3D-Mem: 3D Scene Memory for Embodied Exploration and Reasoning

Yuncong Yang[1,*], Han Yang[2,*], Jiachen Zhou[3], Peihao Chen[1], Hongxin Zhang[1], Yilun Du[4], Chuang Gan[1,5]
[1]UMass Amherst, [2]CUHK, [3]Columbia University
[4]MIT, [5]MIT-IBM Watson AI Lab
yuncongyang@umass.edu

https://umass-embodied-agi.github.io/3D-Mem/

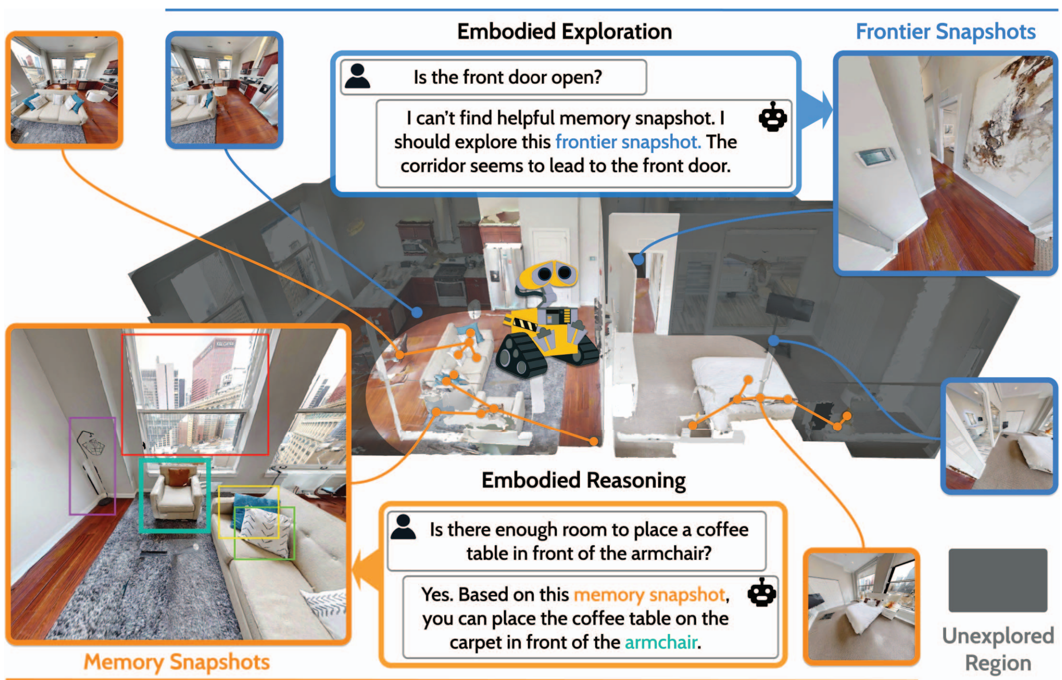

Figure 1. With 3D-Mem, explored regions are represented by a set of Memory Snapshots capturing clusters of co-visible objects, *i.e.*, the objects observable in a single image observation, along with their spatial relationships and background context, as shown in the bottom-left example. Unexplored regions are represented by navigable frontiers along with image observations, referred to as Frontier Snapshots.

Abstract

Constructing compact and informative 3D scene representations is essential for effective embodied exploration and reasoning, especially in complex environments over extended periods. Existing representations, such as object-centric 3D scene graphs, oversimplify spatial relationships by modeling scenes as isolated objects with restrictive textual relationships, making it difficult to address queries requiring nuanced spatial understanding. Moreover, these representations lack natural mechanisms for active exploration and memory management, hindering their application to lifelong autonomy. In this work, we propose 3D-Mem, a novel 3D scene memory framework for embodied agents. 3D-Mem employs informative multi-view images, termed Memory Snapshots, to capture rich visual information of explored regions. It further integrates frontier-based exploration by introducing Frontier Snapshots—glimpses of unexplored areas—enabling agents to make decisions by considering both known and potential new information. To support lifelong memory in active exploration settings, we present an incremental construction pipeline for 3D-Mem, as well as a memory retrieval technique for memory man-

[*]Equal Contribution

agement. Experimental results on three benchmarks demonstrate that 3D-Mem significantly enhances agents' exploration and reasoning capabilities in 3D environments, highlighting its potential for advancing applications in embodied AI.

1. Introduction

Embodied agents operating in complex 3D environments require robust and lifelong scene memory to store rich visual information about the environment, supporting effective exploration and reasoning over extended periods. Currently, there are two main streams of scene representations. The first stream focuses on object-centric representations, particularly 3D scene graphs [8, 36], that represent scenes using nodes for objects and edges for inter-object relationships. Another stream directly uses dense 3D representations, such as point clouds [4, 5, 14, 42] or neural fields [15, 23, 35], filling 3D space with dense visual features for querying.

However, these scene representations exhibit significant limitations when employed as scene memory for embodied agents. Object-centric representations like 3D scene graphs tend to oversimplify 3D scenes by modeling them as individual objects and quantizing inter-object relationships into restrictive textual descriptions. This approach results in the loss of critical spatial information, making it difficult to answer questions that require an understanding of spatial relationships. As illustrated in the bottom-left example from Figure 1, when attempting to answer the question *"Is there enough room to place a coffee table in front of the armchair?"* using a 3D scene graph, the only spatial-related information we can access are the numerical 3D bounding boxes of the objects and the textual descriptions of the spatial relationships among them. It is challenging to measure unoccupied space or determine oriented spatial relationships such as *"in front of"* based solely on this information. On the other hand, dense 3D representations like point clouds or neural fields [10, 12, 38, 40] are computationally expensive and lack scalability as the scene grows during the agent's exploration. Meanwhile, unlike vision-language models (VLMs) and large language models (LLMs) that effectively reason over images and text, current foundation models lack sufficient reasoning capabilities for dense 3D modalities due to limited training data. Additionally, neither type of existing scene representation can model unexplored regions, thus failing to support agents in active exploration, which restrains agents from leveraging their scene memory to expand their knowledge and achieve their goals for embodied tasks.

