

ReMEMbR: Building and Reasoning Over Long-Horizon Spatio-Temporal Memory for Robot Navigation

Abrar Anwar^{1,2}, John Welsh¹, Joydeep Biswas^{1,3}, Soha Pouya¹, Yan Chang¹

Abstract— Navigating and understanding complex environments over extended periods of time is a significant challenge for robots. People interacting with the robot may want to ask questions like where something happened, when it occurred, or how long ago it took place, which would require the robot to reason over a long history of their deployment. To address this problem, we introduce a Retrieval-augmented Memory for Embodied Robots, or ReMEMbR, a system designed for long-horizon video question answering for robot navigation. To evaluate ReMEMbR, we introduce the NaVQA dataset where we annotate spatial, temporal, and descriptive questions to long-horizon robot navigation videos. ReMEMbR employs a structured approach involving a memory building and a querying phase, leveraging temporal information, spatial information, and images to efficiently handle continuously growing robot histories. Our experiments demonstrate that ReMEMbR outperforms LLM and VLM baselines, allowing ReMEMbR to achieve effective long-horizon reasoning with low latency. Additionally, we deploy ReMEMbR on a robot and show that our approach can handle diverse queries. The dataset, code, videos, and other material can be found at the following link: <https://nvidia-ai-iot.github.io/remembr>

I. INTRODUCTION

Robots are increasingly being deployed in a wide variety of environments, including buildings, warehouses, and outdoor settings. During their deployments, robots perceive a range of objects, dynamic events, and phenomena that are challenging to encapsulate within conventional representations like metric or semantic maps. Additionally, these robots exist for long periods of time, typically on the magnitude of hours, but there is currently no way to query the robot on what it has seen over this long period of time. In this work, we address the challenge of efficiently building this long-horizon memory for robot navigation and responding to questions by framing it as a long-horizon video question-answering task. Our system enables robots to respond to free-form questions and to perform actions based on what they have observed.

Existing approaches to spatio-temporal video memory in robotics are constrained by their capacity to handle only short durations, typically limited to 1-2 minutes [1,2]. As the time span increases, the inference time memory requirements grow for transformer-based methods, rendering them impractical for processing arbitrarily long videos. Concurrent work [3] has focused on leveraging extremely large context windows of large language models to answer questions given

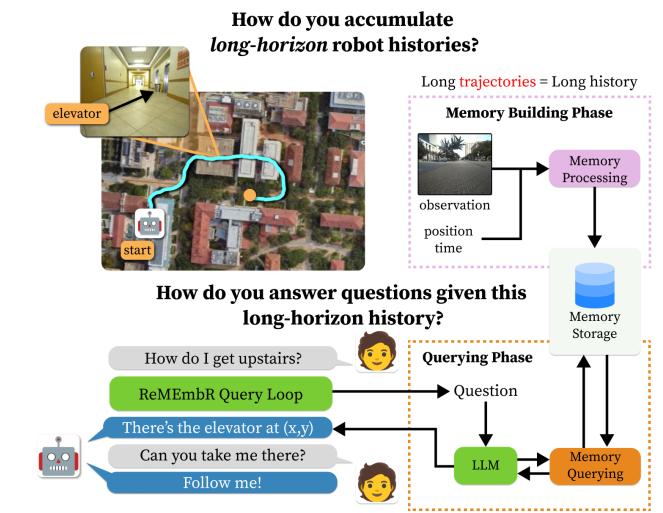


Fig. 1: Robots continuously operate for long periods of time, where they gather long histories. In this work, we investigate how to aggregate these robot histories over time efficiently, and how to utilize that memory representation for answering spatio-temporal questions and generating navigational goals.

a long robot history; however, this is not a scalable solution. No matter the length of the context window, unbounded length histories will not fit in fixed context sizes. In this work, we propose a Retrieval-augmented Memory for Embodied Robots, or ReMEMbR, which uses a retrieval-based LLM-agent capable of querying memory across arbitrary lengths by formulating text-based, spatial, and temporal queries. As shown in Figure 1, ReMEMbR consists of a memory-building phase and a querying phase.

Episodic memory in robotics has been framed mostly as a question-answering task, where systems are evaluated based on their ability to answer questions from a given video [4,5]. While useful for assessing QA capabilities, the text-based outputs of these systems may fall short of providing actionable information for a navigation robot.

For example, a question like “Where did you see my phone?” might yield a response such as “I saw it on the coffee table.” While informative, this answer does not translate into actionable data for the robot. Our work, therefore, also incorporates reasoning over explicit spatial (e.g., xy positions) and temporal (e.g., “10 minutes ago”) information.

To evaluate our system, we construct the Navigation Video Question Answering dataset NaVQA where methods must output position, temporal information, or free-form text. Our

¹ NVIDIA, ² University of Southern California, ³ University of Texas at Austin. This work was done while Abrar Anwar was an intern at NVIDIA. Contact: abrar.anwar@usc.edu, {jwelsh, jbiswas, spouya, yachang}@nvidia.com

dataset consists of 210 questions sampled from subsets of 7 long-horizon navigation videos. This dataset is intended to foster further research in long-horizon memory building and reasoning for navigation robots.

