

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

Supplementary Material

Demo Video and Qualitative Results Available in Supplementary Materials. We provide a demo video that illustrates the core concept of 3D-Mem with animations, along with qualitative results for the continuous embodied question-answering task. To access all content, please open “index.html” in a web browser.

6. Full-Set Evaluation

Following the common practice and due to resource limitations, we only evaluate baselines and our method on a subset of A-EQA and GOAT-Bench in our main paper. For reference, we also evaluate 3D-Mem on the complete benchmarks, as shown in the following table.

GOAT-Bench	Whole Set		Subset	
	Success Rate \uparrow	SPL \uparrow	Success Rate \uparrow	SPL \uparrow
3D-Mem (Ours)	62.9	44.7	69.1	48.9

A-EQA	Whole Set		Subset	
	LLM-Match \uparrow	LLM-Match SPL \uparrow	LLM-Match \uparrow	LLM-Match SPL \uparrow
3D-Mem (ours)	53.3	38.0	52.6	42.0

7. Discussion

7.1. Detailed Experiment Results

A-EQA. Table 4 presents a detailed breakdown of results on A-EQA across the seven OpenEQA question categories. As demonstrated in the table, 3D-Mem significantly outperforms ConceptGraph w/ Frontier Snapshots for questions requiring spatial reasoning, including spatial understanding and object localization where the relative positions of surroundings is needed to generate better answers. Such performance gain is attributed to Memory Snapshot, which visually stores both the foreground inter-object spatial relationships and background room-level spatial cues. In contrast, ConceptGraph relies solely on object-centric representations, limiting its ability to capture broader spatial context. For other question categories focus on identifying object-specific variables, i.e., object recognition, attribute recognition, object state recognition, or heavily rely on external knowledge embedded within VLMs, i.e., world knowledge, 3D-Mem also showcases comparable performance as it ensures the capture of all informative objects and effectively utilizes the capability of VLMs.

Compared with Explore-EQA, 3D-Mem generally exhibits higher LLM-Match scores in object-related question categories as we explicitly represents major objects within the scene by Memory Snapshots, enabling the agent to concentrate on relevant elements that may contribute to the final answer. On the other hand, Explore-EQA has consistently lower SPL due to its inefficient semantic-map-based explo-

ration mechanism where explicit visual information of frontiers is encoded into an implicit semantic map. 3D-Mem addresses this limitation by visually capturing glimpses of unexplored areas with Frontier Snapshots and integrating them with Memory Snapshots in the decision-making phase, which provides a more intuitive and holistic view, enabling it to make more informed and effective choices during exploration.

GOAT-Bench. Table 5 presents a detailed breakdown of results on GOAT-Bench across the three question modalities. Comparing 3D-Mem with CG w/ Frontier Snapshots, we observe that 3D-Mem significantly outperforms CG in the Object Category and Language modalities. This improvement is attributed to 3D-Mem’s memory snapshots, which provide explicit spatial relationships among objects and their surroundings, enabling the agent to locate targets more effectively. The detailed spatial context captured in the snapshots enhances the agent’s ability to interpret instructions that rely on spatial cues. In contrast, 3D-Mem’s SPL in the Image modality is slightly lower than CG’s, despite a similar Success Rate. This decrease is likely due to current vision-language models (VLMs) struggling to relate images of complex scenes taken from different angles. When the memory snapshots and the image prompts depict the same region from different perspectives, the VLM may become distracted, leading to less efficient navigation paths. This may highlight a limitation in current VLMs’ ability to match images across varying viewpoints in complex environments.

3D-Mem consistently outperforms both methods across all modalities in terms of Success Rate and SPL scores. By enabling the agent to recall previously observed regions and objects, memory significantly enhances the effectiveness and efficiency of exploration and reasoning. These results highlight memory’s essential role in lifelong object navigation tasks.

7.2. Decision Frequency

Experimentally, the agent queries the VLM and makes a new decision after moving 1m towards the target. We also tested an alternative approach where the agent makes a new decision only after reaching the navigation target. For example, if the agent selects a frontier, it navigates directly to that frontier’s location before making its next choice. However, this approach results in LLM-match scores and SPL values of 50.5 and 36.2, respectively, on A-EQA, which are suboptimal particularly for SPL. Under this setting, we observed numerous cases where the agent initially selects an incorrect frontier and must fully navigate to it before revis-

