MEMORY TREE GUIDED KEY FRAME QUERYING FOR EFFICIENT 3D QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Answering questions accurately and efficiently in embodied scenarios presents significant challenges due to limited computational and GPU memory resources. Current embodied systems struggle with the GPU memory overhead of Vision-Language Model (VLM) processing extensive video frames collected during scene exploration. An intuitive solution is to select question-related key frames for VLM inference. Existing key frame selection approaches adopt the visual search-based key frame selection paradigm, which is inefficient since the vision model must infer over every frame for each individual query. In this work, we propose a novel memory tree guided key frame selection paradigm for 3D question answering in embodied scenarios. Our method leverages a compact and reusable 3D scene representation, termed MemTree3D, which supports real-time online construction leveraging camera 6-DoF pose. MemTree3D captures multi-level 3D scene information, enabling a Large Language Model to efficiently query and retrieve question-relevant key frames through our scoring-based frame selection without reprocessing the entire video stream. On OpenEQA, our method improves the accuracy of GPT-4o by 17.4%, achieving state-of-the-art performance and outperforms existing visual search methods in both accuracy and efficiency, demonstrating our work's potential as an effective solution for real-world embodied applications requiring fast and accurate scene understanding. Our code will be released with the final version of the paper.

1 INTRODUCTION

Understanding the 3D scene and performing 3D question answering (Majumdar et al., 2024; Azuma et al., 2022; Ma et al., 2023) efficiently and accurately has been challenging in the embodied agent scenario due to restricted computational and GPU memory resources. Vision-Language Models (VLMs) face significant computational and GPU memory overhead when processing the large number of video frames collected from a 3D scene. A straightforward approach to reduce computational and memory usage in 3D question answering is through frame sampling (Cheng et al., 2025; Zhu et al., 2024a; Huang et al., 2025b), which significantly lowers the number of input frames for the VLM. However, uniform sampling can lead to visual information loss, especially for long video sequences, where sparse sampling may omit critical key frames relevant to the question. To address this, recent research has focused on selecting question-related key frames (Fan et al., 2024; Yang et al., 2025b; Xu et al., 2024a; Zhu et al., 2024a; Song et al., 2024) as VLM input to balance computational cost and question answering accuracy.

Despite recent progress in long-form video understanding (Caba Heilbron et al., 2015; Xiao et al., 2021; Fu et al., 2025; Xu et al., 2017; Ye et al., 2025), existing VLMs still require key frame sampling (Zhu et al., 2024a; Yang et al., 2025b; Hu et al., 2025) or token compression (Bolya et al., 2023; Huang et al., 2025a; Wu, 2024) to reduce memory and computational costs. Most key frame sampling approaches follow what we term the **visual search key frame selection** paradigm, where a vision model conditioned on the user-provided question is used to infer over video frames and select those critical for answering the question. Common vision models include open-vocabulary object detectors (Cheng et al., 2024a; Liu et al., 2024c), which are conditioned on question-relevant objects to perform object detection over video frames (Xu et al., 2024a). Image-text retrieval approaches (WANG et al., 2025; Song et al., 2024) have also been proposed, leveraging vision language models (Radford et al., 2021; Li et al., 2023a) to select frames with high similarity to the

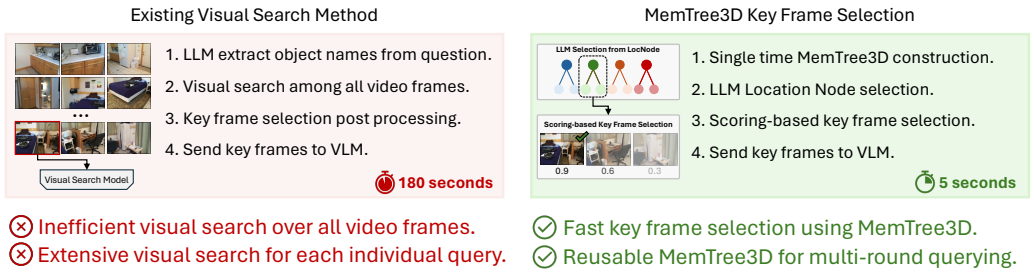


Figure 1: Comparison between existing visual search key frame selection and our proposed MemTree3D key frame selection. Our method bypasses the extensive visual search over the video, achieving higher efficiency.

text query, but the retrieval focus on image-level similarity and omits visual details, resulting in sub-optimal performance. Some methods adopt VLMs to select key frames (Yang et al., 2025b; Hu et al., 2025) based on the given query. However, these methods suffer from several drawbacks, such as heavy reliance on the vision model, whose predictions can be noisy and insufficiently robust for reliable key frame selection, resulting in error propagation. Another major limitation is inefficiency in multi-round user querying—a common scenario in embodied settings—where each new query requires re-running the vision model over all video frames, leading to high latency that scales with video length, as illustrated in Fig. 1. In this work, we propose a novel MemTree3D guided key frame selection strategy, with improved efficiency, advanced performance, and strong potential for adaptation to real-world embodied scenario.

In this paper, we address two key limitations of the existing visual search key frame selection paradigm: (1) the overreliance on vision models, with no mechanism to recover from detection failures or missing objects; and (2) inefficiency, as current methods require running the vision model over all video frames for each user query to select key frame. To overcome these challenges, we leverage a tree based architecture that serves as a compact, and reusable 3D scene representation. This structure can be queried by an LLM to efficiently retrieve the most relevant key frames, which are then passed to a VLM for question answering.

Our approach begins with the online and real-time construction of a hierarchical representation **MemTree3D**. The MemTree3D encodes 3D scene at multiple levels, including frame-level detections, temporally-aware object relationships, and spatially localized segments derived from 6-DoF camera poses. This tree-based structure enables the LLM to reason over the scene content and identify critical frames without scanning every video frame. It is also reusable across multi-round queries: once constructed, the tree can be queried repeatedly by the LLM for different questions without re-running the vision model. Finally, it is robust to detection failures. Because the LLM reasons over the symbolic and structural information in the tree (an example in Fig. 3), it can still identify relevant frames even when detections of target objects are missing. In conclusion, we present a novel tree guided key frame selection paradigm that improves efficiency through MemTree3D querying and enhances robustness to perception failures via LLM-based reasoning. We summarize our contributions as follows:

- We propose a novel, efficient, and performant MemTree3D guided key frame selection paradigm for 3D question answering task, advancing the efficiency beyond existing visual search key frame selection approaches.
- We introduce **MemTree3D**, a compact and reusable 3D scene representation that supports real-time construction and enables LLM-driven key frame selection—paving the way for scalable, real-world embodied applications.
- Our strong performance across multiple benchmarks using fewer input frames offers new insights into video key frame selection, challenging the dominance of visual search as the primary paradigm.