In particular, we

- design the NaVQA dataset for evaluating 1) whether a robot had seen events or objects over the course of its deployment, 2) when it saw certain events or objects, 3) where they happened, and 4) how to reason about these spatio-temporal aspects to answer questions;
- introduce ReMEMbR, a retrieval-augmented LLM-agent capable of forming function calls to retrieve relevant memories and answer questions based on a real-time memory-building process;
- provide qualitative results on a real-world deployment of ReMEMbR on a robot, testing whether ReMEMbR is able to reason over its long-horizon deployment.

II. RELATED WORK

Embodied question answering. Embodied Question Answering (EQA) [4,6–10] is an extension of video question answering to egocentric, and possibly interactive, environments, requiring agents to navigate and gather information to answer questions. Most similar to the question answering ability of our work is OpenEQA [7], which answers questions about what a robot has seen. However, their questions consider only a short 30-second memory. This formulation falls short when applied to robotics scenarios that involve extended time horizons and continuous interaction with the environment. In our work, we focus on answering questions and generating navigational goals on longer lengths of history and leverage robot-centric data such as position and time.

Language and navigation. Classical navigation typically uses metric maps and does not focus on navigating to semantic goals. Most recent work in vision-and-language navigation [11–15], object-goal navigation [16–19], and various forms of language-guided navigation [20–22] focus on navigating in unseen spaces. These works focus more on exploration; however, robots typically are deployed for extended periods of time in the same area. Forms of memory such as scene graphs [23,24], topological memory [25,26], or queryable map representations [1,27,28] may also allow for semantic goal generation, but may fall short in answering questions about a robot’s experience over time about dynamic, non-static objects. As such, our work uses a robot’s video to capture these details over a robot’s deployment.

MobilityVLA [3] is a concurrent work where a long-horizon robot video tour is given to the 1M length context window of a Gemini LLM from which the robot must generate a topological goal. In this work, we solve a more general problem of answering spatial, temporal, and descriptive questions while also generating metric navigation goals. Additionally, simply increasing the context window length is not scalable to unbounded history lengths. Using retrieval-based methods, our approach can scale better to long histories.

Large language models and robotics. Recent years have seen advancements in large language models (LLMs) and vision-and-language models (VLMs), significantly expanding their capabilities across various tasks [29,30]. Prompting techniques such as chain-of-thought [31] and others [32,33] has further enhanced LLMs’ problem-solving abilities, enabling more complex reasoning. Retrieval-augmented generation [34,35] and LLM-agents [36–39] allow the LLM to leverage external information to provide further context to the LLM. In robotics, past work have used the reasoning ability of LLMs for task planning [40–43], generating plans as code [44–46], or to generate navigational goals [3,20]. Rather than focusing on planning, our work focuses open horizon perception, and builds an LLM-agent to enable scalable multi-step reasoning over long-horizon robot histories.

III. PROBLEM FORMULATION

We formulate our problem as a variation of a long-horizon video question answering task for robots. Unlike standard video question answering, robots are deployed for K minutes and actively accumulate a history $H_{1:K}$ from various sensors. Due to their continuous deployment, the size of the history is monotonically increasing over time. Thus our work focuses on two problems: efficiently building a representation this long history $H_{1:K}$ over time and then querying the representation to answer questions and generate navigational goals.

To efficiently build a memory, we consider a history of images H_I , positions H_P , and timestamps H_T . We assume that the robot has localization capabilities, such as using LIDAR-based localization, GPS, or odometry information to provide metric coordinates. After a memory representation is built, a user asks the robot a question Q about spatial, temporal, or descriptive information which the robot may have seen. Specifically, our goal is to predict an answer A given the history and a question $p(A|Q, H)$.

Questions. Robots need to localize information in their histories; however, we focus on making this information actionable. For **spatial** questions such as “Where is closest bathroom?”, the robot must reason about all the bathrooms and signs for bathrooms it has seen. Then, the system must provide the specific (x,y) location to go to the closest bathroom. By formulating spatial questions with coordinates, robots can act on this information to navigate to these goals.

Users may also want to query how long ago an event had occurred or understand how long a robot has done a task. Thus, we define two types of **temporal** questions: point-in-time questions and duration questions. Point-in-time questions such as “When did you see the boxes fall?” with the answer “15 minutes ago” refer to a specific point-in-time relative to the present. Duration questions focus on the length of an activity such as “How long were you inside the building for?” with the answer “10 minutes”. These temporal questions allow robots to retrospectively consider their previous actions.

