient to build and query for real-time deployment. Finally, embodied observations are redundant and repetitive, which can confuse the retriever when attempting to select the correct context using semantic similarity alone, requiring extra reasoning steps during inference.

To address these challenges, we present Embodied-RAG. Embodied-RAG has two components: *Bottom-up Memory*

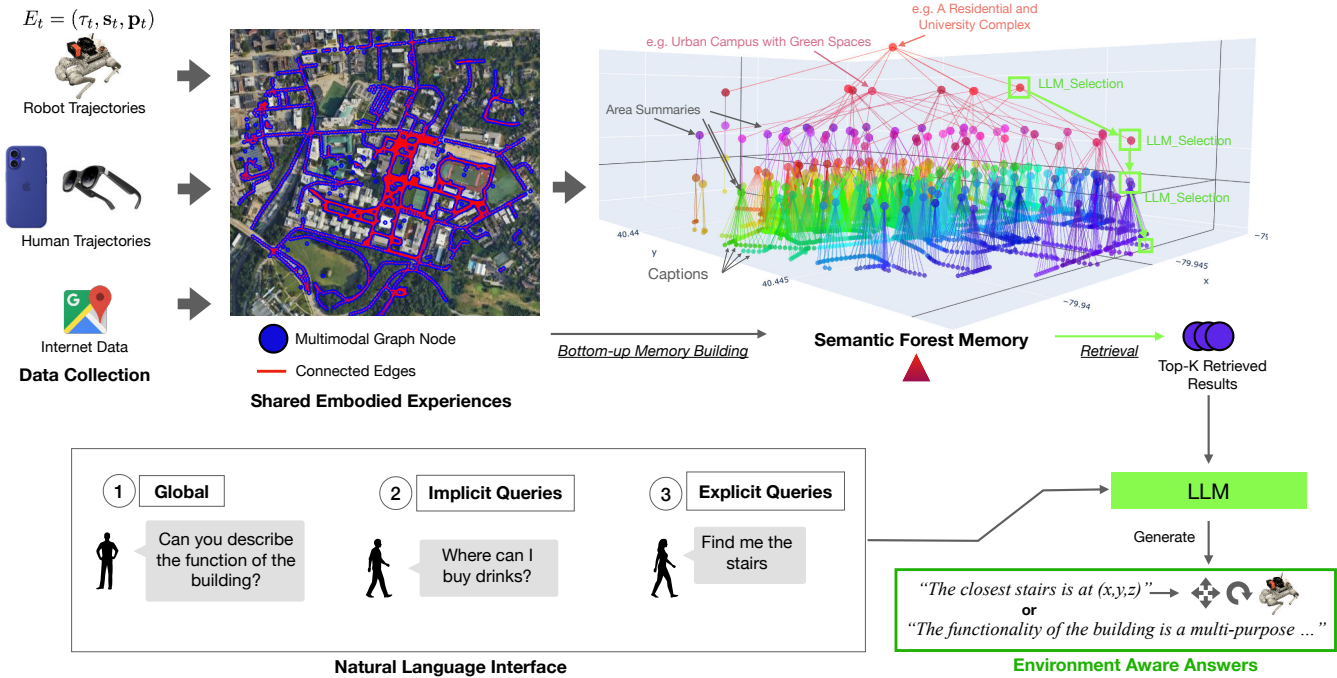


Fig. 1: **Overview:** Our goal is for robots to navigate and communicate effectively in any environment where humans are present. We introduce Embodied-RAG, a framework for automatically building hierarchical spatial memory and providing both explanations and navigation across multiple levels of query abstraction. Embodied-RAG supports robotic operations regardless of the query’s abstraction level, the platform, or the environment.

Building (Fig.2(a)) and *Top-down Retrieval* (Fig.2(b, c)). During *Bottom-up Memory Building*, we address the first two problems: multimodal representation and efficient integration of structure into embodied experiences. The system first represents embodied experiences with a multimodal topological graph, where each node contains robot poses, robot observations (images), and timestamps. Based on these topological nodes, a *semantic forest* is hierarchically clustered based on spatial proximity. This graph-building process is 7.38X faster than Graph-RAG and 9.76X faster than Light-RAG on the same dataset size, and can be extended in real-time. This two-stage memory system creates an efficient, large-scale, globally aware, interpretable, and multimodal memory representation for embodied agents to retrieve from.

In the *Top-down Retrieval* process, to overcome the third challenge, we enhance retrieval performance across three different query types (explicit, implicit, and global), outperforming the state-of-the-art RAG baselines [1], [5], [6]. Instead of relying solely on semantic similarity, Embodied-RAG incorporates a robust reasoning component for retrieval. This involves parallelized tree traversals with a selection-LLM. This retrieval process uses abstracted information from the top nodes to guide retrieval to the correct lower nodes with equal probabilities. For example, a toothbrush node is more likely inside a bathroom node, and a bench in a backyard is less desirable than a bench in a park for quietly reading a book.