To address these challenges, we introduce **3D-Mem**, a more capable 3D scene memory for embodied agents. 3D-Mem employs a set of informative multi-view images, termed **Memory Snapshots**, to encompass the explored regions of a 3D scene. Each memory snapshot captures all objects visible in that snapshot along with their surroundings. Our intuition is that a snapshot image alone is sufficient to capture rich visual information of a region. For example, revisiting the challenge illustrated in Figure 1, a memory snapshot clearly shows that there is sufficient space *"in front of"* the armchair to place a coffee table. With the recent advancements in VLMs, these models can extract such spatial information from images [2, 22], much like humans do through intuitive observations.

Building upon the concept of employing multi-view images as 3D scene memory, 3D-Mem integrates with frontier-based exploration frameworks and extends the concept of "frontiers" to **Frontier Snapshots**, which represent glimpses of unexplored regions in a 3D scene. In the scenario depicted in the top-left example from Figure 1, we can observe that the frontier snapshot corresponds to a corridor, which is the most probable path leading to the front door. By maintaining these frontier snapshots, the agent can choose either to complete tasks based on its accumulated knowledge or navigate to unexplored regions for new information, as illustrated in the embodied question-answering tasks in Figure 1. Meanwhile, representing both explored and unexplored regions through multi-view images allows us to better leverage the decision-making capability of VLMs, enhancing the agent's ability to reason and plan effectively.

Additionally, as agents in lifelong settings continuously explore and expand their knowledge, the 3D scene memory system must operate effectively as the memory grows. To achieve this, 3D-Mem supports real-time incremental memory aggregation. To efficiently manage the ever-expanding scene memory, we introduce Prefiltering as an effective memory retrieval mechanism to select relevant memory during decision-making. This framework enables the agent to perform continuous exploration and navigation over extended periods without excessive computational burdens.

Extensive experiments and superior performance on three benchmarks demonstrate that 3D-Mem significantly enhances agents' capabilities in reasoning and lifelong exploration within 3D environments.

Our contributions can be summarized as follows:

- We introduce 3D-Mem, a compact scene memory that constructs informative multi-view snapshot images to capture diverse and robust information among co-visible objects and their surroundings in 3D scenes.
- By introducing frontier snapshots to include unexplored regions, 3D-Mem enables agents to actively explore and acquire new information.
- We incorporate 3D-Mem with incremental memory aggregation and prefiltering strategies that enable agents to expand their knowledge and adapt over extended periods,

supporting lifelong learning in 3D environments.

2. Related Work

3D Scene Representations. Recent works [24, 32, 39, 42] have focused on establishing dense 3D representations by grounding 2D representations captured by Vision-Language Foundation Models (*e.g.* CLIP [26], BLIP [19], SEEM [44]) into 3D scenes, which showcase impressive results on tasks such as open-vocabulary object segmentation and language-guided object grounding [9]. However, such representations are limited due to high resource consumption and the inability to support dynamic updates. 3D scene graphs address these limitations by formulating the scene as a compact graph, where nodes represent objects, and edges encode inter-object relationships as textual descriptions [1, 7, 8, 18, 36], enabling real-time establishment and dynamic update for hierarchical scene representations [13, 29, 37]. While such object-centric representations have demonstrated effectiveness in various tasks, they remain constrained for oversimplifying inter-object relationships with restrictive text descriptions. To tackle this challenge, our work leverages a set of informative memory snapshots to visually capture spatial and semantic relationships among objects, offering a more sophisticated understanding of the scene.

VLM for Exploration and Reasoning. Vision-Language Models (VLMs) have shown promising results in solving embodied exploration and reasoning tasks by leveraging commonsense reasoning and internet-scale knowledge. Existing exploration approaches can be divided into two categories. The former directly employs consecutive observations together with instructions as input, requiring the VLM to predict the next-step action [43] while the latter grounds the exploration target in the 3D scene through visual prompting, establishing a semantic map to guide the exploration process [21, 28, 33, 41]. However, both approaches are constrained by their memory representations. For the former, vanilla past observations can only serve as short-term memory. For the latter, their semantic maps are target-specific and cannot be generalized to future tasks. To address these limitations, our work introduces the first lifelong and target-agnostic scene memory that can be seamlessly integrated with VLM for further reasoning, stepping closer to the ultimate goal of lifelong autonomy.

Topological Mapping. Besides the 3D scene representations mentioned above, prior 2D topological mapping methods like TSGM [17] and RoboHop [6] provide valuable context. They construct navigation graphs from images and objects but do not fully capture all objects or their interrelationships. In contrast, 3D-Mem clusters co-visible objects to represent inter-object spatial relationships, enabling tasks such as embodied Q&A that extend beyond route planning. Moreover, its efficient memory retrieval mechanism scales to lifelong exploration, leveraging VLMs for advanced reasoning and decision-making.

3. Approach

3D-Mem contains two types of snapshots: **Memory Snapshots** and **Frontier Snapshots**. A memory snapshot represents a cluster of objects and their surroundings in the explored regions, while a frontier snapshot represents an unexplored region along with its exploration direction. During exploration, we construct the set of memory snapshots from a stream of RGB-D images with poses and maintain frontier snapshots using frontier algorithms and occupancy maps. Given an objective and the set of memory and frontier snapshots, a VLM agent iteratively selects the most promising frontier snapshot and moves toward it, continually updating both sets of snapshots while making decisions, until the objective is achieved based on the current memory snapshots.