Lastly, **descriptive** questions ask about the environment, activities the robot may have seen, or the robot’s state in the

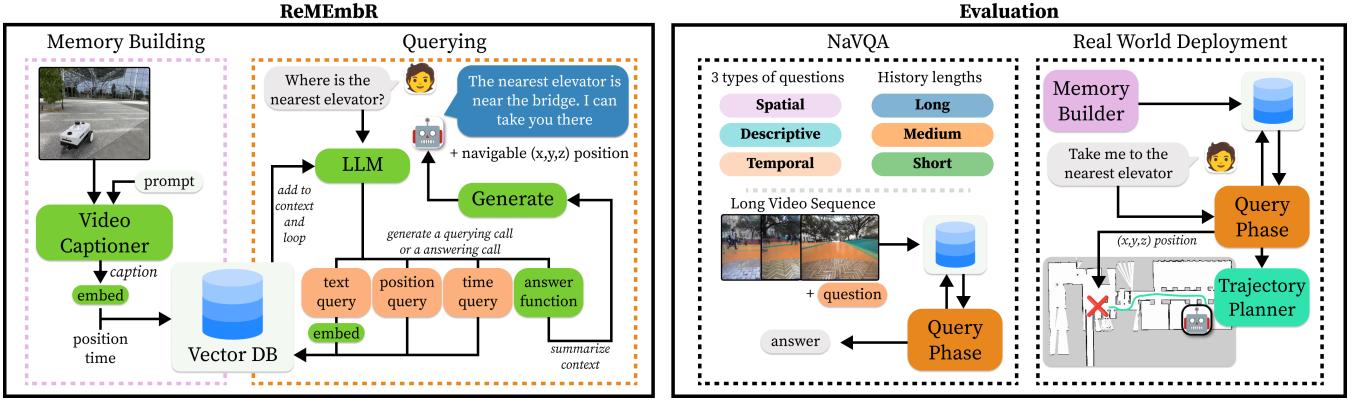


Fig. 2: (Left) We design ReMEmBR with a memory building phase and a querying phase. The memory building phase runs a VILA [47] video captioning model, embeds the caption, then stores the caption embedding, position, and time vectors into a vector database. Then, when a user asks a question, a vector database querying loop starts with an LLM. (Right) Then, we evaluate ReMEmBR on the NaVQA dataset which we construct. NaVQA consists of three types of questions as shown above. Then we deploy ReMEmBR on a robot.

past. This general category can be yes or no questions such as “Was the sidewalk busy today?” or be more descriptive like “What side of the street are you driving on?” These descriptive questions ensure that our robots can effectively remember pertinent details that users ask for.

To capture these questions, we build the NaVQA dataset. We then design ReMEmBR as a step towards solving this task.

IV. REMEMBR

Since robots are embodied and continually persist in the environment, we decompose the task into two distinct phases: memory building and querying.

The computation of $p(A|Q, H_{1:K})$ is often difficult, as long histories are computationally expensive for Transformer-based models or can lead to forgetting in state-space models such as LSTMs. We note that for a given question, a large history is often not required to provide a correct answer. Instead, only a subset of the history $R \subseteq H_{1:K}$ is needed.

Therefore, we can compute the answer given an optimal history subset $R^* \subseteq H_{1:K}$. In practice, we cannot compute R^* and must sample an R such that it contains the same information as R^* . To do so, we build a memory representation V that is sampled using $F : V \rightarrow R$, where $F(V) = \{h | h \in H_{1:K}\}$. We decompose the problem as follows:

$$p(A|H_{1:K}, Q) = p(A|R^*, Q) \approx p(A|R, Q), \quad (1)$$

s.t. $R \sim F(V)$. Then, our goal is to estimate R^* such that the answers derived from R and H are consistent. To do so, we must minimize the size of R while ensuring that the answer can be predicted from both the history H and the subset R :

$$\begin{aligned} R^* &= \operatorname{argmin}_R |R| \\ \text{s.t. } \operatorname{argmax}_A p(A|R, Q) &= \operatorname{argmax}_{A'} p(A'|H, Q) \end{aligned} \quad (2)$$

Using a memory representation V and a sampling strategy F makes the computation more tractable given a long history. Next, we detail how ReMEmBR aggregates the memory representation V during a memory building phase and how it samples $R \sim F(V)$ during a querying phase.

Memory Building. As robots aggregate information over time, we define the queryable memory representation V as a vector database. Vector databases are commonly used to store millions of vector embeddings and search efficiently through them using quantized approximate nearest neighbor methods. Since these databases are efficient in search, we use a vector database to store time, position, and visual representations.

Robots perceive static objects, scenes, and dynamic events, over the course of their deployments. We would like to note that the memory representation V must be constructed without knowing the question Q in advance, and thus must be general enough for any potential question. As the robot is moving in real-time, we aggregate t seconds of image frames $H_{I_{i:i+t}}$ to compute an embedding representation for that segment of memory. We use video captioning using VILA [47] over each consecutive t -second segment, which generates a caption for each temporal segment $L_{i:i+t}$. These captions capture low-level details of what the robot sees over time, which we then embed using a text embedding function E . We use the mxbai-embed-large-v1 [48] embedding model to embed the captions. Over time, the robot adds the vector representation of the text captions, the position, and the timestamps $E(L_{I_{i:i+t}}), H_{P_{i:i+t}}, H_{P_{i:i+t}}$ into the vector database V .

Querying With the vector database V in place, the querying phase can begin. To gather a history subset R , we use an LLM-agent as the sampling function F to sample the database V .

