Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation

Abdelrhman Werby[1*], Chenguang Huang[1*], Martin Büchner[1*], Abhinav Valada[1], Wolfram Burgard[2]

[1]University of Freiburg [2]University of Technology Nuremberg

***Abstract*—Recent open-vocabulary robot mapping methods enrich dense geometric maps with pre-trained visual-language features. While these maps allow for the prediction of point-wise saliency maps when queried for a certain language concept, large-scale environments and abstract queries beyond the object level still pose a considerable hurdle, ultimately limiting language-grounded robotic navigation. In this work, we present HOV-SG, a hierarchical open-vocabulary 3D scene graph mapping approach for language-grounded indoor robot navigation. Leveraging open-vocabulary vision foundation models, we first obtain state-of-the-art open-vocabulary segment-level maps in 3D and subsequently construct a 3D scene graph hierarchy consisting of floor, room, and object concepts, each enriched with open-vocabulary features. Our approach is able to represent multi-story buildings and allows robotic traversal of those using a cross-floor Voronoi graph. HOV-SG is evaluated on three distinct datasets and surpasses previous baselines in open-vocabulary semantic accuracy on the object, room, and floor level while producing a 75% reduction in representation size compared to dense open-vocabulary maps. In order to prove the efficacy and generalization capabilities of HOV-SG, we showcase successful long-horizon language-conditioned robot navigation within real-world multi-story environments. We provide code and trial video data at: https://hovsg.github.io.**

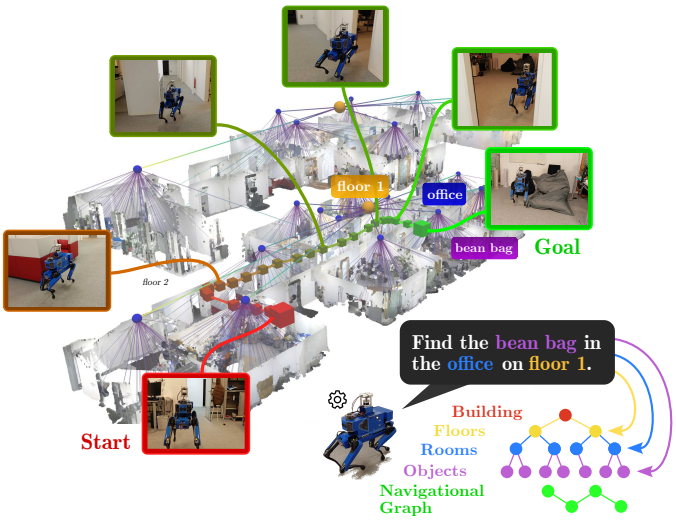


Fig. 1: HOV-SG enables the construction of accurate open-vocabulary 3D scene graphs for large-scale and multi-story environments and enables robots to navigate in them effectively.

I. Introduction

Humans acquire conceptual knowledge about the world as a whole and concrete objects in particular through multi-modal experiences. These semantic experiences are paramount to object recognition and language as well as reasoning and planning [1, 2]. Cognitive maps store this information based on sensor fusion, fragmentation, and hierarchical structure. This is central to the human ability to navigate the physical world [3, 4, 5]. Recently, language proved to be an effective link between intelligent systems and humans and can enable robot autonomy in complex human-centered environments [6, 7, 8, 9, 10, 11, 12, 13, 14].

Classical methods for robot navigation build dense spatial maps of high accuracy using approaches to simultaneous localization and mapping (SLAM) [15]. Those give rise to fine-grained navigation and manipulation based on geometric goal specifications. Recent advances have combined dense maps with pre-trained zero-shot vision-language models, which facilitates open-vocabulary indexing of observed environments [9, 10, 11, 16, 13, 17, 12]. While these approaches marry the area of classical robotics with modern open-vocabulary semantics, representing larger scenes while abstracting still poses a considerable hurdle. Scalable scene representations generated from real-world perception inputs should generally fulfill the following requirements: 1) Object-centricity and abstraction in terms of hierarchies, 2) efficiency regarding storage capacity as well as actionability, 3) true open-vocabulary indexing and easy querying.

A number of works approach this using 3D scene graph structures [18, 19, 20] that excel at representing larger environments efficiently. At the same time, they constitute a useful interface to semantic tokens used for prompting large language models (LLM). Nonetheless, most approaches rely on closed-set semantics with the exception of ConceptGraphs [14] that focuses on smaller scenes.

In this work, we present **H**ierarchical **O**pen-**V**ocabulary 3D **S**cene **G**raphs, short HOV-SG. Our approach abstracts from dense open-vocabulary maps and allows the indexing of three distinct concepts, namely floors, rooms as well as objects. We utilize open-vocabulary vision-language models [21, 22] across all concepts in order to construct 3D scene graph hierarchies that span multi-story environments while maintaining a small memory footprint. Given its concept-centric nature, the HOV-SG representation is promptable using LLMs.

* Equal contribution.

Different from previous work, our approach relates to different conceptual levels by first decomposing abstract queries such as *"towel in the bathroom on the upper floor"* and scoring the obtained tokens against the different levels of the hierarchy. We complement this with a navigational Voronoi graph that covers multiple floors including stairs, which allows actionable grounding of decomposed queries in the environment. This enables object retrieval and long-horizon robotic navigation in large-scale indoor environments from abstract queries as shown in Fig. 1.

In summary, we make the following contributions:

1) We introduce a novel fusion scheme using feature clustering of zero-shot embeddings that yields state-of-the-art results in open-set 3D semantic segmentation.
2) We present an algorithm that enables the construction of truly actionable open-vocabulary 3D scene graphs of multi-floor buildings.
3) We evaluate the semantic segmentation performance of our method on the Replica [23] and ScanNet [24] dataset and analyze key properties of our scene graphs on the Habitat-Matterport 3D Semantics dataset [25]. Furthermore, we present a detailed ablation study to justify our design choices.
4) We conduct real-world multi-floor object navigation experiments based on long natural language queries.
5) We introduce a novel evaluation metric for measuring open-vocabulary semantics termed $\text{AUC}_{\text{top-k}}$.
6) We make our code and evaluation protocol publicly available at https://hovsg.github.io to foster future research and introduce comparability in open-set mapping.

II. Related Work

A. Semantic 3D Mapping

Enriching a geometric map with semantic information is a stepping stone to a flexible and versatile navigation system [26, 27, 16, 10, 4, 9]. In the past, researchers created semantics-enhanced or instance-level maps by learning sensor observation features [28], matching pre-built object shapes to the geometric map [29], back-projecting 2D semantic predictions into the 3D space [30, 31, 32], or instantiating 2D detections with basic 3D elements such as cubes or quadrics [33, 34]. These methods have shown their capabilities of reconstructing scenes with both accurate geometric structure and precise semantic meaning. However, most of these methods only work with a fixed category set constrained by either the trained semantic prediction models or the pre-defined set of relevant object primitives.

On account of recent advancements in large vision-language models such as CLIP [21] and their fine-tuned counterparts, a number of works proposed map representations that integrate visual-language features into geometric maps, enabling open-vocabulary indexing of objects [11, 16, 10, 13, 35, 17, 12], audio data [16, 13] and images [26, 16] in an unstructured environment. While lifting the constraints of fixed semantic categories, these approaches often necessitate the storage of a visual-language embedding for each geometric element such as points, voxels, or 2D cells in the map, resulting in a significant increase in storage overhead.

B. 3D Scene Graphs

3D scene graphs have emerged as an effective, object-centric representation of large-scale indoor [36, 19, 20] and outdoor scenes [18]. By representing objects or spatial concepts as nodes and their relations as edges, 3D scene graphs allow to efficiently represent larger scenes [36, 18, 19]. Both edges and nodes can hold geometric and semantic attributes, which are often inferred from certain off-the-shelf networks [37]. Decomposing scenes into objects and their relations enables higher-level reasoning for robotic navigation and manipulation. This is particularly useful in the realm of reasoning, planning, and navigation given the object-centric nature of these tasks [19, 38]. Often combined with odometry estimates from simultaneous localization and mapping (SLAM) [39, 18, 40], 3D scene graphs also allow a tight coupling between semantics and highly accurate mapping approaches utilizing e.g. meshes to represent environments [19].