First, in Section 3.1, we introduce how the set of memory snapshots is constructed from a series of RGB-D observations with poses. Next, in Section 3.2, we extend this algorithm to incremental construction in active exploration settings. By integrating frontier snapshots, we enable the agent to actively conduct frontier-based exploration. We also design methods to efficiently retrieve memory as it scales up. Finally, in Section 3.3, we detail how the agent utilizes the constructed memory and frontier snapshots for exploration and reasoning.

Algorithm 1 Co-Visibility Clustering for Memory Snapshots

Initial clusters $\mathcal{C} = \{\mathcal{O}\}$
Memory snapshot set $\mathcal{S} = \varnothing$
All frame candidates $\mathcal{I}$
Score function $\mathcal{F}$
while $\mathcal{C}$ is not empty **do**
  $\mathcal{O}^* = \arg\max_{\mathcal{O} \in \mathcal{C}} |\mathcal{O}|$
  $\mathcal{I}^* = \{I | I \in \mathcal{I}, \mathcal{O}^* \subseteq \mathcal{O}_I\}$
  if $\mathcal{I}^*$ is not empty **then**
    $I^* = \arg\max_{I \in \mathcal{I}^*} \mathcal{F}(I)$
    $S^* = \langle \mathcal{O}^*, I^* \rangle$
    $\mathcal{S} = \mathcal{S} \cup \{S^*\}$
  else
    Use K-Means to split $\mathcal{O}^*$ into two clusters $\mathcal{O}^* = \mathcal{O}_1^* \cup \mathcal{O}_2^*$ based on 2D horizontal positions (x, y)
    $\mathcal{C} = \mathcal{C} \cup \{\mathcal{O}_1^*, \mathcal{O}_2^*\}$
  end if
  $\mathcal{C} = \mathcal{C} - \{\mathcal{O}^*\}$
end while

while exists $S_j, S_k \in \mathcal{S}$ such that $I_{S_j} = I_{S_k}$ **do**
  $\mathcal{S} = \mathcal{S} \setminus \{S_j, S_k\}$
  $S_l = \langle \mathcal{O}_{S_j} \cup \mathcal{O}_{S_k}, I_{S_j} \rangle$
  $\mathcal{S} = \mathcal{S} \cup \{S_l\}$
end while
return $\mathcal{S}$

3.1. 3D-Mem Construction

Inspired by the idea that an image itself inherently contains rich and robust information to represent a small area of a scene, we propose a novel way that utilizes a set of multi-view snapshot images to cover the whole informative areas of a scene. Instead of the object-centric representation proposed by ConceptGraph[8], in which only object-level visual features are stored and managed, we propose using one snapshot image to represent a cluster of co-visible objects within that image, namely a **Memory Snapshot**. This approach not only encapsulates foreground object-level details but also captures contextual room-level cues embedded in the background, providing a more holistic representation of the scene.

3.1.1. Memory Snapshot Formulation

Specifically, given a set of N image observations $\mathcal{I}^{obs} = \{I_1^{obs}, I_2^{obs}, ..., I_N^{obs}\}$, where each I_i^{obs} consists of an RGB-D image and its pose, we aim to construct a set of memory snapshots, a minimal subset of $\mathcal{I}^{obs}$, to cover all salient observed objects. We first follow ConceptGraph [8] to perform a series of object segmentation, spatial transformations, matching, and merging, resulting in an object set that contains all detected objects from the observations $\mathcal{O} = \{o_1, o_2, ..., o_M\}$, where each object is characterized by an object category, detection confidence, and 3D location. Meanwhile, we obtain a set of frame candidates $\mathcal{I} = \{I_1, I_2, ..., I_N\}$, where each $I_i = \langle I_i^{obs}, \mathcal{O}_{I_i}\rangle$ consists of the image observation I_i^{obs} together with a set of all detected objects $\mathcal{O}_{I_i}$, *i.e.*, all co-visible objects in I_i^{obs}.

We define $\mathcal{S}$ as a set of memory snapshots $\{S_1, S_2, ..., S_K\}$ of size $K \leq N$, where each memory snapshot $S_k = \langle \mathcal{O}_{S_k}, I_{S_k}\rangle$ is characterized by a frame candidate $I_{S_k} \in \mathcal{I}$ and a cluster of objects $\mathcal{O}_{S_k}$, a subset of all detected objects $\mathcal{O}_{I_{S_k}}$ in the image $I_{S_k}^{obs}$. Therefore, an image $I_{S_k}^{obs}$ serves as a shared visual feature among $\mathcal{O}_{S_k}$. Since $\mathcal{S}$ needs to cover the whole object set $\mathcal{O}$, and each object o_j needs to be uniquely mapped to one memory snapshot S_k (although it may still be visible in other memory snapshots), we require $\mathcal{O}_{S_1} \cup \mathcal{O}_{S_2} \cup ... \cup \mathcal{O}_{S_K} = \mathcal{O}$, and $\mathcal{O}_{S_i} \cap \mathcal{O}_{S_j} = \varnothing$ for $\forall S_i, S_j \in \mathcal{S}$.

