2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

KeySG: Hierarchical Keyframe-Based 3D Scene Graphs

Abdelrhman Werby Dennis Rotondi Fabio Scaparro Kai O. Arras

***Abstract*— In recent years, 3D scene graphs have emerged as a powerful world representation, offering both geometric accuracy and semantic richness. Combining 3D scene graphs with large language models enables robots to reason, plan, and navigate in complex human-centered environments. However, current approaches for constructing 3D scene graphs are semantically limited to a predefined set of relationships, and their serialization in large environments can easily exceed an LLM's context window. We introduce KeySG, a framework that represents 3D scenes as a hierarchical graph consisting of floors, rooms, objects, and functional elements, where nodes are augmented with multi-modal information extracted from keyframes selected to optimize geometric and visual coverage. The keyframes allow us to efficiently leverage VLMs to extract scene information, alleviating the need to explicitly model relationship edges between objects, enabling more general, task-agnostic reasoning and planning. Our approach can process complex and ambiguous queries while mitigating the scalability issues associated with large scene graphs by utilizing a hierarchical multi-modal retrieval-augmented generation (RAG) pipeline to extract relevant context from the graph. Evaluated across three distinct benchmarks, 3D object semantic segmentation, functional element segmentation, and complex query retrieval KeySG outperforms prior approaches on most metrics, demonstrating its superior semantic richness and efficiency. See our project page at https://keysg-lab.github.io/**

I. Introduction

A long-standing goal in robotics is to create autonomous agents that can operate effectively in human-centered environments such as homes or offices. These environments are characterized by high object density, semantic richness, and a variety of potential tasks. A key challenge for this goal is the development of a 3D world representation that is simultaneously detailed for precise manipulation and abstract enough for high-level reasoning and long-horizon planning.

3D scene graphs (3DSGs) [1]–[5] have gained significant attention as a powerful representation to address the limitations of purely geometric maps. By modeling the world as a graph where nodes represent entities and edges represent their relationships, 3DSGs impose a structure on raw perception, explicitly linking geometry to semantics. However, current 3D scene graph approaches have two main limitations: first, they are restricted to a predefined set of geometric or semantic relationships, reducing the diversity of tasks and queries they can support. For instance, a 3DSG [6] with edges representing spatial relationships between objects and places would excel in locating objects in large buildings. In contrast, a 3DSG [7] encoding functional relationships would be well suited for tasks that require understanding how functional elements control their objects (e.g, "turn off the oven," where knowing the relationship between the knob and the oven is necessary). A 3D scene graph designed with a predefined set of relationships for a specific task is inherently suboptimal for others.

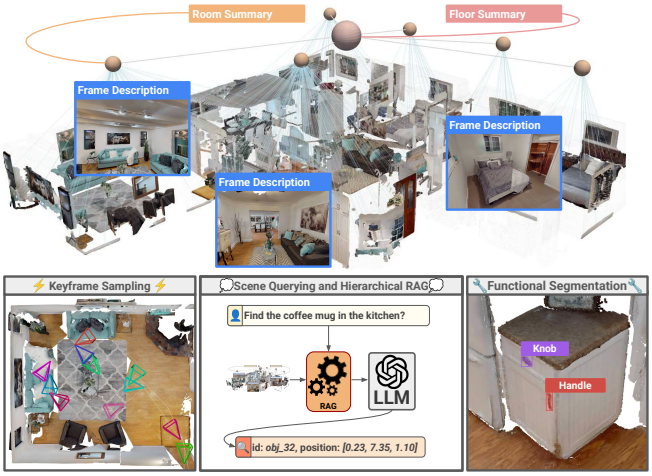

Fig. 1. As illustrated (top), KeySG is a hierarchical, keyframe-based 3D scene graph comprising floors, rooms, objects, and functional elements (bottom right). Each node is augmented with contextual information efficiently extracted from scene keyframes via adaptive keyframe sampling (bottom left). Leveraging a multimodal RAG pipeline, KeySG enables users to ask complex natural language queries and receive answers grounded in the 3D scene (bottom middle).

Second, scalability is a major bottleneck. 3D scene graphs are often paired with large language models (LLMs), serving as a persistent world model that the LLM uses for planning and high-level reasoning. However, providing a complete, detailed scene graph of a large-scale environment, such as a multi-story office building, directly to an LLM can exceed the context window limits of even the most advanced models. Even if the graph fits within the context window, LLMs suffer from attentional biases and a "lost in the middle" problem [8], where performance degrades as the model is distracted by the vast amount of task-irrelevant information present in the prompt. This makes it challenging for the LLM to identify the crucial entities and environmental states necessary for robust reasoning and planning.

To this end, we present the Hierarchical KeyFrame-Based 3D Scene Graphs (KeySG), a novel framework that resolves the semantic and scalability dilemma by augmenting the 3D scene graph with multi-modal contextual information, implicitly capturing the geometry, semantics, affordances, and states of objects. Our key idea is to sample keyframes

All the authors are with the Socially Intelligent Robotics Lab, Institute for Artificial Intelligence University of Stuttgart, Germany. Email: {first.last}@ki.uni-stuttgart.de. A. Werby and D. Rotondi are also part of the International Max Planck Research School for Intelligent Systems (IMPRS-IS).

that ensure comprehensive visual coverage of each room in the environment. A Vision-Language Model (VLM) then generates a detailed description for each keyframe. To address scalability, these descriptions are recursively summarized into concise textual overviews for rooms and, subsequently, entire floors. This hierarchy is queried using a multi-modal retrieval-augmented generation (RAG) pipeline, which ensures that only the most relevant information is retrieved and provided to the LLM planner, enabling efficient and accurate reasoning.