To evaluate Embodied-RAG, we present a new dataset called the **Embodied-Experiences Dataset** for Embodied-

RAG tasks. It contains topological graphs collected from 14 photorealistic simulated and 5 real environments of varying scales paired with two hundred queries with ground truth labeled information.

Furthermore, our experiments demonstrate that Embodied-RAG is a more efficient graphical non-parametric memory for embodied data, surpassing the baselines [5], [6] in memory building time, and achieve better performance across all query types. In addition, semantic forest is a more versatile memory, as shown in Table I, and it is capable of taking multi types of input, while also able to be applied to various forms of embodiment (drones, locobots, quadrupeds) as global planner, seamlessly integrated with existing low-level autonomous navigation pipelines. This highlights Embodied-RAG’s potential as a general system capable of task-, environment-, and platform-agnostic operation, enabling robots to effectively navigate and communicate in any environment where humans are present.

The key contributions and implications of this paper include:

- **Task:** We extend RAG into embodied settings and highlight the unique challenges of retrieval from embodied experiences.
- **Dataset:** We present the *Embodied-Experience Dataset*, formulating semantic navigation and question answering under a single paradigm (Table I, Figure 1).
- **Method:** We show an initial step toward solving the challenges in representing and retrieving from embodied experiences, outperforming Naive-RAG, GraphRAG [5],

and LightRAG [6] on different query types across 19 diverse real and simulated environments. In addition, the high-speed memory building process make it applicable in real-time navigation and mapping.

- **Implications:** Our results and discussion provide a basis for rethinking approaches to generalist robot agents based on language-form non-parametric memories.

II. RELATED WORKS

A. Retrieval and Generation

Retrieval-Augmented Generation (RAG) systems integrate large language models (LLMs) with external text corpora to enhance factual grounding and relevance in generated outputs [1], [7]–[9]. Traditional RAG models embed user queries and document chunks into a shared vector space, retrieving the top-k most semantically similar text fragments to augment the model’s context window [10], [11]. This approach effectively improves performance on tasks requiring domain-specific or up-to-date information. However, naive RAG systems [9] rely heavily on fragmented text chunks and simple similarity-based retrieval, limiting their ability to capture comprehensive and globally coherent information. Advanced RAG models such as GraphRAG [5] and LightRAG [6] have been developed to overcome these limitations by extract entities and their relationships, organize them into graph structure for more complete and globally aware retrieval. However, due to the intrinsic nature of embodied experiences are often redundant, hierarchically correlated, and spatially grounded, these purely textual graph building approaches don’t perform that well. In contrast, Embodied-RAG utilized spatial correlations to build spatially related scene graphs.

B. Existing Methods of Semantic Memory and Retrieval

Several methods have been proposed for storing and querying semantic memory in spatial environments, but they remain limited and task-specific compared to the potential of foundation models. Approaches like [12]–[14] associate voxels with predefined object categories, enabling fixed vocabulary retrieval, while methods such as [15], [16] map voxels to image embeddings, allowing for open vocabulary queries. Systems like [17] store images per voxel, supporting queries about people, language/image inputs, and object categories. However, a common challenge across these approaches is aligning the semantic abstraction with the spatial resolution. Queries such as “cup,” “red cup,” or “I want to heat my lunch” are object-level, but methods like [18], [19] focus primarily on local retrieval during exploration, using structured frontiers based on object layouts. Scene graphs [20], [21], while free from dense memory issues, rely on human-engineered schemas (e.g. floor \rightarrow room \rightarrow object \rightarrow asset), making them unsuitable for novel or outdoor environments.

Other approaches, such as OCTREE maps [22] and their semantic versions [23]–[25], organize occupancy data efficiently but still limit semantics to the object level. Methods like Semantic OCTREE [23], [25] and GENMos [24] use fixed object categories, lacking support for free-form language queries or varying levels of spatial and semantic resolution needed for holistic understanding.

C. Semantic Navigation and Question Answering

Tasks like ObjectNav [14], [19], [26], ImageNav [27]–[29], and Visual Language Navigation [30] assess a robot’s ability to navigate towards semantic targets based on object categories, images, or language descriptions. While recent efforts like GOATBench [31] combine multiple input types, these tasks still focus on object-level queries and lack the flexibility to handle broader, more abstract user requests. Embodied Question Answering (EQA) [32]–[35] and Video Question Answering (VideoQA) [36]–[39] extend navigation by requiring text-based answers within actionable or video environments, though EQA is limited to indoor settings and VideoQA lacks active navigation. Our approach expands these paradigms by integrating action-based and question-answering capabilities across a wider range of environments and user queries.

III. METHOD: EMBODIED RETRIEVAL AND GENERATION

A. Bottom-up 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:

- Pose information: The (x, y, z) position and yaw angle θ on the map where the image was captured. Blue nodes in 1 are the topological nodes, and they are connected according to agent’s path history or within a threshold α
- Timestamps
- Images: Ego-centric images.
- Captions: Generated by a VLM (GPT-4o), these captions provide detailed textual descriptions of the image.