3.1.2. Co-Visibility Clustering

To acquire the desired set of memory snapshots, we designed Co-Visibility Clustering based on the classical hierarchical clustering [30] to split $\mathcal{O}$ into clusters, each of which is a subset of the detected object set $\mathcal{O}_{I_i}$ of a certain frame candidate I_i. As detailed in the pseudocode in Algorithm 1, we define a cluster set $\mathcal{C}$ composed of all unsettled object clusters that haven't been matched with observations, initialized to contain the full object set $\{\mathcal{O}\}$, and the memory snapshot set $\mathcal{S}$, initialized to $\varnothing$. Each time, we pick the largest unsettled cluster $\mathcal{O}^*$ from $\mathcal{C}$ and search through all frame candidates for capable candidates I^* such that $\mathcal{O}^*$ is a subset of the detected object list of I^*. When such candidates exist, we rank them based on a score function $\mathcal{F}$ and pick the top-ranked frame candidate I^* to create a new memory snapshot $S^* = \langle \mathcal{O}^*, I^*\rangle$ and add it to $\mathcal{S}$. In practice, we choose $\mathcal{F}(I_i) = |\mathcal{O}_{I_i}|$ to select the frame candidate that contains the most objects, and when more than one frame candidate has the same largest object number, we choose the one with the highest sum of detection confidence. If there is no feasible frame candidate, we then use K-Means to further divide $\mathcal{O}^*$ into two subclusters $\mathcal{O}_1^*$ and $\mathcal{O}_2^*$ based on the 2D horizontal positions of the objects, and add them to $\mathcal{C}$. We repeat the above process until no clusters remain in $\mathcal{C}$. Note that the process is guaranteed to terminate since every object has been captured in certain observations. Ultimately, after all objects have been assigned to corresponding snapshots, memory snapshots sharing the same frame candidates are merged to achieve the final compact memory representation $\mathcal{S}$.

In each memory snapshot, not only is the visual information of each object stored, but also the spatial relationships between objects and the room-level information are provided by visual cues in the background. With the increasing perception abilities of VLMs, such multi-view images representations can provide richer and more robust visual information for VLMs to complete difficult tasks.

3.2. 3D-Mem with Frontier-based Exploration

We adapt the memory construction algorithm introduced above to the frontier-based exploration pipeline [28], where we introduce frontier snapshot as an extension of frontier (Section 3.2.1). In the frontier-based exploration episode, an agent is initialized in an unknown scene and explores the environment step by step. At each step, the agent receives a series of observations, including rgb, depth, and pose, which are used to update frontier and memory snapshots (Section 3.2.2). As the 3D scene memory grows during exploration, the decision-making becomes computationally heavy. Therefore we introduce Prefiltering as an efficient memory retrieval mechanism (Section 3.2.3).

3.2.1. Introduction of Frontier Snapshot

In a common frontier-based exploration framework, an occupancy map is kept to record the explored regions, defined as the nearby areas along the agent's trajectory, and the unexplored regions, defined as navigable but yet-to-be-explored areas. A frontier is then defined to represent such an unexplored region that could be further explored. In this work, we extend this concept by using a snapshot to represent a frontier, similar to memory snapshots. We define a **Frontier Snapshot** $F = \langle r, p, I^{obs}\rangle$, consisting of the unexplored region r it represents, a navigable location p, and an image observation I^{obs} from the agent's position toward that unexplored region. Therefore, the frontier and memory

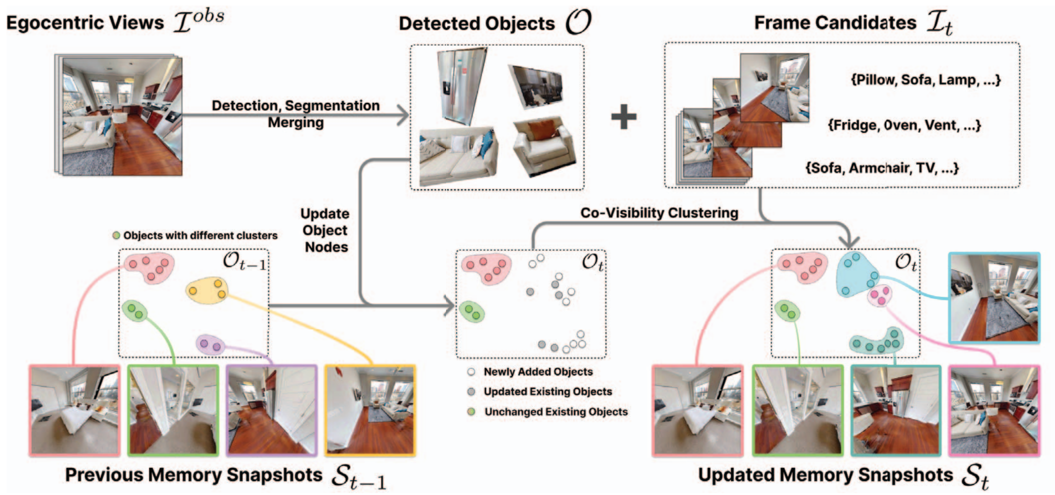

Figure 2. **The memory aggregation process of 3D-Mem.** At each step t, the object set $\mathcal{O}_t$ is first updated using the object-wise update pipeline introduced in Section 3.1. The newly detected objects and the updated existing objects are then jointly clustered into new memory snapshots using co-visibility clustering (Algorithm 1), which are used to update the memory snapshot set $\mathcal{S}_t$.

snapshots, which are all raw images, can be directly used as visual input for the VLMs. More details about frontier-based exploration are provided in Appendix 9.

3.2.2. Incremental Construction of 3D-Mem

Throughout the exploration process, the scene memory is dynamically and incrementally constructed. At each exploration step, the agent observes its surroundings and updates the scene memory and frontiers. At step t, we denote the current object set as $\mathcal{O}_t$, the frontier set as $\mathcal{F}_t$, the memory snapshot set as $\mathcal{S}_t$, and the frame candidate set as $\mathcal{I}_t$, all of which are initialized as $\varnothing$ at the beginning of the episode.