In summary, we make the following contributions:

- We introduce KeyFrame-Based 3DSGs (KeySG), the first 3D scene graph framework designed to hierarchically represent environments spanning from full buildings, to floors, rooms, objects, down to functional elements.
- We propose a new pipeline to augment 3DSGs with multi-modal context from keyframes, featuring hierarchical scene summarization, and a RAG-based retrieval mechanism that efficiently provides task-relevant context to LLMs.
- We conduct a comprehensive evaluation of KeySG across diverse benchmarks, including open-vocabulary 3D segmentation (Replica [9]), functional element segmentation (FunGraph3D [10]), and 3D object grounding from natural language queries (Habitat Matterport [11], Nr3D [12]).

II. Related Work

A. 3D Scene Graphs

The idea of 3D Scene Graphs (3DSGs) [13], [14] is to represent a scene as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V}$ correspond to objects and spatial entities (e.g., cameras, rooms, floors, buildings), while the edges $\mathcal{E}$ encode relationships such as `hierarchical` (e.g., A "is part of" B), `spatial` (e.g., A "is next to" B), and `comparative` (e.g., A "is larger than" B). Unlike geometric maps fused with language features [15]–[17], 3DSGs provide higher-level abstraction, scale naturally to large environments [18], and, thanks to detailed node captions and semantic relationships, have proved useful in robotics tasks such as planning [5], [19]–[21], manipulation [22], [23], and navigation [1], [6].

3DSGs are typically constructed either from dense sequences of RGB-D images [3], [24], [25] or from class-agnostic segmented point clouds [4], [26], [27].

In general, all available input is used to construct the graph, and in some approaches, images that meet specific criteria –for example, those focusing on a particular pair of objects [1], [27], [28]– are used to establish spatial edges, while others are selected to extract functional interactive relationships [7], [10] (e.g., a button "turns on" a monitor). In contrast to our approach, all these solutions explicitly model edges and disregard the RGB input once the graph is constructed, thereby constraining the 3DSG to the specific application for which it was built.

To address this problem, the works of [29], [30] attempt to reason about object primitives at the resolution required for a specific task by mitigating the information bottleneck. However, a key limitation of these methods is that even a slight change in the task requires reconstructing the entire graph from scratch. In contrast, [31] generates a 3DSG from a NeRF, capturing the necessary relationships between pairs of objects, but these relationships are restricted to those modeled by the NeRF at training time. Our method overcomes both issues by simply rendering new edges from our keyframes to answer specific queries.

B. Object Localization with 3D Scene Graphs

3D object localization [32], also known as object grounding, is the task of predicting a target 3D object given a point cloud and a natural language expression as input. Contrary to other approaches [33]–[35], 3DSGs do not need to be trained on ground-truth point clouds, nor do they require fine-tuning of the LLM itself for object grounding. In practice, object localization with 3DSGs is often realized by serializing the graph into JSON [1], [5], [7] and prompting an LLM to find the target object using the graph structure and node features. Different works have tried to optimize the JSON by adding hierarchical structure [22], using the concept of schema [36] or by applying a deductive scene reasoning algorithm [28] that prunes the search space using spatial relationships. All these methods are limited to the information contained in the 3DSG, which is often insufficient to disambiguate complex queries that include specific edges.

III. Technical Approach

This work aims to build a hierarchical 3D scene graph for a large-scale environment, consisting of floors, rooms, objects, and functional elements, where each node is augmented with multi-modal knowledge extracted from keyframes in the environment. Given a posed RGB-D image sequence, we first reconstruct a dense 3D point cloud of the environment and segment it into floors and rooms (Sec. III-A). Within each room, we select keyframes that maximize both geometric coverage and visual informativeness (Sec. III-B). A VLM then extracts textual descriptions, object labels, and functional element labels, which guide an open-vocabulary segmentation pipeline to produce 3D object and functional element segments (Sec. III-C). An LLM subsequently condenses keyframe descriptions into room summaries and aggregates them into floor-level summaries, thereby building contextual information across multiple levels of abstraction within the 3DSG (Sec. III-D). Finally, we introduce a multi-modal hierarchical Retrieval-augmented generation (RAG) pipeline inspired by [37], which exploits the graph's topology to support top-down querying, ensuring that LLMs receive task-relevant information without exceeding their context window (Sec. III-E). Fig. 2 provides an overview of our approach.

A. Hierarchical Scene Segmentation

Given a posed RGB-D sequence $\mathcal{I} = \{P_t, I_t, D_t\}_{t=1}^{T}$ of the environment, where $P_t = [R_t|t_t] \in SE(3)$ is the camera pose, I_t is the RGB image, and D_t is the depth map, we first reconstruct a global 3D point cloud $\mathcal{P}_{scene} \in \mathbb{R}^{N\times 3}$ of the entire scene. Second, we segment the full scene point cloud $\mathcal{P}_{scene}$ into a set of N_F floor point clouds, $\{\mathcal{F}_i\}_{i=1}^{N_F}$. Then we segment each floor point cloud $\mathcal{F}_i$ into a