Early works have shown how to encode hierarchies via abstraction in both the spatial and the semantic domain using offline approaches [36, 20]. Successive works investigated learning-based scene graph construction [41, 37] as well as dynamic indoor scenes [19]. Several approaches such as SceneGraphFusion [37] and S-Graphs [39] also investigate the real-time capabilities of their proposed approaches. Most recently, ConceptGraphs [14] was the first to show how to combine 3D scene graphs with open-vocabulary vision-language features. In addition, the authors show how to query the graph using LLMs and demonstrate various downstream applications.

C. Scene Graphs for Planning

Several recent works have investigated the use of scene graphs for robotic planning. The earliest approaches rely on pre-explored environments and perform iterative scene graph decomposition to retrieve grounded plans [38, 42]. RobLM [42] decomposes the planning stage by relying on a fine-tuned GPT-2 instance that proposes high-level sub-problems from scene graphs, which are in turn solved through PDDL task planners. SayPlan [38] directly utilizes GPT-4 [43] for iterative search on a scene graph to generate grounded plans, which requires feasibility constraints on the manipulated entities and actions. Another line of work investigates robotic navigation from scene graphs. SayNav [44] obtains LLM-generated plans from scene graphs and executes short-distance point-goal navigation sub-tasks. Contrary to that, VoroNav [45] constructs a Voronoi graph that is attributed to camera observations in order to solve object navigation. Orthogonal to that, MoMa-LLM [46] tackles mobile manipulation objectives using scene graphs fed to GPT-4 in a task-specific manner. Similarly, GRID [47] uses a graph neural network to predict actions from scene graphs and LLM encodings.

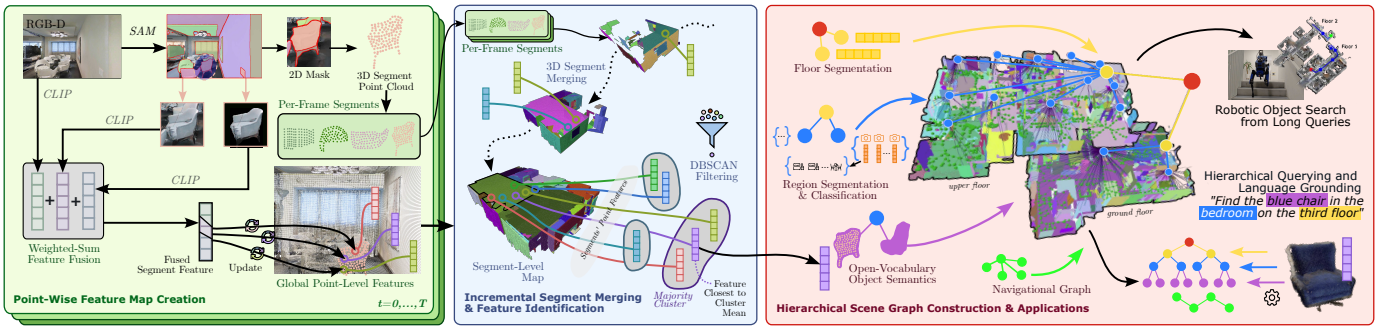


Fig. 2: HOV-SG builds hierarchical open-vocabulary 3D scene graphs of indoor household scenes. We first use SAM to extract object masks per frame while obtaining vision-language features via CLIP. In the next step, we aggregate these features on a point level in the map. Secondly, we segment the full point cloud based on merged 3D masks. To generate more meaningful semantic object features, we employ a DBSCAN-based filtering approach to obtain a majority vote feature for each object. To construct an actionable 3D scene graph, we segment the obtained panoptic map into multiple floors, segment and classify distinct regions using several view embeddings, and identify object names via querying. As a result, HOV-SG allows hierarchical querying and navigation using mobile robots even in complex multi-floor environments.

Conceptually, our work is most similar to ConceptGraphs and Hydra. While ConceptGraphs is only evaluated on small scenes and mostly validated by human evaluators in terms of semantic accuracy of nodes etc., our work proposes not only a novel metric for measuring the semantic accuracy of object features but also introduces open-vocabulary hierarchies. Different from ConceptGraphs and similar to Hydra [19], which does not operate on open-vocabulary features, our approach demonstrates how to efficiently represent actionable, hierarchical 3D scene graphs that are attributed with open-vocabulary features.

III. Technical Approach

This work aims to develop a concise and efficient visual-language graph representation for large-scale multi-floor indoor environments given RGB-D observations and odometry. The graph should facilitate the indexing of multi-level semantic concepts through natural language queries such as "the first floor" (floor level), "the office on the first floor" (room level), and "the plant in the office on the second floor" (object level). Additionally, the graph should be actionable and enable a robot to localize and navigate semantically and spatially in the environment without additional geometric maps. We address this by introducing **H**ierarchical **O**pen-**V**ocabulary **S**cene **G**raphs, in short HOV-SG. The overall pipeline consists of two stages. We first create a 3D segment-level open-vocabulary map and then build a hierarchical open-vocabulary scene graph based on the map. In the following sections, we describe (i) the construction of the 3D segment-level open-vocabulary map (Sec. III-A), (ii) the creation of the hierarchical open-vocabulary scene graph (Sec. III-B), and (iii) how to use the graph for language-conditioned navigation across a large-scale environment (Sec. III-C). Fig. 2 presents an overview of our method.

A. 3D Segment-Level Open-Vocabulary Mapping

The main idea of building a segment-level open-vocabulary map is to create a list of 3D point clouds, namely segments, from an RGB-D video with odometry and assign an open-vocabulary feature generated by a pre-trained visual-and-language model (VLM) to each segment. Unlike previous works that equip each 3D point with an independent visual-language feature [11, 10, 13, 12], we leverage the fact that neighboring points in the 3D world often share the same semantic information. This implies the potential of reducing the required semantic features to represent the scene while maintaining expressiveness.

Frame-Wise 3D Segment Merging: Given a sequence of RGB-D observations, we utilize Segment Anything [22] to obtain a list of class-agnostic 2D binary masks at each timestep. The pixels in each mask are then backprojected to 3D using the depth information, resulting in a list of point clouds, or 3D segments. Based on accurate odometry estimates, we transform all 3D segments into the global coordinate frame. These frame-wise segments are either initialized as new global segments or merged with existing ones based on an overlap metric:

$$R(m,n) = max(overlap(S_m, S_n), overlap(S_n, S_m)), \quad (1)$$

where S_m and S_n indicate segment (or point cloud) m and n, $overlap(S_a, S_b)$ is computed by taking the number of points in S_a showing a neighbor in S_b within a certain distance divided by the total number of points in S_a. Different from Gu *et al.* [14], who incrementally merge new segments with one global segment that has the largest overlapping ratio, we construct a fully connected graph where each segment serves as a node and their edge weights are the corresponding overlapping ratios. Based on these weights, highly-connected subgraphs are subsequently merged. In this way, one segment can be merged with multiple segments, which is useful in situations in which an incoming segment is, e.g., filling a gap between two already registered global segments.

Segment-Level Open-Vocabulary Features Computation: For each obtained 2D SAM mask per frame, we obtain an image crop based on its bounding box as well as an image of the isolated mask without background. We encode the full

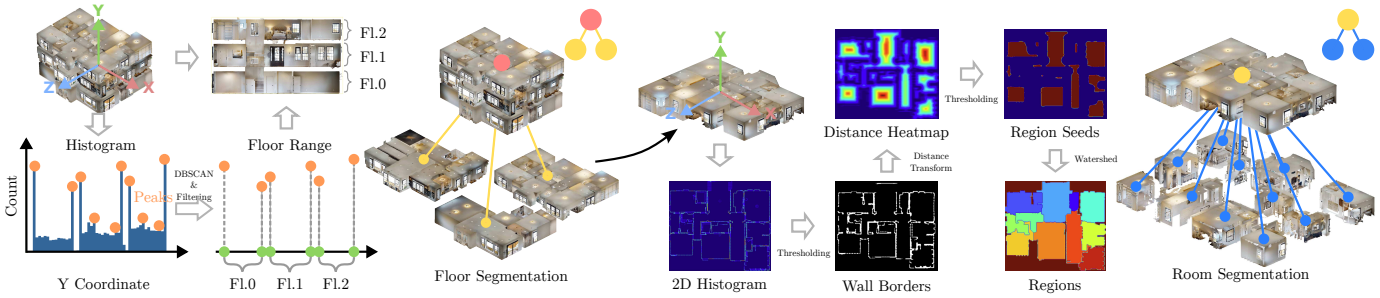


Fig. 3: Floor and Room Segmentation. Given the point cloud of the whole environment, floor and room nodes are subsequently derived based on geometric heuristics. Floor boundaries are computed by finding peaks of the pixel density along the height direction followed by filtering while room segment masks are extracted using the Watershed algorithm.