The nodes form a topological map (blue nodes in Fig. 2), 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 [12], [14], [15], allowing for efficient scaling in both large outdoor and complex indoor environments. Our experiments show that this approach successfully operates on kilometer-scale topological graphs.

Semantic Forest The concept of a semantic forest leverages the observation of intrinsic structure of embodied data, where objects and scenes naturally exhibit spatial and semantic organization. By capturing higher-level spatial and semantic information in a hierarchical tree structure, known as a *semantic forest*, we can effectively model these relationships through a two-step iterative process: clustering and summarization.

First, we employ complete-linkage hierarchical clustering (CLINK) [40], [41] with a novel hybrid distance metric to group leaf nodes (level 0 nodes in Fig. 2(a)). The hybrid similarity matrix computation integrates both spatial and semantic relationships between nodes through a weighted combination approach. The similarity (S_{ij}) of two nodes i and j is defined as the weighting of two terms:

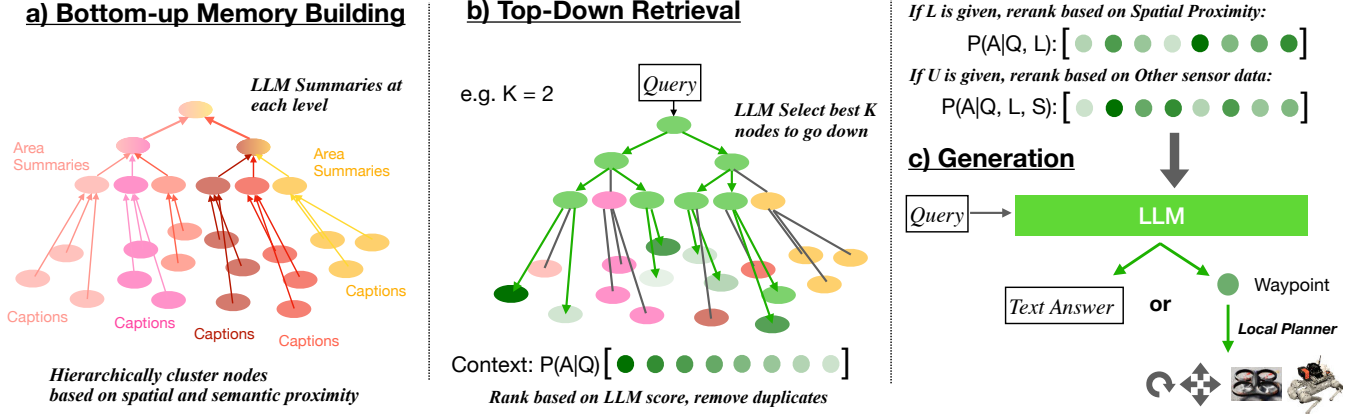


Fig. 2: **Embodied-RAG method overview.** (a) Memory is constructed by hierarchically organizing the nodes of the topological map into a *semantic forest*. (b) The memory in (a) can be retrieved for a query, with parallelized tree traversals. (c) Navigation actions with text outputs, or global explanations can be generated for the query, with using the retrieval results as LLM contexts.

$$S_{ij} = (1 - \alpha)S_{ij}^{\text{spatial}} + \alpha S_{ij}^{\text{semantic}}$$

The components of the similarity metric are as follows:

Spatial Similarity (S_{ij}^{spatial}): The spatial similarity is computed using the haversine distance with exponential decay:

$$S_{ij}^{\text{spatial}} = \exp\left(-\frac{d_{\text{haversine}}(i, j)}{\theta}\right)$$

Where $d_{\text{haversine}}(i, j)$ is the great-circle distance between locations i and j , and θ is the base distance threshold.

Semantic Similarity (S_{ij}^{semantic}): The semantic similarity is computed as the cosine similarity between neural language embeddings $\mathbf{e}_i, \mathbf{e}_j$ of the text descriptions for nodes i and j .

$$S_{ij}^{\text{semantic}} = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \|\mathbf{e}_j\|}$$

Once the clusters are formed at each level, we generate semantic summaries for each cluster using an LLM summarizer (e.g., GPT-4). The summary and the average distance between nodes are saved as new nodes (level 1-3 nodes in Fig. 2(a)). This bottom-up clustering process continues until either root nodes are formed or no further meaningful clusters can be created. The summarization process is parallelized across clusters at the same hierarchical level, ensuring efficient processing of the entire forest structure.

Unlike traditional 3D Scene Graph approaches [4], [20], which often rely on manual rules for identifying rooms and functional spaces, our hierarchical structure and corresponding semantic trees enable the automatic creation of meaningful semantic regions. This approach is particularly advantageous for outdoor navigation, where walls and physical structures are absent and cannot be used to infer functional areas.