Method	object recognition		object localization		attribute recognition		spatial understanding		object state recognition		functional reasoning		world knowledge		overall	
<i>Blind LLMs</i>																
GPT-4*	25.3	-	28.4	-	27.3	-	37.7	-	47.2	-	54.2	-	29.5	-	35.5	-
GPT-4o	22.0	-	25.0	-	27.3	-	40.8	-	50.9	-	61.8	-	38.4	-	35.9	-
<i>Question Agnostic Exploration</i>																
CG Scene-Graph Captions*	25.3	-	16.5	-	29.2	-	37.0	-	52.2	-	46.8	-	37.8	-	34.4	6.5
SVM Scene-Graph Captions*	29.0	-	17.2	-	31.5	-	31.5	-	54.2	-	39.8	-	38.9	-	34.2	6.4
LLaVA-1.5 Frame Captions*	25.0	-	24.0	-	34.1	-	34.4	-	56.9	-	53.5	-	40.6	-	38.1	7.0
Multi-Frame*	34.0	-	34.3	-	51.5	-	39.5	-	51.9	-	45.6	-	36.6	-	41.8	7.5
<i>VLM Exploration</i>																
Explore-EQA	44.0	19.6	37.1	29.6	55.3	36.0	42.1	6.6	46.3	9.2	63.2	35.7	45.5	22.0	46.9	23.4
CG w/ Frontier Snapshots	45.0	42.0	32.1	25.0	50.8	35.2	32.9	18.7	68.5	38.4	58.8	42.2	45.5	33.5	47.2	33.3
3D-Mem (Ours)	49.0	45.2	48.6	41.3	47.7	38.6	43.4	33.3	69.4	50.3	64.7	47.2	49.1	38.9	52.6	42.0
Human Agent*	89.7	-	72.8	-	85.4	-	84.8	-	97.8	-	78.9	-	88.5	-	85.1	-

Table 4. **Performance on A-EQA by Question Categories.** For each question categories, there are two columns. The first column stands for the LLM-Match Score, while the second column represents the SPL score. “CG” denotes ConceptGraphs. Methods with * are reported from OpenEQA [22]. Columns represent different category of questions in the dataset.

Method	Object Category		Language		Image		Overall	
	Success Rate	SPL	Success Rate	SPL	Success Rate	SPL	Success Rate	SPL
<i>Open-Sourced VLM Exploration</i>								
3D-Mem w/o memory	55.6	16.0	33.3	15.5	31.8	12.2	40.6	14.6
3D-Mem (Ours)	62.6	33.3	49.5	31.7	35.2	22.7	49.6	29.4
<i>GPT-4o Exploration</i>								
Explore-EQA	64.7	48.4	42.9	22.7	56.8	41.8	55.0	37.9
CG w/ Frontier Snapshots	65.3	44.7	55.0	38.9	64.0	52.8	61.5	45.3
3D-Mem w/o memory	69.9	45.4	50.35	30.1	54.4	39.5	58.6	38.5
3D-Mem (Ours)	79.2	55.8	61.9	46.0	65.2	44.2	69.1	48.9

Table 5. **Performance on GOAT-Bench by Question Modalities.** Evaluated on the “Val Unseen” split. “CG” denotes ConceptGraphs. Methods denoted by * are from GOAT-Bench.

ing its decision, leading to significant wasted exploration distance. In contrast, with our default setting, the agent can adjust its decision en route, mitigating such inefficiencies. One advantage of the alternative setting, however, is that it prevents the agent from oscillating between two frontiers, a problem that can arise in our default setting, particularly during longer exploration episodes. For this reason, in our demo, we opted to have the agent navigate to the target before making the next decision.

7.3. Limitations

We acknowledge several key constraints of 3D-Mem: (1) Similar to most 3D scene representations, 3D-Mem is designed for static environments and is not robust to moving objects. (2) The performance of 3D-Mem depends on object detection results and VLM reasoning capabilities. (3) The precise location of the agent is required to accurately locate the objects in the scene and construct scene memory, which could be challenging to acquire after long-time exploration. (4) The latency of the whole pipeline during embodied QA is still noticeable. We time the detailed la-

tency of each component of our pipeline in Table ?? . We also argue that, although 2D-3D lifting (including SLAM and object detection) and Prefiltering consume considerable time, in real-world scenarios, they run concurrently during navigation, and caching Prefiltering results minimizes their impact on throughput. The primary bottleneck is VLM inference, which is caused by both heavy model computation and network latency. This can be mitigated by running the model locally or optimizing VLMs with techniques such as model quantization. Additionally, 3D-Mem requires VLM inference only after each high-level step, thereby reducing the VLM query frequency.

Component	2D-3D Lift	Clustering	Prefiltering	VLM Inference
A-EQA	2.43	0.04	1.12	3.34
GOAT-Bench	2.79	0.09	1.35	3.58

Table 6. Time cost of each component on A-EQA, evaluated in seconds.