RGB frame and the two mask-wise images with CLIP [21] and fuse them in a weighted-sum manner (Fig. 2, left). Previous work [48] proposed to use the CLIP feature of the masked image without background, while others [13] approach this by combining the CLIP features of the whole image and the target mask's crop including background. In our work, we empirically show that encoding the full RGB frame and the two mask-wise images with CLIP and fusing them in a weighted-sum manner achieves improved results (Sec. IV-E). The fusion scheme can be formulated as:

$$f_i = w_g f_g + w_l f_l + w_m f_m, \tag{2}$$

where f_i indicates the fused features for the i-th 2D mask in the frame, f_g, f_l, and f_m indicate the CLIP features extracted from the entire RGB frame, the image crop of the 2D mask, and the image crop of the 2D mask excluding the background, respectively. Furthermore, w_g, w_l, and w_m represent their respective weights, which sum up to 1.

Assuming a single CLIP feature for each mask, we transform the 2D mask into global 3D coordinates and associate the obtained fused CLIP feature with the nearest 3D points in a pre-computed reference point cloud. Based on this association, we register the obtained segment features on a global point-wise feature map. The final point-wise features are determined by averaging each reference point's associated features. Based on the 3D segments obtained in the independent merging step, we can finally infer open-vocabulary vision-language features for all 3D segments as outlined in Fig.2. In the subsequent step, we match point-wise features with the obtained 3D segments. For each point within a segment, we identify the nearest points in the reference point cloud and collect their CLIP features. We leverage DBSCAN to cluster all the point-wise features of the segment and assign the feature that is closest to the majority cluster's mean to the segment (Fig. 2, middle). This circumvents mode collapse while removing noise and thus produces more semantically meaningful segment features.

B. 3D Scene Graph Construction

In this section, we describe how to build a hierarchical open-vocabulary scene graph given a global reference point cloud of the scene, a list of global 3D segments, and their associated CLIP features as described in Sec. III-A.

We formalize our graph as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ denotes the nodes and $\mathcal{E}$ denotes the edges. The nodes can be expressed as $\mathcal{N} = \mathcal{N}_S \cup \mathcal{N}_F \cup \mathcal{N}_R \cup \mathcal{N}_O$, consisting of a root node $\mathcal{N}_S$, floor nodes $\mathcal{N}_F$, room nodes $\mathcal{N}_R$, and object nodes $\mathcal{N}_O$. Each node in the graph except the root node $\mathcal{N}_S$ contains the point cloud of the concept it refers to and the open-vocabulary features associated with it. The edges can be written as $\mathcal{E} = \mathcal{E}_{SF} \cup \mathcal{E}_{FR} \cup \mathcal{E}_{RO}$. Here, $\mathcal{E}_{SF}$ represents the edges between the root node and the floor nodes, $\mathcal{E}_{FR}$ represents the edges between the floor nodes and the room nodes, and lastly, $\mathcal{E}_{RO}$ denotes the edges between the room and object nodes.

Floor Segmentation: In order to separate floors, we identify peaks of a height histogram over all points contained in the point cloud. Given the point cloud of the whole environment, we construct the histogram over all points along the height axis using a bin size $0.01\,\mathrm{m}$. Next, we identify peaks in this histogram (within a local range of $0.2\,\mathrm{m}$) and select only peaks that exceed a minimum of 90% of the highest intensity peak. We apply DBSCAN and select the two highest-ranking peaks in each cluster. After that, every two consecutive values in the sorted height vector represent a single floor (floor and ceiling) in the building. The floor segmentation process is shown in Fig. 3. Using the obtained height levels, we can extract floor point clouds for each floor $\mathcal{P}_l$ where l is the floor number. In addition, we equip each floor node with a CLIP text embedding using the template "floor {#}". A graph edge between the root node and each floor node $(N_S, N_l) \in \mathcal{E}_{SF}$ is established.

Room Segmentation: Based on each obtained floor point cloud, we construct a 2D bird's-eye-view (BEV) histogram, from which a binary wall skeleton mask is extracted by thresholding the histogram. After dilating the wall mask and computing an Euclidean distance field (EDF), a number of isolated regions is derived by thresholding the EDF. Taking these regions as seeds, we apply the Watershed algorithm to obtain 2D region masks. The room segmentation process is further shown in Fig. 3. Given the 2D region masks, we extract the 3D points that fall into the floor's height interval as well as the BEV room segment to form room point clouds that are used to associate objects to rooms later.

To enrich room nodes with open-vocabulary features, we associate RGB-D observations whose camera poses reside within a room segment to those rooms (see Fig. 4). The

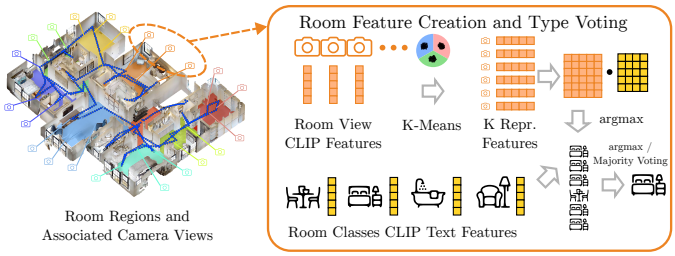
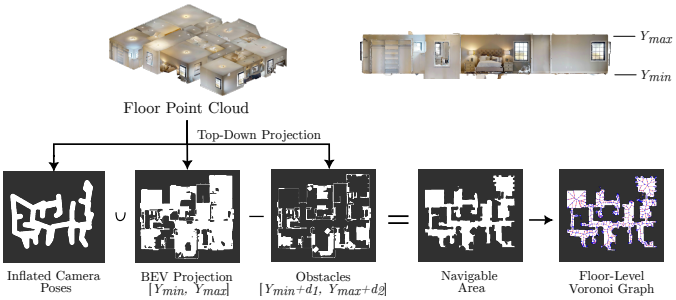
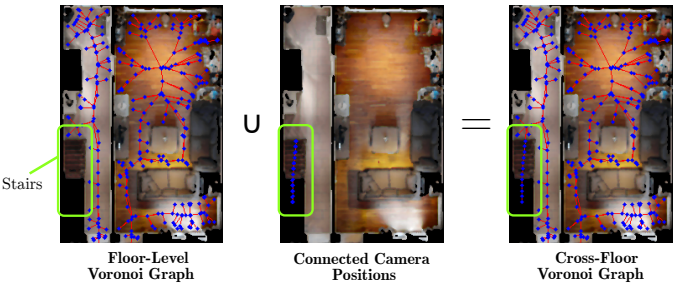


Fig. 4: Room embedding computation and room type voting. We enrich each room node with open-vocabulary embeddings by associating the observations with it. Given the segmented room region and the contained camera poses we extract 10 distinct CLIP features that represent the semantic distribution of a room.



(a) Single-floor Navigational Graph



(b) Cross-floor Navigational Graph

Fig. 5: Actionable navigational graph: The creation of the actionable navigational graph involves constructing single-floor and cross-floor navigational graphs: (a) By deducting the set of obstacles from the union of camera poses and the per-floor obtained BEV projection of the floor point cloud, we obtain the navigable area. Within this area we construct a Voronoi diagram as shown right. (b) In order to equip our navigational graph with cross-floor navigation capabilities, we extract the camera positions within regions classified as *stairs*. This subgraph is connected with the corresponding floor-level Voronoi graphs.

CLIP embeddings of these images are distilled by extracting k representative view embeddings using the k-means algorithm. During inference, given a list of room categories encoded via CLIP, we construct a cosine similarity matrix between the k representative features and all room category features. Next, we take the argmax along the category axis and obtain the most probable room type for each representative separately, resulting in k votes per room. Given these votes, we obtain the predicted room category by either taking the maximum-score vote or the majority vote across all k representatives per room.

These k representative embeddings and the room point cloud attribute the room node $N_{f,r}$ of room r on floor f. An edge between the floor node and each room node $(N_f, N_{f,r}) \in \mathcal{E}_{FR}$ is established. The construction and querying of view embeddings are illustrated in Fig. 4.