2 Related Work

3D Scene Representations Capturing complex 3D scenes in a compact representation remains a significant challenge in current research. Recent 3D-LLMs (Zhu et al., 2023; 2024b; Xu et al., 2024b) leverage point cloud representations, but require extensive 3D data training and still fall short in performance and generalizability compared to 2D models (Huang et al., 2025b; WANG et al., 2025). Object-centric representations (Wang et al., 2023; Huang et al., 2024a; Hong et al., 2024) and 3D scene graphs (Gu et al., 2024; Armeni et al., 2019; Cherian et al., 2022; Wu et al., 2021) are also popular approaches, yet they suffer from perception failures such as missing objects or false positives. More recent video-based 3D-LLMs (Zheng et al., 2025; Zhu et al., 2024a) utilize 3D scene videos directly for 3D understanding, but face significant computational overhead due to the length of 3D scan videos. To address these challenges, we propose MemTree3D, an object-centric 3D scene representation that differs from traditional scene graphs by incorporating temporal information. Rather than serving as the final scene representation, MemTree3D acts as an intermediate structure for LLM-driven key frame retrieval, mitigating visual information loss typically seen in existing scene graph-based methods.

Key Frame Selection for Video Understanding To address the computational and memory overhead in long-form video, recent research has developed key frame selection techniques to select question-relevant frames for VLM input. Most existing key frame selection methods follow the visual search key frame selection paradigm, which leverages the input question to identify relevant frames using techniques such as detecting question-related objects (Ye et al., 2025; Xu et al., 2024a), VLM-based selection (Hu et al., 2025; Yang et al., 2025b), or computing image-text similarity scores (WANG et al., 2025) via vision foundation models (Radford et al., 2021; Li et al., 2023a). While these approaches generally outperform naive uniform sampling, the visual search paradigm significantly limits their efficiency in multi-round user query scenarios, as each query requires re-running visual search across the entire video.

3D Question Answering MLLM Existing 3D question answering MLLMs can be broadly categorized into video-LLM and 3D-LLM, with the former taking the 3D scan video as input, while the latter consumes 3D scene point cloud or voxel representation. Video-LLMs (Li et al., 2023b; Maaz et al., 2024; Zhang et al., 2023; Liu et al., 2024a) can be directly applied to the 3D question answering task under zero-shot setting, yet they suffer when handling a large number of frames, which incur extensive memory and computational cost. 3D-LLM methods (Fu et al., 2024; Hong et al., 2023; Huang et al., 2024b; Zhu et al., 2023; Chen et al., 2024; Xu et al., 2024b), despite using a more compact point cloud or voxel representations, require extensive fine-tuning on 3D data, and do not achieve significant performance advantages when compared with zero-shot, 2D video-LLM (Huang et al., 2025b; WANG et al., 2025).

3 Method

In this work, we propose the MemTree3D guided key frame selection paradigm. The motivation is to bypass the visual search key frame selection, which requires the vision model to infer on all video frames for every user query. Our work introduces MemTree3D, a compact 3D scene representation for LLM querying and key frame selection. We introduce MemTree3D in the following sections.

3.1 MemTree3D

MemTree3D is a 3D representation with three levels of tree nodes. We describe our three-level node design as follows:

Location Node. We split the 3D scene video into multiple segments and construct a location node (`LocNode`) for each segment. Each `LocNode` is assigned a unique location ID. While several existing works (He et al., 2024; Wang et al., 2025) rely on image semantic to cluster video into segments, such approaches pose challenges in the embodied AI setting. These clustering methods are effective in general video question answering tasks, where inter-frame differences often correspond to changes in video content or camera shots. However, in the embodied scenario, the camera

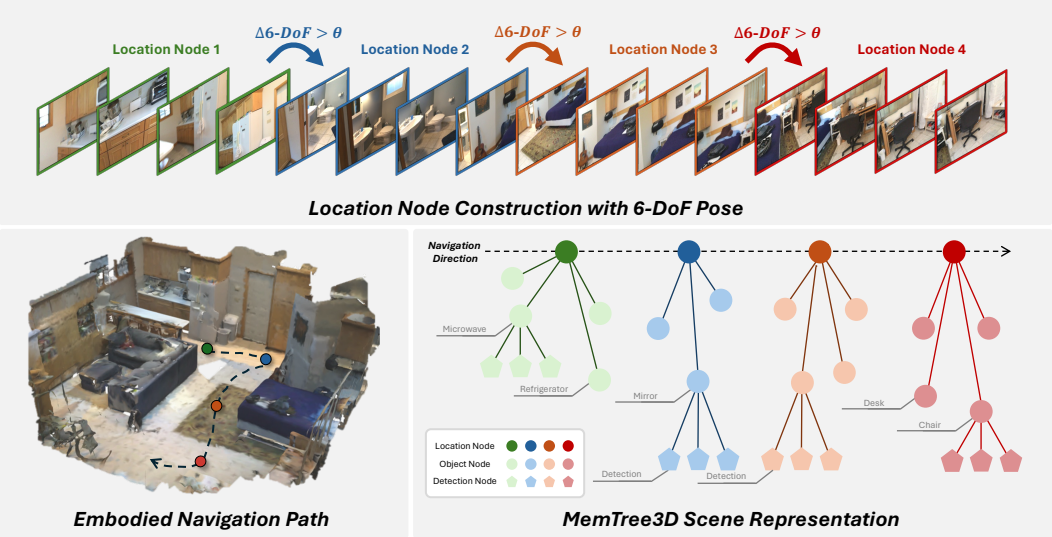


Figure 2: Our proposed MemTree3D construction process. During the embodied navigation, the camera 6-DoF poses are used to construct the location node, each location node contains multi-level 3D scene information for later LLM key frame selection.

captures a continuous 3D scan with smaller semantic variation between adjacent frames, especially in visually uniform scene. This makes semantic-based clustering less reliable for identifying spatial transitions in 3D scenes. The second challenge lies in the computational overhead of extracting visual semantics, which relies on another off-the-shelf vision foundation model. This further poses additional difficulties in resource-constrained embodied scenarios.

To address these limitations, we propose to segment the video stream in a way that (1) introduces minimal additional computational cost, and (2) ensures that each segment corresponds to a distinct location in the 3D scene. In our work, we utilize the 6-DoF pose, which is information that can be obtained directly from most existing embodied AI devices. We keep track of each frame's 6-DoF pose during video processing and maintain a previous 6-DoF pose, P_{prev}, to calculate the translation and rotation with respect to the 6-DoF pose at the current frame, P_t. Whenever the 6-DoF changes exceed translation threshold T_{thres} or rotation threshold R_{thres}, we construct a new `LocNode` and update P_{prev} with the current pose P_t; we provide a detailed pseudo code in Algorithm. 1.