8. Failure Case Analysis

In Figure 5 to 10, we analyze and categorize the types of questions where 3D-Mem performs poorly in A-EQA. For each example question, we provide the ground truth answer, 3D-Mem’s predicted answer, and the memory snapshot selected for answering the question. Each memory snapshot includes object detection annotations to better visualize which objects are detected and incorporated into the scene graph. These annotations can be zoomed in for a clearer view. Note that these annotations are not part of the actual input provided to the VLM.

We generally classify the failures into the following three categories:

- **Dataset Issues.** As shown in Figure 5, some questions in the A-EQA dataset are inherently vague and allow for multiple reasonable answers. Although the ground truth answers are generally more appropriate, the VLM often exhibits overconfidence in its current predictions and terminates the episode prematurely.
- **Limitations of the VLM.** Many questions fail due to the limited perception capabilities of the VLM, which can be further divided into two subcategories. In the first case, as shown in Figure 6, the correct memory snapshot is selected, but the predicted answer is incorrect. This often occurs when the target objects in the snapshots are small, and the limited image resolution (360×360) makes it difficult for the VLM to identify them. In the second case, as shown in Figure 7, the VLM selects the wrong memory snapshot entirely and produces unreasonable answers.
- **Limitations of the Object Detection Model.** 3D-Mem relies on an object detector to identify and add new objects to the scene graph. However, the object detector can sometimes produce incorrect labels, as shown in Figure 8. In most cases, the prefiltering process successfully filters out these incorrect labels, as they are often highly irrelevant. Additionally, the VLM is generally capable of recognizing and ignoring such errors. However, certain situations, as in Figure 9, illustrate cases where the detector mislabels objects—such as detecting a TV as a fan or a cloth rack as a ladder. These misclassifications can confuse the VLM, especially when the incorrect labels closely align with the expected answer. In addition to incorrect detections, the target objects for answering the questions are detected at all. As shown in Figure 9, if the car or the monitor had been detected in the relevant snapshots, the question could have been answered correctly. However, since these objects were not detected and included in memory, the VLM could not select an appropriate memory snapshot, and the agent eventually exceeded the step limit. Interestingly, in some cases, as shown in Figure 10, even when the target objects are not detected, they remain visible in other memory snapshots that pass the prefiltering process. In these cases, the VLM is still

able to answer the question successfully. This demonstrates that 3D-Mem is more robust than traditional 3D scene graph approaches.

9. Details of Frontier-based Exploration Framework

Our frontier-based exploration framework is based on the framework in Explore-EQA [28]. We enhance its robustness and adapt it to our multi-view images representation framework. A 3D grid-based occupancy map M , representing the length, width and height of the entire room, is used to record the occupancy, with each voxel having a side length of 0.1 meters. During exploration, each depth observation, together with its corresponding observation pose, is used to map unoccupied spaces onto the initially fully occupied M . The navigable region is then defined as the layer of unoccupied voxels at the height of 0.4 meters above the ground where the agent moves. Within this navigable region, the area within 1.7 meters of the agent’s trajectory is defined as the explored region, while the remainder is designated as the unexplored region, as illustrated in Figure 11.

Frontiers are defined as clusters of pixels in the unexplored region. Pixels in the unexplored region are clustered into different groups using Density-Based Spatial Clustering of Applications with Noise (DBSCAN), with each group consisting of connected pixels. Each frontier $F = \langle r, p, I^{obs} \rangle$ represents such a pixel group r . The navigable location of the frontier p is determined at the boundary between the frontier region and the explored region, and an image observation I^{obs} is captured once the frontier has been updated. As shown in Figure 11, each purple arrow together with a green region it points to is a frontier. For a frontier to be meaningful, r must contain more than 20 pixels; otherwise, the frontier will not be created. A frontier is considered updated if the intersection-over-union (IoU) between the new and previous regions r is less than 0.95. Additionally, if r spans more than 150° in the agent’s field of view, it is split into two regions using K-Means clustering, resulting in two separate frontiers. This approach allows for more flexibility in choosing navigation directions. Also, it is important to note that this format for representing 3D space does not currently support scenes with multiple floors. Consequently, our results in Table 1 fall significantly short of human performance, as many of the questions in A-EQA require exploration across different floors.

When prompting the VLM, only the image observations are included in the prompt. If the VLM chooses a frontier F , the location p is used as the agent’s navigation target.

10. More Details in Experiments

At each step t , we take $N = 3$ egocentric views, each with a gap of 60° . The egocentric views are captured at a resolu-

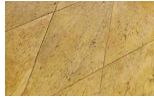


Figure 5. Failure Case 1: Some questions in A-EQA are vague and may have multiple reasonable answers.



Question: What is on the top shelf to the right side of the garage?