Object Identification: Given the room point cloud, we associate object-level 3D segments that show a point cloud overlap with a potential candidate room in the bird's-eye-view. Whenever a segment shows zero overlap with any room, we associate it with the room showing the smallest Euclidean distance. To reduce the number of nodes, we merge 3D segments of significant pair-wise partial overlap (Sec. III-A) that produce equal object labels when queried against a chosen label set. Each merged point cloud constitutes an object node $N_{f,r,o}$ that is connected to its corresponding room node $N_{f,r} \in \mathcal{N}_R$ by an edge $(N_{f,r}, N_{f,r,o}) \in \mathcal{E}_{RO}$. Each object node holds its corresponding 3D segment feature as described in Sec. III-A, its 3D segment point cloud as well as a maximum-score object label for intermediate naming.

Actionable Graph Creation: In addition to the open-vocabulary hierarchy, the scene graph also contains a navigational Voronoi graph that serves robotic traversability of the mapped surroundings [49] spanning multiple floors. This enables high-level planning and low-level execution based on the Voronoi graph. The creation of actionable graphs involves constructing per-floor and cross-floor navigation graphs. For the floor-level graph, the approach entails computing the free space map of the floor and creating a Voronoi graph [49] based on it. To construct per-floor graphs, we first obtain all camera poses and project them as 2D points onto a BEV map of each floor, assuming areas within a certain radius of two nodes are pair-wise navigable. Subsequently, the entire floor's region is obtained by projecting all floor-wise points into the BEV plane. An obstacle map is generated based on points within a predefined height range $[y_{min} + \delta_1, y_{min} + \delta_2]$, where y_{min} is the minimal height of the floor points while $\delta_1 = 0.2$, $\delta_2 = 1.5$ are empirically tuned. By taking the union of the pose region map and the floor region map and subtracting the obstacle region map, the free space map of each floor is derived. The Voronoi graph of this free map yields the floor graph. (see Fig. 5a). To build cross-floor navigational graphs, camera poses on stairs are connected to form stair-wise graphs. Subsequently, the closest nodes between the stairs graph and the floor-wise graph are selected respectively and connected, thereby completing the construction of cross-floor navigational graphs as shown in Fig. 5b.

C. Navigation with Scene Graph

HOV-SG extends the scope of potential navigation goals to more specific spatial concepts such as regions and floors compared to simple object goals [14, 16, 10, 13]. Language-guided navigation with HOV-SG involves processing complex queries such as *"find the toilet in the bathroom on floor 2"* using a large language model (prompts are given in the supplementary material Sec. S.1-A). We break down such lengthy instructions into three separate queries: one for the floor level, the room level, and the object level, respectively.

TABLE I: OPEN-VOCABULARY 3D SEMANTIC SEGMENTATION

Method	CLIP Backbone	Replica mIOU	Replica F-mIOU	Replica mAcc	ScanNet mIOU	ScanNet F-mIOU	ScanNet mAcc
MinkowskiNet [50]		-	-	-	0.42	0.47	0.56
ConceptFusion [13]	OVSeg	0.10	0.21	0.16	0.08	0.11	0.15
	Vit-H-14	0.10	0.18	0.17	0.11	0.12	0.21
ConceptGraph [14]	OVSeg	0.13	0.27	0.21	0.15	0.18	0.23
	Vit-H-14	0.18	0.23	0.30	0.16	0.20	0.28
HOV-SG (ours)	OVseg	0.144	0.255	0.212	0.214	0.258	0.420
	Vit-H-14	**0.231**	**0.386**	**0.304**	**0.222**	**0.303**	**0.431**

Higher values are better. The used evaluation metrics are defined in Sec. S.2-B in the supplementary material. The ConceptFusion pipeline evaluated against made use of instance masks predicted by SAM [22]. The MinkowskiNet [50] is a privileged method that was trained on the full set of ScanNet [24] scenes to demonstrate the gap between zero-shot and fully-supervised methods.

Leveraging the explicit hierarchical structure of HOV-SG, we sequentially query against each hierarchy level to progressively narrow down the solution corridor. This is done by taking the cosine similarity between the identified query floor, query region, and query object as well as all objects, rooms, and floors given in the graph, respectively. Once a target node is identified via scoring, we utilize the navigational graph mentioned above to plan a path from the starting pose to the target destination, which is demonstrated in Fig. S.1 and visualized in Fig. 1.

IV. EXPERIMENTAL EVALUATION

The goals of our experiments are five-fold: (i) we quantitatively compare HOV-SG with recent open-vocabulary map representations in 3D semantic segmentation on ScanNet and Replica (Sec. IV-A), (ii) we investigate the semantic and geometric accuracy of HOV-SG at the floor, room, and object level on the Habitat Matterport 3D Semantic Dataset (Sec. IV-B), (iii) we study how HOV-SG enables large-scale language-grounded navigation in the real-world (Sec. IV-C), (iv) we demonstrate the compact memory footprint of HOV-SG compared to previous open-vocabulary representations (Sec. IV-D), and lastly, (v) we justify our design choices through an ablation study (Sec. IV-E).

A. 3D Semantic Segmentation on ScanNet and Replica

To test the semantic expressiveness of our HOV-SG method, we evaluate the open-vocabulary 3D semantic segmentation performance on ScanNet [24] and Replica [23]. We compare our method with two alternative vision-language representations (ConceptFusion [13] and ConceptGraphs [14]) while using different CLIP backbones. We consider ViT-H-14 and a fine-tuned backbone ViT-L-14 released with the work OVSeg [48]. The used evaluation metrics are defined in Sec. S.2-B. To demonstrate the existing gap between zero-shot and fully supervised methods we also evaluated a MinkowskiNet [50] instance trained on ScanNet [24].

Prediction Generation: We generate the CLIP text embedding for each category contained in the dataset by using a template of the form "There is the {category} in the scene." as well as the category name "{category}" itself. Next, we average the two to obtain the embedding of each specific category. We obtain predicted labels for each object node by computing the cosine similarity between all object nodes' embeddings and all category embeddings and lastly apply the argmax operator. In the following, we concatenate all objects' point clouds to create our predicted point cloud $\mathcal{P}_{pred}$ and transform it to the same coordinate frame as that of the point cloud with ground-truth (GT) semantic labels $\mathcal{P}_{GT}$. Given that the predicted point cloud may exhibit varying point densities compared to the ground truth (GT), we iterate through each GT point to locate its five nearest points in $\mathcal{P}_{pred}$, and then determine the majority label among these points as the predicted label for each GT point.

Evaluation Scenes: For consistency, we evaluate the same scenes evaluated in [14, 13]. For ScanNet, we evaluate scenes: `scene0011_00`, `scene0050_00`, `scene0231_00`, `scene0378_00`, `scene0518_00`. Regarding Replica, we evaluate on `office0-office4` and `room0-room2`.

Results: The semantic segmentation results on both Replica and ScanNet are provided in Table I. Regarding mIOU and F-mIOU, HOV-SG outperforms the open-vocabulary baselines by a large margin. This is primarily due to the following improvements we made: First, when we merge segment features, we consider all point-wise features that each segment covers and use DBSCAN to obtain the dominant feature, which increases the robustness compared to taking the mean as done by ConceptGraphs [14]. Secondly, when we generate the point-wise features, we use the mask feature which is the weighted sum of the sub-image and its contextless counterpart. This mitigates the impact of salient background objects. Further qualitative results are given in Fig. 6.

B. Scene Graph Evaluation on Habitat 3D Semantics

We evaluate our scene graph on four aspects. To analyze the geometric accuracy of the scene graph, we analyze the floor and class-agnostic region segmentation performance in Sec. IV-B1. To evaluate the semantic accuracy, we evaluate predicted region semantics (Sec. IV-B2) as well as open-vocabulary object-level semantics (Sec. IV-B3). To scrutinize the downstream navigation capabilities of HOV-SG, we conduct hierarchical object retrieval and navigation experiments given abstract language queries in Sec. IV-B4. We display two exemplary constructed hierarchical 3D scene graphs in Fig. 7. The 3D scene graph visualization of the remaining scenes is shown in Fig S.4.