Algorithm 1: LocNode Construction

Input: Poses $P[0{:}N{-}1]$, Detections $D[0{:}N{-}1]$
Output: Location nodes $\mathcal{L}$

$\mathcal{L} \leftarrow \emptyset$, $loc\_id \leftarrow 0$;
$P_{\text{prev}} \leftarrow P[0]$;
$d \leftarrow \emptyset$ `// Detection buffer`
for $t \leftarrow 0$ **to** $N{-}1$ **do**
  $P_t \leftarrow P[t]$;
  $d \leftarrow d \cup \{D[t]\}$;
  $t_\Delta \leftarrow \Delta\text{Translation}(P_{\text{prev}}, P_t)$;
  $r_\Delta \leftarrow \Delta\text{Rotation}(P_{\text{prev}}, P_t)$;
  if $t_\Delta > T_{thres}$ ***or*** $r_\Delta > R_{thres}$ **then**
    `// Create new location node from buffered detections`
    $\mathcal{L} \leftarrow \mathcal{L} \cup$ **LocNode**(loc_id, d);
    $loc\_id \leftarrow loc\_id + 1$;
    $d \leftarrow \emptyset$, $P_{\text{prev}} \leftarrow P_t$;
return $\mathcal{L}$;

Object Node. In the 3D scene scan, we continuously leverage an open-vocabulary object detector (Cheng et al., 2024a) and a multi-object tracker (Aharon et al., 2022) to obtain object detections and their temporal dependencies across time stamps. In each location node, we can obtain multiple object tracklets from the detector and tracker. For each location, we wrapped each observed tracklet into an object node `ObjNode`. Each `ObjNode` stores a compact trajectory derived from the tracker. The combinations of `ObjNode` at each location ensure a compact, high-level coarse representation, which can enable the LLM querying to reason over the coarse perception content for each location in the scene.

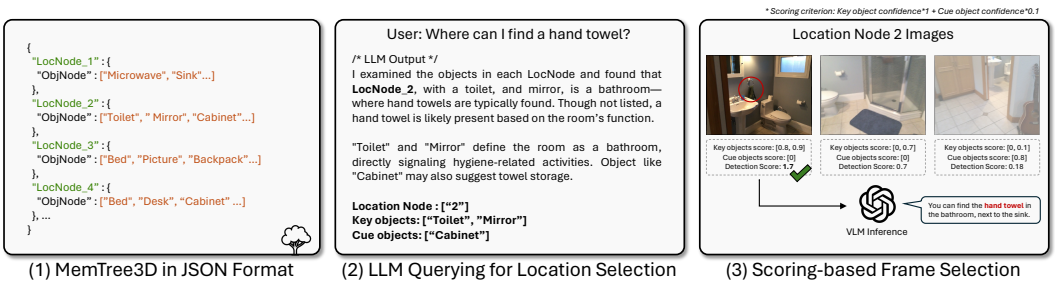


Figure 3: Our LLM key frame selection consists of 3 steps. (1) We transform the constructed MemTree3D representation into JSON format for LLM comprehension and reasoning. (2) The JSON format MemTree3D and the user query will be sent to the LLM to jointly predict the potential Location Node that contains the answer, as well as the Key Object List and Cue Object List for later frame selection. (3) We use the Key/Cue Object List and detection confidence to select the key frame for VLM input. Note that the LLM select ***k* location nodes** in our default implementation, here we only present one node for simplicity.

Detection Node. Detection node `DetNode` serves as the leaf component in MemTree3D, representing the 2D detection obtained in each frame. Each `DetNode` contains only the basic attribute, including the detection time-stamp, detection results like bounding box coordinates, and detection confidence representing the quality of the detection, which will be used in the later frame selection.

3.2 LLM Querying for Location Selection

After the MemTree3D construction, we now obtain a compact 3D scene representation. Given a user question Q, we leverage the LLM to determine which locations might contain the answer to this given question Q. Here, we transform the constructed MemTree3D representation into JSON format and feed it to the LLM, as shown in Fig. 3. Note that this JSON format only includes the first two levels of nodes (`LocNode` and `ObjNode`), and `DetNode` is not included as it is not heavily relevant to the LLM location node selection process. Nextly, we prompt the LLM to conduct reasoning based on the `ObjNode` in each location, and select the top k locations that might contain important visual information for the answer. In the scenario where the object in the user question is detected, LLM can directly predict the correct locations without extensive reasoning. However, when an object is not detected due to a detection failure, our framework demonstrates its robustness through the LLM reasoning over the MemTree3D and predicts the most possible locations that contain the answer to the query. This distinct our work from existing visual search and scene-graph methods, which are not able to answer the question correctly when the vision model fails to identify the critical objects. Besides predicting the top k possible locations, we also prompt the LLM to provide two object lists, including key objects O_{key} and cue objects O_{cue}. Key objects are listed as `ObjNode` that can help locate the answer to the question, and cue objects listed `ObjNode` that might be close or near the key objects. These two object lists serve as the criterion for the following scoring-based frame selection.

3.3 Scoring-based Frame Selection

For the k locations suggested by the LLM in previous step, we select **one** key frame from the video frames captured in each location node, resulting in a total k key frames that are related to the user's question. The sampling process involves selecting frames that are most informative with respect to the user question, using our scoring-based frame selection illustrated in section (3) of Fig. 3. In our scoring-based frame selection strategy, we leverage the O_{key} and O_{cue} from the last step to select the key frame using the detection confidence score as criterion. Within each frame, the detected object $o \in O_{key}$ and cue objects O_{cue} are assigned with different weight factors to reflect frame importance. This weighted scoring mechanism helps prioritize frames that are semantically aligned with the query while also providing relevant contextual cues. Formally, we compute a score for each frame f using the sets of **key objects** O_{key} and **cue objects** O_{cue}:

Table 1: Performance comparison of our MemTree3D with other SoTA methods on OpenEQA, including the performance on ScanNet and HM3D subset. Performance of MemTree3D with different number of used frame can be found in ablation study.

Method	Avg. Frame	LLM-Match		
		ScanNet	HM3D	ALL
Open-source VLM				
Video-LLaMA (Zhang et al., 2023)	8	20.1	19.8	20.0
Video-ChatGPT (Maaz et al., 2024)	100	32.9	30.4	32.1
LLaMA-2 w/ Concept Graph (Majumdar et al., 2024)	50	31.0	24.2	28.7
LLaMA-2 w/ Sparse Voxel Map (Majumdar et al., 2024)	50	36.0	30.9	34.3
LLaMA-2 w/ LLaVA-1.5 caption (Majumdar et al., 2024)	50	39.6	31.1	36.8
Qwen-2.5-VL-7B (Bai et al., 2025)	12	49.4	40.2	46.2
Video-LLaMA2 (Cheng et al., 2024b)	16	50.1	47.5	49.2
LLaVA-3D (Zhu et al., 2024a)	32	–	–	53.2
Closed-source VLM				
Claude-3 Opus (Anthropic, 2024)	20	–	–	36.3
Gemini 1.0 Pro Vision (Team et al., 2023)	15	–	–	44.9
Claude-3.5 Sonnet (Anthropic, 2024)	20	–	–	48.7
GPT-4V w/ Concept Graph (Majumdar et al., 2024)	50	37.8	34.0	36.5
GPT-4V w/ Sparse Voxel Map (Majumdar et al., 2024)	50	40.9	35.0	38.9
GPT-4V w/ LLaVA-1.5 caption (Majumdar et al., 2024)	50	45.4	40.0	43.6
GPT-4V (Achiam et al., 2023)	50	57.4	51.3	55.3
GPT-4o (Hurst et al., 2024)	12	63.9	58.4	62.0
Visual Search Key Frame Sampling Method				
Image-Text Retrieval (Radford et al., 2021)	3	55.1	41.9	50.6
3D-Mem (Yang et al., 2025b)	3.1	–	–	57.2
VLM-Grounder (Xu et al., 2024a)	6	65.1	58.8	63.0
Ours (with zero-shot 2D VLM)				
LLaVA-OneVision-7B (Li et al., 2024a)	3	52.5	42.9	49.2
w/ MemTree3D	3	57.9 (+5.4)	49.3 (+6.4)	55.0 (+5.8)
GPT-4o (Hurst et al., 2024)	3	55.8	37.1	49.4
w/ MemTree3D	3	**69.4 (+13.6)**	**61.7 (+24.6)**	**66.8 (+17.4)**