The LLM-agent acts as a state machine that iteratively calls the LLM as shown in Figure 2. Our approach begins with a retrieval node which queries the vector database in three different ways, using position, timestamp, or text

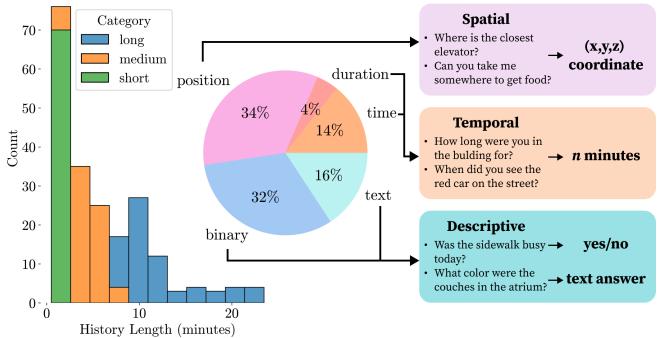


Fig. 3: We introduce the NaVQA dataset, which is composed of 210 examples across three different time ranges up to 20 minutes in length. The dataset consists of spatial, temporal, and descriptive questions, each of which has different types of outputs as shown above.

embeddings. The LLM considers the current set of memories $R_{0:i}$ and the question Q to generate a function call f and a query q which retrieves m memories. Each memory contains position, time, and caption information to be used as further context. These m retrieved memories are then added into R :

$$R_{i:i+m} = f(q), \text{ where } q = LLM(R_{0:i}, Q).$$

We define three functions which the LLM could call:

- **Text retrieval:** $f_t(\text{object}) \rightarrow m$ memories
- **Position retrieval:** $f_p((x, y, z)) \rightarrow m$ memories
- **Time retrieval:** $f_t(\text{"HH:MM:SS"}) \rightarrow m$ memories

At each iteration, the LLM can formulate up to k queries that may help it answer the question. Once $k \times m$ memories are retrieved, the LLM assesses whether the question can be answered with the updated context. If the question is not answerable, the LLM uses the current context and executes the querying phase again retrieve new memories. If the question is answerable, the LLM summarizes any relevant information, and then generates an answer given the entire based on all the retrieved memories. The output is formatted as a JSON with keys for text, position, time, or duration answers. This structured output ensures simple evaluation on NaVQA and makes it easy to generate goals for a robot deployment.

V. DATASET

We introduce the NaVQA dataset, a long-horizon navigation video question answering dataset built on top of the CODa robot navigation dataset [49]. As described in the previous section, this dataset is annotated with spatial, temporal, and descriptive questions and answers. We use these questions to evaluate models' ability to handle robot-centric long-horizon reasoning. We are excited for the robot learning community to leverage this robot-centric QA dataset to improve the long-horizon reasoning capability of robots.

CODa Dataset. The CODa dataset is a large urban navigation dataset consisting of long-horizon sequences in indoor and outdoor settings on a university campus. The

dataset was collected using a Clearpath Husky robot [50], where the robot navigated during the morning, afternoon, and evening. This data is also realistic to what outdoor robots may encounter, with sunny, cloudy, low-light, and rainy sequences. Though the dataset provides various sensor information such as LIDAR, GPS, LIDAR, and multiple cameras, we consider only the GPS coordinate and a front-facing camera. We select 7 of the 23 sequences in the CODa dataset for building the NaVQA dataset. Each sequence ranged in length from 15 to 30 minutes.

Data Annotation. We are interested in how varying the length of a robot trajectory may impact the question answering ability of a system. In our work, we subsample the 7 sequences into three length-based categories: less than 2 minutes (*short*), between 2 and 7 minutes (*medium*), and longer than 7 minutes (*long*) segments. For each sequence, we subsample 10 segments of each length category, which we then provide to annotators to design questions and answers for. This process leads to 30 questions per sequence, for a total of 210 total questions. As these videos are long and require an understanding of robot perception, we recruited 5 robot experts to annotate spatial, temporal, and descriptive questions.

Data Statistics. The NaVQA dataset consists of five types of question outputs: binary yes/no questions (32%), point-in-time questions (14%), duration questions (4%), spatial position questions (34%), and descriptive text questions (16%). Figure 3 depicts the distribution over time of the videos and examples of questions. These questions focused on spatial understanding, object detection, sign reading, dynamic event understanding, and contextual reasoning.

VI. EXPERIMENTAL SETUP

We use NaVQA to evaluate the ability of ReMEmbR and other LLM-based approaches.

Methods. ReMEmbR uses a retrieval module to aggregate relevant parts of the long-horizon history. We show the ability of ReMEmbR with a closed-source LLM (GPT-4o), various open-source LLMs (Codestral [51], Command-R [52]), and a smaller 8 billion parameter Llama3.1 [53] model. ReMEmbR uses up to 3 retrieval steps to construct R . We compare these models to using GPT-4o with all the captions provided at once and a version using frames sampled at 2 FPS from the video itself. For captioning, we use the VILA1.5-13b over 3 seconds of video, leading to 2 FPS.

Metrics. The NaVQA dataset consists of four types of answers, for which we compute different metrics. To unify each of these types of metrics into one metric and reduce the impact of outliers, we threshold the temporal and spatial metrics to determine whether an instance is correct or not to create an **Overall Correctness** metric.