B. Top-down Retrieval

To address the challenges posed by redundant and repetitive embodied observations, which make retrieval difficult when

relying solely on semantic similarities as in Naive RAG [1], and to enhance reasoning capabilities over hierarchies of abstraction constructed for a given environment, we modified RAG’s relevancy scoring mechanism in a manner inspired by Tree-of-Thoughts [42]. Specifically, the two phase retrieval mechanism transitions from semantic similarity to LLM-based selections at each hierarchical level:

Phase 1: Semantic-Guided Hierarchical Traversal: The first phase involves a top-down parallel exploration of the semantic forest, where node selection is guided purely by semantic relevance. For a given query q , we define a selection function:

$$N_l = f_{\text{LLM}}(q, C_l, k)$$

Here, N_l represents the set of selected nodes at level l , C_l denotes the candidate nodes at that level, and f_{LLM} is the LLM-based selection function. The parameter k specifies the branching factor. The algorithm recursively explores the children of each selected non-base node until the base nodes are reached. By focusing on semantic relevance during traversal, this phase prunes irrelevant branches early and ensures computational efficiency, while still comprehensive exploration over the entire semantic forest.

Phase 2: Hybrid Re-ranking: Once the base nodes are collected, they are scored individually by another LLM and ranked in descending order of relevance, kept in a set to remove duplicates. If location information is provided, the probability distribution is updated from $P(A | Q)$ to $P(A | Q, L)$. A re-ranking based on spatial proximity is then conducted by computing spatial scores and combining them with semantic scores, weighted by a factor α . The spatial scores are calculated in the same manner as the spatial similarity described in Section III-A.

Additionally, if the embodied experiences include more sensor data, these data can be preprocessed to refine the scoring mechanism. For instance, in this study, Normalized Difference Vegetation Index (NDVI) data is used to determine the quality of the grass. This score is integrated into the re-

ranking process to prioritize regions with low-quality grass. The preprocessing and re-ranking steps can be adapted to the specific requirements and its operational goals, allowing the framework to remain flexible across different applications.

C. Generation

The retrieved nodes are passed as part of the context, along with the user query, to a generation LLM. The LLM is prompted to handle two types of queries: (1) For “*find*” queries (explicit or implicit), it outputs a desired waypoint in a JSON format, along with reasoning for its choice; (2) For “*explain*” queries, it generates a textual response. The detailed prompt used for the generation process is available on our project website. The Embodied-RAG pipeline can be conceptualized as a global planner for “*find*” queries. Since the topological graph contains connectivity information, a Dijkstra’s path planning algorithm is employed to compute the minimum-distance path between the current location and the selected waypoint. For navigation between waypoints, any local planner can be integrated. In this paper, we use the Unitree-Go2’s “go-to-waypoint” API as our local planner.

IV. EXPERIMENTS

A. Embodied-Experiences Dataset

The datasets utilized in the Embodied-RAG task are structured as topological graphs (see Section III for node details).

The dataset is composed of diverse environmental settings collected through complementary data collection techniques. Real-world environments were explored using autonomous robots to construct three detailed indoor graphs and one mixed outdoor-indoor graph, capturing realistic navigation scenarios. To model large-scale urban spaces, a comprehensive street-view graph was created using imagery from Google Street View, providing broad and complex spatial data. Additionally, fourteen object-centric topological graphs were generated using the photo-realistic AirSim [43] simulator, enabling controlled simulation of varied and intricate environments. On average, the topological graphs contain approximately 50 nodes, reflecting moderate complexity in most environments, while the large-scale street-view graph consists of 3,525 nodes, offering extensive spatial coverage for evaluating more complex navigation and retrieval tasks.

The dataset is further divided based on modality for experimentation. In the *E-image* setting, each topological graph, denoted as $E_t = (\tau_t, \mathbf{s}_t, \mathbf{p}_t)$, includes nodes where \mathbf{s}_t contains only image data. In contrast, the *E-multimodal* setting incorporates nodes where \mathbf{s}_t includes both images and additional sensory data. In the experiments shown in Table III, \mathbf{s}_t contains both image data and NDVI readings, reflecting the multimodal nature of the environment.

B. Embodied-RAG Task

We include two query types: *Find* and *Explain*.

Find queries has two subcategories: (1) Explicit Queries. These involve searching for a specific object instance or a clearly defined target (e.g., “Find a bench”), (2) Implicit

Queries: These require a more nuanced, pragmatic understanding, such as assessing adequacy or interpreting instructions with contextual reasoning (e.g., “Find a quiet spot to read”).

For *Explain* queries, the request may pertain to global information, such as describing a specific location or providing a general understanding of the environment (e.g., “What’s the vegetation trend of this environment?”).

Example tasks are shown in Fig. 1. *Find* queries are navigational tasks that expect navigation actions and text descriptions of the retrieved location. *Explain* queries are QA tasks requiring text generation at a more holistic level.

The queries were collected by four human annotators familiar with the Embodied-Experience datasets. The annotators created queries by reviewing the dataset’s images and leveraging their understanding of the environmental context.