$$\text{Score}(f) = \sum_{d \in D_f} \begin{cases} s(d), & \text{if } d \in O_{\text{key}}, \\ \lambda \cdot s(d), & \text{if } d \in O_{\text{cue}}, \\ 0, & \text{otherwise.} \end{cases}$$

where D_f is the set of all detections in frame f, $s(d)$ is the confidence score of detection d, and λ is a weighting factor that down-weights cue objects relative to key objects which we we set to 0.1. This design ensures that target objects contribute their full confidence while cue objects provide auxiliary context at a reduced weight, encouraging the selection of frames that capture both direct evidence and relevant supporting cues. For each location, we then choose the frame with the highest score. Finally, the user question and the k selected key frames across different locations and viewpoints are passed to the VLM for question answering.

4 Experiments

4.1 Implementation Details

MemTree3D. We use YOLO-World (Cheng et al., 2024a) and BoT-SORT (Aharon et al., 2022) as tracker for MemTree3D construction. The tree construction can be run in online and real-time (**25+ FPS**) on a single GPU, and can be easily integrate to an embodied agent during the observation collection stage compared to existing methods that require multiple vision foundation models (Gu et al., 2024; Jatavallabhula et al., 2023). The construction threshold T_{thres} and R_{thres} is set to `1.5m` and `45°` for all the experiments, note that these thresholds are not carefully tuned but selected based on reasonable spatial and angular displacements to segment the 3D scene. Despite this, tuning these parameters for each scene could potentially yield better `LocNode` segmentation. However, we use the same parameter setting across all 3D scenes, which makes our design more closely aligned with

Table 2: EM@1 comparison on ScanQA and SQA3D.

Method	Avg. Frame	EM@1 ScanQA	EM@1 SQA3D
3D Fine-tuned Model			
Scan2Cap (Chen et al., 2021)	–	–	41.0
ScanRefer+MCAN (Chen et al., 2020)	–	18.6	–
ClipBERT (Lei et al., 2021)	–	–	43.3
ScanQA (Azuma et al., 2022)	–	21.1	47.2
3D-VisTA (Zhu et al., 2023)	–	22.4	48.5
3D-LLM (Hong et al., 2023)	–	20.5	48.1
3D-VLP (Jin et al., 2023)	–	21.6	–
LEO (Huang et al., 2024b)	–	24.5	50.0
ChatScene (Huang et al., 2024a)	–	21.6	54.6
Zero-shot 2D VLM			
Agent3D-zero (Zhang et al., 2024)	24	17.5	–
LLaVA-Next-Video (Liu et al., 2024a)	32	18.7	34.2
VideoChat2 (Li et al., 2024b)	16	19.2	37.3
Ours (with zero-shot 2D VLM)			
LLaVA-OneVision-7B (Li et al., 2024a)	3	25.1	46.2
w/ MemTree3D	3	**28.0 (+2.9)**	**49.6 (+3.4)**

real-world applications, as the number of `LocNode` can adjust adaptively according to the size of the 3D scene (see Tab. 6 in appendix). Also note that we keep our MemTree3D design from merging the `LocNode` even when the observation revisits the same location. This preserves the temporal dimension of the memory that more closely reflects real-world settings. Furthermore, the resulting growth in JSON size remains negligible in practice compared to the computational savings gained from our approach. We conducted all the experiments on a single V100.

LLM Querying. The LLM select top k locations from MemTree3D after reasoning. And the scoring-based key frame selection select one frame for each location. In our experiments, we default k to 3, with ablation study on different value of k (selecting more `LocNode`) in Fig. 4. We include more implementation details including LLM prompt for location selection in appendix.

Benchmark. We evaluate MemTree3D on OpenEQA, ScanQA and SQA3D. OpenEQA is a recent Embodied Question Answering (EQA) benchmark focusing on spatial understanding and embodied reasoning. It contains 187 episode histories collected from ScanNet (Dai et al., 2017) and HM3D (Ramakrishnan et al., 2021), with over 1,600 human-generated questions. Furthermore, OpenEQA adopts the automatic LLM evaluation protocol to evaluate the performance of the method. We follow the official setting and report the GPT-4 (Achiam et al., 2023) LLM-Match score. ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2023) are another two large-scale benchmarks that focus on 3D scene spatial understanding, with ScanQA contains 4,675 and SQA3D contains 3,519 QA pairs. We follow previous works (Zhu et al., 2024a; Huang et al., 2024a; 2025b) and evaluate on ScanQA validation and SQA3D test set using Exact Match (EM@1). More benchmark details and statistics can be found in our appendix.

Baselines. On OpenEQA (Table. 1), we compared with multiple SOTA VLMs, captioning-based Socratic methods (Liu et al., 2024b), Concept Graph and Sparse Voxel Map (Gu et al., 2024) LLM methods. We also compared with three visual search methods, including detector-based VLM Grounder (Xu et al., 2024a), VLM-based frame selection 3D-Mem (Yang et al., 2025b) and an image-text retrieval baseline with CLIP (Radford et al., 2021) that retrieve key frames by selecting the frame with highest image-query similarity. On ScanQA and SQA3D (Table. 2), we compared with 3D fine-tuned models, including task-specific models (Chen et al., 2021; 2020; Lei et al., 2021;

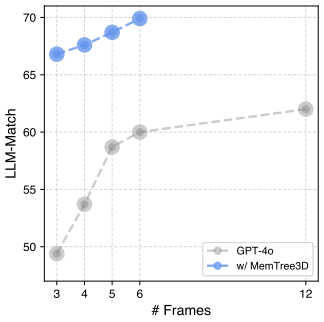


Figure 4: OpenEQA performance comparison with uniform sampling using different number of frames.

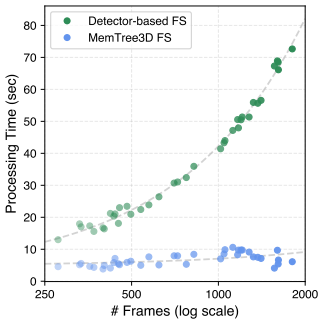


Figure 5: Runtime comparison with Detector-based frame sampling method. Runtime is measured after receiving question.

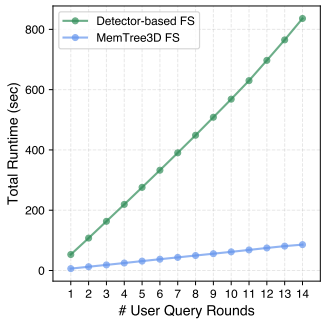


Figure 6: Runtime comparison with Detector-based frame sampling method when processing multi-round query.