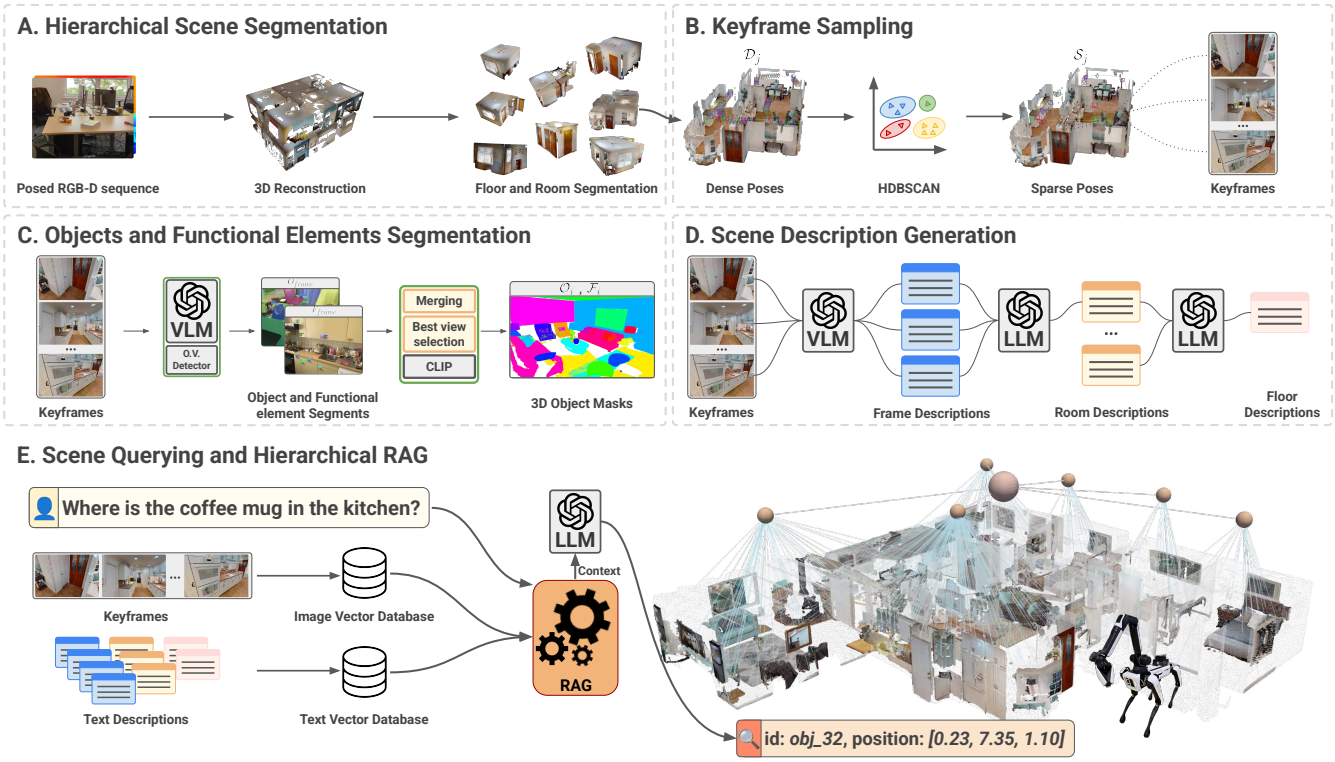

Fig. 2. Overview of KeySG: (A) We first reconstruct the full point cloud of the 3D scene and segment it into floors and rooms. (B) For each room, we select a minimum set of keyframes that provide geometric coverage of the entire space while maximizing visual information. (C) We leverage VLMs to extract object and functional element tags from the keyframes, which guide an open-vocabulary segmentation pipeline to obtain 3D segments of objects and functional elements. (D) We generate geometrically-grounded frame descriptions using a VLM, and employ LLMs to recursively summarize them into dense room and floor-level overviews. (E) To enable efficient querying, we introduce a hierarchical retrieval mechanism grounded in RAG that performs a top-down search, ensuring LLMs receive rich, task-relevant context without exceeding their context window.

set of N_{R_i} room point clouds $\{\mathcal{R}_{ij}\}_{j=1}^{N_{R_i}}$. This hierarchical segmentation follows the approach from [6], segmenting floors by computing a height histogram of the point cloud, then selecting dominant peaks. For room segmentation, we compute a 2D histogram from the bird's-eye view of the floor and apply the Watershed algorithm.

B. Keyframe Sampling

A key principle of KeySG is to augment graph nodes with both raw sensory data and semantic context using VLMs. However, storing and processing the entire data sequence for a large-scale environment is computationally impractical. To address this, KeySG employs a down-sampling procedure that balances computational efficiency with the preservation of critical spatial and visual information. While this problem is known from keyframe-based visual SLAM [38], we select frames based on visual room coverage rather than geometric reconstruction accuracy.

We start by associating each frame from the original sequence with a specific segmented room $\mathcal{R}_{ij}$ (Room j in Floor i). A frame's camera pose is assigned to a room if its associated 3D camera center falls within the room's volumetric boundary, i.e., $t_t \in \text{Vol}(\mathcal{R}_{ij})$. This process yields a dense set of poses for each room $\mathcal{R}_{ij}$, denoted by $\mathcal{D}'_{ij} = \{P_t \in \mathcal{I} \mid t_t \in \text{Vol}(\mathcal{R}_{ij})\}$. To avoid corner cases where the frame's camera pose lies in one room but is viewing a different room or corridor, we further filter $\mathcal{D}'_{ij}$ by requiring that a significant fraction η of the frame's back-projected 3D points also fall within the room's 2D polygon. The resulting refined set is denoted by $\mathcal{D}_{ij}$.