Dataset: In order to evaluate various aspects of the produced scene graph hierarchy, we have chosen the Habitat-Semantics dataset (HM3DSem) as it provides true open-vocabulary labels across large multi-floor scenes and also provides object-region assignments. Since our approach operates on RGB-D frames, we manually record random walks of 8 scenes of the Habitat Semantics dataset [25], which span multiple rooms and floors: `00824`, `00829`, `00843`, `00861`, `00862`, `00873`, `00877`, `00890`. To construct ground-truth maps to compare against, we fuse the RGB-D and panoptic data across all frames given accurate odometry and obtain RGB and panoptic global

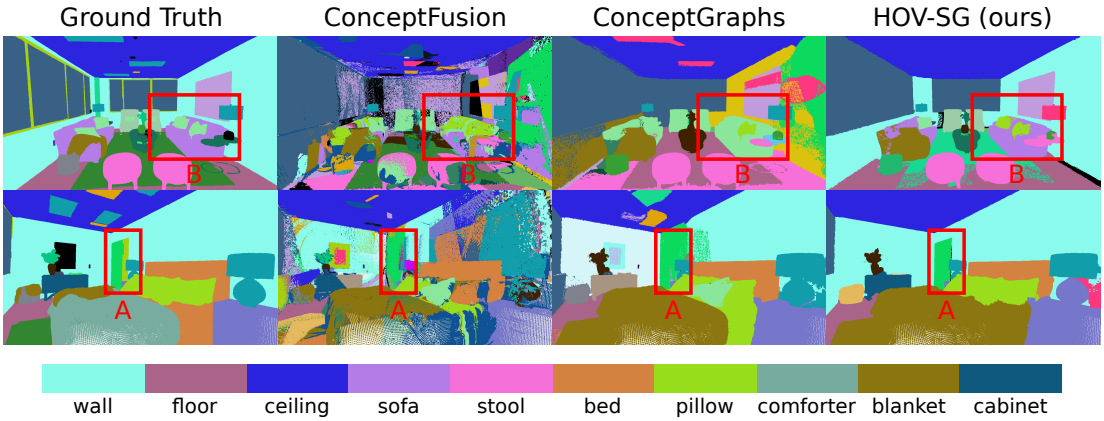


Fig. 6: Qualitative results for 3D semantic segmentation on Replica dataset. For conciseness, the legend only shows ten out of 101 categories in the Replica dataset. Observing the position "A" indicated in the images, only HOV-SG predicts the door with correct boundaries. At position "B" in the images, only HOV-SG predicts the sofa correctly.

ground truth. These maps are finally voxelized using a $0.02\,\mathrm{cm}$ resolution.

1) Floor and Class-Agnostic Room Segmentation: In order to evaluate both floor and class-agnostic room segmentation, we identified several heuristics on top of the provided metadata of the dataset. Since HM3DSem does not include floor height labels, we hand-labeled all upper and lower floor boundaries. In addition, we pooled the annotated objects contained in our constructed ground truth point clouds based on their associated region labels. Based on this, we obtain region-wise point clouds we utilize as ground truth. As shown in Table II, our method achieves 100% accuracy in retrieving the number of floors, both in single-floor as well as multi-floor scenes. In addition, we evaluate the region segmentation performance of HOV-SG and compare with Hydra [19] across eight scenes on HM3DSem. We observe slightly lower precision but a significantly greater recall of 83.59% compared to 77.55%. In addition to the overall results given in Table II, we provide scene-wise results in Table S.1 in the supplementary material. In general, we obtain higher precision and recall on smaller scenes comprising fewer regions. Similar to Hydra [19], our approach utilizes a naïve morphological heuristic to segregate regions, which does not work well on more complex, semantically ambiguous room layouts such as combined kitchen and living rooms. Nonetheless, our approach does not suffer from this drawback too drastically as we equip each segmented region with 10 representative embeddings. This allows adaptive prompting without directly setting a fixed room category.

2) Semantic Room Classification: We quantitatively evaluate our proposed view embedding-based room category labeling method (See Fig. 4) by comparing it against two strong baselines across the set of eight scenes on HM3DSem. Both baselines rely on object labels to classify room categories. To draw a fair comparison, all methods rely on ground truth room segmentation, namely the class-agnostic mask of each room. Thus, all objects are assigned to rooms based on ground truth room layouts. This is different from the general HOV-SG method, which also estimates room masks.

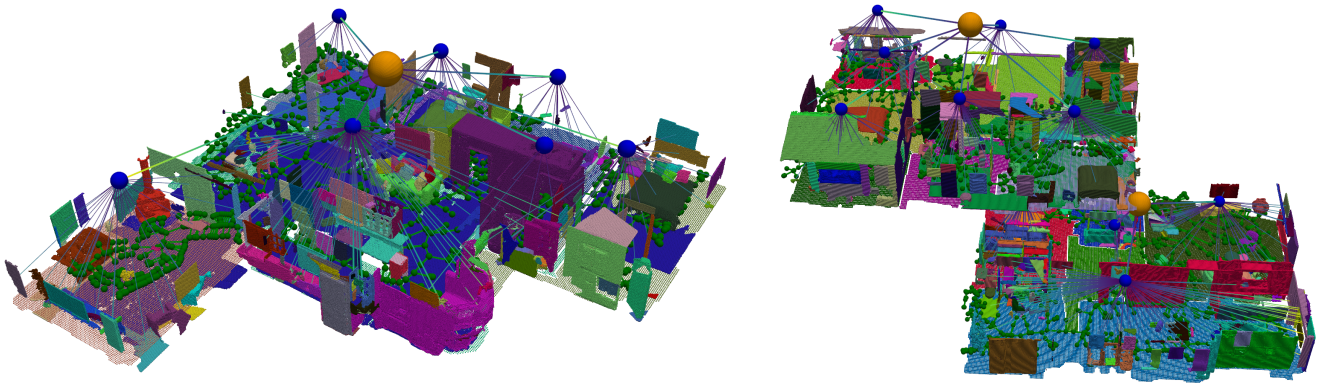
TABLE II: Floor and Region Segmentation on HM3DSem

Method	Floors	Regions	
	Acc_F [%]	Precision [%]	Recall [%]
Hydra [19]	-	**86.18**	77.55
HOV-SG (ours)	100	84.10	**83.59**

Evaluation of the floor and room segmentation: We present the accuracy of correctly predicted floors using a threshold of $0.5\,\mathrm{m}$. The region segmentation precision (P) and recall (R) are calculated based on the metric in Hydra [19].

Dataset: In this evaluation, we utilize a closed set of room categories. The HM3DSem dataset does not provide annotated room categories but merely educated votes, which are often not sufficient. Therefore, we manually labeled the regions of the eight scenes detailed in Sec. IV-B. The list of room categories is provided in Sec. S.2-D.

Baselines: We compare the HOV-SG approach of using filtered view embeddings for labeling rooms against a privileged and an unprivileged baseline. The privileged baseline operates on ground truth object categories contained within each room. In order to obtain room labels, the baseline prompts an LLM (GPT-3.5 and GPT-4 [43]) to provide a room category guess out of the closed set of room categories given the objects per room in a few-shot manner (prompts are detailed in Sec. S.2-D). The second and unprivileged baseline applies the same principle of prompting an LLM but only has access to the predicted object categories obtained using HOV-SG. This means that each predicted object is labeled as the category showing the highest similarity to the object feature among a category list. In general, we expect that the number of objects will be different from the privileged baseline because of under- and over-segmentation of HOV-SG's predictions. In comparison, our view embedding method relies on 10 distinct view embeddings, which are scored against the chosen set of room categories. The final predicted room category is defined by the room category that showed the highest similarity across all view embeddings as described in Fig. 4.



(a) *Scene 00824 (single floor)* (b) *Scene 00861 (two floors)*

Fig. 7: Qualitative visualization of the hierarchical 3D scene graphs produced by HOV-SG of two apartments of the Habitat Matterport 3D Semantics dataset [25]. Yellow nodes represent floors, while blue nodes represent rooms. The green graph right above the respective floor represents the produced navigational graph. For more visualizations, please refer to Fig. S.4.

Metrics: We apply two different evaluation criteria: The first accuracy called $\text{Acc}_{=}$ fosters replicability by evaluating whether the predicted and the ground truth room categories are text-wise equal. Different from that, the performance regarding the $\text{Acc}_{\approx}$ metric is produced via human evaluation. This is crucial as room categories are not always fully determinable when labeling, e.g., combined kitchen and living room areas. Moreover, the answers provided by the LLM do not always state definitive categories because of frequent hallucinations. A high number of objects per room exacerbates this. This particularly applies to the unprivileged baselines when facing under-segmentation. In order to circumvent this, we manually filter all outputs across the set of eight scenes and check whether the LLM *leaned* towards the correct answer such as predicting a synonym of the GT room type, which boosts results in favor of the LLM-based methods. In addition to this, we also evaluated the same task using the current state-of-the-art LLM, GPT-4, which shows significantly fewer hallucinations and increased accuracy.