Azuma et al., 2022) and 3D understanding LLMs (Zhu et al., 2023; Hong et al., 2023; Jin et al., 2023; Huang et al., 2024b;a). We also compared with recent VLM-based key frame selection methods (Zhang et al., 2024; Yang et al., 2025b) and 2D VLMs (Liu et al., 2024a; Li et al., 2024b;a) using uniform sampling.

Analysis. In Table. 1, we show our MemTree3D performance implemented with open-source VLM LLaVA-One-Vision-7B and proprietary VLM `GPT-4o` on the OpenEQA benchmark. Our methods largely boost the performance of existing VLMs, with **5.8%** accuracy gain for LLaVA-One-Vision, and **17.4%** over `GPT-4o` under the same number of input frame. Moreover, we achieve stronger performance compared with existing visual search methods using the same VLM `GPT-4o`. On ScanQA and SQA3D results in Table. 2, MemTree3D also brings accuracy improvements, and achieve strong zero-shot performance compared with existing 3D fine-tuned models and 2D VLM. The performance gain on ScanQA and SQA3D is rather moderate compared with the large performance improvements on OpenEQA, this is likely caused by the smaller scene size in ScanNet, where the uniform sampling can already cover most visual information. This observation can also be justified by the improvement gap on the ScanNet subset (avg. scene size 82.6 m^3) and HM3D subset (avg. scene size 556.0 m^3) in OpenEQA. This suggests our method is more effective in a more challenging and larger 3D scene setting.

4.2 Ablation Studies

Does the number of frames affect performance? We evaluate the impact of using a larger k value for inference in Fig. 4. Our MemTree3D outperforms uniform sampling across different frame usage, highlighting the effectiveness of our method. Moreover, the performance of `GPT-4o` when using frames selected by MemTree3D consistently improves as more frames are included, due to access to richer visual information from diverse spatial locations in the scene.

How efficient is MemTree3D? To demonstrate that our proposed paradigm is more efficient than visual search key frame selection, we compare MemTree3D Frame Selection (MemTree3D FS) with the Detector-based Frame Selection (Detector-based FS) method (Xu et al., 2024a) in Fig. 5. We use the open source LLM Qwen3 (Yang et al., 2025a) for both methods and run them on the same V100 GPU for fair comparison and reproducibility. The runtime of Detector-based FS increases with the number of video frames, while the runtime of MemTree3D FS remains relatively stable, thanks to its pre-build compact representation, and more efficient LLM key frame querying paradigm. The reported runtime is measured on videos from OpenEQA, starting from the moment the system receives the user query. Furthermore, we compare the latency of MemTree3D and Detector-based FS in multi-round user query scenario in Fig. 6, the latency gap highlights the better efficiency our MemTree3D key frame selection paradigm compared with existing visual search approach. This efficiency stems directly from our design, which bypasses extensive per-query visual search and thereby minimizes computation. More experiment details can be found in the appendix.

Table 3: Performance of different LLM/VLM combinations. Qwen3-8B* experiment is run on 98.8% of the OpenEQA question as it fails to follow our specified JSON format in the location selection process for a small portion (1.2%) of questions despite our best effort in prompt engineering.

LLM	VLM	ScanNet	HM3D	ALL
–	GPT-4o (Uniform Sampling)	55.8	37.1	49.4
Qwen3-8B*	LLaVA-OneVision-7B	56.8	46.0	53.1
GPT-4o	LLaVA-OneVision-7B	57.9	49.3	55.0
GPT-4o	GPT-4o	69.4	61.7	66.8

Can MemTree3D work well with smaller models? In Table. 3, we conduct experiments and investigate the performance of various open source and proprietary LLM and VLM combinations with our MemTree3D. We found that switching proprietary LLM `GPT-4o` to open source small LLM Qwen3 (Yang et al., 2025a) does not introduce large performance degradation, with its 53.1 LLM-Match score on OpenEQA outperforms uniform sampling `GPT-4o`.

Is MemTree3D robust to detector failure? To evaluate the robustness of our framework under detector failures during the MemTree3D construction, we analyze the performance gap between whether the question-mentioned objects are successfully detected in Table. 4. We adopt a heuristic string matching to check whether the detected object class names appeared in the question. On OpenEQA, our key frame selection is mostly operate under detection failure, with only 25.8% contain object class name that are successfully detected, while for 74.2% of questions the detector fails to directly identify the question-mentioned objects. However, our method still achieves 65.9 LLM-Match on these undetected cases—just a 3.1 point drop compared to those cases where question-related objects are directly detected. This highlights the robustness of our LLM key frame selection, which does not over-rely on vision model, and further differentiates our work with the existing visual search and scene graph based methods. We presents more qualitative results of the LLM deduct the critical key frames through our MemTree3D in Fig. 8.

Table 4: Robustness evaluation under detector failure.

Success Detection	# questions	ScanNet	HM3D	All
✓	358 (25.8%)	71.4	64.3	69.0
✗	1,029 (74.2%)	67.9	62.1	65.9

Does 3D scene complexity affect performance? A critical evaluation for our framework is its robustness in complex 3D scenes with significant clutter. To quantify scene complexity, we introduce a clutter level, defined as the total number of unique objects divided by the number of `LocNode` in a 3D scene. A higher clutter level indicates more objects per location, thereby increasing the difficulty of selecting the correct location and frame. As shown in Fig. 7, our method's LLM-Match score on OpenEQA benchmark does not degrade significantly as the clutter level increases. This result demonstrates that our framework's LLM-based selection mechanism is not confounded by distractor objects, enabling it to intelligently identify the most salient locations and keyframes for efficient and accurate question answering.

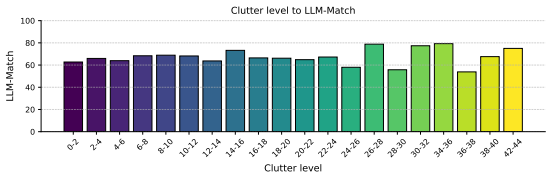


Figure 7: Performance of MemTree3D under different level of scene complexity (scene clutter level).