Next, we extract a subset $\mathcal{S}_{ij} \subseteq \mathcal{D}_{ij}$ such that $|\mathcal{S}_{ij}| \ll |\mathcal{D}_{ij}|$. For each pose matrix $P_t \in \mathcal{D}_{ij}$, we extract pose features as 7D vectors $f_t = (t_t; w \cdot q_t)$, where $q_t \in \mathbb{R}^4$ is the quaternion derived from its rotation matrix R_t and w is a scalar rotation weight. The features are then standardized:

$$\tilde{f}_t = \frac{f_t - \mu}{\sigma} \tag{1}$$

where μ and σ are the mean and standard deviation computed across all pose features. We apply HDBSCAN [39] clustering to the set of standardized features, utilizing a minimum cluster size of 15 and a rotation weight $w = 1.5$. This yields a set of clusters $\mathcal{C}_{ij}$. For each cluster $c_k \in \mathcal{C}_{ij}$, we compute the medoid, i.e., the point $f_k^* \in c_k$ that minimizes the sum of distances to all other points within that cluster:

$$f_k^* = \arg\min_{f_j \in c_k} \sum_{f_t \in c_k} \|\tilde{f}_j - \tilde{f}_t\|_2. \tag{2}$$

The final set of selected keyframes, $\mathcal{S}_{ij} = \{f_k^* \mid k = 1, \ldots, |C_{ij}|\}$, is formed by collecting the medoid from each cluster.

C. Objects and Functional Elements Segmentation

After obtaining the representative keyframes for each room, the pipeline proceeds with the 3D segmentation of

objects and their corresponding functional elements. For the keyframes $\mathcal{S}_{ij}$ extracted from each room $\mathcal{R}_{ij}$, we first utilize a VLM [40] to generate a list of object tags and functional element tags for every keyframe. These lists are then aggregated and deduplicated across all keyframes to form the comprehensive tag sets $\mathcal{O}_{ij}$ (objects) and $\mathcal{E}_{ij}$ (functional elements) for the entire room. Next, we use open-vocabulary object detection and segmentation models [41], [42], guided by the object tags $\mathcal{O}_{ij}$, to perform 3D segmentation on the keyframe set of room frames $\mathcal{S}_{ij}$. This yields a set of redundant point clouds in global coordinates for each object. To consolidate these, we incrementally merge objects with significant geometric overlap, following the approach in [6]. Each merged object incorporates 2D masks from multiple viewpoints. For deriving a canonical semantic feature, we apply a best-view selection strategy: each 2D mask is scored based on its size and distance from the image boundaries, prioritizing large, centrally located views. The highest-scoring view is then used to compute a CLIP [43] embedding, ensuring a high-quality feature representation. Finally, we extract the 3D segments of associated functional elements using the best view from each object, leveraging open-vocabulary models [41], [42] with the functional element tags $\mathcal{E}_{ij}$.

D. Scene Description Generation

In KeySG, we argue that scene details, such as layout, semantics, object relationships, state, and affordance, are implicitly stored within the keyframes and their corresponding text descriptions, alleviating the need to explicitly model these specific relations as edges in the 3D scene graph. In this step, we utilize the sparse room keyframes and a VLM to generate a detailed description of each frame. To force the VLM to generate a geometrically-grounded frame description, we determine if an object point cloud is visible from a keyframe's perspective, and we perform a multi-step visibility check. First, object points are transformed into the camera's coordinate system. These points are then projected onto the keyframe 2D image plane, and any points falling outside the image are culled. For the remaining points, we perform an occlusion check by comparing each point's depth to the corresponding value in the keyframe depth map. The object is considered visible if the fraction of its visible points exceeds a minimum threshold θ_{vis}. We feed the VLM with a keyframe image and the list of objects that are visible in it, resulting in a description that is geometrically grounded to the 3D objects we found in the scene. We then aggregate all keyframe descriptions to generate a comprehensive summary for each room. Subsequently, we aggregate all room summaries to generate the overall floor summary.

To construct the final hierarchical 3D scene graph, we assign each extracted context entity to its appropriate level based on spatial and semantic containment. Floors form the top-level nodes, encapsulating aggregated floor summaries. Within each floor node, room nodes are nested, each containing the segmented room's point cloud, keyframe multi-modal data, and room summary. Objects are assigned as child nodes under their respective rooms, incorporating their 3D merged point clouds, CLIP embeddings, brief descriptions, and associated functional elements as sub-nodes. The KeySG hierarchy reflects the environment's physical organization, and it is semantically rich, making it applicable for a wide range of tasks.

E. Scene Querying and Hierarchical RAG

In general, 3D scene graphs are often paired with LLMs [1], [6], [7] to answer user queries. However, serializing the entire scene graph into a single prompt can overwhelm an LLM's context window. Moreover, LLMs' performance degrades when processing long contexts, making it harder for the model to retrieve relevant information. These issues hinder scalability to large environments.

To address this, we design a hierarchical multi-modal Retrieval-Augmented Generation (RAG) pipeline aligned with the structure of KeySG. This pipeline enables users to query the environment using KeySG (e.g., "Where is the coffee mug in the kitchen?") and receive answers grounded in the recovered 3D geometry and semantic data. It also supports indirect references via attributes, status, or spatial relations to other objects in the scene. We initialize our RAG system through three main indexing steps: first, we create text chunks from all information extracted from KeySG keyframes and group them by their graph level, yielding four chunk types: floor, room, frame, and object. Each chunk contains text as described in Sec. III-D. Second, we compute vector embeddings for all chunks and index them by type, using OpenAI's embedding models [40] and FAISS [44]. Third, we build a visual vector database over all keyframes in the scene, as well as a separate database of object-level CLIP [43] embeddings. Grounding a target can be challenging for complex queries, particularly those involving indirect references, attribute descriptions, or implicit targets. To handle this, we first employ an LLM to decompose the raw query q into a set of semantic entities: `<target floor>`, `<target room>`, `<target object>`, and `<anchor objects>`. These entities guide the retrieval of a context bundle C^* comprising a floor F_i, a room R_{ij}, and relevant contents $\mathcal{O}_{\text{rel}}, \mathcal{S}_{\text{rel}}$ that maximizes the posterior probability

$$P(C|q) \approx \underbrace{P(F_i|q)P(R_{ij}|F_i,q)}_{\text{Location Prior}} \prod_{o\in\mathcal{O}_{\text{rel}}} \underbrace{P(o|R_{ij},q)}_{\text{Object}} \prod_{s\in\mathcal{S}_{\text{rel}}} \underbrace{P(s|R_{ij},q)}_{\text{Visual}}$$

We approximate these conditional probabilities using cosine similarity between the query and the entity embeddings. We perform a top-k selection at each level of the hierarchy, filtering objects and keyframes by the identified room R_{ij} to ensure geometric consistency. The retrieved context C^* can be supplied to an LLM to answer object-grounding or general 3D scene question-answering tasks.