Ground Truth: An ice cooler

Prediction: There is a black bag and a small box on the top shelf to the right side of the garage.



Question: What is between the fruit bowl and knife set?

Ground Truth: A container of spices

Prediction: Between the fruit bowl and knife set, there is a toaster.

Figure 6. Failure Case 2: Due to limitations in perception capabilities and image resolution, the VLM cannot provide the correct answer even when the memory snapshot is correctly chosen.

tion of 1280×1280 for better object detection and are then resized to 360×360 as frame candidates for VLM input. Frontier snapshots are initially captured at 360×360 . We

use YOLOv8x-World, implemented by Ultralytics, as our detection model and a 200-class set from ScanNet [3] as the detection class set. We set a maximum of 50 steps for each

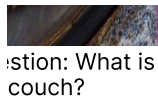
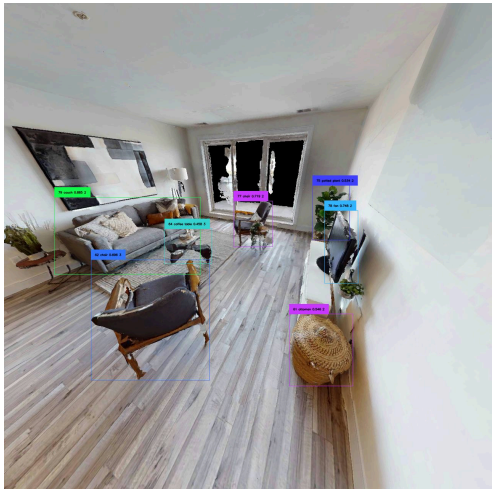


Figure 7. Failure Case 3: In some cases, the VLM selects an entirely incorrect memory snapshot and produces unreasonable answers.



Question: What can I use to cool down the living room?

Ground Truth: The thermostat

Prediction: You can use the fan to cool down the living room.



Question: How can I reach the wine glasses if I'm too short?

Ground Truth: Using the step-stool in the laundry room.

Prediction: You can use the ladder to reach the wine glasses.

Figure 8. Failure Case 4: Incorrect labels predicted by the object detector can mislead the VLM. Although the VLM can often ignore such errors, in certain cases, these misclassifications cause confusion.

task.

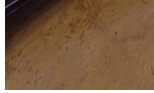


Figure 9. Failure Case 5: Some target objects are not added to the scene graph due to missed detections by the object detector. The memory snapshot shown above is where the target object should have been detected and assigned to.

Question: What is
TV?

Ground Truth: A c

Figure 10. A similar scenario to Figure 9, where the target objects are not detected. However, as they are still visible in other memory snapshots, the VLM still successfully answers the questions.

10.1. Implementation Details for A-EQA

As explained in detail in Section 3.2, we integrate 3D-Mem into the frontier-based exploration framework. The VLM directly returns an answer after identifying visual

clues from certain memory snapshots. We set the number of egocentric observations at each step $N = 3$, the maximum distance for objects to be included in the scene graph $max_dist = 3.5$, and the number of prefiltered classes

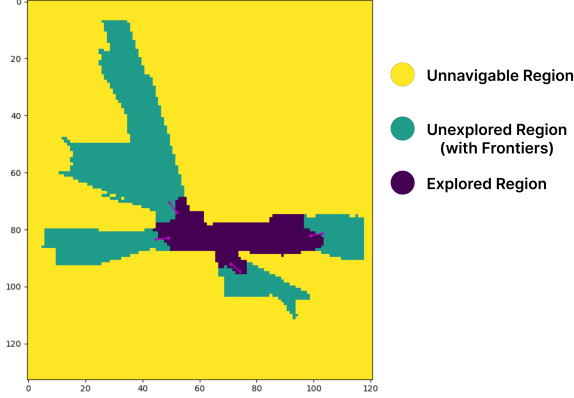


Figure 11. A illustration of different regions and frontiers in the frontier-based exploration framework. Note that navigable region consists of explored and unexplored regions.

$K = 10$.

10.2. Implementation Details for EM-EQA

To adapt 3D-Mem to the EM-EQA benchmark, we first construct 3D-Mem for each scene using the given RGB-D observations and corresponding camera poses. For each question, we then apply prefiltering to the memory snapshots using different K values (1, 2, 3, 5, 10), and utilize the resulting filtered snapshots as prompts for GPT-4o to generate the answers.