Frame Storage Feasibility and Efficiency. One potential concern with our approach is the memory requirement introduced by storing the RGB frames for later retrieval and VLM inference. However, we emphasize that the benefits of our method substantially outweigh this cost. First, the Video RAM (VRAM) requirement of running the VLM on a large number of frames imposes far greater cost than storing frames on conventional storage media such as SSD/HDD or System RAM. Second, modern embodied devices can easily accommodate extended video storage: for example, a standard memory card can store several days of high-resolution footage. Therefore, the primary

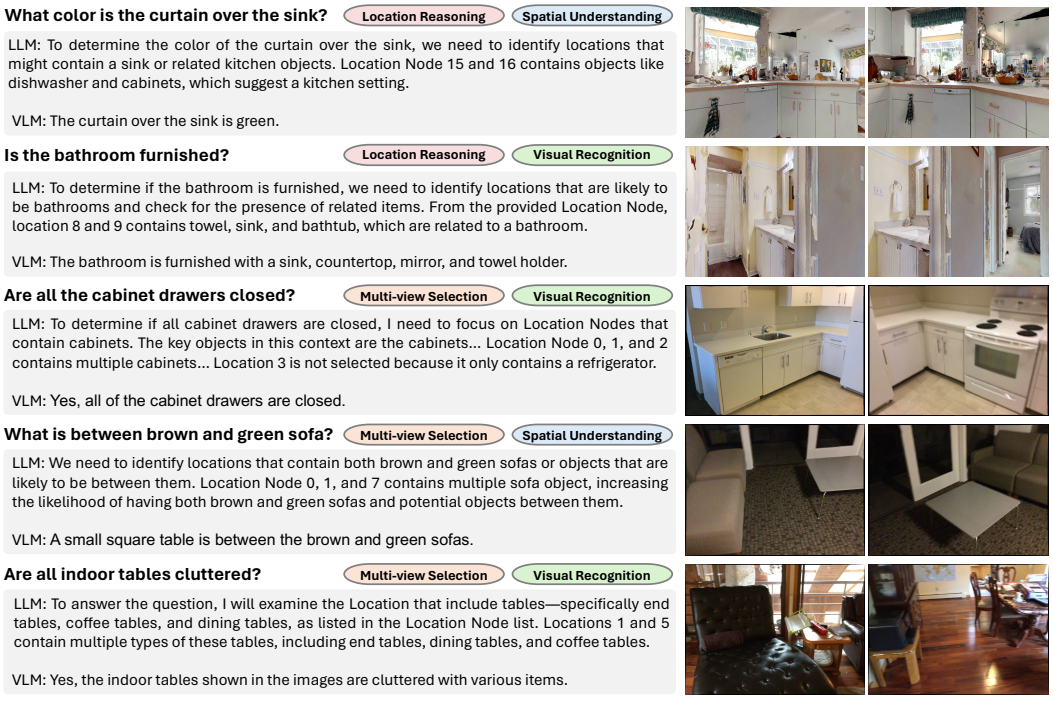


Figure 8: Qualitative results from the OpenEQA. We present the top two selected key frames by our MemTree3D framework, as well as the corresponding LLM location selection and VLM output. More results can be found in supplemental materials.

challenge lies in the efficient retrieval of relevant frames, which is precisely addressed by our work's memory-efficient mechanism for 3D question answering task.

4.3 Qualitative Results

We present some qualitative results in Fig. 8, with part of the LLM location selection process and the VLM output. These qualitative results cover multiple question types including:

Location Reasoning. Question that does not directly indicate the key objects due to detection failure or user query does not include any object, and it requires the LLM to conduct reasoning over the MemTree3D and determine the most possible locations from the objects within each location.

Multi-view Selection. Require retrieving multiple images from various viewpoints. Our method can select the critical images from multiple location nodes to answer the question.

Spatial Understanding & Visual Recognition. Our proposed MemTree3D key frame selection enables identifying the most visually informative frames with respect to user's question across different locations in the scene, enhancing the VLM's question answering accuracy.

5 Conclusion

In this work, we introduce a memory tree guided key frame selection framework for efficient and accurate 3D question answering. By building a compact, hierarchical 3D scene representation, our MemTree3D enables symbolic LLM reasoning over the scene and key frame selection without exhaustive visual search. This reduces computational cost and latency, supporting multi-turn querying in embodied settings. Experiments on OpenEQA, ScanQA, and SQA3D show consistent performance gains across both open source and proprietary VLMs with fewer input frames, demonstrating potential for real-world embodied AI applications.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.

Anthropic. Claude 3 opus, March 2024. URL `https://www.anthropic.com/news/claude-3-family`. Family of Next-Generation AI Models.

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*, pp. 5664–5673, 2019.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19129–19139, 2022.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 961–970, 2015.

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pp. 202–221. Springer, 2020.

Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26428–26438, 2024.

Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3193–3203, 2021.

Jen-Hao Cheng, Vivian Wang, Huayu Wang, Huapeng Zhou, Yi-Hao Peng, Hou-I Liu, Hsiang-Wei Huang, Kuang-Ming Chen, Cheng-Yen Yang, Wenhao Chai, et al. Tempura: Temporal event masked prediction and understanding for reasoning in action. *arXiv preprint arXiv:2505.01583*, 2025.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16901–16911, 2024a.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024b.

Anoop Cherian, Chiori Hori, Tim K Marks, and Jonathan Le Roux. (2.5+ 1) d spatio-temporal scene graphs for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 444–453, 2022.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pp. 75–92. Springer, 2024.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24108–24118, 2025.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13504–13514, 2024.

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. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

Yining Hong, Zishuo Zheng, Peihao Chen, Yian Wang, Junyan Li, and Chuang Gan. Multiply: A multisensory object-centric embodied large language model in 3d world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26406–26416, 2024.

Jian Hu, Zixu Cheng, Chenyang Si, Wei Li, and Shaogang Gong. Cos: Chain-of-shot prompting for long video understanding. *arXiv preprint arXiv:2502.06428*, 2025.

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Hsiang-Wei Huang, Wenhao Chai, Kuang-Ming Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. Tosa: Token merging with spatial awareness. *arXiv preprint arXiv:2506.20066*, 2025a.

Hsiang-Wei Huang, Fu-Chen Chen, Wenhao Chai, Che-Chun Su, Lu Xia, Sanghun Jung, Cheng-Yen Yang, Jenq-Neng Hwang, Min Sun, and Cheng-Hao Kuo. Zero-shot 3d question answering via voxel-based dynamic token compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19424–19434, 2025b.

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. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024b.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

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.

Zhao Jin, Munawar Hayat, Yuwei Yang, Yulan Guo, and Yinjie Lei. Context-aware alignment and mutual masking for 3d-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10984–10994, 2023.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7331–7341, 2021.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023a.

Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023b.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024b.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024b.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024c.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=IDJx97BC38.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16488–16498, 2024.

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*, pp. 8748–8763. PMLR, 2021.

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.

Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. Moviechat+: Question-aware sparse memory for long video question answering. *arXiv preprint arXiv:2404.17176*, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

FENGYUN WANG, Sicheng Yu, Jiawei Wu, Jinhui Tang, Hanwang Zhang, and Qianru Sun. 3d question answering via only 2d vision-language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=IkhJApkJQ3.

Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3272–3283, 2025.

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 (CVPR)*, pp. 7515–7525, June 2021.

Wenhao Wu. Freeva: Offline mllm as training-free video assistant. *arXiv preprint arXiv:2405.07798*, 2024.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9777–9786, 2021.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1645–1653, 2017.

Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. VLM-grounder: A VLM agent for zero-shot 3d visual grounding. In *8th Annual Conference on Robot Learning*, 2024a. URL https://openreview.net/forum?id=IcOrwlXzMi.