- Spatial questions output (x,y,z) coordinates, from which we compute an L2 distance. We define a spatial question to be correct if it is within 15 meters of the goal.
- Temporal point-in-time and duration questions produce answers such as “15 minutes”, for which we compute

Method	LLMs	Descriptive Question Accuracy ↑			Positional Error (m) ↓			Temporal Error (s) ↓		
		Short	Medium	Long	Short	Medium	Long	Short	Medium	Long
Ours	GPT4o	0.62±0.5	0.58±0.5	0.65±0.5	5.1±11.9	27.5±26.8	46.25±59.6	0.3±0.1	1.8±2.0	3.6±5.9
	Codestral	0.25±0.4	0.24±0.4	0.11±0.3	151.3±109.7	189.0±109.6	212.4±121.3	4.8±5.6	8.4±6.8	14.8±7.5
	Command-R	0.36±0.5	0.32±0.5	0.14±0.3	158.7±129.6	172.2±119.4	188.7±107.1	4.5±17.3	14.3±6.7	15.3±11.7
	Llama3.1:8b	0.31±0.5	0.33±0.5	0.21±0.4	159.9±123.2	151.2±121.1	165.3±115.1	9.5±27.5	7.9±16.3	18.7±10.8
LLM with Caption	GPT4o	0.57±0.5	0.66±0.5	0.55±0.5	5.1±8.2	33.3±47.3	56.0±61.7	0.5±0.5	1.9±2.2	8.0±6.7
Multi-Frame VLM	GPT4o	0.55±0.5	X	X	7.5±11.4	X	X	0.5±2.2	X	X

TABLE I: **Results.** We compare ReMEMbR to an approach that processes all captions at once and another that processes all frames at once. We find that GPT4o-based approaches perform the best, and that ReMEMbR outperforms the LLM-based method and remains competitive to the VLM-based approach on the Short videos. The Medium and Long videos are too long for the VLM to process, and thus is marked with an X.

L1 temporal error. We define a temporal question to be correct if it is within 2 minutes of the goal.

- Descriptive questions produce either yes/no or textual answers, for which we compute a binary accuracy. This accuracy also determines correctness.

To make evaluation faster, text answers are evaluated by an LLM to be correct or not, similar to other work [7].

All ReMEMbR experiments are run over three seeds while the baseline results are over one seed due to cost. Since seeds are not as reproducible, we micro-average the results across all seeds. The variance is high due to the differences in difficulty between questions.

VII. RESULTS

ReMEMbR performs strongly given a long-horizon memory at a lower latency. As shown in the results in Table I, ReMEMbR improves performance on long-horizon tasks compared to traditional LLM methods. For long-duration videos, ReMEMbR using GPT4o achieves better descriptive question accuracy, positional error, and temporal error compared to the LLM with captions and Multi-Frame VLM baselines. ReMEMbR performed similarly to the VLM for short category; however, the VLM is unable to process the long videos and most of the medium length videos.

ReMEMbR scales to longer videos with higher overall correctness. Figure 4 shows the overall correctness over time.

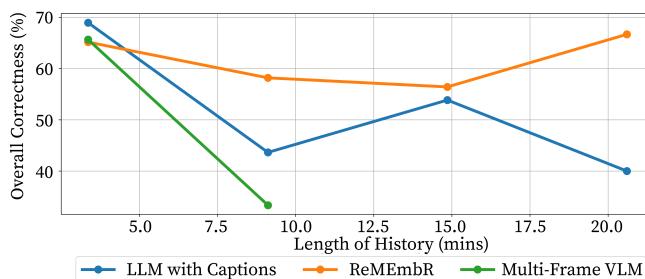


Fig. 4: **Overall correctness over time.** We discretize time into 4 bins and average overall correctness scores in each. Note that although the Medium category in Table I is incomplete, some test instances did complete. We find that ReMEMbR is more correct as the amount of time increases.

Although ReMEMbR does not have the highest performance for short videos compared to the VLM and LLM with captions, ReMEMbR is able to maintain a higher overall correctness score as the video length scales to be longer.

ReMEMbR performs with low latency. We found that for a 21.5 minute video, ReMEMbR takes approximately 25 seconds per question, while the VLM took around 90 seconds per question for a shorter 5.5 minute video. In fact, since ReMEMbR only calls retrieval functions, the amount of time to answer a question remains relatively static regardless of the video duration. Despite their lower performance, we also note that Command-R and Codestral running on a local desktop takes around 40 seconds, while the smaller Llama3.1-8b takes around 15 seconds.

Open-source LLMs perform worse than GPT-4o. As shown in Table I, we found that LLMs trained specifically for code or function calling work well for generating queries. However, our results imply that these LLMs struggle largely with arithmetic reasoning required for answering temporal and spatial questions, leading to lower performance.

Longer caption lengths hurt performance. We captioned with VILA1.5-13b during memory building by passing the model 6 frames for every 3 seconds of accumulated video, effectively operating at 2 FPS. We chose 6 frames as this is the max number of frames VILA can process. To evaluate the effect of frame rate, we also tested a lower rate of 6 frames every 12 seconds, or 0.5 FPS. We observed that captioning at this reduced frame rate led to a drop in performance, likely due to information loss from the coarser sampling.