10.3. Implementation Details for GOAT-Bench

We reformulate the navigation task into the embodied question answering format by filling in templates for three types of target descriptions: “Can you find the {category}?”, “Can you find the object described as {language description}?”, and “Can you find the object captured in the following image? {image}”. We adapt the prompt for navigation tasks as described in Section 3.3, allowing the VLM to choose an object directly from a memory snapshot. After the VLM identifies an object in such a way, the agent navigates to a location near that object to complete the task. We evaluate both GPT-4o and open-sourced VLM (specifically LLaVA-7B [20]) as the choice of VLM. For LLaVA-7B model, we further fine-tune it on our generated dataset for better performance (see Appendix 12 for more details). Other hyperparameter settings are the same as the experiments on A-EQA.

11. Details of the Active Exploration

When prompting the VLM for embodied question answering (A-EQA Benchmark), as shown in Figure 17, we append each memory snapshot with the object classes it contains. However, we only append classes that are within the prefiltered class list. For frontier snapshots, only the raw

Figure 12. Overview of the frontier-based exploration pipeline with 3D-Mem on embodied question-answering task.

snapshot images are used as input. The VLM will then respond with either a frontier snapshot or a memory snapshot. If the VLM returns a frontier, we set the location p as the navigation target. If the VLM returns a memory snapshot along with the answer, although we directly conclude the navigation episode in our A-EQA experiments, we also set a navigation target for that memory snapshot. This allows the agent to move closer to the snapshot region, refine the selected memory snapshot, and potentially reconsider its choice.

The navigation location for a memory snapshot is determined by several conditions. We set the observation distance, obs_dist , to 0.75 meters. If the snapshot contains only one object, the location is set obs_dist away from the object, in the direction from the object’s location toward the center of the navigable area that is obs_dist around the object. If the memory snapshot contains two objects, the location is set obs_dist away from the midpoint of the two objects, in the direction of the perpendicular bisector of the line segment connecting the objects. If the memory snapshot contains more than two objects, we first perform Principal Component Analysis (PCA) on the object cluster to obtain the principal axis with the smallest eigenvalue. The navigation location is then set obs_dist away from the center of the object cluster, in the direction of this principal axis. Note that, in all cases for determining the navigation location, we always ignore the height of the objects and treat them as 2D points. Additionally, the above algorithm can be randomized by assigning the highest probabilities to the aforementioned positions.

Embodied navigation tasks (GOAT-Bench Benchmark) work similarly, with the following differences: 1) we append the object crop after each class name when prompting the VLM, as shown in the prompt in Figure 18; 2) when the VLM returns an object choice, we treat that object as a memory snapshot containing one object and follow a similar method to set the navigation location.

After a navigation target is set (either a frontier or a

memory snapshot), the agent moves towards it along a path generated by the pathfinder in habitat-sim [25, 31, 34]. Although we utilize the pathfinder, which uses prior information from a global navmesh to find the shortest paths, we can easily replace it with a simple path-finding algorithm based on the navigable map described in Appendix 9. Step t ends after the movement. Then in the new step $t + 1$, the agent updates the frontiers and memory snapshots and makes the next decision.

12. Details of Training Open-Sourced VLMs for GOAT-Bench Navigation

12.1. Training Dataset Collection

In GOAT-Bench [16], each navigation target is described by three types of descriptors: category, language, and image. We generate training data based on their provided exploration data, sourced from 136 scenes in HM3D [27] training set. In each scene, a set of navigation targets is provided, each consisting of an object ID, location, category, language description, and multiple viewpoints and angles for capturing image observations. In total, the training set includes 3669 such objects, which we use as navigation targets to generate training data in our framework’s format.

We adapt our exploration pipeline for data generation. For each navigation target, we first randomly select an initial point on the same floor. We then use the pathfinder in habitat-sim [25, 31, 34] to find the shortest trajectory to the target. At each step, if the target object is present in a memory snapshot, we use that memory snapshot as the ground truth and move one step toward a location near it; if the target object is not present in any memory snapshot, we select the frontier closest to the shortest trajectory as the ground truth for that step and move one step toward that frontier. On average, we collect 4 exploration paths per target object from different initial points, with each path consisting of approximately 12 steps.

We also collect the ground truth for prefiltering by prompting GPT-4o. For each navigation target, we collect all objects that can be seen along the exploration path and feed them, together with the description, into GPT-4o. We ask GPT-4o to rank all visible objects based on their helpfulness in finding the navigation target. For each navigation target, we collect three such rankings corresponding to three types of descriptions.

12.2. Training Process

We fine-tune our model based on the LLaVA-1.5-7B checkpoint[20] using the collected training dataset for 5 epochs with a learning rate of $4e-6$ and a batch size of 1. We use the AdamW optimizer with no weight decay. During training, DeepSpeed ZeRO-2 and LORA [11] are used to save GPU memory and accelerate training. FP16 is en-

abled to balance speed and precision. We train our model with 6×24 Tesla V100 GPUs, and the fine-tuning process is completed within 6 hours.