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*, pp. 131–147. Springer, 2024b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3d-mem: 3d scene memory for embodied exploration and reasoning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17294–17303, 2025b.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Re-thinking temporal search for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8579–8591, 2025.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

Sha Zhang, Di Huang, Jiajun Deng, Shixiang Tang, Wanli Ouyang, Tong He, and Yanyong Zhang. Agent3d-zero: An agent for zero-shot 3d understanding. In *European Conference on Computer Vision*, pp. 186–202. Springer, 2024.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8995–9006, 2025.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024a.

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2911–2921, 2023.

Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. Unifying 3d vision-language understanding via promptable queries. In *European Conference on Computer Vision*, pp. 188–206. Springer, 2024b.

A Appendix

We present more details and experiment results in the supplementary material structured as follows:

- Evaluation benchmark details.
- LLM prompt for key frame querying.
- More results on efficiency analysis.
- Ablation study on location node construction.
- Additional qualitative results.
- Limitations and failure cases.
- Disclosure of AI Assistance

B Evaluation Benchmarks Details

Table 5: Scale comparison of the evaluated 3D question answering benchmarks in our work.

Benchmark	# of scenes	# of questions
OpenEQA (Majumdar et al., 2024)	**152**	1,636
ScanQA (Azuma et al., 2022)	71	**4,306**
SQA3D (Ma et al., 2023)	67	3,519

Table 6: Statistics of the ScanNet and HM3D subset from the OpenEQA. The table shows the number of scenes, questions, and the average scene size.

Subset	# scenes	avg. size (m^3)	avg. LocNode	Δ LLM-Match
ScanNet	89	82.6	5.9	+13.6
HM3D	63	556.0	20.3	+24.6

In this work, we evaluate on 3 different benchmarks, including OpenEQA (Majumdar et al., 2024), ScanQA (Azuma et al., 2022), and SQA3D (Ma et al., 2023). We provide detailed statistics of these benchmarks in Tab. 5. Among these benchmarks, OpenEQA has the most 3D scenes, featuring 3D scene collected from ScanNet (Dai et al., 2017) and HM3D (Ramakrishnan et al., 2021). And ScanQA having the largest scale with 4,306 questions.

We also compare the two subsets from OpenEQA in Table. 6, including the number of 3D scene, average 3D scene size, average number of constructed `LocNode`, and the performance gain when using our MemTree3D. As discussed in our main paper, we notice a larger performance gain in the HM3D subset compare with the ScanNet subset, we believe this is caused by the larger 3D scene size in HM3D. In ScanNet, the 3D scene size is much smaller, and uniform sampling serve as a strong baseline strategy to cover all the visual information from the 3D scene. However, in HM3D, the 3D scene size is much larger, and uniform sampling fails to cover all the visual details. Our proposed key frame selection strategy can address this challenges by selecting the most relevant key frame to the question, with a significant 24.6% performance gain on HM3D subset.

C Efficiency Analysis

We compare the runtime efficiency of our MemTree3D key frame selection framework against existing visual search-based approach, as shown in Fig. 5. Specifically, we measure the latency between receiving a user question and producing a response—an essential factor in real-world embodied AI scenarios. In such settings, an agent typically navigates in a 3D scene to collect visual observations and then answers user queries based on the visual stream. The key performance metric is

the response time after receiving the user question. Visual search-based methods, such as VLM-Grounder (Xu et al., 2024a), involve a multi-stage pipeline after receiving a question, resulting in substantial latency. The process typically includes:

1. Using an LLM to parse the question and extract a list of target objects relevant to the query.
2. Running open-vocabulary object detector on all video frames using the identified target objects as text prompts.
3. Sending the subset of frames containing detected objects to a VLM for question answering.

Among these steps, step 2 leads to large latency issue due to the extensive computational cost growing proportionally with the number of video frames. In contrast, our tree based frame selection framework pre-build the 3D representation before receiving the user's question, instead of searching the answer in the video frames, our work search the key frame from the constructed MemTree3D, therefore bypass this extensive visual search process, and incur minimal latency regardless of the video length, as illustrated in Fig. 5.

D Implementation Details

LLM/VLM Inference Parameters. We set the inference temperature of `GPT-4o` and LLaVA-OneVision-7B to 0.0, and 0.6 for Qwen3 following the default setting. The rest of the inference parameters follow the default generation configuration from their corresponding huggingface repositories and official API setting.

LLM Location Selection Prompt. We present the full prompt used for location selection with `GPT-4o` in Fig. 9. The prompt includes a one-shot example illustrating the expected JSON response format, along with custom tags such as `<think>` and `<answer>` to guide the model's reasoning and output parsing.

Image-text Retrieval Baseline. We implement the Image-text retrieval baseline using CLIP (Radford et al., 2021) from the `openai-clip-vit-base-patch32` checkpoint. We directly use the user question as input text and compare the image-text cosine similarity to retrieve the top k images for `GPT-4o` VLM input.

E Additional Qualitative Results

We present more qualitative results in Fig. 10. With more frame selection results comparison with VLM image-text retrieval baseline method (Radford et al., 2021). Here, we present several failure cases of the VLM image-text retrieval baseline, including 1. VLM fails to retrieve any key frames related to question (row 1), where the key frame selection required question reasoning and location understanding. 2. VLM samples key frames from the similar location and viewpoint, and failed to retrieve correct key frame for the question (row 2-4), as direct selecting top-k frames based on similarity score can leads to over sample on the similar location and viewpoint, our MemTree3D segments the 3D scene into multiple `LocNode` and alleviate this issue.

F Limitations and Failure Cases

Despite the better efficiency and significant performance gain over existing visual search key frame selection method, our MemTree3D still have its limitation. Our tree based key frame selection paradigm can occasionally failed in some challenging object localization question where the object is not directly detected. More specifically, when the object localization question is querying an entirely novel object that is not part of MemTree3D, the LLM during key frame querying process will conduct its best deduction and reasoning to select several most possible locations that the target object might appear, but this best effort of reasoning does not always yield success object localization. We illustrate several failure cases in Fig. 11, where the LLM select several reasonable locations but failed to retrieve the target objects.

Location Node (LocNode): contains Object Node from multiple locations with similar spatial coordinates. It contains a unique Location ID.
Object Node (ObjNode): contains an object's multiple Detection Node across a period of time span. Object Node include information such as object class name, object tracking ID.

Now, I will prompt you a question, and you will be given a list of Location Node.

Based on the provided Location Node list, observe the object node contain in each location node, select <top_k_locs> location node that you think might contain critical visual information that can help answer the given question.
Next, select Object Node within each Location Node that belongs to Key objects (critical object that is related to the answer)
And then, select Object Node within each Location Node that belongs to Cue objects (objects that might be near key objects)

Provide the Location ID (key) and key_objects and cue_objects (value) in the following format.

<think> Conclude the <top_k_locs> location you want to select based on your best guess. You must select <top_k_locs>!</think>
<answer>
{
 "0": {
  "key_objects": ["tv"],
  "cue_objects": ["picture", "clock"]
 },
 "1": {
  "key_objects": ["tv"],
  "cue_objects": ["picture", “desk"]
 },
 "2": {
  "key_objects": [“tv”, ”monitor”],
  "cue_objects": ["picture” , “tv stand”]
 }
}
</answer>

Question: <question>
Location Node list : <MemTree3D_JSON>

Figure 9: LLM prompt for MemTree3D querying and location selection.