Object Update. As illustrated in Figure 2, at each time step t, the agent first captures N egocentric views $\mathcal{I}^{obs} = \{I_1^{obs}, I_2^{obs}, ..., I_N^{obs}\}$, which are used to extract the object set $\mathcal{O}$ and the frame candidate set $\mathcal{I}$ similar to Section 3.1. Specifically, a threshold "max_dist" is set to ensure only objects within a certain distance from the agent are added to the scene graph, as the memory snapshot should only represent objects from a local area. It is important to note that the object set $\mathcal{O}$ detected in these egocentric views may contain both newly identified objects and those already present in the previous set $\mathcal{O}_{t-1}$. Subsequently, the full object set and frame candidate set are updated as $\mathcal{O}_t = \mathcal{O}_{t-1} \cup \mathcal{O}$ and $\mathcal{I}_t = \mathcal{I}_{t-1} \cup \mathcal{I}$, respectively.

Memory Snapshot Update. We implement the co-visibility clustering in Section 3.1 incrementally. At each time step t, instead of performing clustering on the entire object set $\mathcal{O}_t$, we focus on clustering objects related to $\mathcal{O}$, the objects detected from the egocentric views at this step. In $\mathcal{O}$, some objects may have already been assigned to specific memory snapshots in $\mathcal{S}_{t-1}$. We refer to those memory snapshots as $\mathcal{S}_{prev} = \{S | S \in \mathcal{S}_{t-1}, \mathcal{O}_S \cap \mathcal{O} \neq \varnothing\}$. All objects from $\mathcal{S}_{prev}$, along with the newly detected objects in $\mathcal{O}$, are used as input for clustering, denoted as $\mathcal{O}_{input}$. Then, the memory snapshot set is updated as $\mathcal{S}_t = (\mathcal{S}_{t-1} \setminus \mathcal{S}_{prev}) \cup \text{Cluster}(\mathcal{O}_{input}, \mathcal{I}_t)$

Frontier Snapshot Update. At each step t, an existing frontier from $\mathcal{F}_{t-1}$ may be modified if the unexplored region it represents has been updated, or it may be removed if the region has been fully explored. Additionally, new frontiers may be introduced. For each newly added or modified frontier, a new snapshot is taken to update its image representation. Then, $\mathcal{F}_{t-1}$ is updated to $\mathcal{F}_t$. More implementation details are in Appendix 11.

3.2.3. Memory Retrieval with Prefiltering

During exploration, when we incrementally construct memory snapshots, the large volumes of memory can make the agent's decision-making inefficient and introduce noise unrelated to the objectives. Therefore, we designed a memory retrieval mechanism called **Prefiltering**. Since most memory snapshots are irrelevant to a given instruction and processing them consumes substantial computational resources without meaningful benefit, we present the VLM agent with the question along with all object categories in $\mathcal{O}_t$. The VLM agent then outputs the relevant object categories ranked by importance, retaining only the top K categories determined by a hyperparameter K. Memory snapshots that do not contain any of these K selected categories are filtered out. Figure 3 illustrates Prefiltering in an embodied question-answering task, showing how the original set of memory snapshots is reduced based on relevancy. This technique significantly reduces resource consumption, allowing us to include images directly within the prompt and enhancing the agent's efficiency.

3.3. Exploration and Reasoning with 3D-Mem

With the updated frontier and memory snapshots, we can directly leverage the perception and reasoning capabilities

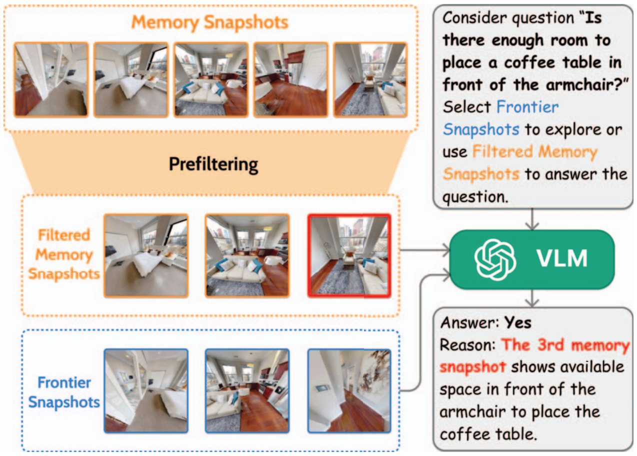

Figure 3. **3D-Mem as visual input for the VLM in embodied question answering.** The VLM first retrieves relevant memory snapshots with prefiltering, then utilizes the frontier snapshots and memory snapshots to perceive the scene and reason about the embodied questions.

of large VLMs, as the image nature of frontier and memory snapshots makes them easily interpretable by VLMs. This is a core advantage of 3D-Mem over other 3D representations such as point clouds, as recent VLMs, trained on internet-scale data, often have stronger visual capabilities than specially fine-tuned models such as 3D-LLM [10]. A pseudo-code overview of the whole exploration and reasoning framework is in Figure 12 in Appendix.

Versatility in Task Applications. 3D-Mem is versatile and can be applied to various tasks. In the case of embodied question answering (illustrated in Figure 3), the VLM agent decides whether to explore a frontier or answer the question based on the current memory snapshots. If the agent chooses a frontier, it provides a rationale for exploring that direction; otherwise, it directly answers the question, concluding the exploration episode. In object navigation tasks, where the agent aims to find a specific object, we augment each memory snapshot with image crops of the objects it contains. The agent then selects an object from one of the memory snapshots to navigate toward. Detailed experiments on these two tasks are presented in Section 4.1 and 4.3, with the complete prompt provided in Appendix 15.