We use the default CLIP vision encoder of LLaVA to encode all memory snapshots, frontier snapshots, egocentric views and image navigation targets. And the encoded vision features are further compressed to 12×12 (for image targets and egocentric views) and 3×3 (for memory snapshots and frontier snapshots) tokens in the training prompt.

During fine-tuning, we simultaneously optimize the model for exploration task and prefiltering task with cross-entropy loss. The loss weights for exploration and prefiltering are set to 1 and 0.3, respectively. The training goal of exploration is to correctly predict the ground truth choice of memory snapshot or frontier at each step. The training goal of prefiltering is to select the top 10 helpful objects that have been observed, based on the ground truth we collected earlier.

13. Ablation Study

13.1. Ablation on Hyperparameter Choices

We mainly evaluate on the number of egocentric observations at each step (N), the maximum distance an object should be included in the memory snapshot (max_dist), and the number of prefiltered classes (K).

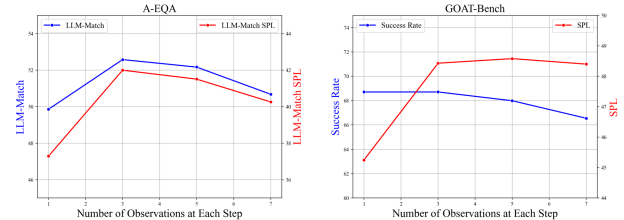


Figure 13. Ablation on the number of observation each step (N) for A-EQA and GOAT-Bench.

In Figure 13, we present the evaluation metrics for different choices of N on both A-EQA and GOAT-Bench. We can observe that increasing the number of observations does not necessarily lead to better performance. This is mainly because the additional views often provide repeated and redundant information. Furthermore, as the number of frame candidates increases, a cluster of objects that would originally be assigned to one memory snapshots may instead be assigned to separate memory snapshots, resulting in confusion. Based on the results, we choose $N = 3$ for both datasets.

In Figure 14, we present the evaluation metrics for different choices of max_dist on both A-EQA and GOAT-Bench, where we observe different tendencies across the two benchmarks. Evaluation metrics on GOAT-Bench generally improve with an increase in max_dist , while metrics

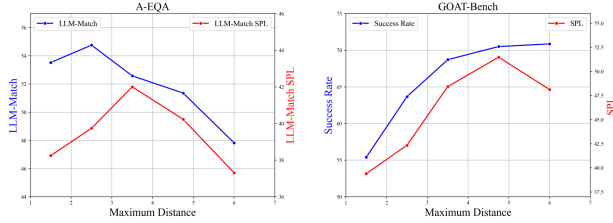


Figure 14. Ablation on the maximum distance for including an object to the scene graph (*max_dist*) for A-EQA and GOAT-Bench.

on A-EQA decline. This is because, under normal circumstances, a memory snapshot should only represent objects within a local area. Objects in more distant regions should either remain in unexplored areas or be captured by another memory snapshot that is closer to them. A large *max_dist* imposes a looser distance restriction, which can introduce disorder. However, in the navigation task of GOAT-Bench, the earlier the target object is added to the scene graph as a choice for the VLM, the faster the VLM can select it as the direct navigation target, resulting in faster arrival at the target objects. Balancing both accuracy and efficiency across the two benchmarks, we choose *max_dist* to be 3.5 meters.

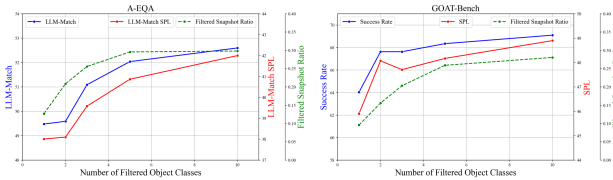


Figure 15. Ablation on the number of prefiltered classes (K) for A-EQA and GOAT-Bench.

In Figure 15, we present the evaluation metrics for different choices of K on both A-EQA and GOAT-Bench. In addition to the metrics introduced in the experiment sections, we include the average ratio of the number of remaining memory snapshots after prefiltering to the total number of memory snapshots as a measure of the effectiveness and intensity of prefiltering. The results on both benchmarks align with our intuition: allowing more prefiltered classes leads to better performance. Moreover, even when $K = 10$, on average only 3.26 and 4.66 memory snapshots are left after prefiltering for A-EQA and GOAT-Bench respectively, accounting for 29.8% and 28.1% of the total memory snapshots, and 8.2% and 5.1% of the total frame candidates. These statistics demonstrate the effectiveness of prefiltering as a memory retrieval mechanism, as well as 3D-Mem’s compactness as a scene representation. Furthermore, we observe that the overall performance does not drop significantly when K is small, highlighting the robustness of our framework.

