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

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

***Abstract*— Typically, robotic mapping relies on highly accurate dense representations obtained via approaches to simultaneous localization and mapping. While these maps allow for point/voxel-level features, they do not provide language grounding within large-scale environments due to the sheer number of points. In this work, we present HOV-SG, a hierarchical open-vocabulary 3D scene graph mapping approach for robot navigation. Using open-vocabulary vision foundation models, we first obtain state-of-the-art open-vocabulary maps in 3D. We then perform floor as well as room segmentation and identify room names. Finally, we construct a 3D scene graph hierarchy. Our approach is able to represent multi-story buildings and allows robots to traverse them by providing feasible links among floors. We demonstrate long-horizon robotic navigation in large-scale indoor environments from long queries using large language models based on the obtained scene graph tokens and outperform previous baselines.**

I. Introduction

Humans acquire conceptual knowledge through multi-modal experiences. These 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]–[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]–[9].

Classical methods for robot navigation build dense spatial maps of high accuracy using approaches to simultaneous localization and mapping (SLAM) [10]–[12]. 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 [7], [13]–[18]. While these approaches marry the area of classical robotics with modern open-vocabulary semantics, representing larger scenes while abstracting still poses a considerable hurdle. A number of works approach this using 3D scene graph structures [19]–[21] 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 [22] that focuses on smaller scenes.

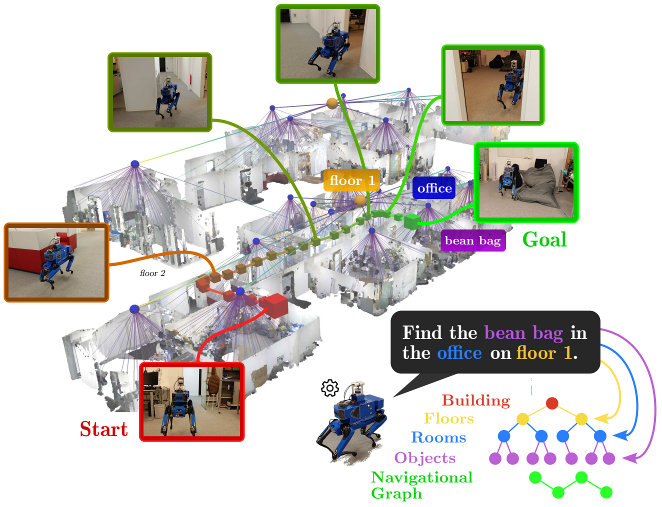


Fig. 1. HOV-SG allows the construction of accurate, open-vocabulary 3D scene graphs for large-scale and multi-story environments and enables robots to effectively navigate in them.

With HOV-SG, we demonstrate the construction of hierarchical 3D scene graphs with open-vocabulary vision-language features [23], [24] across multi-story environments. By abstracting from dense maps and indexing floors, rooms as well as objects, our actionable 3D scene graph hierarchies are promptable using LLMs. This enables object retrieval as well as long-horizon robotic navigation in large-scale indoor environments from long queries as shown in Fig. 1. By doing so, we present state-of-the-art results in open-set 3D semantic segmentation, object retrieval from long queries as well as room identification from perception inputs.

In summary, we make the following main contributions:

- We introduce a pipeline for constructing hierarchical, open-vocabulary 3D scene graphs from a multi-story environment that enables LLM-based prompting, planning as well as robotic navigation.
- We extensively evaluate our approach across three diverse datasets as well as a real-world environment. We achieve state-of-the-art performance in 3D open-set semantic segmentation and demonstrate successful robotic navigation from abstract language queries.
- We introduce a novel evaluation metric for measuring open-vocabulary semantics termed AUC$_{\text{top-k}}$ and publish code at http://hovsg.github.io.

[*] Equal contribution.

[1] Department of Computer Science, University of Freiburg, Germany.

[2] Department of Eng., University of Technology Nuremberg, Germany.

Supplementary material can be found at http://hovsg.github.io

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.