Navigation Strategy. If the VLM agent chooses a frontier F_i to explore, the agent then directly navigates towards the location of the frontier p_i from its current pose. The agent stops either when it has moved a maximum distance of $1.0m$ from its original pose or when it arrives within $0.5m$ of p_i. We assume a collision-free planner is available to guide the agent along the shortest path within the explored region between the two locations. Experimentally, we adopt the pathfinder provided by Habitat-sim to compute such paths and move the agent incrementally at each time step. This strategy is similar to that of ExploreEQA [28], which also uses VLMs to guide exploration.

4. Experiments

As a form of 3D scene memory, 3D-Mem stores rich and compact visual information, serving as an effective framework for a lifelong agent to explore and reason about a 3D scene. To comprehensively evaluate 3D-Mem, we begin with Active Embodied Question Answering (Section 4.1), where the scene is initially unknown. This assessment tests 3D-Mem's overall performance in scenarios that require both embodied exploration and reasoning. Next, we examine 3D-Mem's efficiency in representing 3D scene information through Episodic Memory Embodied Question Answering (Section 4.2). In this evaluation, the scene scan of the ground truth region is provided and no exploration is needed for question answering. Following this, we evaluate 3D-Mem on GOAT-Bench (Section 4.3), a multi-modal lifelong navigation benchmark, to demonstrate 3D-Mem's effectiveness as a lifelong memory system. Finally, we conduct a series of ablation studies to determine key components and hyperparameter choices in Appendix 13.

Since 3D-Mem is a versatile scene memory, we adapt it to different benchmarks in slightly different ways, as detailed in Appendix 10. We also include more discussion and qualitative analysis in Appendix 7.

4.1. Active Embodied Question Answering

On A-EQA [22] benchmark (Table 1), we evaluate 3D-Mem's ability to dynamically construct scene representations for exploration and reasoning given complex questions. More implementation details are in Appendix 10.1.

Method	LLM-Match ↑	LLM-Match SPL ↑
Blind LLMs		
GPT-4*	35.5	N/A
GPT-4o†	35.9	N/A
Question Agnostic Exploration		
CG Scene-Graph Captions*	34.4	6.5
SVM Scene-Graph Captions*	34.2	6.4
LLaVA-1.5 Frame Captions*	38.1	7.0
Multi-Frame*†	41.8	7.5
VLM Exploration		
Explore-EQA†	46.9	23.4
CG w/ Frontier Snapshots†	47.2	33.3
3D-Mem (Ours)†	**52.6**	**42.0**
Human Agent*	85.1	N/A

Table 1. **Experiments on A-EQA.** "CG" denotes ConceptGraphs. Methods with * are reported from OpenEQA [22], and with † are evaluated on the 184-question subset.

Benchmark. A-EQA consists of 557 questions drawn from 63 scenes in HM3D [27]. Due to resource limitations, our evaluation focuses on a subset of 184 questions, as mentioned in the OpenEQA benchmark [22]. We also include the evaluation of 3D-Mem on the full set in Appendix 6.

The open-vocabulary and open-ended questions in A-EQA encompass diverse daily tasks such as object recognition, functional reasoning, and spatial understanding. For each question, an agent is initialized at a specific location and is required to explore the scene to gather the necessary information for answering the question.

Metrics. Following OpenEQA, we employ LLM-Match and LLM-Match SPL for quantitative evaluation. We first rate each predicted answer from 1 to 5 using GPT-4 to compare ground-truth and predicted answers. Given the predicted answers, LLM-Match, which measures the answer accuracy, is calculated as the average score for each question, mapped to a 0-100 scale. LLM-Match SPL, which measures the exploration efficiency, is then calculated by weighting the LLM-Match score by exploration path length. For the questions where the VLM Exploration methods failed to provide an answer, we ask GPT-4o to directly guess an answer without visual inputs, setting the SPL to 0.0.

Baselines. For baselines that use VLM for exploration, we mainly compare 3D-Mem with Explore-EQA [28] and ConceptGraph [8] w/ frontier snapshots. We adapt Explore-EQA for open-ended questions by halting exploration and answering the question with the egocentric view once the VLM's confidence in the question exceeds a predetermined threshold. We integrate ConceptGraph into our exploration pipeline by replacing memory snapshots with object image crops, while maintaining other settings the same, including prefiltering and how answers are obtained. We adopt GPT-4o as the choice of VLM by directly utilizing the OpenAI API. Besides the methods that can do active exploration above, we also include other simple baselines implemented by OpenEQA. The group of question-agnostic exploration baselines employ question-agnostic frontier exploration to obtain an episodic memory of image frames. These frames are subsequently used to prompt VLMs directly (Multi-Frame), generate frame captions as prompts for LLMs (LLaVA-1.5 Frame-Captions), or construct textual scene-graph representation using ConceptGraph (CG) and Sparse Voxel Map (SVM) to prompt LLMs. Additionally, blind LLM experiments are included, where the LLM is tasked with answering questions without any visual information. Note that the Multi-Frame baseline uses 75 frames for each question, and is evaluated on the 184-question subset. Other baselines from OpenEQA are evaluated on the full 557-question set.

Analysis. As shown in Table 1, 3D-Mem significantly outperforms previous methods in both accuracy and efficiency. The major takeaways are as follows: 1) The superior performance in open-ended embodied question answering highlights the advantages of using snapshots as a memory format, which store richer and more flexible visual information for the VLM to address complex questions. In contrast, object-based memory systems, which use either image crops or language captions to represent objects and spatial relationships, are less robust when handling diverse questions, as they rely on rigid object-level features. 2) The multi-frame VLM implemented by OpenEQA also achieves inferior results, despite using a similar multi-view imagesrepresentation, because this baseline linearly selects episodic frames and includes excessive repetitive or irrelevant information for the questions. In contrast, 3D-Mem creates an average of 10.94 memory snapshots from 39.76 egocentric observations after each episode. After prefiltering with $K = 10$, only 3.26 memory snapshots are retained as direct input for VLMs. These results demonstrate the compactness and efficiency of 3D-Mem as scene memory.