Results: As presented in Table III, the view embedding method of HOV-SG outperforms all unprivileged baselines both in terms of the strict accuracy ($\text{Acc}_{=}$) as well as the approximate accuracy ($\text{Acc}_{\approx}$) by a significant margin. In addition, HOV-SG also outperforms the privileged baseline relying on GPT-3.5 while showing similar approximate accuracy as GPT-4 of 84.10% vs. 84.25%. Thus, we conclude that our room labeling method is robust and outperforms comparable unprivileged methods by a significant margin. Furthermore, we provide additional scene-wise evaluations in Table S.2 in the supplementary material.

3) Object-Level Semantics: Existing open-vocabulary evaluations usually circumvent the problem of measuring true open-vocabulary semantic accuracy. This is due to arbitrary sizes of the investigated label sets, a potentially enormous and challenging amount of object categories [25], and the ease of use of existing evaluation protocols [10, 13]. While human-level evaluations such as Amazon Mechanical Turk (AMT) partly solve this problem, robust results replication and scaling remain challenging [14].

TABLE III: SEMANTIC ROOM CLASSIFICATION RESULTS (HM3DSEM)

	Room Identification Method	$\text{Acc}_{=}$ [%]	$\text{Acc}_{\approx}$ [%]
Privileged	GPT-3.5 w/ GT object categories	66.89	81.49
	GPT-4 w/ GT object categories	79.86	84.25
Unprivileged	GPT-3.5 w/ predicted object categories	28.48	42.95
	GPT-4 w/ predicted object categories	<u>59.47</u>	<u>62.55</u>
	HOV-SG (ours) w/ view embeddings	**73.93**	**84.10**

We present the room classification performance on HM3DSem of HOV-SG (view embeddings) and two baselines (GPT-3.5 / GPT-4) using either privileged or unprivileged information. In the privileged case, rooms are labeled based on ground truth object categories per room. The unprivileged baselines take the predicted masks and categories as input. We consider two different evaluation criteria: $\text{Acc}_{=}$ measures whether the exact text-wise room category was predicted while $\text{Acc}_{\approx}$ measures semantically correct room predictions based on qualitative human evaluation through manual inspection.

Metrics: In this work, we propose the novel AUC_k^{top} metric that quantifies the area under the top-k accuracy curve between the predicted and the actual ground-truth object category (see Fig. S.2). This entails computing the ranking of all cosine similarities between the predicted object feature and all possible category text features, which are in turn encoded using a vision-language model (CLIP). Thus, the metric encodes how many *erroneous shots* are necessary on average before the ground-truth label is predicted correctly. Based on this, the metric encodes the actual open-set similarity while scaling to large, variably-sized label sets. We envision a future use of this metric in diverse open-vocabulary tasks.

We visualize the AUC_k^{top} curve HOV-SG on the *00824* scene of HM3DSem in Fig. S.2. The closer the curve is

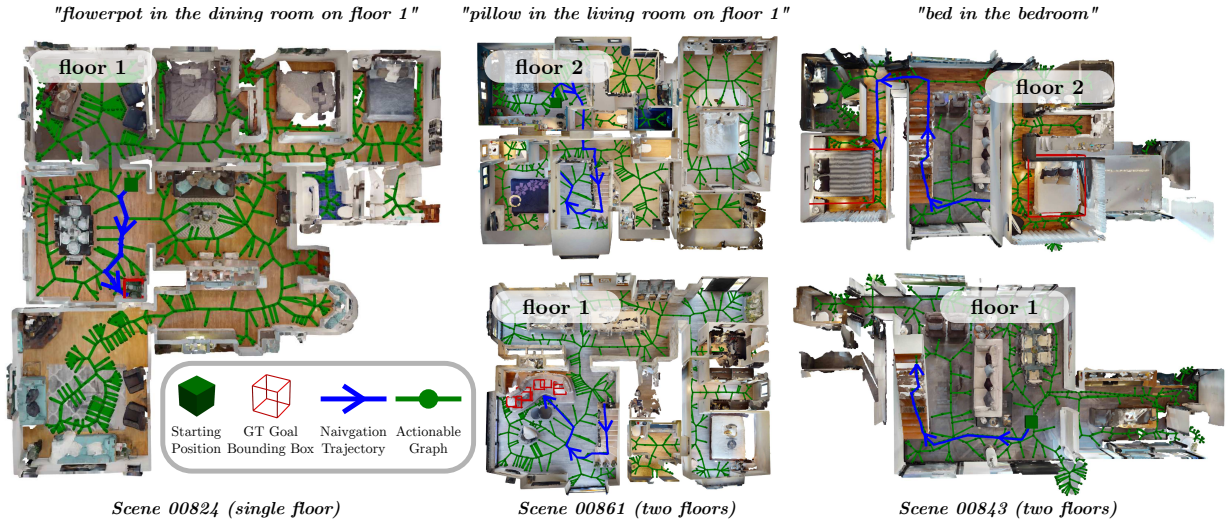


Fig. 8: Qualitative visualization of the language-conditioned navigation in multi-floor environments in HM3DSem dataset. In some cases, there are multiple ground truth objects matching the instruction. Stopping at any of them with a distance within 1 meter is regarded as a success.

to the upper left corner of the plot, the higher the open-vocabulary similarity. Instead of showing the accuracy at distinct values of k, we normalize k over the extent of the label set, which contains 1624 categories for HM3DSem. This also shows visually how the AUC_k^{top} metric provides a dependable measure for large but variably sized label sets.

Baselines: In order to show the applicability of the AUC_k^{top} metric, we compare HOV-SG against two strong baselines VLMaps [10] and ConceptGraphs [14] on the Habitat-Semantics dataset [25], which comprises an enormous label set of 1624 object categories. To allow for a fair comparison, we perform a linear assignment among predicted and GT objects and only consider predicted objects that show an IoU $> 50\%$ with the ground truth. Since VLMaps [10] does not predict masks by design, it takes the object masks predicted by HOV-SG and averages all masked voxels' features as the object feature.

Results: The overall AUC_k^{top} score as well as various top-k thresholds are given in Table IV. We observe that VLMaps [10] performs inferior, which is presumably due to its dense feature aggregation as well as its dependence on a fine-tuned VLM, LSeg [51], limiting its generalization in challenging open-vocabulary scenarios. It does not only score merely 5% of objects correctly given its top-5 choices but also only predicts the correct class given its top-500 predicted classes in 40.02% of all cases. In comparison, ConceptGraphs [14] obtains a competitive score of 84.07% while HOV-SG achieves 84.88%. Especially, up to the top-100 highest ranking classes, HOV-SG outperforms ConceptGraphs.

4) Hierarchical Concept Retrieval: To take advantage of the hierarchical character of our proposed representation, we evaluate to what extent we can retrieve objects from hierarchical queries of the form: *"pillow in the living room on the second floor"* or *"bottle in the kitchen"*. To do so, we decompose the query using GPT-3.5 into its sought-after hierarchical concepts, e.g., `[floor 2, living room, pillow]` or `[-, kitchen, bottle]`), and compute the corresponding CLIP embeddings. In the next step, we hierarchically query against the most suitable floor, the most appropriate room, and lastly, the most suitable object given the query at hand (see Table V). While floor prompting is done naively, we select the room producing the highest maximum cosine similarity to the query room across its ten embeddings. On average, this produces higher success rates compared with mean- or median-based schemes.

TABLE IV: OBJECT-LEVEL SEMANTICS EVALUATION ON HM3DSEM

Method	top_5	top_{10}	top_{25}	top_{100}	top_{250}	top_{500}	AUC_k^{top}
VLMaps [10]	0.05	0.17	0.54	15.32	26.01	40.02	56.20
ConceptGraphs [14]	18.11	24.01	33.00	55.17	**70.85**	**81.55**	84.07
HOV-SG (ours)	**18.43**	**25.73**	**36.41**	**56.46**	69.95	80.86	**84.88**

We provide object-level semantic accuracies across all eight considered scenes within HM3DSem using both the overall AUC_k^{top} metric across 1624 categories as well as accuracies at a few selected thresholds k. To allow for a fair comparison, we perform a linear assignment among predicted and GT objects and only consider predicted objects that show an IoU $> 50\%$ with the ground truth. Since VLMaps [10] does not predict masks by design, it takes the masks predicted by HOV-SG and evaluates wrt. those.