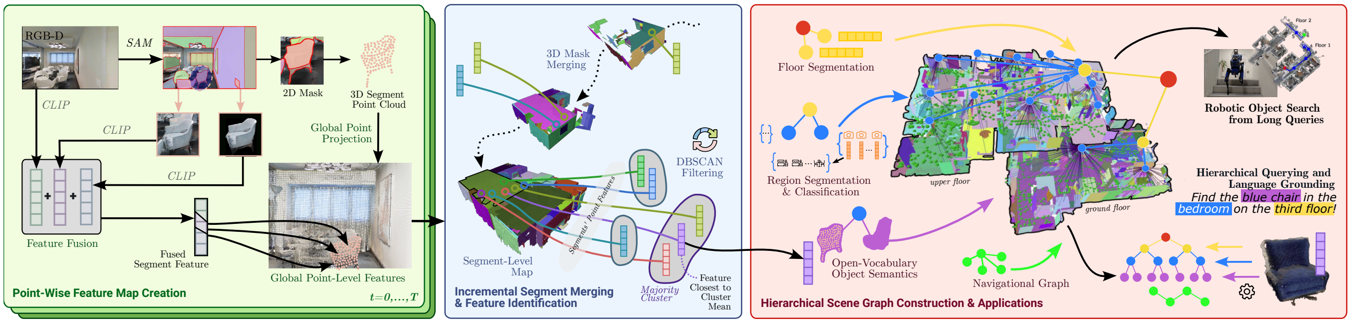

RGB-D
SAM
2D Mask
3D Segment Point Cloud
CLIP
CLIP
CLIP
Global Point Projection
Feature Fusion
Fused Segment Feature
Global Point-Level Features
Point-Wise Feature Map Creation
t=0,...,T
3D Mask Merging
DBSCAN Filtering
Segments' Point Features
Segment-Level Map
Incremental Segment Merging & Feature Identification
Majority Cluster
Feature Closest to Cluster Mean
Floor Segmentation
Region Segmentation & Classification
upper floor
ground floor
Open-Vocabulary Object Semantics
Navigational Graph
Robotic Object Search from Long Queries
Floor 2
Floor 1
Hierarchical Querying and Language Grounding
Find the blue chair in the bedroom on the third floor!
Hierarchical Scene Graph Construction & Applications

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 produce 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.

II. 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 semantic concepts through natural language queries and enable robotic navigation in multi-floor environments. We address this by proposing **H**ierarchical **O**pen-**V**ocabulary **S**cene **G**raphs, in short HOV-SG. In the following, we describe (i) the construction of a 3D segment-level open-vocabulary map (Sec. II-A), (ii) the generation of the hierarchical open-vocabulary scene graphs (Sec. II-B), and (iii) language-conditioned navigation across large-scale environments (Sec. II-C). Fig. 2 presents an overview of our method.

A. 3D Segment-Level Open-Vocabulary Mapping

Frame-Wise 3D Segment Merging: Given a sequence of RGB-D observations, we utilize Segment Anything [24] to obtain a list of class-agnostic 2D binary masks at each timestep. The pixels in each mask are then backprojected into 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 detailed in Sec. S.1-A.

Open-Vocabulary Segment Features: 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 RGB observation and the two mask-wise images with CLIP [23] and fuse them in a weighted-sum manner. Assuming constant CLIP features across 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 then 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.1. In the subsequent step, we match point-wise features with previously computed segments. For each point within a segment, we identify the nearest points in the reference point cloud and collect their CLIP features. To mitigate potential high variance, instead of directly averaging these features, we employ DBSCAN clustering to handle cases like under-segmentation or imperfect views, ensuring a more representative feature by selecting the feature closest to the mean of the majority cluster.

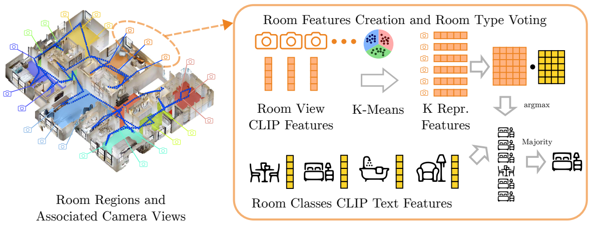

Room Regions and Associated Camera Views
Room Features Creation and Room Type Voting
Room View CLIP Features
K-Means
K Repr. Features
argmax
Majority
Room Classes CLIP Text Features

Fig. 3. By associating 10 camera views, i.e., its CLIP embeddings to each room we obtain a set of 10 open-vocabulary embeddings per segmented room. This serves as the attributed room feature in the scene graph.