Methods	Avg. Frames	LLM-Match
Blind LLM*	0	35.5
CG Captions*	0	34.4
SVM Captions*	0	34.2
Frame Captions*	0	38.1
Multi-Frame	3.0	48.1
3D-Mem (Ours)	3.1	**57.2**
Human	Full	86.8

Table 2. **Experiments on EM-EQA.** Frame Efficiency and performance. Methods denoted by * use GPT-4 to generate answers, as reported in OpenEQA

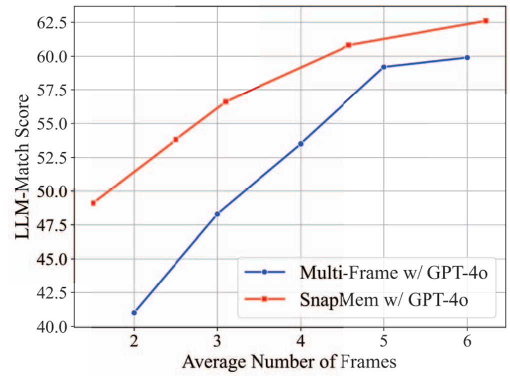

Figure 4. **Frame Efficiency of 3D-Mem on EM-EQA.** LLM-Match Score vs. Average Number of Frames for 3D-Mem and Multi-Frame both using GPT-4o

4.2. Episodic-Memory Embodied Q&A

We evaluate the representation capability of 3D-Mem on EM-EQA [22] to further demonstrate 1) the effectiveness of image memory compared to captions, 2) the compact and informative nature of our method. More implementation details are in Appendix 10.2.

Benchmark. EM-EQA is an Embodied Q&A benchmark that contains over 1600 questions from 152 ScanNet [3] and HM3D[27] scenes. The open-vocabulary and open-ended questions in EM-EQA encompass diverse daily tasks such as object recognition, functional reasoning, and spatial understanding. For each question, a trajectory comprising RGB-D observations and the corresponding camera poses at each step is provided, offering necessary contextual information needed to answer the questions.

Baselines. We compare against language-only scene representations, including ConceptGraphs captions, Sparse Voxel Maps Captions, and Frame Captions. We also compare against Multi-Frame, which directly processes 2 to 6 linearly sampled frames using GPT-4o.

Analysis. From the results, we can observe that: 1) As shown in Table 2, both 3D-Mem and Multi-Frame significantly outperform methods that rely on captions to represent a 3D scene while using only approximately three frames. This demonstrates the effectiveness of using a set of images to represent a 3D scene and highlights the limitations of 3D scene graph captions in addressing complex queries involving relationships between objects. 2) In both Table 2 and Figure 4, 3D-Mem surpasses Multi-Frame in frame efficiency, underscoring the compact and informative nature of our proposed 3D scene memory.

4.3. GOAT-Bench

On GOAT-Bench [16] (Table 3), we evaluate 3D-Mem's effectiveness as a lifelong memory system that facilitates efficient exploration and reasoning. More implementation details are in Appendix 10.3.

Benchmark. GOAT-Bench is a multimodal lifelong navigation benchmark, where an agent is tasked with sequentially navigating to several objects in an unknown scene, with each target described by either a category name (*e.g.*, microwave), a language description (*e.g.*, the microwave on the kitchen cabinet near the fridge), or an image of the target object. Due to the large size of GOAT-Bench and the resource limitations, we assess a 1/10-size subset of the "Val Unseen" split, consisting of one exploration episode for each of the 36 scenes, totaling 278 navigation subtasks. For reference, we also include the evaluation of 3D-Mem on the full test set in Appendix 6.

Metrics. GOAT-Bench employs the Success Rate and Success weighted by Path Length (SPL), similar to A-EQA dataset. A navigation task is deemed success if the agent's final location is within 1 meter from the navigation goal. SPL is the success score weighted by exploration distances.

Baselines. Similar to the experiments in A-EQA, we compare 3D-Mem with Explore-EQA [28] and ConceptGraph [8] baselines. Due to implementation differences in Explore-EQA, we introduce an additional success criterion for this baseline: a subtask is considered successful if the target object is visible in the final observation. This extra criterion leverages ground truth grounding, thereby enhancing the baseline's capability. To demonstrate the effectiveness of 3D-Mem's lifelong memory, we include another baseline (3D-Mem w/o memory) in which we clear the constructed scene graph after each subtask. We also directly include baselines implemented in GOAT-Bench. However, these baselines are simple RNN-based models trained via reinforcement learning, which causes their performance to lag behind the baselines we implemented.

Analysis. As shown in Table 3, 3D-Mem achieves the highest scores compared to previous methods in both accuracy and efficiency. Our major observations are as follows: 1) Even though GOAT-Bench is an object-based navigation benchmark, which is well-suited for ConceptGraph settings, 3D-Mem still outperforms ConceptGraph w/ frontier snapshots. This can be attributed 3D-Mem's multi-view images representation, which captures more comprehensive information, making it easier to match with the diverse descriptions in GOAT-Bench. 2) When compared with the original 3D-Mem, the performance of 3D-Mem w/o memory declines for both GPT-4o and LLaVA-7B models, particularly in efficiency (SPL), indicating that 3D-Mem is beneficial as a memory system for lifelong learning. 3) Explore-EQA, which uses a traditional value map for each subtask to indicate regions of interest, also performs worse, as it lacks the mechanism to memorize information in explored regions. 4) In terms of efficiency, after each episode, 3D-Mem generates an average of 16.58 memory snapshots from 91.37 egocentric observations, and after prefiltering with $K = 10$, only 4.66 memory snapshots are kept for each query.