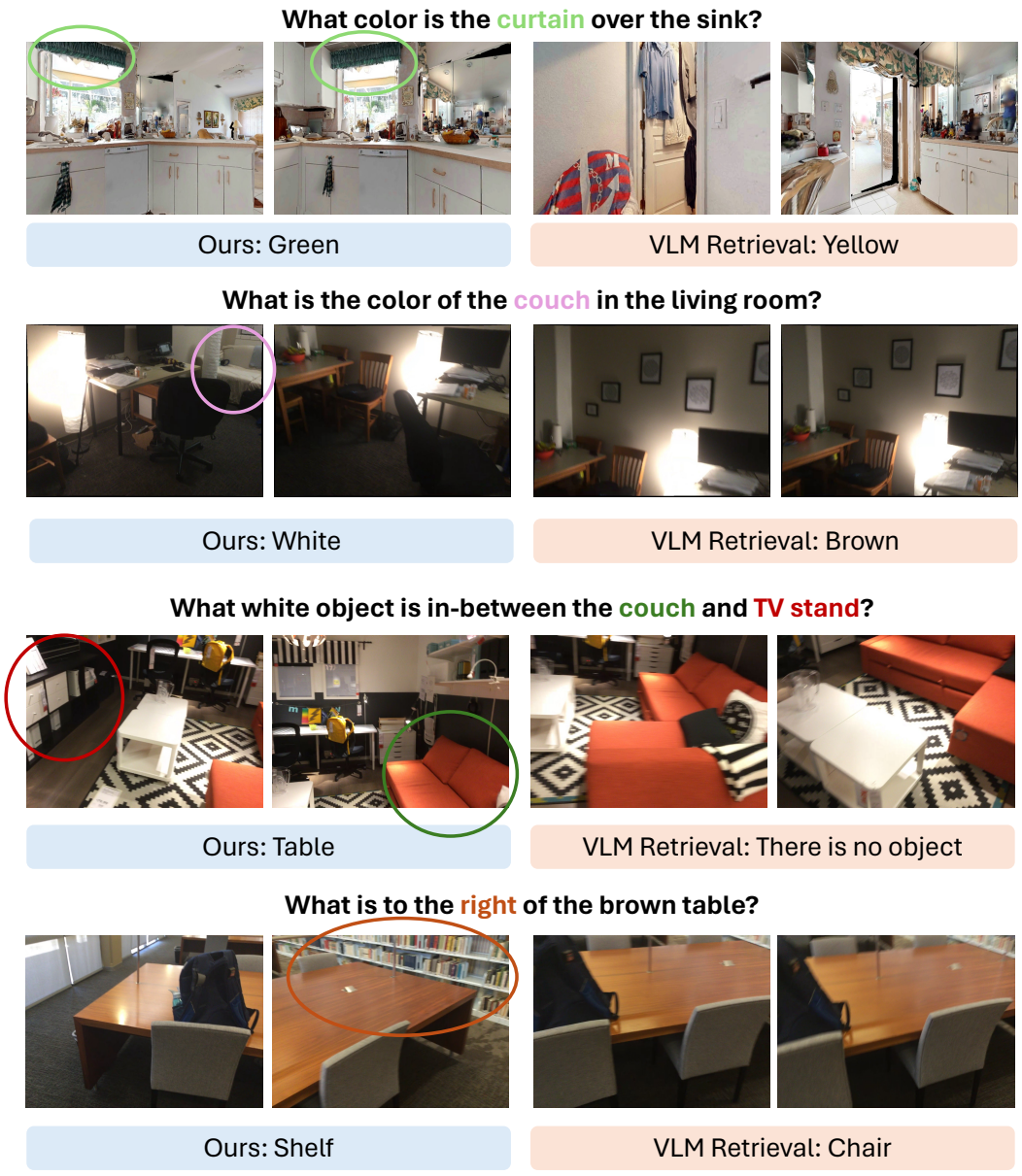
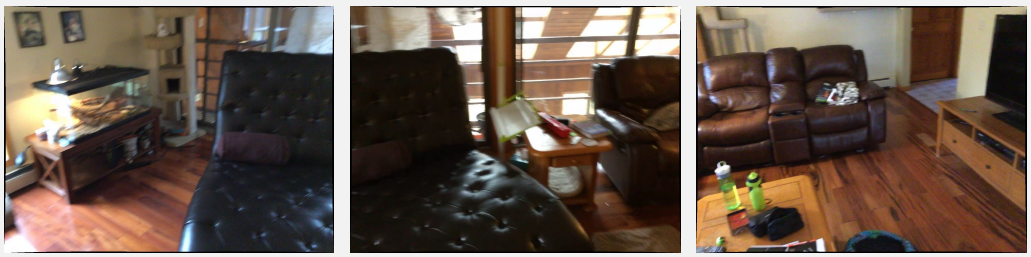


Figure 10: Additional qualitative results comparison between our MemTree3D framework and VLM retrieval baseline. We showcase several cases, including 1. VLM fails to retrieve any key frames related to question (row 1). 2. VLM samples key frames from the similar location and viewpoint, and failed to retrieve correct key frame for the question (row 2-4).

Where is the cardboard cat scratcher?



LLM Location Selection

To determine where the cardboard cat scratcher might be, I need to identify locations that are likely to contain pet-related items or areas where a cat might be present. Common places for a cat scratcher could be near furniture like couches, chairs, or in living areas. Based on the provided Location Node list, I will look for locations with objects that suggest a living room or common area setting, such as couches, chairs, or other furniture.

1. Location 0: Contains objects like couch, pillow, tv_stand, and end_table, which are typical in a living room setting where a cat scratcher might be found.
2. Location 1: Contains multiple couches, armchairs, and coffee tables, indicating a living room or lounge area.
3. Location 3: Contains a couch, tv, and armchair, which are also indicative of a living room setting. These locations seem most likely to contain a cardboard cat scratcher based on the presence of furniture typically found in living areas.

VLM Inference on Key Frames

In the first image, there is a cat tree visible, but no cardboard scratcher is apparent. In the second image, under the small table, there is a white object that resembles a cardboard cat scratcher. The third image does not show any additional objects that match the description of a cardboard cat scratcher.

Figure 11: A failure case of an novel object localization question from the OpenEQA. The question asked to locate where is the cardboard cat scratcher, because the cat scratcher is not part of the `ObjNode` in MemTree3D, our LLM conduct its best reasoning effort to select the most possible locations that the cat scratcher can be. However, the final selected frames does not contain a clear view for VLM to answer the question.

G DISCLOSURE OF AI ASSISTANCE

In accordance with the ICLR 2026 policy on the disclosure of language model usage, we acknowledge the use of Large Language Models in the preparation of this manuscript. Specifically, ChatGPT (developed by OpenAI) was employed to assist in refining the clarity, grammar, and style of the text. All research ideas, methodology, experimental design, analysis, and conclusions are solely the work of the authors. The role of ChatGPT was limited to editorial assistance, and it did not contribute substantively at the level of a co-author.