V. RESULTS

A. Evaluation

To comprehensively assess the system’s performance, we employ distinct evaluation metrics tailored to the nature of the queries: *Find* and *Explain*.

1) *Find Queries*: To effectively evaluate the system’s capabilities as a global planner, we separate *navigation success* from *generation success*. The system outputs an image path as the result and calculates the probability $P(Q | A)$, representing the likelihood of finding the queried object given the generated answer (image path). This probability is determined using a **cross-voting technique** among five Vision-Language Models (VLMs), ensuring unbiased scoring for open-ended queries.

Instead of binary checks, we use probabilities to account for the inherent ambiguity in implicit queries (e.g., “Find me a place to eat”), where deterministic answers are not always feasible. Additional details about the VLM prompts and evaluation methodology are available on our project website.

If *location information* is provided $P(Q | A, L)$, we extend the evaluation by weighting $P(Q | A)$ by the *path length*, similar to established metrics like Success Weighted by (normalized inverse) Path Length (SPL) [26]. Specifically:

$$P(Q | A, L) = P(Q | A) \times \frac{\text{path length}}{\text{radius of the environment}}$$

This adjustment ensures the evaluation reflects not only the success of finding the object but also the efficiency of the navigation path.

2) *Explain Queries*: For explanation queries, we constructed a **golden dataset** by collecting answers from expert annotators for each query. The system’s generated responses are evaluated by computing the *semantic similarity* between the generated answers and the corresponding golden answers $SS(A, A_e)$ A_e represents the expert-provided answer.

B. Baselines

To evaluate our Embodied-RAG approach, we conducted comparative experiments against three baseline methods: *Naive-RAG*, *GraphRAG*, and *LightRAG*.

For compatibility with Naive-RAG [1], we converted the graph files from the Embodied-Experience dataset into plain

Query Types	Metrics	Input Types	Naive-RAG	Graph-RAG	Light-RAG	Embodied-RAG
Explicit	$P(Q A)\uparrow$	Q only	0.08	0.06	0.08	0.55
	$P(Q A, L)\uparrow$	Q, L	0.041	0.029	0.027	0.28
Implicit	$P(Q A)\uparrow$	Q only	0.10	0.12	0.13	0.62
	$P(Q A, L)\uparrow$	Q, L	0.07	0.06	0.07	0.25
Global	$SS(A, A_e)\uparrow$	Q only	0.31	0.68	0.65	0.67

TABLE II: Performance Comparison on the *E-image* Dataset across different methods. Embodied-RAG consistently outperforms all other methods across Explicit, Implicit, and Global query types, achieving the highest scores in both retrieval probabilities ($P(Q | A)$ and $P(Q | A, L)$) and semantic similarity ($SS(A, A_e)$) metrics. Input types used in evaluation are specified for each query type.

text (.txt) files. The dataset is divided into text chunks, and the system retrieves semantically relevant chunks to populate the context window of GPT-4o (with a token limit of 16k). These retrieved chunks are used to generate enhanced responses. Naive-RAG does not leverage any structural knowledge of the dataset, treating it as flat text.

GraphRAG [5] incorporates graph structures into the Naive-RAG system. After preprocessing the dataset into text chunks (similar to Naive-RAG), GraphRAG utilizes an LLM to extract entities and relationships from the text and aggregates them into different communities. A graph is then constructed to capture global relationships, with community reports summarizing the entities and their connections. During retrieval, GraphRAG generates multiple intermediate answers in parallel, one for each chunk, ranks them based on a helpfulness score, and iteratively adds the most relevant answers to the context window until the token limit is reached. The final answer is generated based on this enriched context.

Finally, *LightRAG* [6] is a state-of-the-art graphical RAG approach designed for efficiency. LightRAG’s contribution is more efficient retrieval by indexing the graph using a dual-level key system (low-level and high-level keys). During retrieval, this dual-level retrieval method is used to retrieve relationships between key entities. Like the other baselines, the same preprocessing steps were applied to the dataset.

C. Quantitative Results

The main results are presented in Table II and Table III, where we evaluate performance on both the *E-image* and *E-multimodal* datasets using three different query types and three different input types. The performance of the baselines is notably poor for explicit and implicit query types. This is primarily because chunking multimodal embodied data into text often fails to retrieve the correct images, resulting in retrieval failures in most cases. For global queries, however, the baselines *LightRAG* and *GraphRAG* outperform *Naive-RAG*, demonstrating the effectiveness of graph structures in generating holistic responses about the environment.