Different sizes of captioning models slightly reduces performance. As shown in Table II, using the 13b captioning model performs slightly better than smaller 8b and 3b models with respect to overall correctness. The minimal performance loss for using the 3b model is important as smaller models have a higher throughput when deployed on a robot.

Iterative function calls are required for good performance. ReMEMbR uses up to three iterations to find the answer. We found that with only one iteration, which is similar to traditional retrieval-augmented generation, overall correctness decreases. This is likely due to some questions requiring multi-step reasoning, or if the first retrieval did not provide relevant information, ReMEMbR can try again.

LLMs	Overall Correctness ↑		
	Short	Medium	Long
ReMEmbR	0.72±0.5	0.56±0.5	0.61±0.5
- 1 call only	0.67±0.5	0.48±0.4	0.50±0.5
- 12-sec captions	0.54±0.5	0.50±0.5	0.38±0.5
- Llama-VILA1.5-8b	0.58 ±0.5	0.52 ±0.5	0.54±0.5
- VILA1.5-3b	0.60±0.5	0.58±0.5	0.50±0.5

TABLE II: **Ablations.** We provide various ablations of different components of ReMEmbR. We find that the iterative querying process, 3-second captions, and the size of the captioning model are important components to making ReMEmbR work.

VIII. REAL WORLD DEPLOYMENT

Though NaVQA is useful for prototyping and validating new methods, it is important to deploy such methods on robots. In this section, we demonstrate that ReMEmbR can also be deployed in real time on a robot in the real world.

Robot Deployment. We deploy ReMEmbR on a Nova Carter robot [54]. We run the memory building phase on Jetson Orin 32GB, and use GPT-4o as the LLM backend for the ReMEmbR agent. We run a quantized version of VILA-3b to aggregate captions over time. We use ROS2’s Nav2 stack with AMCL for computing localization over a pre-mapped metric map. We run a Whisper automatic speech recognition model [55] that was optimized for a Jetson to enable interaction with ReMEmbR. VILA-3b, Whisper, the Nav2 stack with 3D LiDAR, and the vector database querying runs on-device. In the code release, we will provide various LLM backends such as cloud-based LLMs like NVIDIA NIM APIs [56] or OpenAI APIs, local large LLMs like Command-R that can run on a local desktop, and smaller function-calling LLMs that can run on-device. We hope that our code release can enable researchers to build and query long-horizon robot histories across arbitrary embodiments.

Qualitative results. We deployed the system in a large office space by first building a memory by driving the robot around for 25 minutes. Then we began querying the robot with various navigation-centric questions. We found that our robot was able to execute tasks such as “Where can I get some chips” where the robot took the user to a cafeteria shelf that contained chips. In contrast to searching for specific objects, we also found that our system can guide users to more general areas such as food courts if asked about food or drinks. Our system can also handle more vague questions. We asked the robot to “Take me somewhere with a nice view”, and observed the function calls looking for tall glass windows, plants, and open spaces. Then the robot navigated to a lobby with large glass windows and greenery. We also found that for questions such as “Take me to the soda machine”, the robot would go to a water fountain, as it was captioned as a “silver machine”. This is likely an artifact of using a quantized 3B captioning model that was unable to caption the water fountain properly.

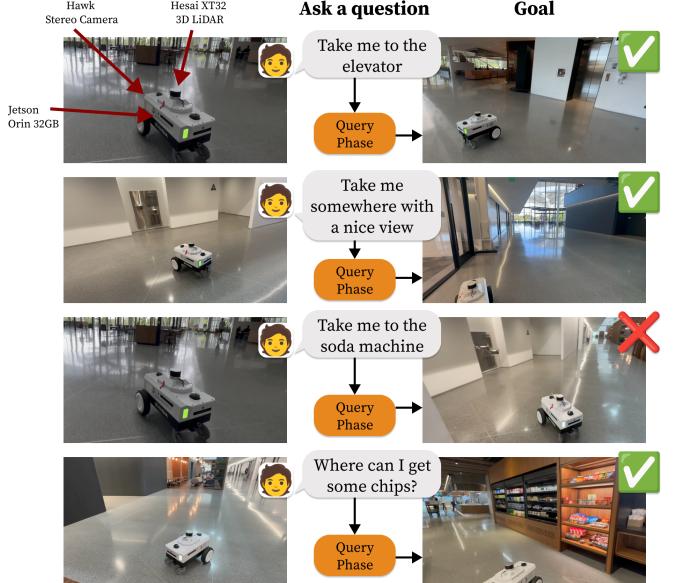


Fig. 5: **Robot deployment.** We deploy ReMEmbR on a Nova Carter robot. We run the memory building phase for 25 minutes, and then begin to ask navigation-centric questions. The robot successfully handles various instructions, including those with more ambiguous instructions such as going to somewhere with a nice view. However, we found that ReMEmbR often confuses some objects such as soda machines and water fountains, leading to incorrect goals.