IV. Experimental Evaluation

In this section, we evaluate KeySG on four different benchmarks to demonstrate both the quality of the geometric

TABLE I

Method	mAcc	F-mIoU
Mask2former [45] + CLIP [46]	10.42	13.11
ConceptFusion [16]	24.16	31.31
ConceptFusion [16] + SAM [42]	31.53	38.70
ConceptGraphs [1]	<u>40.63</u>	35.95
ConceptGraphs-Detector [1]	38.72	35.82
Clio [29]	37.95	36.26
HOV-SG [6]	38.07	<u>40.16</u>
KeySG (ours)	**45.81**	**46.16**

Results for open-vocabulary 3D semantic segmentation of Replica dataset [9]. We report the mAcc and F-mIoU metrics (%).

and semantic data stored in the graph nodes and its ability to accurately handle complex queries without explicit relationship modeling. First, we assess its open-vocabulary 3D semantic segmentation capabilities against recent methods on the Replica dataset (Sec. IV-A). Second, we compare it to recent approaches in functional element segmentation on the FunGraph3D dataset (Sec. IV-B). Third, we evaluate its ability to retrieve objects from hierarchical queries in large-scale indoor environments using the Habitat Matterport 3D Semantic Dataset (Sec. IV-C). Finally, we examine its capabilities to ground objects from complex natural language queries that require understanding of the scene and its objects' shape, location, color, and affordance, using the Nr3D dataset (Sec. IV-D).

A. Open-vocabulary 3D Semantic Segmentation

To evaluate the objects' visual semantic embeddings generated by KeySG, we utilized scenes `office0-office4` and `room0-room2` from the Replica dataset [9] to facilitate comparison with other researchers. We followed the evaluation protocol from [1]: first, we extracted all semantic class names and modified them to "an image of {class name}". Then, we computed the CLIP text embeddings for each class. Finally, we calculated the cosine similarity between these text embeddings and the object embeddings in the scene, assigning each object the class with the maximum similarity. We report mAcc as the class-mean recall and f-mIOU as frequency-weighted mean intersection over union (IoU). Tab. I shows that KeySG surpasses all recent approaches with a notable margin, demonstrating the effectiveness of extracting the visual CLIP embedding from the best object view.

B. Functional Elements 3D Segmentation

To evaluate the segmentation performance of functional elements in KeySG, we use the FunGraph3D dataset [10], a collection of annotated functional interactive elements within 3D scenes. We compare KeySG against FunGraph [7] and OpenFunGraph [10], the only two 3DSG methods specifically designed for this task. Our primary evaluation metric is Recall@K (R@K) for K=$\{3, 5, 10\}$ across all scenes in the dataset. A prediction is considered a true positive if the open-vocabulary class assigned to a segmented functional element has its embedding ranked within the top-k closest to the ground-truth class among the available labels, and if the IoU with the ground-truth segment exceeds a specified threshold. We report results under different IoU thresholds $(0.0, 0.10, 0.25)$ in Tab. II. KeySG outperforms both FunGraph and OpenFunGraph across all metrics except Recall@10 with $\text{IoU}_{\geq 0.0}$, highlighting that while OpenFunGraph tends to detect more functional elements, it is far less accurate in segmenting their geometry.

C. Object Retrieval From Large-Scale Environment

To evaluate KeySG's capabilities for retrieving objects in large-scale multi-floor environments, we utilized the Habitat Matterport 3D Semantic Dataset [11] , which features scenes with multiple floors and rooms. We selected RGB-D sequences from scenes `00824`, `00829`, `00843`, `00861`, `00862`, `00873`, `00877`, `00890`, following the protocol in [6]. Following with [6], we assessed performance on two query types: floor-room-object (e.g., "the toilet in the bathroom on the ground floor") and room-object (e.g., "oven in the kitchen"). The evaluation spans 345 object categories with 2,809 queries per type. As shown in Tab. III, we compared two variants of KeySG against HOV-SG [6]. We report Recall@K (R@K) for $K \in \{1, 5, 10\}$. A retrieval is considered successful if the target object appears within the top-K results and its Intersection over Union (IoU) with the ground truth meets or exceeds a specified threshold $\tau \in \{0.0, 0.10, 0.50\}$. In the first variant (KeySG w/o RAG), we relied on an LLM to decompose the hierarchical query into `[<floor>, <room>, <object>]`, mirroring the baseline [6]. We computed CLIP text embeddings for each parsed concept and hierarchically compared their cosine similarities with the corresponding CLIP embedding at each level of the scene graph. In the second variant (KeySG w/ RAG), we retrieved the multi-modal context C^* by computing the posterior probability $P(C|q)$ given the query, as detailed in Sec. III-E, utilizing the augmented text and image data for each graph level. We then selected the top-k objects from the context C^* and computed the IoU. KeySG outperformed HOV-SG [6] in both variants, with the RAG-based approach achieving competitive results.