Notably, Embodied-RAG outperforms all baselines for explicit and implicit queries across all input types. For global queries, particularly under $P(A | Q, L)$, Embodied-RAG

Query Types	Metrics	Input Types	Naive-RAG	Graph-RAG	Light-RAG	Embodied-RAG
Explicit	$P(Q A)\uparrow$	Q only	0.08	0.09	0.12	0.58
	$P(Q A, L)\uparrow$	Q, L	0.03	0.04	0.03	0.28
	$P(Q A, L)\uparrow$	Q, L, S	0.04	0.04	0.03	0.36
Implicit	$P(Q A)\uparrow$	Q only	0.10	0.12	0.13	0.67
	$P(Q A, L)\uparrow$	Q, L	0.04	0.05	0.07	0.29
	$P(Q A, L)\uparrow$	Q, L, S	0.04	0.04	0.08	0.41
Global	$SS(A, A_e)\uparrow$	Q only	0.60	0.72	0.75	0.74
	$SS(A, A_e)\uparrow$	Q, S	0.46	0.68	0.78	0.95

TABLE III: Performance on the *E-multimodal* Dataset with Input Types. This table includes sensor data as an additional input, represented in rows with Q, L, S or Q, S . The results show that Embodied-RAG achieves significant performance improvements over metrics $P(Q | A, L)$ when sensor data is incorporated. Embodied-RAG consistently outperforms other methods across all query types and metrics.

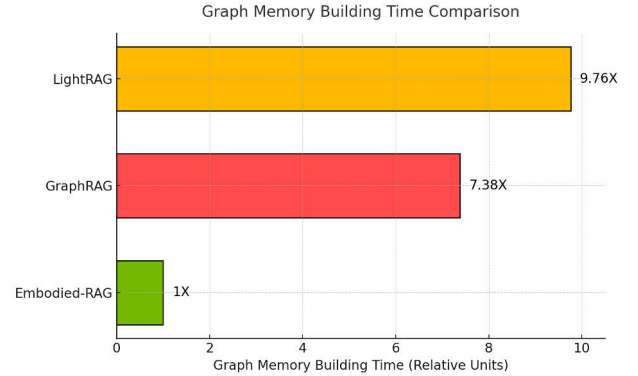


Fig. 3: Graph Memory Building Time Comparison. The relative graph memory building time for Embodied-RAG, GraphRAG, and LightRAG is displayed, normalized to Embodied-RAG (1X). Embodied-RAG demonstrates significantly faster graph construction, being 7.38 times faster than GraphRAG and 9.76 times faster than LightRAG

shows superior performance, highlighting its flexibility in the hybrid re-ranking step described in Section III-B. This flexibility enables it to adapt actively to spatial constraints during retrieval.

In Table III, we observe that when sensor data is provided to Embodied-RAG, its performance further improves, while the baselines remain unaffected. This result emphasizes the advantages of integrating multimodal information, offering insights into how sensor data enhances the system’s ability to understand and respond effectively to complex queries.

D. Computation Results

We conducted a computational comparison of the graphical memory building time between Embodied-RAG, *LightRAG*, and *GraphRAG*. The results, shown in Fig. 3, demonstrate that Embodied-RAG’s graph-building process is 7.38 times faster than GraphRAG and 9.76 times faster than LightRAG. This efficiency is attributed to Embodied-RAG’s semantic forest design, which leverages the inherent properties of embodied

Metrics	Naive-RAG		Graph-RAG		Light-RAG		Embodied-RAG	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Explicit Queries								
P(Q A)	0.51	0.55	0.64	0.522	0.53	0.43	0.65	0.44
NI. Path Length	0.38	0.47	0.44	0.45	0.39	0.37	0.48	0.68
P(Q A, L)	0.19	0.26	0.28	0.23	0.20	0.16	0.31	0.30
Implicit Queries								
P(Q A)	0.60	0.51	0.66	0.60	0.64	0.60	0.62	0.57
NI. Path Length	0.32	0.45	0.65	0.57	0.28	0.27	0.69	0.68
P(Q A)	0.19	0.23	0.38	0.34	0.18	0.16	0.41	0.39

TABLE IV: Retrieval performance comparison for explicit and implicit queries on Embodied-RAG graph memory with $P(A | Q, L)$ input. Metrics include $P(Q | A)$, normalized inverse path length (NI. Path Length), and $P(Q | A, L)$, evaluated for Top-1 and Top-5 results. Embodied-RAG consistently outperforms other methods across most metrics, particularly in $P(Q | A, L)$ for both query types, showcasing its strong retrieval and navigation capabilities.

data. Specifically, objects or locations that are spatially close can naturally be abstracted into higher-level clusters, reducing the need for excessive LLM calls to generate relationships between individual text chunks as the baseline do. By clustering larger groups of information at once, Embodied-RAG creates a leaner graph structure.

On average, Embodied-RAG takes approximately 4 minutes and 35 seconds to build a complete semantic forest for a one-kilometer radius environment (e.g., the CMU dataset inside Embodied-Experiences dataset), consisting of 3,353 nodes. Furthermore, the semantic forest can be built incrementally by progressively clustering nodes at each hierarchical level.