Retrieval Results: In the following, we compare HOV-SG against an augmented variant of ConceptGraphs [14] that is equipped with privileged floor information and it scores objects against the requested room and object, which allows it to draw answers at the floor and room level. As shown in Table V, HOV-SG shows a significant performance increase of 11.69% on object-room-floor queries and a 2.2% advantage on object-room queries when compared with ConceptGraphs. While ConceptGraphs struggles on larger scenes and under more detailed queries, HOV-SG outperforms it by a significant margin even though it suffers from erroneous room segmentations by design. For more information, we refer to Sec. S.1-A.

Language-Grounded Navigation in Simulation: In addition,

TABLE V: OBJECT RETRIEVAL FROM LANGUAGE QUERIES (HM3DSEM)

Query Type	Method	# Trials	Retrieval-SR$_{10}$ [%]	Navigation-SR [%]
(o, r, f)	ConceptGraphs	40.63	16.31	-
	HOV-SG (ours)		**28.00**	37.32
(o, r)	ConceptGraphs	34.88	29.26	-
	HOV-SG (ours)		**31.48**	40.41

Evaluation over 20 frequent distinct object categories in terms of the top-5 accuracy. A match is counted as a success when the IoU > 0.1 between predicted object and ground truth. The number of trials is an average number of trials across the eight scenes evaluated. It is lower for (o,r) compared to (o,r,f) due to a higher number of query duplicates whenever we drop the floor specification. The 20 evaluated categories are: *picture, pillow, door, lamp, cabinet, book, chair, table, towel, plant, sink, stairs, bed, toilet, tv, desk, couch, flowerpot, nightstand, faucet*.

TABLE VI: REAL-WORLD OBJECT RETRIEVAL AND GOAL NAVIGATION FROM LANGUAGE QUERIES

Query Type	# Trials	Graph Querying # Successes	Graph Querying SR [%]	Goal Navigation Success	Goal Navigation SR [%]
Object	41	29	70.7	23	56.1
Room	9	5	55.6	5	55.6
Floor	2	2	100	2	100

We count a retrieval as successful whenever the robot is in close vicinity to the object sought after (~ 1 m).

to the general querying tasks we also perform navigational trials using the Habitat Simulator. Based on the obtained actionable multi-floor navigational Voronoi graphs introduced in Sec. III-B, we traverse the set of retrieved high-probability objects satisfying the query and report the physical retrieval success rate. If the robot stops by one of the matched ground truth point clouds and its distance to it is smaller than 1 m, we consider the navigation successful. As shown in Table V, the navigational success rates are higher when compared to the retrieval success rates. We found that the robot regularly reaches the locations of the predicted objects that are in close vicinity to actual ground truth target objects. This effect is measurable since we rely on a Euclidean distance-based evaluation for the navigational success assessment. Moreover, imperfectly predicted instance masks increase the chance of retrieving segments that do not provide complete overlap with actual ground truth objects in terms of geometry and semantics, which induces slight offsets in the retrieved positions when querying. Example navigation trials are shown in Fig. 8.

C. Real World Language-Grounded Navigation

To validate the system in the real world, we utilize a Boston Dynamics *Spot* robot with a calibrated Azure Kinect RGB-D camera and a 3D LiDAR attached to it. We collect a stream of RGB-D sequences inside a two-story office building, traversing through a variety of rooms with diverse semantic information as shown in Fig. 9.

Real-World Application of HOV-SG: We calibrate the extrinsics between the LiDAR and the RGB-D camera and apply an off-the-shelf LiDAR SLAM implementation that combines FAST-LIO2 [52] with a loop closure component to obtain LiDAR poses. Subsequently, we leverage the extrinsics

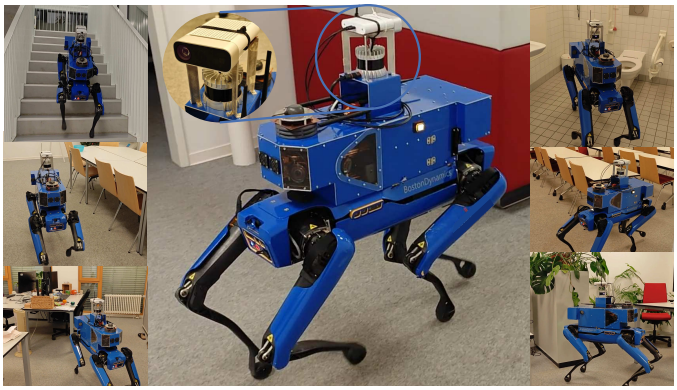


Fig. 9: Boston Dynamics *Spot* robot exploring a two-story office building with multiple types of rooms. The quadruped is equipped with an Azure Kinect RGB-D camera and a 3D LiDAR to collect RGB-D data and odometry.

to derive the associated camera poses. Finally, employing the RGB-D data and odometry, we construct the HOV-SG representation as detailed in Sec. III.

Robot Navigation with Long Queries: Within the two-story building, we select 41 object goals, nine room goals, and two floor goals and use natural language to query the HOV-SG representation built. Some examples of the queries are "go to floor 0", "navigate to the kitchen on floor 1", and "find the plant in the office on floor 0". Unlike previous methods that only retrieved object-level goals, our representation enables long queries containing multiple levels of concepts and facilitates a more detailed constrained retrieval.

To separate the evaluation of our representation and the navigation system, we first evaluate the accuracy of the retrieval tasks qualitatively. Since the building is well structured, we can easily determine the boundary between rooms and regions like offices, corridors, and dining rooms. Meanwhile, we manually label the categories of all regions in the building, using a category set containing `office`, `corridor`, `kitchen`, `seminar room`, `meeting room`, `dining room`, `bathroom`. If the HOV-SG representation returns the correct floor and room point cloud as well as an object point cloud that shows overlap with the correct object, we consider this as retrieval success. We achieve a 100% success rate in floor retrieval, a 55.6% success rate in room retrieval, and a 70.7% success rate in object retrieval. The major failure cases for room retrieval stem from the visual similarity among rooms such as "meeting room", "seminar room", and "dining room" as shown in Fig. S.3.

We query our HOV-SG representation using hierarchical concepts and utilize the *Spot* quadrupedal robot to carry out navigation trials. In our experiments, the robot navigates to queried objects across both floors with a 56.1% success rate while navigating to all successfully retrieved room and floor concepts with language instructions. One failure case occurred for the "go to the whiteboard in the office on the second floor" query. Since the whiteboard is attached to a room-separating wall, the robot achieved the necessary distance to the object but was positioned on the opposite side of the wall as shown in Fig. S.6). In addition to that, we did not consider target locations on stair segments to prevent the robot

TABLE VII: REPRESENTATION SIZE (HM3DSEM)

Scene	# Floors	VLMaps [10]	ConceptGraphs [14]	HOV-SG (ours)
00824	1	568	**143**	**143**
00829	1	407	110	**99**
00843	2	534	143	**125**
00861	2	943	255	**225**
00862	3	1808	**474**	479
00873	2	570	167	**129**
00877	2	556	154	**131**
00890	2	682	192	**162**
Σ	-	6068	1638	**1493**

We compare the storage sizes of the representations produced by VLMaps [10], ConceptGraphs [14], and HOV-SG, measured in megabyte (MB), across eight differently sized scenes of Habitat-Semantics (HM3DSem). The smallest sizes are highlighted **bold**, respectively.

assuming unstable poses. The evaluation results for retrieval and navigation are shown in Table VI. We display a subset of target objects in Fig. S.6 and three trials in more detail in the supplementary material Fig. S.5.

D. Representation Storage Overhead Evaluation

A key advantage of HOV-SG is the compactness of the representation. We compare the storage size of VLMaps [10], ConceptGraphs [14], and HOV-SG created for the eight scenes in the HM3DSem dataset and show the results in Tab. VII. We adapt VLMaps to store LSeg features at 3D voxel locations with 0.05m voxel size. The backbone of the LSeg is ViT-B-32, which has 512-dimensional features. ConceptGraphs and HOV-SG are using the ViT-H-14 CLIP backbones, which requires saving 1024-dimension features in the representation. VLMaps is optimized to only save features at voxels near object surfaces instead of saving redundant features at non-occupied voxels. Nonetheless, thanks to the compact graph structure, ConceptGraphs and HOV-SG are much smaller than their dense counterparts. HOV-SG even reduces as much as 75% in memory footprint on average compared to VLMaps. While ConceptGraphs encodes supplementary object relationships, HOV-SG incorporates hierarchical semantic features. Overall, ConceptGraphs and HOV-SG serve as excellent complementary representations, each emphasizing distinct facets of scene semantics.