D. Object Grounding from Language Queries on Nr3D

Finally, to evaluate KeySG on grounding objects from language queries in indoor cluttered environments, we used the Nr3D dataset [12]. It provides a diverse set of natural language queries categorized into classes based on how the target object is referenced. Following the evaluation protocol of [28], we utilized scenes `0011_00`, `0030_00`, `0046_00`, `0086_00`, `0222_00`, `0378_00`, `0389_00`, and `0435_00`. In Tab. IV, we report grounding accuracy at an IoU threshold of 0.1. The results are presented overall and categorized by query characteristics, including the presence of spatial, color, or shape language, as well as explicit target mentions.

We compared two variants of KeySG against recent scene graph methods [1], [17], [28]. The first variant, KeySG w/ BBQ [28], constructs the KeySG scene graph but replaces our RAG pipeline with a fixed set of semantic and metric relations between objects. The second variant, KeySG w/ RAG, utilizes our full multi-modal RAG pipeline. As shown

TABLE II

Method	R@3			R@5			R@10		
	$IoU_{>0.0}$	$IoU_{\geq 0.10}$	$IoU_{\geq 0.25}$	$IoU_{>0.0}$	$IoU_{\geq 0.10}$	$IoU_{\geq 0.25}$	$IoU_{>0.0}$	$IoU_{\geq 0.10}$	$IoU_{\geq 0.25}$
OpenFunGraph [10]	45.34	5.39	0.31	47.74	6.89	1.50	**60.40**	9.30	1.50
FunGraph [7]	33.56	22.03	13.04	35.79	22.93	**13.64**	39.98	24.28	14.30
KeySG (ours)	**46.44**	**24.23**	**13.33**	**53.06**	**25.19**	**13.64**	57.12	**27.57**	**14.53**

Results for 3D functional elements segmentation on the FunGraph3D dataset. Recall (R) grouped by Top-k and IoU thresholds metrics (%) are reported.

TABLE III

Method	Query Type	R@1			R@5			R@10		
		$IoU_{>0.0}$	$IoU_{\geq 0.10}$	$IoU_{\geq 0.5}$	$IoU_{\geq 0.0}$	$IoU_{\geq 0.10}$	$IoU_{\geq 0.5}$	$IoU_{>0.0}$	$IoU_{\geq 0.10}$	$IoU_{\geq 0.5}$
HOV-SG [6]	(r, o)	23.30	0.60	0.00	44.30	2.00	0.00	55.90	4.50	0.00
	(f, r, o)	22.80	0.60	0.00	44.90	0.20	0.00	56.60	4.30	0.00
KeySG w/o RAG	(r, o)	32.62	26.50	15.50	**70.75**	61.25	40.05	**83.25**	**75.50**	53.00
	(f, r, o)	**35.30**	**30.37**	15.80	69.60	61.90	40.30	**83.10**	75.00	51.50
KeySG w/ RAG	(r, o)	34.00	**30.40**	**20.60**	68.10	**62.00**	**45.90**	80.80	75.10	**58.30**
	(f, r, o)	32.90	28.50	18.40	68.00	61.00	43.40	79.80	73.10	55.50

Results for hierarchical 3D object retrieval on a large-scale environment within the HM3DSem dataset [11]. The evaluation spans 345 object categories with 2,809 queries per type. We report Recall (R) grouped by Top-k and IoU thresholds metrics (%).

TABLE IV

Method	Overall $IoU_{\geq 0.10}$	w Spatial Lang. $IoU_{\geq 0.10}$	w/o Spatial Lang. $IoU_{\geq 0.10}$	w Color Lang. $IoU_{\geq 0.10}$	w/o Color Lang. $IoU_{\geq 0.10}$	w Shape Lang. $IoU_{\geq 0.10}$	w/o Shape Lang. $IoU_{\geq 0.10}$	w Target Mention $IoU_{\geq 0.10}$	w/o Target Mention $IoU_{\geq 0.10}$
OpenFusion [17]	10.7	8.9	22.3	11.8	10.5	9.8	10.9	11.3	4.9
ConceptGraphs [1]	16.0	15.0	22.3	17.6	15.7	10.8	16.9	16.9	6.6
BBQ [28]	28.3	28.1	29.8	25.2	29.0	34.3	27.3	29.6	14.8
KeySG w/ BBQ [28]	44.1	45.8	43.0	46.5	45.2	28.9	48.2	47.5	23.7
KeySG w/ RAG (ours)	**49.8**	**49.2**	**57.0**	**48.3**	**50.7**	**50.0**	**50.3**	**52.6**	**25.4**

Results for 3D object grounding on the Nr3D dataset [12]. We report accuracy at an IoU threshold of 0.1 (%) for different query characteristics.

in Tab. IV, KeySG w/ RAG outperforms all baselines relying on explicit spatial and semantic edges, including the KeySG w/ BBQ variant, across all categories. This demonstrates the strict limitations of explicitly modeled relations, which often fail when queries contain complex spatial cues or implicit boundaries (see Fig. 3). The performance gap is clearly visible in queries without spatial language (57.0% vs. 43.0%), showing that KeySG w/ RAG effectively captures implicit contextual clues that fixed relational edges miss. Furthermore, KeySG w/ RAG shows substantial gains in queries requiring fine-grained visual reasoning, such as shape (50.0% vs. 28.9%) and color (48.3% vs. 46.5%), highlighting the advantage of our multimodal RAG retrieving keyframes. Finally, while queries without an explicit target mention remain the most difficult category overall (25.4%), KeySG w/ RAG still achieves a relative improvement over the strongest baseline, showing robust deductive reasoning capabilities even when the semantic object tag is missing.