E. Ablation

Since traditional RAG systems are not designed to handle embodied experience data, baselines perform poorly on *Find* tasks. This raises the question as to whether the superior performance of Embodied-RAG is due to its memory structure or its retrieval mechanism. To explore this, we modified the baselines to retrieve directly from our *semantic forest* memory, allowing a direct comparison of retrieval performance. For each method, we computed $P(Q | A)$ and path-weighted $P(Q | A, L)$ for the top-1 and top-5 retrieved images. The results reveal that while our retrieval method achieves similar performance on $P(Q | A)$, it consistently produces shorter path lengths. Consequently, Embodied-RAG demonstrates superior performance on $P(Q | A, L)$, underscoring its ability to effectively leverage both semantic and spatial relationships within the semantic forest to build better context for generation.

VI. LIMITATIONS AND FUTURE WORK

Embodied-RAG is not a drop-in replacement for other mapping approaches within the robotics community, but rather a supplement that focuses on open-world hierarchical semantics, building a high-level nonparametric memory system for easy integration of language models and related technologies with embodied agents versus low-level mapping and planning or precise vision reasoning often necessary in robotics.

For example, our work assumes access to a perfect local planner for the navigation task. This results in our system

not guaranteeing robustness in obstacle avoidance involving dynamic objects and people. A natural question for future work is to include dynamic objects in the memory but this requires also reasoning over the concept of stale observations.

On the topic of visual reasoning, Embodied-RAG 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. Generally, current (V)LMs struggle with 3D spatial reasoning, so future work could need to explicitly incorporate multi-view consistency techniques into the hierarchies of the semantic forest with a learned or pre-trained mechanism to cluster with positional information (e.g. utilizing a LLM).

Finally, the reliance on (V)LM APIs creates a deployment dependency of nearly uninterrupted internet access. While we tackled some of this directly in our efficiency evaluation, the question of what knowledge and abilities are lost when models are distilled and quantized to the point of being deployable offline on local compute is an open research topic with the NLP community and is left to future work.

VII. CONCLUSIONS

We present Embodied-RAG, a nonparametric embodied memory system capable of capturing embodied memories at any spatial and semantic resolution in both indoor and outdoor environments, and retrieving and generating responses for navigation and explanation requests. Additionally, we introduce the Embodied-Experiences datasets for allowing the community to continue testing different RAG system for robotics settings. Our findings demonstrate that Embodied-RAG can outperform existing baselines in all explicit, implicit, and global queries, while able to build the structured graph-memory 9.76 times faster than LightRAG. Our results indicate that Embodied-RAG shows potential as the basis for incorporating large nonparameteric embodied memories into foundation models. The memories constructed here are open-world and semantically rich while still being tied to the environment. This provides a fundamentally new resource to robotic systems and we are excited for future extensions to manipulation and dynamic environments that enable robotics tasks out of reach for current approaches.