13.2. Ablation on Pipeline Components

Ablation study on Prefiltering is infeasible because directly querying the VLM would exceed its context length. However, we conduct an ablation on Frontier Snapshots by always navigating to the nearest frontier when Memory Snapshots cannot provide the answer, rather than choosing a frontier via VLM. As shown in Table 7 (SnapMem w/o FS), performance declines on both A-EQA and GOAT-Bench, though the drop is smaller on GOAT-Bench. This is likely due to the lifelong setting of GOAT-Bench, where the agent tends to rely on its memory once the scene is mostly explored. Additional experiments removing both Frontier Snapshots and memory maintenance (SnapMem w/o FS & Mem) confirm this pattern.

Method	A-EQA		GOAT-Bench	
	LLM-Match \uparrow	LLM-Match SPL \uparrow	Success Rate \uparrow	SPL \uparrow
SnapMem w/o FS & Mem	-	-	57.2	33.2
SnapMem w/o FS	49.3	31.0	63.7	46.8
SnapMem	52.6	42.0	69.1	48.9

Table 7. Ablation study of Frontier Snapshot on A-EQA and GOAT-Bench. FS denotes ”Frontier Snapshots”.

14. Other Related Works

While our work focuses on comparing to 3D scene representations, prior research on 2D scene representations, particularly in topological mapping for navigation, is notable. Methods like Topological Semantic Graph Memory [17] and RoboHop [6] similarly represent environments as graphs using images and objects, where nodes correspond to images of navigable places and edges denote navigability.

Our proposed 3D-Mem first differs from these topological mapping methods in its focus and design. 3D-Mem focuses on capturing all salient objects in the scene by a minimum number of memory snapshots. Each memory snapshot is designed to capture the visual features of a cluster of objects in the nearby region, along with their spatial relationships and surrounding environment. Objects are uniquely assigned to one memory snapshot, making the representation informative, comprehensive, and compact—key qualities for leveraging vision-language models (VLMs) with limited context length to interpret and reason over visual data. In contrast, the images in topological map in [17] are primarily designed to represent landmarks for navigation, without attempting to capture all informative aspects of the scene or focusing on visually representing all objects in a 3D environment. The images in the representation in [6], though focused on object segments, are not compact, containing redundancy between consecutive frames, and the images still serve as navigation landmarks rather than visual representations of inter-object relationships.

In addition, 3D-Mem introduces the concept of frontier

snapshots to explicitly model unexplored regions, allowing agents to make informed decisions about where to explore next to expand knowledge—an active exploration capability not addressed in previous 2D methods. Moreover, the structure of 3D-Mem enables the memory retrieval mechanism of Prefiltering, which manages the memory scalability and efficiency over extended operations, supporting life-long learning that is absent in the aforementioned works. Lastly, like other 3D scene graphs, 3D-Mem stores the 3D information of the objects and snapshots. Based on this information, as in the practice of ConceptFusion [14], a set of spatial relationship comparators can be called by LLMs as queries, *e.g.*, querying the distance between A and B by calling “howFar(A, B)”. This information is cannot be stored in those 2D representations.

15. Complete Prompts for VLMs

We present the full prompt for prefiltering in Figure 16, the prompt for embodied question answering (A-EQA dataset) in Figure 17, and the prompt for navigation (GOAT-Bench dataset) in Figure 18.

System Prompt:

You are an AI agent in a 3D indoor scene.

Content Prompt:

Your goal is to answer questions about the scene through exploration.

To efficiently solve the problem, you should first rank objects in the scene based on their importance. These are the rules for the task.

1. Read through the whole object list.
2. Rank objects in the list based on how well they can help your exploration given the question.
3. Reprint the name of all objects that may help your exploration given the question.
4. Do not print any object not included in the list or include any additional information in your response.

Here is an example of selecting helpful objects:

Question: What can I use to watch my favorite shows and movies?

Following is a list of objects that you can choose, each object one line:

painting
speaker
box
cabinet
lamp
tv
book rack
sofa
oven
bed
curtain

Answer:

tv
speaker
sofa
bed

Following is the concrete content of the task and you should retrieve helpful objects in order:

Question: {question}

Following is a list of objects that you can choose, each object one line:

{class_0}
{class_1}

...

Answer:

Figure 16. Prompt for prefiltering. The placeholders {question} and {class_*i*} are replaced by the question and all existing classes in the scene graph, respectively.

System Prompt:

Task: You are an agent in an indoor scene tasked with answering questions by observing the surroundings and exploring the environment. To answer the question, you are required to choose either a Snapshot as the answer or a Frontier to further explore.