B. 3D Scene Graph Construction

In this section, we describe how to build the 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. II-A. The HOV-SG representation comprises a root node, multiple floors, and room nodes as well as object nodes. The edges among nodes represent the hierarchy of the obtained graph. For more details, we refer to Sec. S.1-B.

Floor Segmentation: In order to separate floors, we identify peaks within a height histogram over all points contained in the point cloud. We apply adaptive thresholding and DBSCAN clustering to obtain potential floors and ceilings. We select the top-2 levels in each cluster. Taken pairwise, these represent individual floors (floor and ceiling) in the building as in Fig. S.1. We equip each floor node with a CLIP text embedding using the template "floor {#}". An edge between the root node and the respective floor node is established.

Room Segmentation: Based on each obtained floor point cloud, we construct a 2D bird's-eye-view (BEV) histogram as outlined in Fig. S.1. Next, we obtain walls and apply the Watershed algorithm to obtain a list of 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. Each room constitutes a node and is connected to its corresponding floor.

In order to attribute room nodes, we associate RGB-D observations whose camera poses reside within a BEV room segment to those rooms, see Fig. 3. The CLIP embeddings of these images are filtered by extracting k representative view embeddings using the k-means algorithm. During the query, we compute the cosine similarity between the CLIP text embeddings of a room categories list and each representative feature, resulting in a similarity matrix. With the argmax operation, we obtain opinions from all representatives, allowing retrieval of the room type voted by the majority. These K representative embeddings and the room point cloud are jointly stored in the root node in the graph. An edge between the floor node and each room node and its parent floor node is established. The construction and querying of room features are illustrated in Fig. 3.

Object Identification: We associate global object segments to rooms in the bird's-eye-view based on point cloud overlaps, which is further described in Appendix S.1-C. To reduce the number of nodes, we merge 3D segments of partial overlap that produce equal object labels when queried against a chosen label set. Finally, each obtained 3D segment translates to an object node, and an edge is established between the object and its parent room node.

Actionable Navigational Graph: In addition to the open-vocabulary hierarchy, the scene graph also contains a navigational Voronoi graph that serves robotic traversability of the mapped surroundings [25] spanning multiple floors. This enables high-level planning and low-level execution based on the Voronoi graph. The details of the navigation graph creation are provided in Sec. S.1-D.

C. Object Navigation from Long Queries

HOV-SG extends the scope of potential navigation goals to more specific spatial concepts like regions and floors compared to simple object goals [7], [15], [16], [22]. 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 (GPT-3.5). We break down such lengthy instructions into three separate queries: one for the floor level, one for the room level, and one for the object level. Leveraging the explicit hierarchical structure of HOV-SG, we sequentially query against each hierarchy to progressively narrow down the solution space. Once a target node is identified, 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.4.

III. Experimental Evaluation

In the following, we first evaluate HOV-SG against recent open-vocabulary map representations on the task of 3D semantic segmentation. Secondly, we investigate the accuracy of the constructed hierarchical, open-vocabulary 3D scene graphs from scenes of the Habitat Matterport 3D Semantics Dataset [26]. Finally, we study open-vocabulary object retrieval and demonstrate large-scale language-grounded robotic navigation in the real world.

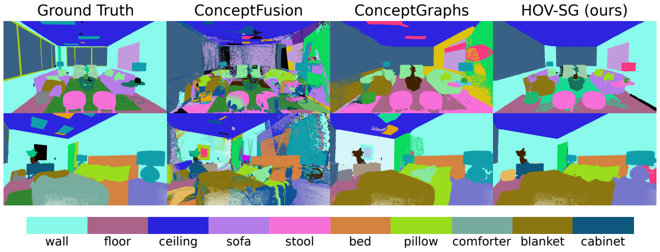


Fig. 4. Qualitative results for 3D semantic segmentation on Replica

TABLE I

Open-Vocabulary 3D Semantic Segmentation

Method	CLIP	Replica			ScanNet		
	Backbone	mIOU	F-mIOU	mAcc	mIOU	F-mIOU	mAcc
ConceptFusion [16]	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 [22]	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 ConceptFusion pipeline evaluated against made use of instance masks predicted by SAM [24]. We consider ViT-H-14 and a fine-tuned backbone ViT-L-14 released with the work OVSeg [29].