REFERENCES

- [1] P. Lewis, D. Kiela *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” 2021.
- [2] A. Asai, S. Min, Z. Zhong, and D. Chen, “Acl 2023 tutorial: Retrieval-based language models and applications,” *ACL 2023*, 2023.
- [3] J. Chen, H. Lin, X. Han, and L. Sun, “Benchmarking large language models in retrieval-augmented generation,” 2023.
- [4] N. Hughes *et al.*, “Hydra: A real-time spatial perception system for 3d scene graph construction and optimization,” *RSS*, 2022.
- [5] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson, “From local to global: A graph rag approach to query-focused summarization,” *arXiv preprint arXiv:2404.16130*, 2024.
- [6] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, “Lightrag: Simple and fast retrieval-augmented generation,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.05779>
- [7] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham, “In-context retrieval-augmented language models,” 2023. [Online]. Available: <https://arxiv.org/abs/2302.00083>
- [8] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, “A survey on rag meeting llms: Towards retrieval-augmented large language models,” 2024. [Online]. Available: <https://arxiv.org/abs/2405.06211>
- [9] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, “Retrieval-augmented generation for large language models: A survey,” 2024. [Online]. Available: <https://arxiv.org/abs/2312.10997>
- [10] L. Gao, X. Ma, J. Lin, and J. Callan, “Precise zero-shot dense retrieval without relevance labels,” 2022. [Online]. Available: <https://arxiv.org/abs/2212.10496>
- [11] C.-M. Chan, C. Xu, R. Yuan, H. Luo, W. Xue, Y. Guo, and J. Fu, “Rq-rag: Learning to refine queries for retrieval augmented generation,” 2024. [Online]. Available: <https://arxiv.org/abs/2404.00610>
- [12] S. Y. Min, D. S. Chaplot, P. Ravikumar, Y. Bisk, and R. Salakhutdinov, “Film: Following instructions in language with modular methods,” *ICLR*, 2021.
- [13] S. Y. Min, Yonatan *et al.*, “Don’t copy the teacher: Data and model challenges in embodied dialogue,” *EMNLP*, 2022.
- [14] Chaplot, R. R *et al.*, “Object goal navigation using goal-oriented semantic exploration,” *NeurIPS*, vol. 33, 2020.
- [15] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, “Clip-fields: Weakly supervised semantic fields for robotic memory,” *arXiv: Arxiv-2210.05663*, 2022.
- [16] C. Huang, O. Mees, A. Zeng, and W. Burgard, “Visual language maps for robot navigation,” in *Proceedings of the ICRA*, London, UK, 2023.
- [17] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, S. Y. Min, K. Shah, C. Paxton, S. Gupta, D. Batra *et al.*, “Goat: Go to any thing,” *arXiv:2311.06430*, 2023.
- [18] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, “Poni: Potential functions for objectgoal navigation with interaction-free learning,” in *ICCV*, 2022, pp. 18 890–18 900.
- [19] S. Y. Min, Y.-H. H. Tsai, W. Ding, A. Farhadi, R. Salakhutdinov, Y. Bisk, and J. Zhang, “Self-supervised object goal navigation with in-situ finetuning,” in *2023 IROS*. IEEE, 2023, pp. 7119–7126.
- [20] Li, Fuchun *et al.*, “Embodied semantic scene graph generation,” in *CoRL*, A. Faust, D. Hsu, and G. Neumann, Eds. PMLR, 2022.
- [21] K. Rana *et al.*, “Sayplan: Grounding large language models using 3d scene graphs for scalable task planning,” in *CoRL*, 2023.
- [22] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: An efficient probabilistic 3d mapping framework based on octrees,” *Autonomous robots*, vol. 34, pp. 189–206, 2013.
- [23] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, “Semantic slam based on object detection and improved octomap,” *IEEE Access*, vol. 6, pp. 75 545–75 559, 2018.
- [24] K. Zheng, A. Paul, and S. Tellex, “Asystem for generalized 3d multi-object search,” in *2023 ICRA*. IEEE, 2023, pp. 1638–1644.
- [25] K. Liu, Z. Fan, M. Liu, and S. Zhang, “Object-aware semantic mapping of indoor scenes using octomap,” in *2019 Chinese Control Conference (CCC)*. IEEE, 2019, pp. 8671–8676.
- [26] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, “On evaluation of embodied navigation agents,” *arXiv preprint arXiv:1807.06757*, 2018.
- [27] L. Mezghan, S. Sukhbaatar, T. Lavril, O. Maksymets, D. Batra, P. Bojanowski, and K. Alahari, “Memory-augmented reinforcement learning for image-goal navigation,” in *2022 IROS*. IEEE, 2022, pp. 3316–3323.
- [28] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, “Target-driven visual navigation in indoor scenes using deep reinforcement learning,” in *2017 ICRA*. IEEE, 2017, pp. 3357–3364.
- [29] J. Krantz, S. Lee, J. Malik, D. Batra, and D. S. Chaplot, “Instance-specific image goal navigation: Training embodied agents to find object instances,” *CVPR*, 2022.
- [30] J. Gu, E. Stefani, Q. Wu, J. Thomason, and X. E. Wang, “Vision-and-language navigation: A survey of tasks, methods, and future directions,” *arXiv:2203.12667*, 2022.
- [31] Khanna, Roohbeh *et al.*, “Goat-bench: A benchmark for multi-modal lifelong navigation,” *arXiv:2404.06609*, 2024.
- [32] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, “Embodied question answering,” in *CVPR*, 2018, pp. 1–10.
- [33] Padalkar *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models,” *arXiv:2310.08864*, 2023.
- [34] L. Yu, X. Chen, G. Gkioxari, M. Bansal, T. L. Berg, and D. Batra, “Multi-target embodied question answering,” in *ICCV*, 2019, p. 6309.
- [35] S. Tan, M. Ge, D. Guo, H. Liu, and F. Sun, “Knowledge-based embodied question answering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [36] Zhong, Tat-Seng *et al.*, “Video question answering: Datasets, algorithms and challenges,” *arXiv:2203.01225*, 2022.
- [37] H. Yang, L. Chaisorn, Y. Zhao, S.-Y. Neo, and T.-S. Chua, “Videoqa: question answering on news video,” in *Proceedings of the eleventh ACM international conference on Multimedia*, 2003, pp. 632–641.
- [38] Castro, Rada *et al.*, “Lifeqa: A real-life dataset for video question answering,” in *LREC*, 2020.
- [39] J. Xiao, X. Shang, A. Yao, and T.-S. Chua, “Next-qa: Next phase of question-answering to explaining temporal actions,” in *ICCV*, 2021.
- [40] P. H. Sneath and R. R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W.H. Freeman, 1973.
- [41] D. Müllner, “Modern hierarchical, agglomerative clustering algorithms,” *ArXiv*, vol. abs/1109.2378, 2011. [Online]. Available: <https://api.semanticscholar.org/CorpusID:8490224>
- [42] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, “Tree of thoughts: Deliberate problem solving with large language models,” *NeurIPS*, vol. 36, 2024.
- [43] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” in *FSR*, 2018.