E. Ablation Study

In order to shed light on the contributions of various key components in our approach, we present an ablation study on the Replica dataset [23] in Table VIII. One key component of the 3D open-vocabulary segment-level mapping pipeline (Sec. III-A) is the DBSCAN clustering we apply to pixel-wise CLIP embeddings associated with each segment to select the most representative features among them. This design, inspired by the principle of majority voting, has proven effective in mitigating outlier CLIP features caused by the inherent limitations of CLIP and the noise originating from SAM's outputs, thereby enhancing semantic accuracy. A different key component of our approach involves fusing CLIP features extracted from various sources: the global image, the masked image of an object based on its SAM mask, and the masked object image given the SAM mask without background. In contrast to ConceptGraphs [14], which only integrate the global image embedding and the sub-image embedding, we hypothesize that incorporating salient features from the sub-image into CLIP embeddings could enhance accuracy. Based on this, we tested three setups: utilizing only the CLIP embedding of the masked object including background (L-CLIP), employing only the CLIP embedding of the masked object without background (M-CLIP), and third, combining both of these CLIP embeddings by fusing them as done in HOV-SG. Our findings indicate that combining both embeddings yields the highest semantic accuracy as given in Table VIII.

TABLE VIII: ABLATION STUDY ON REPLICA

DBSCAN	L-CLIP	M-CLIP	mIOU ↑	F-mIoU ↑	mAcc ↑
✗	✓	✓	0.212	0.340	0.290
✓	✗	✓	0.136	0.178	0.170
✓	✓	✗	0.215	0.337	0.298
HOV-SG (ours)			**0.231**	**0.386**	**0.304**

DBSCAN indicates whether we apply DBSCAN clustering to select segment features, L-CLIP indicates the use of only masked images including background, and M-CLIP refers to only the masked CLIP embeddings without background.

V. CONCLUSION

We presented HOV-SG, a novel hierarchical open-vocabulary 3D scene graph representation for indoor robot navigation. Through the semantic decomposition of environments into floors, rooms, and objects, we demonstrate effective concept retrieval from abstract language queries and perform long-horizon navigation across a multi-story environment in the real world. With extensive experiments conducted across multiple datasets, we showcase that HOV-SG surpasses previous baselines in terms of semantic accuracy, open-vocabulary capability, and compactness. Nevertheless, HOV-SG is not without limitations. Consisting of several stages and components, our approach necessitates a large number of hyperparameters. Moreover, the construction process of HOV-SG is time-consuming, rendering the method unsuitable for real-time mapping. Furthermore, it assumes a static environment and thus cannot handle dynamic environments. Future research directions may involve developing an open-vocabulary dynamic representation of the environment or integrating a reactive embodied agent to enhance reasoning and grounding in the physical world. To foster future research, we make the code publicly available at https://hovsg.github.io.

VI. ACKNOWLEDGEMENT

This work was funded by the German Research Foundation (DFG) Emmy Noether Program grant number 468878300, the BrainLinks-BrainTools Center of the University of Freiburg, and an academic grant from NVIDIA.

References

[1] E. Jefferies and X. Wang, "Semantic cognition: semantic memory and semantic control," in *Oxford Research Encyclopedia of Psychology*, 2021.

[2] A. A. Kumar, "Semantic memory: A review of methods, models, and current challenges," *Psychonomic Bulletin & Review*, vol. 28, pp. 40–80, 2021.

[3] S. C. Hirtle and J. Jonides, "Evidence of hierarchies in cognitive maps," *Memory & cognition*, vol. 13, no. 3, pp. 208–217, 1985.

[4] B. Kuipers, "The spatial semantic hierarchy," *Artificial Intelligence*, vol. 119, no. 1, pp. 191–233, 2000.

[5] H. Voicu, "Hierarchical cognitive maps," *Neural Networks*, vol. 16, no. 5-6, pp. 569–576, 2003.

[6] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[7] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11576–11582, IEEE, 2023.

[8] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, 2023.

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

[10] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (London, UK), 2023.

[11] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler, "Open-vocabulary queryable scene representations for real world planning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11509–11522, IEEE, 2023.

[12] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "Clip-fields: Weakly supervised semantic fields for robotic memory," *arXiv preprint arXiv: Arxiv-2210.05663*, 2022.

[13] 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," *Robotics: Science And Systems*, 2023.

[14] 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.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics.* MIT Press, 2005.

[16] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio visual language maps for robot navigation," in *Proceedings of the International Symposium on Experimental Robotics (ISER)*, (Chiang Mai, Thailand), 2023.

[17] S. Peng, K. Genova, C. M. Jiang, A. Tagliasacchi, M. Pollefeys, and T. Funkhouser, "Openscene: 3d scene understanding with open vocabularies," in *CVPR*, 2023.

[18] E. Greve, M. Büchner, N. Vödisch, W. Burgard, and A. Valada, "Collaborative dynamic 3d scene graphs for automated driving," *arXiv preprint arXiv:2309.06635*, 2023.

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

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

[21] 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 *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[22] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[23] 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.

[24] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," pp. 5828–5839, 2017.

[25] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4927–4936, 2023.

[26] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, S. Y. Min, K. Shah, C. Paxton, S. Gupta, D. Batra, *et al.*, "Goat: Go to any thing," *arXiv preprint arXiv:2311.06430*, 2023.

[27] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23171–23181, 2023.

[28] Ó. M. Mozos, C. Stachniss, A. Rottmann, and W. Burgard, "Using adaboost for place labeling and topological map building," in *Robotics Research: Results of the 12th*

International Symposium ISRR, pp. 453–472, Springer, 2007.

[29] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1352–1359, 2013.

[30] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric instance-aware semantic mapping and 3d object discovery," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3037–3044, 2019.

[31] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *2018 international conference on 3D vision (3DV)*, pp. 32–41, IEEE, 2018.

[32] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5231–5237, IEEE, 2019.

[33] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2018.

[34] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[35] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19729–19739, 2023.

[36] 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*, pp. 5663–5672, 2019.

[37] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "SceneGraphFusion: Incremental 3D scene graph prediction from RGB-D sequences," in *CVPR*, pp. 7515–7525, June 2021.

[38] 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," *arXiv preprint arXiv:2307.06135*, 2023.

[39] H. Bavle, J. L. Sanchez-Lopez, M. Shaheer, J. Civera, and H. Voos, "Situational graphs for robot navigation in structured indoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9107–9114, 2022.

[40] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3607–3613, 2011.

[41] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3d semantic scene graphs from 3d indoor reconstructions," in *CVPR*, 2020.

[42] G. Chalvatzaki, A. Younes, D. Nandha, A. T. Le, L. F. Ribeiro, and I. Gurevych, "Learning to reason over scene graphs: a case study of finetuning gpt-2 into a robot language model for grounded task planning," *Frontiers in Robotics and AI*, vol. 10, 2023.

[43] R. OpenAI, "Gpt-4 technical report," *arXiv*, pp. 2303–08774, 2023.

[44] A. Rajvanshi, K. Sikka, X. Lin, *et al.*, "Saynav: Grounding large language models for dynamic planning to navigation in new environments," *arXiv preprint arXiv:2309.04077*, 2023.

[45] P. Wu, Y. Mu, B. Wu, Y. Hou, J. Ma, S. Zhang, and C. Liu, "Voronav: Voronoi-based zero-shot object navigation with large language model," *arXiv preprint arXiv:2401.02695*, 2024.

[46] D. Honerkamp, M. Buchner, F. Despinoy, T. Welschehold, and A. Valada, "Language-grounded dynamic scene graphs for interactive object search with mobile manipulation," *arXiv preprint arXiv:2403.08605*, 2024.

[47] Z. Ni, X.-X. Deng, C. Tai, X.-Y. Zhu, X. Wu, Y.-J. Liu, and L. Zeng, "Grid: Scene-graph-based instruction-driven robotic task planning," *arXiv preprint arXiv:2309.07726*, 2023.

[48] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu, "Open-vocabulary semantic segmentation with mask-adapted clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7061–7070, 2023.

[49] S. Thrun and A. Bücken, "Integrating grid-based and topological maps for mobile robot navigation," in *AAAI*, 1996.

[50] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084, 2019.

[51] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," *arXiv preprint arXiv:2201.03546*, 2022.

[52] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.