IX. CONCLUSION

In this work, we introduced ReMEmbR, a system designed to address the challenge of long-horizon video question answering for robots. By decomposing the task into a memory building phase using a VLM and a vector database then a querying phase with an LLM-agent, ReMEmbR efficiently handles the extensive histories that robots accumulate over time. This approach makes it feasible for robots to leverage long-term memory in dynamic and complex environments.

Limitations and Future Work. While NaVQA ensures a unique answer for each question, real-world deployments often involve situations where multiple potential answers could be valid, which would require more focus on contextual reasoning. Additionally, our memory-building approach relies solely on video captioning. However, real-world environments contain rich spatial information such as room numbers, equipment labels, and other details that could be manually annotated. Semantic maps, scene graphs, and queryable scene representations can also provide useful spatial information. We hope to integrate other kinds of memory as function calls so that the ReMEmbR agent can reason spatially and contextually across a broader range of information. A limitation of our approach is that it constantly adds potentially repetitive information into the vector database which would dilute useful information over time. We believe that efficient memory aggregation of pertinent information is an interesting area of future research.

REFERENCES

- [1] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler, “Open-Vocabulary Queryable Scene Representations for Real World Planning,” *International Conference on Robotics and Automation (ICRA)*, 2023.
- [2] P. Sermanet, T. Ding, J. Zhao, F. Xia, D. Dwibedi, K. Gopalakrishnan, C. Chan, G. Dulac-Arnold, S. Maddineni, N. J. Joshi, P. Florence, W. Han, R. Baruch, Y. Lu, S. Mirchandani, P. Xu, P. Sanketi, K. Hausman, I. Shafran, B. Ichter, and Y. Cao, “RoboVQA: Multimodal Long-Horizon Reasoning for Robotics,” *International Conference on Robotics and Automation (ICRA)*, 2024.
- [3] H.-T. L. Chiang, Z. Xu, Z. Fu, M. G. Jacob, T. Zhang, T.-W. E. Lee, W. Yu, C. Schenck, D. Rendleman, D. Shah *et al.*, “Mobility VLA: Multimodal Instruction Navigation with Long-Context VLMs and Topological Graphs,” *arXiv preprint arXiv:2407.07775*, 2024.
- [4] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, “Embodied Question Answering,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [5] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu *et al.*, “Ego4D: Around the World in 3,000 Hours of Egocentric Video,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [6] A. Das, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, “Neural Modular Control for Embodied Question Answering,” *Conference on Robot Learning (CoRL)*, 2018.
- [7] A. Majumdar, A. Ajay, X. Zhang, P. Putta, S. Yenamandra, M. Henaff, S. Silwal, P. Mcvay, O. Maksymets, S. Arnaud *et al.*, “OpenEQA: Embodied Question Answering in the Era of Foundation Models,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [8] A. Z. Ren, J. Clark, A. Dixit, M. Itkina, A. Majumdar, and D. Sadigh, “Explore Until Confident: Efficient Exploration for Embodied Question Answering,” *arXiv preprint arXiv:2403.15941*, 2024.
- [9] J. Thomason, D. Gordon, and Y. Bisk, “Shifting the Baseline: Single Modality Performance on Visual Navigation & QA ,” *North American Association for Computational Linguistics (NAACL)*, 2019.
- [10] E. Wijmans, S. Datta, O. Maksymets, A. Das, G. Gkioxari, S. Lee, I. Essa, D. Parikh, and D. Batra, “Embodied Question Answering in Photorealistic Environments with Point Cloud Perception,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [11] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, “Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [12] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, “Vision-and-Dialog Navigation,” *Conference on Robot Learning (CoRL)*, 2020.
- [13] G. Zhou, Y. Hong, and Q. Wu, “NavGPT: Explicit Reasoning in Vision-and-Language Navigation with Large Language Models,” *AAAI Conference on Artificial Intelligence*, 2024.
- [14] J. Krantz, S. Banerjee, W. Zhu, J. Corso, P. Anderson, S. Lee, and J. Thomason, “Iterative Vision-and-Language Navigation,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [15] Z. Gan, L. Li, C. Li, L. Wang, Z. Liu, J. Gao *et al.*, “Vision-Language Pre-Training: Basics, Recent Advances, and Future Trends,” *Foundations and Trends in Computer Graphics and Vision*, 2022.
- [16] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, “Object Goal Navigation Using Goal-Oriented Semantic Exploration,” *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.
- [17] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, “Cows on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [18] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, “ZSON: Zero-Shot Object-Goal Navigation Using Multimodal Goal Embeddings,” *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- [19] R. Ramrakhy, E. Undersander, D. Batra, and A. Das, “Habitat-Web: Learning Embodied Object-Search Strategies from Human Demonstrations at Scale,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [20] D. Shah, B. Osiński, S. Levine *et al.*, “LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action,” *Conference on Robot Learning (CoRL)*, 2022.
- [21] D. Shah, M. R. Equi, B. Osiński, F. Xia, B. Ichter, and S. Levine, “Navigation with Large Language Models: Semantic Guesswork as a Heuristic for Planning,” *Conference on Robot Learning (CoRL)*, 2023.
- [22] V. S. Dobrala, G. Sigurdsson, R. Piramuthu, J. Thomason, and G. S. Sukhatme, “CLIP-Nav: Using CLIP for Zero-Shot Vision-and-Language Navigation,” *Workshop on Language and Robot Learning (LangRob) @ CoRL*, 2022.
- [23] J. Wald, H. Dhamo, N. Navab, and F. Tombari, “Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [24] X. Li, D. Guo, H. Liu, and F. Sun, “Embodied Semantic Scene Graph Generation,” *Conference on Robot Learning (CoRL)*, 2022.
- [25] B. Eysenbach, R. R. Salakhutdinov, and S. Levine, “Search on the Replay Buffer: Bridging Planning and Reinforcement Learning,” *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [26] N. Savinov, A. Dosovitskiy, and V. Koltun, “Semi-Parametric Topological Memory for Navigation,” *International Conference on Learning Representations (ICLR)*, 2018.
- [27] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, “Scene Memory Transformer for Embodied Agents in Long-Horizon Tasks,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [28] N. M. M. Shafullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, “CLIP-Fields: Weakly Supervised Semantic Fields for Robotic Memory,” *arXiv preprint arXiv:2210.05663*, 2022.
- [29] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambo, F. Azhar *et al.*, “Llama: Open and Efficient Foundation Language Models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [30] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “GPT-4 Technical Report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [31] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- [32] 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,” *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [33] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le *et al.*, “Least-to-Most Prompting Enables Complex Reasoning in Large Language Models,” *arXiv preprint arXiv:2205.10625*, 2022.
- [34] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” *arXiv preprint arXiv:2312.10997*, 2023.
- [35] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection,” *arXiv preprint arXiv:2310.11511*, 2023.
- [36] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, “Reflexion: Language Agents with Verbal Reinforcement Learning,” *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [37] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “React: Synergizing Reasoning and Acting in Language Models,” *arXiv preprint arXiv:2210.03629*, 2022.
- [38] J. S. Park, J. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, “Generative Agents: Interactive Simulacra of Human Behavior,” *ACM Symposium on User Interface Software and Technology (UIST)*, 2023.
- [39] T. Schick, J. Dwivedi-Yu, R. Dessi, R. Raileanu, M. Lomeli, E. Hambo, L. Zettlemoyer, N. Cancedda, and T. Scialom, “Toolformer: Language Models Can Teach Themselves to Use Tools,” *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [40] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” *Conference on Robot Learning (CoRL)*, 2023.
- [41] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, “Inner Monologue: Embodied Reasoning through Planning with Language Models,” *Conference on Robot Learning (CoRL)*, 2022.
- [42] I. Singh, D. Traum, and J. Thomason, “TwoStep: Multi-agent task planning using classical planners and large language models,” *arXiv preprint arXiv:2403.17246*, 2024.