E. Ablation

In order to evaluate the capability of different keyframe-selection strategies to preserve spatial and visual information while minimizing the number of keyframes, we design an ablation study comparing our proposed pose-based **HDBSCAN** [39] clustering against six baseline strategies. To quantify the effectiveness of each method, we define three metrics: first, *Object Recall*, defined as the percentage of unique ground-truth objects visible in the selected subset, serving as a proxy for preserving visual information; second, *Geometric Coverage*, defined as the fraction of surface points in the full reconstruction within a distance threshold $\tau = 0.05$m of the sparse keyframe-based point cloud, serving as a proxy for preserving spatial information; and finally, *Sampling Ratio*, defined as the ratio of retained keyframes over the total number of available frames. We evaluate seven distinct sampling methods: naive *Uniform* and *Random* baselines; visual feature-based clustering leveraging *DINOv2* embeddings [47]; an *Adaptive DBSCAN* approach that iteratively relaxes parameters to meet geometric coverage target; and two covisibility optimization strategies, a *Local* sliding-window approach solving a set-cover problem, and a *Global* greedy approach maximizing total object quality and redundancy. In Fig 4, we report the results for the ablation study. The results show that pose-based clustering with **HDBSCAN** [39] is the sweet spot between *Object Recall*, *Geometric Coverage*, and *Sampling Ratio*.

V. Limitations

Our framework, while effective, has several limitations. First, its reliance on computationally expensive large language and vision-language models makes graph construction an offline process that requires a pre-reconstructed scene. However, once the graph is built, it can be deployed on a robot as a persistent knowledge base that can be efficiently

Fig. 3. Qualitative illustration of KeySG for grounding 3D objects from ambiguous queries. We compare our approach (KeySG w/ RAG, bottom) against a baseline that relies on explicit, predefined spatial relations (KeySG w/ BBQ [28], middle). The fixed-edge scene graph fails to find objects when queries contain multiple complex spatial cues or implicit boundaries. KeySG w/ RAG successfully disambiguates objects as it leverages the implicit information stored in the text descriptions and raw images. These results demonstrate that relationships inferred on demand are superior in handling open-set, complex queries to edges that are rigidly predefined at construction time.

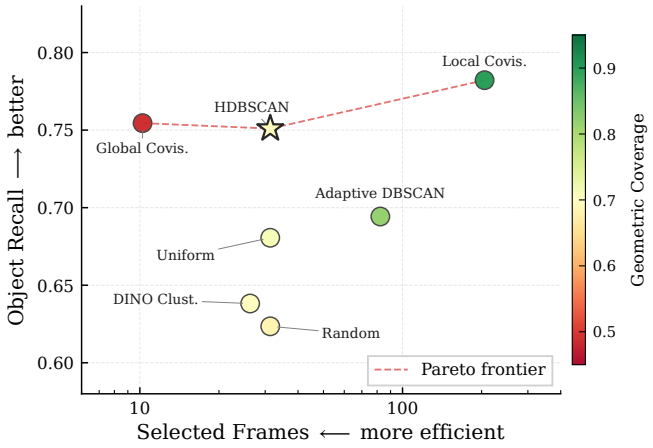

Fig. 4. **Efficiency vs. Recall Frontier.** We compare keyframe-selection strategies based on three metrics defined in Sec. IV-E. **X-axis:** Selected frames (average frame count, log scale; left is more efficient). **Y-axis:** Object Recall (visual proxy). **Bubble Color:** Geometric Coverage (spatial proxy). **HDBSCAN** achieves the optimal trade-off, attaining 96% of the peak recall (found by Covisibility) while using 6.5× fewer frames, avoiding the severe coverage drop observed in Covisibility Global.

queried in real time thanks to the RAG pipeline. Second, our method currently assumes a static environment and does not handle dynamic objects or changes in object states. These areas offer clear directions for future research.

VI. Conclusion

In this work, we addressed a fundamental limitation of existing 3D Scene Graphs (3DSGs): their reliance on predefined relationship edges that restrict open-set semantic reasoning. We introduced KeyFrame-Based 3DSGs (KeySG), the first framework to extend 3DSGs across multiple resolutions, ranging from full buildings to functional elements. By combining keyframe sampling, hierarchical scene summarization, and a retrieval-augmented generation mechanism, KeySG efficiently grounds task-relevant context for large language models while overcoming traditional scalability limits. This enables the framework to handle a wide variety of open-ended tasks and queries. Finally, our experiments across open-vocabulary 3D segmentation, functional element extraction, and 3D object grounding benchmarks demonstrate that KeySG outperforms prior approaches on most metrics.

VII. Acknowledgements

We would to like thank the anonymous reviewers for the constructive comments regarding the ablation study. This work has been supported by the German Federal Ministry of Research, Technology, and Space (BMFTR) under the Robotics Institute Germany (RIG).

References

[1] Q. Gu, A. Kuwajerwala, S. Morin, K. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, C. Gan, C. de Melo, J. Tenenbaum, A. Torralba, F. Shkurti, and L. Paull, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *arXiv*, 2023.

[2] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *RSS*, 2020.

[3] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," in *RSS*, 2022.

[4] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions," in *CVPR*, 2020.

[5] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning," in *CoRL*, 2023.

[6] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation," in *RSS*, Delft, Netherlands, 2024.

[7] D. Rotondi, F. Scaparro, H. Blum, and K. O. Arras, "Fungraph: Functionality aware 3d scene graphs for language-prompted scene interaction," *IROS*, 2025.

[8] C.-Y. Hsieh, Y.-S. Chuang, C.-L. Li, Z. Wang, L. T. Le, A. Kumar, J. Glass, A. Ratner, C.-Y. Lee, R. Krishna *et al.*, "Found in the middle: Calibrating positional attention bias improves long context utilization," *arXiv preprint arXiv:2406.16008*, 2024.