Method	Success Rate ↑	SPL ↑
GOAT-Bench Baselines		
Modular GOAT*	24.9	17.2
Modular CLIP on Wheels*	16.1	10.4
SenseAct-NN Skill Chain*	29.5	11.3
SenseAct-NN Monolithic*	12.3	6.8
Open-Sourced VLM Exploration		
3D-Mem w/o memory†	40.6	14.6
3D-Mem (Ours)†	49.6	29.4
GPT-4o Exploration		
Explore-EQA†	55.0	37.9
CG w/ Frontier Snapshots†	61.5	45.3
3D-Mem w/o memory†	58.6	38.5
3D-Mem (Ours)†	**69.1**	**48.9**

Table 3. **Experiments on GOAT-Bench.** Evaluated on the "Val Unseen" split. "CG" denotes ConceptGraphs. Methods denoted by * are from GOAT-Bench, and with † are evaluated on the subset.

5. Conclusion

We present 3D-Mem, a 3D scene memory framework that uses a set of informative multi-view snapshot images to store robust visual information of a 3D scene. With the integration of the frontier-based exploration framework, 3D-Mem allows the agent to either leverage the memory of explored regions to solve tasks or further explore the scene to expand its knowledge. With its incremental construction and efficient memory retrieval mechanism, 3D-Mem serves as an effective memory system for lifelong agents. Extensive experiments demonstrate the significant advantages of 3D-Mem over traditional scene representations.

References

[1] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5664–5673, 2019. 3

[2] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024. 2

[3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 7, 3

[4] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7010–7019, 2023. 2

[5] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Lowis3d: Language-driven open-world instance-level 3d scene understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[6] Sourav Garg, Krishan Rana, Mehdi Hosseinzadeh, Lachlan Mares, Niko Suenderhauf, Feras Dayoub, and Ian Reid. Robohop: Segment-based topological map representation for open-world visual navigation. *arXiv*, 2023. 3, 9

[7] Paul Gay, James Stuart, and Alessio Del Bue. Visual graphs from motion (vgfm): Scene understanding with object geometry reasoning. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 330–346. Springer, 2019. 3

[8] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028, 2024. 2, 3, 4, 7, 8

[9] Yining Hong, Yilun Du, Chunru Lin, Josh Tenenbaum, and Chuang Gan. 3d concept grounding on neural fields. *Advances in Neural Information Processing Systems*, 35:7769–7782, 2022. 3

[10] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *arXiv*, 2023. 2, 6

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 8

[12] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 2

[13] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. *arXiv preprint arXiv:2201.13360*, 2022. 3

[14] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023. 2, 10

[15] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 2

[16] Mukul Khanna*, Ram Ramrakhya*, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. Goat-bench: A benchmark for multi-modal lifelong navigation. In *CVPR*, 2024. 8

[17] Nuri Kim, Obin Kwon, Hwiyeon Yoo, Yunho Choi, Jeongho Park, and Songhawi Oh. Topological Semantic Graph Memory for Image Goal Navigation. In *CoRL*, 2022. 3, 9

[18] Ue-Hwan Kim, Jin-Man Park, Taek-Jin Song, and Jong-Hwan Kim. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE transactions on cybernetics*, 50(12):4921–4933, 2019. 3

[19] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 3

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 7, 8

[21] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352, 2022. 3

[22] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 6, 7

[23] Kirill Mazur, Edgar Sucar, and Andrew J Davison. Feature-realistic neural fusion for real-time, open set scene un-

derstanding. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8201–8207. IEEE, 2023. 2

[24] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023. 3

[25] Xavi Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander William Clegg, Michal Hlavac, Tiffany Min, Theo Gervet, Vladimir Vondrus, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023. 7, 8

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[27] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 6, 7, 8

[28] Allen Z Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until confident: Efficient exploration for embodied question answering. *arXiv preprint arXiv:2403.15941*, 2024. 3, 4, 6, 7, 8

[29] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. Kimera: From slam to spatial perception with 3d dynamic scene graphs. *The International Journal of Robotics Research*, 40(12-14):1510–1546, 2021. 3

[30] Sergio M Savaresi and Daniel L Boley. On the performance of bisecting k-means and pddp. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–14. SIAM, 2001. 4

[31] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 7, 8

[32] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022. 3

[33] Dhruv Shah, Michael Robert Equi, Błażej Osiński, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. In *Conference on Robot Learning*, pages 2683–2699. PMLR, 2023. 3

[34] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 7, 8

[35] Nikolaos Tsagkas, Oisin Mac Aodha, and Chris Xiaoxuan Lu. Vl-fields: Towards language-grounded neural implicit spatial representations. *arXiv preprint arXiv:2305.12427*, 2023. 2

[36] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3961–3970, 2020. 2, 3

[37] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7515–7525, 2021. 3

[38] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2025. 2

[39] Kashu Yamazaki, Taisei Hanyu, Khoa Vo, Thang Pham, Minh Tran, Gianfranco Doretto, Anh Nguyen, and Ngan Le. Open-fusion: Real-time open-vocabulary 3d mapping and queryable scene representation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9411–9417. IEEE, 2024. 3

[40] Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. Llmgrounder: Open-vocabulary 3d visual grounding with large language model as an agent. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7694–7701. IEEE, 2024. 2

[41] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 42–48. IEEE, 2024. 3

[42] Junbo Zhang, Runpei Dong, and Kaisheng Ma. Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2048–2059, 2023. 2, 3

[43] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and Wang He. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024. 3

[44] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36, 2024. 3