Definitions:

Snapshot: A focused observation of several objects. Choosing a Snapshot means that this snapshot image contains enough information for you to answer the question. If you choose a Snapshot, you need to directly give an answer to the question. If you don't have enough information to give an answer, then don't choose a Snapshot.

Frontier: An observation of an unexplored region that could potentially lead to new information for answering the question. Selecting a frontier means that you will further explore that direction. If you choose a Frontier, you need to explain why you would like to choose that direction to explore.

Content Prompt:

Question: {question}

Select the Frontier/Snapshot that would help find the answer of the question.

The following is the egocentric view of the agent in forward direction: [img]

The followings are all the snapshots that you can choose (followed with contained object classes). Please note that the contained classes may not be accurate (wrong classes/missing classes) due to the limitation of the object detection model. So you still need to utilize the images to make decisions.

Snapshot 0 [img] {class_0}, {class_1}, ...

Snapshot 1 [img] {class_0}, {class_1}, ...

...

The followings are all the Frontiers that you can explore:

Frontier 0 [img]

Frontier 1 [img]

...

Please provide your answer in the following format: "Snapshot i\n[Answer]" or "Frontier i\n[Reason]", where i is the index of the snapshot or frontier you choose. For example, if you choose the first snapshot, you can return "Snapshot 0\nThe fruit bowl is on the kitchen counter.". If you choose the second frontier, you can return "Frontier 1\nI see a door that may lead to the living room.".

Note that if you choose a snapshot to answer the question, (1) you should give a direct answer that can be understood by others. Don't mention words like "snapshot", "on the left of the image", etc; (2) you can also utilize other snapshots, frontiers and egocentric views to gather more information, but you should always choose one most relevant snapshot to answer the question.

Figure 17. Prompt for embodied question answering. The placeholders {question} and {class_i} are replaced by the question and the object classes contained in the corresponding memory snapshots, respectively. [img] are replaced by the egocentric views, memory snapshots or frontier snapshots.

System Prompt:

Task: You are an agent in an indoor scene that is able to observe the surroundings and explore the environment. You are tasked with indoor navigation, and you are required to choose either a Snapshot or a Frontier image to explore and find the target object required in the question.

Definitions:

Snapshot: A focused observation of several objects. It contains a full image of the cluster of objects, and separate image crops of each object. Choosing a snapshot means that the object asked in the question is within the cluster of objects that the snapshot represents, and you will choose that object as the final answer of the question. Therefore, if you choose a snapshot, you should also choose the object in the snapshot that you think is the answer to the question.

Frontier: An unexplored region that could potentially lead to new information for answering the question. Selecting a frontier means that you will further explore that direction.

Content Prompt:

Question: {question}

Select the Frontier/Snapshot that would help find the answer of the question.

The following is the egocentric view of the agent in forward direction: [img]

The followings are all the snapshots that you can choose. Following each snapshot image are the class name and image crop of each object contained in the snapshot. Please note that the class name may not be accurate due to the limitation of the object detection model. So you still need to utilize the images to make the decision.

Snapshot 0 [img] Object 0: {class_0} [img_crop_0], Object 1: {class_1} [img_crop_1] ...

Snapshot 1 [img] Object 0: {class_0} [img_crop_0], Object 1: {class_1} [img_crop_1] ...

...

The followings are all the Frontiers that you can explore:

Frontier 0 [img]

Frontier 1 [img]

...

Please provide your answer in the following format: "Snapshot i, Object j" or "Frontier i", where i, j are the index of the snapshot or frontier you choose. For example, if you choose the fridge in the first snapshot, please return "Snapshot 0, Object 2", where 2 is the index of the fridge in that snapshot. You can explain the reason for your choice, but put it in a new line after the choice.

Figure 18. Prompt for GOAT-Bench dataset. The placeholders {question} and {class_*i*} are replaced by the question and the object classes contained in the corresponding memory snapshots, respectively. [img] are replaced by the egocentric views, memory snapshots or frontier snapshots, and [img_crop_*i*] are replaced by the corresponding object crops, which are directly cropped from the memory snapshots based on the detection bounding boxes.

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, 4
- [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. Lrf: 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, Sri-ram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. Goat-bench: A benchmark for multi-modal life-long navigation. In *CVPR*, 2024. 8
- [17] Nuri Kim, Obin Kwon, Hwiyeon Yoo, Yunho Choi, Jeongho Park, and Songhwi 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. 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. 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. 8
- [35] Nikolaos Tsagkas, Oisín Mac Aodha, and Chris Xiaoxuan Lu. VI-fields: Towards language-grounded neural implicit spatial representations. *arXiv preprint arXiv:2305.12427*, 2023. 2
- [36] Johanna Wald, Helisa Dhama, 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. Llm-grounder: 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