A. 3D Semantic Segmentation on ScanNet and Replica

We evaluate the open-vocabulary 3D semantic segmentation performance on the ScanNet [27] and Replica [28] datasets. We compare our method with two competitive vision-and-language representations, namely ConceptFusion [16] and ConceptGraphs [22], and ablate over two CLIP backbones, see Table I. In terms of 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. Second, 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, to some extent mitigate the impact of salient background objects. Further qualitative results are shown in Fig. 4.

B. Scene Graph Evaluation on Habitat 3D Semantics

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 amount of object categories [26], and the ease of use of existing evaluation protocols [7], [16]. While human-level evaluations solve this problem partly, robust replication of results remains challenging [22].

TABLE II

OBJECT-LEVEL SEMANTICS EVALUATION ON HM3DSEM

Method	top_5	top_{10}	top_{25}	top_{100}	top_{250}	top_{500}	AUC$_k^{top}$
VLMaps [7]	0.05	0.17	0.54	15.32	26.01	40.02	56.20
ConceptGraphs [22]	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 8 considered scenes within HM3DSem [26] using both the overall AUC$_k^{top}$ metric across 1624 categories as well as accuracies at a few selected thresholds k.

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. This means 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 various open-vocabulary tasks.

In order to show the applicability of the AUC$_k^{top}$ metric, we compare HOV-SG against two strong baselines on the Habitat-Semantics dataset [26] in Tab. II, which comprises an enormous label set of 1624 object categories. We observe that VLMaps [7] performs inferior, which is presumably due to its dense feature aggregation. In comparison, ConceptGraphs [22] obtains a competitive score of 84.07% while HOV-SG achieves 84.88%. Additional top-k values shed light on how probable it is to score the correct class within a few tries.

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 concepts 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 III). 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. In the following, we compare HOV-SG against an augmented variant of ConceptGraphs [22]. We equip it with privileged floor information while it scores objects against the requested room and object, which allows it to draw answers at the floor and room level. As shown in Tab. III, 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.2-D.

TABLE III

OBJECT RETRIEVAL FROM LANGUAGE QUERIES (HM3DSEM)

Query Type	Method	# Floors	# Regions	# Trials	SR$_{10}$[%]
`(obj, room, floor)`	ConceptGraphs HOV-SG (ours)	1.88	15.63	40.63	16.31 28.00
`(obj, room)`	ConceptGraphs HOV-SG (ours)	1.88	15.63	34.87	29.26 31.48

Evaluation of 20 frequent distinct object categories across 8 scenes. Success rate criterium: IoU > 0.1. The floor and room counts refer to the ground-truth labels.

TABLE IV

REAL-WORLD OBJECT RETRIEVAL FROM LANGUAGE QUERIES

Query	#	Graph Querying		Goal Navigation	
Type	Trials	# Successes	SR [%]	Success	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 ($\sim 1\,\mathrm{m}$).

C. Real-World Experiments

To validate the system in the real world, we conduct multiple navigation trials using a Boston Dynamics *Spot* quadruped. First, we collect an RGB-D sequence of a two-storage office building comprising a variety of room types as well as objects. Using this data, we create the hierarchical 3D scene graph presentation as introduced in Sec. II and Fig.1.

Robot Navigation from Long Queries: We select 41 distinct object goals, 9 room goals, and 2 floor goals and use natural language to query the HOV-SG representation as detailed above. Some examples of the queries are *go to floor 0*, *navigate to the kitchen on floor 1*, or *find the plant in the office on floor 0*. Similar to the evaluation on Habitat-Semantics, we first evaluate general object retrieval given the scene graph. HOV-SG achieves a 100% success rate on floor retrieval, a 55.6% for room retrieval, and a 70.7% success rate for object retrieval. The major failure cases for room retrieval stem from the visual ambiguity among “meeting room”, “seminar room”, and “dining room”. Based on this, we evaluated the object navigation capabilities from abstract, hierarchical queries in the real world using the *Spot* quadruped. We observe a 56.1% success rate in object navigation while traversing multiple rooms as well as floors given an abstract long query. These results prove the efficacy of HOV-SG in enabling real-world agents to navigate to language-conditioned goals across multiple floors.