- [43] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, “LLM+P: Empowering Large Language Models with Optimal Planning Proficiency,” *arXiv preprint arXiv:2304.11477*, 2023.
- [44] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, “ProgPrompt: Generating situated robot task plans using large language models,” *International Conference on Robotics and Automation (ICRA)*, 2023.
- [45] Z. Hu, F. Lucchetti, C. Schlesinger, Y. Saxena, A. Freeman, S. Modak, A. Guha, and J. Biswas, “Deploying and Evaluating LLMs to Program Service Mobile Robots,” *IEEE Robotics and Automation Letters (RAL)*, 2024.
- [46] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, “Code as policies: Language model programs for embodied control,” *International Conference on Robotics and Automation (ICRA)*, 2023.
- [47] J. Lin, H. Yin, W. Ping, P. Molchanov, M. Shoeybi, and S. Han, “VILA: On Pre-Training for Visual Language Models,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [48] S. Lee, A. Shakir, D. Koenig, and J. Lipp, “Open source strikes bread - new fluffy embeddings model,” 2024. [Online]. Available: <https://www.mixedbread.ai/blog/mxbai-embed-large-v1>
- [49] A. Zhang, C. Eranki, C. Zhang, J.-H. Park, R. Hong, P. Kalyani, L. Kalyanaraman, A. Gamare, A. Bagad, M. Esteva *et al.*, “Towards Robust Robot 3D Perception in Urban Environments: The UT Campus Object Dataset,” *IEEE Transactions on Robotics (T-RO)*, 2024.
- [50] Clearpath Robotics, “Husky UGV - Outdoor Field Research Robot,” 2024. [Online]. Available: <https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>
- [51] Mistral AI, “Codestral model card,” 2024. [Online]. Available: <https://huggingface.co/mistralai/Codestral-22B-v0.1>
- [52] Cohere for AI, “Command-model card,” 2024. [Online]. Available: <https://huggingface.co/CohereForAI/c4ai-command-r-v01>
- [53] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, “The Llama 3 Herd of Models,” *arXiv preprint arXiv:2407.21783*, 2024.
- [54] Segway Robotics, “Nova Carter - Complete Robotics Development Platform,” 2024. [Online]. Available: <https://robotics.segway.com/nova-carter/>
- [55] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” in *International Conference on Machine Learning (ICML)*, 2023.
- [56] NVIDIA, “NVIDIA NIM,” 2024. [Online]. Available: <https://docs.nvidia.com/nim/index.html>