[9] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[10] C. Zhang, A. Delitzas, F. Wang, R. Zhang, X. Ji, M. Pollefeys, and F. Engelmann, "Open-Vocabulary Functional 3D Scene Graphs for Real-World Indoor Spaces," in *CVPR*, 2025.

[11] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre, A. X. Chang, D. Batra, M. Savva *et al.*, "Habitat-matterport 3d semantics dataset," in *CVPR*, 2023.

[12] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. J. Guibas, "ReferIt3D: Neural listeners for fine-grained 3d object identification in real-world scenes," in *ECCV*, 2020.

[13] I. Armeni, Z.-Y. He, A. Zamir, J. Gwak, J. Malik, M. Fischer, and S. Savarese, "3D scene graph: A structure for unified semantics, 3D space, and camera," in *ICCV*, 2019.

[14] U.-H. Kim, J.-M. Park, T.-J. Song, and J.-H. Kim, "3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents," *TOC*, 2019.

[15] D. Shah, B. Osiński, S. Levine *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *CoRL*. PMLR, 2023, pp. 492–504.

[16] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *RSS*, 2023.

[17] K. Yamazaki, T. Hanyu, K. Vo, T. Pham, M. Tran, G. Doretto, A. Nguyen, and N. Le, "Open-fusion: Real-time open-vocabulary 3d mapping and queryable scene representation," in *ICRA*, 2024.

[18] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone, "Foundations of spatial perception for robotics: Hierarchical representations and real-time systems," *IJRR*, 2024.

[19] C. Agia, K. M. Jatavallabhula, M. Khodeir, O. Miksik, V. Vineet, M. Mukadam, L. Paull, and F. Shkurti, "Taskography: Evaluating robot task planning over large 3d scene graphs," *CoRL*, 2022.

[20] E. Bartoli, D. Rotondi, K. O. Arras, and I. Leite, "Long-term planning around humans in domestic environments with 3d scene graphs," *arXiv 2503.09173*, 2025.

[21] E. Bartoli, D. Rotondi, B. He, P. Jensfelt, K. O. Arras, and I. Leite, "Social 3d scene graphs: Modeling human actions and relations for interactive service robots," *arXiv 2509.24966*, 2025.

[22] D. Honerkamp, M. Büchner, F. Despinoy, T. Welschehold, and A. Valada, "Language-grounded dynamic scene graphs for interactive object search with mobile manipulation," *IEEE RA-L*, 2024.

[23] Z. Yan, S. Li, Z. Wang, L. Wu, H. Wang, J. Zhu, L. Chen, and J. Liu, "Dynamic open-vocabulary 3d scene graphs for long-term language-guided mobile manipulation," *IEEE RA-L*, 2025.

[24] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *RSS*, 2020.

[25] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From slam to spatial perception with 3d dynamic scene graphs," *IJRR*, 2021.

[26] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences," in *CVPR*, 2021.

[27] S. Koch, N. Vaskevicius, M. Colosi, P. Hermosilla, and T. Ropinski, "Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships," in *CVPR*, 2024.

[28] S. Linok, T. Zemskova, S. Ladanova, R. Titkov, D. Yudin, M. Monastyrny, and A. Valenkov, "Beyond bare queries: Open-vocabulary object grounding with 3d scene graph," *ICRA*, 2025.

[29] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time task-driven open-set 3d scene graphs," *IEEE RA-L*, 2024.

[30] Y. Chang, L. Fermoselle, D. Ta, B. Bucher, L. Carlone, and J. Wang, "Ashita: Automatic scene-grounded hierarchical task analysis," in *CVPR*, 2025.

[31] S. Koch, J. Wald, M. Colosi, N. Vaskevicius, P. Hermosilla, F. Tombari, and T. Ropinski, "Relationfield: Relate anything in radiance fields," in *CVPR*, 2025.

[32] D. Z. Chen, A. X. Chang, and M. Nießner, "Scanrefer: 3d object localization in rgb-d scans using natural language," *ECCV*, 2020.

[33] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Language conditioned spatial relation reasoning for 3d object grounding," in *NeurIPS*, 2022.

[34] A. e. a. Majumdar, "Openeqa: Embodied question answering in the era of foundation models," in *CVPR*, 2024.

[35] H. Huang, Y. Chen, Z. Wang, R. Huang, R. Xu, T. Wang, L. Liu, X. Cheng, Y. Zhao, J. Pang *et al.*, "Chat-scene: Bridging 3d scene and large language models with object identifiers," *NeurIPS*, 2024.

[36] J. Loo, Z. Wu, and D. Hsu, "Open scene graphs for open-world object-goal navigation," *The International Journal of Robotics Research*, 2025.

[37] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson, "From local to global: A graph rag approach to query-focused summarization," *arXiv preprint arXiv:2404.16130*, 2024.

[38] G. Younes, D. Asmar, E. Shammas, and J. Zelek, "Keyframe-based monocular slam: design, survey, and future directions," *Robotics and Autonomous Systems*, 2017.

[39] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds., 2013.

[40] OpenAI, "GPT-4 technical report," *CoRR*, vol. abs/2303.08774, 2023.

[41] S. Fu, Q. Yang, Q. Mo, J. Yan, X. Wei, J. Meng, X. Xie, and W.-S. Zheng, "Llmdet: Learning strong open-vocabulary object detectors under the supervision of large language models," *arXiv preprint arXiv:2501.18954*, 2025.

[42] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *CVPR*, 2023.

[43] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa *et al.*, "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," *arXiv preprint arXiv:2502.14786*, 2025.

[44] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The faiss library," *arXiv*, 2024.

[45] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," *CVPR*, 2022.

[46] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[47] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.