IV. CONCLUSION

We presented a novel pipeline for constructing hierarchical open-vocabulary 3D scene graphs for robot navigation. Through the semantic decomposition of environments into floors and rooms, we demonstrate effective object retrieval from abstract queries and perform long-horizon navigation in the real world. By doing so, we outperform previous baselines in open-set mapping.

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] 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.

[8] 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)*. IEEE, 2023, pp. 11 576–11 582.

[9] 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.

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

[11] N. Vödisch, D. Cattaneo, W. Burgard, and A. Valada, "Continual slam: Beyond lifelong simultaneous localization and mapping through continual learning," in *The International Symposium of Robotics Research*, 2022, pp. 19–35.

[12] J. Arce, N. Vödisch, D. Cattaneo, W. Burgard, and A. Valada, "Padloc: Lidar-based deep loop closure detection and registration using panoptic attention," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1319–1326, 2023.

[13] 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*. PMLR, 2023, pp. 492–504.

[14] 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)*. IEEE, 2023, pp. 11 509–11 522.

[15] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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*. PMLR, 2021, pp. 8748–8763.

[24] 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.

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

[26] 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*, 2023, pp. 4927–4936.

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

[28] 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.

[29] 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*, 2023, pp. 7061–7070.

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

- Supplementary Material -

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

In this supplementary material, we expand upon multiple aspects of our main paper. In Sec. S.1, we detail several design choices regarding the proposed segment merging, the hierarchical scene graph construction, the navigational graph as well as a semantic localization scheme. In Sec. S.2, we additionally present experimental results that support the claims introduced in the manuscript. This includes a more detailed discussion of the proposed open-vocabulary metric, an analysis regarding identified semantic room categories, and additional baselines regarding object retrieval from language queries. Moreover, we provide insightful visualizations of the produced multi-story scene graphs, representing scenes from both Habitat Semantics as well as our real-world environment.

S.1. Method Details

A. Merging Frame-Wise Segments

Given the frame-wise 3D segments created at all timesteps ${}^{W}P_{ik}$ where $i = 1, \ldots, N$, $k = 1, \ldots, K_i$, and ^{W} indicates world coordinates, we merge the overlapping point clouds across frames based on the geometric distances between point clouds, following a similar merging scheme as Gu *et al.* [22]. We maintain a list of global 3D segments that are incrementally constructed $\mathcal{S} = \{S_j\}_{j=1,\ldots,J}$, where J is the total segment number. For each new frame, we add all segments to the global segments list and compute the pair-wise overlapping ratio between all segment pairs in the global segments list. The overlapping ratio $R(m, n)$ between segment m and n is computed as:

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

where *overlap*$(;\dot{})$ is computed by taking the number of points in a that can find a neighboring point in b within a certain distance threshold divided by the total number of points in a. Different from Gu *et al.* [22], who merges new segments with one global segment that has the largest overlapping ratio, we construct a graph based on the overlapping ratios among the segments. When the ratio is above a certain threshold, we establish an edge between the two segments and merge all connected segments. In this way, one segment can merge with more segments, which is useful in situations in which an incoming segment is filling, e.g., a gap between two already registered global segments.

B. Scene Graph Formalization

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 $\mathcal{N} = \mathcal{N}_{root} \cup \mathcal{N}_F \cup \mathcal{N}_R \cup \mathcal{N}_O$, consisting of a root node $\mathcal{N}_{root}$, 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 $\mathcal{N}_{root}$ contains the point cloud of the concept it refers to and the open-vocabulary features associated with it. The edges can be written as $\mathcal{E} = \mathcal{E}_{0F} \cup \mathcal{E}_{FR} \cup \mathcal{E}_{RO}$. Here, $\mathcal{E}_{0F}$ 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.

C. Hierarchical 3D Scene Graph

Floor Segmentation: Given the point cloud of the whole environment, we plot the histogram of all points along the axis indicating the height (bin size $0.01\,\mathrm{m}$). Next, we extract local peaks in this histogram (within a local range of $0.2\,\mathrm{m}$) and select only peaks that are exceed at least 90% of the highest peak. We apply DBSCAN and select the top-2 heights in each cluster. After that, every 2 consecutive values in the sorted height vector represents a single floor (floor and ceiling) in the building. The floor segmentation process is shown in Fig. S.1. Using the heights above, we can extract floor point clouds for each floor $\mathcal{P}_{Fl}$ where l is the floor number. We compute the CLIP text embedding of “floor {#}” and pack it with the floor point cloud as a floor node N_{Fl} in the graph. An edge between the root node and this floor node $E(N_{root}, N_{Fl}) \in \mathcal{E}_{0F}$ is also established.

Room Segmentation: After segmenting the floors, we use each floor point cloud to further segment room regions. We first compute the 2D histogram of the floor point in the bird’s-eye-view. We extract the binary wall skeleton mask $M_w \in \{0, 1\}^{\bar{H} \times \bar{W}}$ in the top-down map by selecting locations where the histogram density is higher than a certain threshold. We apply dilation to the wall mask to enhance the skeleton and compute a Euclidean Distance Field based on this. We extract a list of isolated regions by taking locations that have distance values higher than a certain threshold. These regions are later used as the seeds for the watershed algorithm to obtain a list of region masks $\{M_r\}_r = 1...R$. The room segmentation process is shown in Fig. S.1.

Using each top-down region mask M_r, we extract all points falling into both the region mask as well as the respective floor to form room point clouds. Simultaneously, we collect

[*] These authors contributed equally.

[1] Department of Computer Science, University of Freiburg, Germany.

[2] Department of Eng., University of Technology Nuremberg, Germany

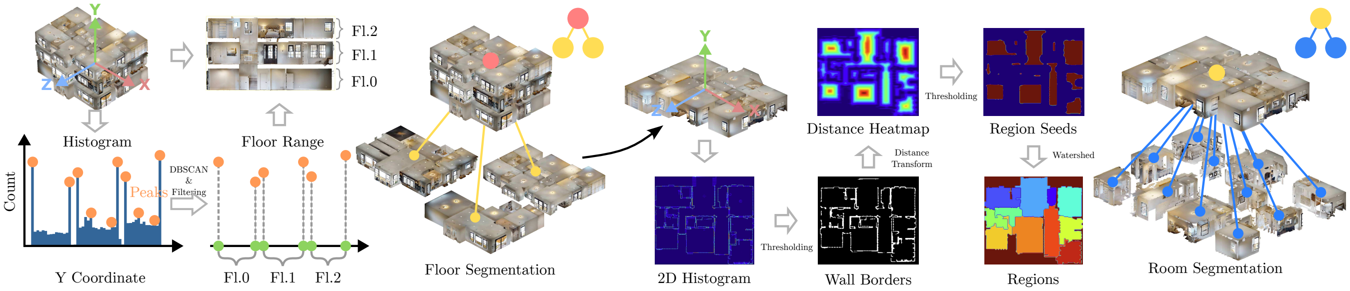


Fig. S.1. Floor and Room Segmentation. Given the point cloud of the whole environment, floor and room nodes are subsequently derived based on geometric heuristics. Floor boundaries are computed by finding peaks of the pixel density along the height direction followed by filtering while room segment masks are extracted with the Watershed algorithm.

camera poses within each room, and encode their images with a CLIP image encoder, generating CLIP features for all room views. Instead of assigning only one feature to each room, we propose applying the k-means algorithm to extract k representative features for each room, covering diverse aspects of each region. During the query, given a list of room categories, we encode them with a CLIP text encoder and compute the cosine similarity between each representative feature and each category, resulting in a similarity matrix. By performing the argmax operation, we obtain scores from all representatives, which allows the retrieval of the room type voted by the majority. These k representative embeddings and the room point cloud are jointly stored in the room node N_{FlRr} in the graph, representing room r on floor l. An edge between the floor node and each room node $E(N_{Fl}, N_{Rr}) \in \mathcal{E}_{FR}$ is established. The room feature construction and querying routines are illustrated in Fig. 3.

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 to the room with the smallest Euclidean distance. We prompt a list of categories to the segment features to classify the segments. Then we compute the pair-wise overlapping ratio for all segments as in Sec. S.1-A. We then merge segments that are in the same category and with overlapping ratios above a certain threshold. Each merged point cloud is an object node N_{FlRrOo}, denoting object o in room r on floor l, and an edge $E(N_{FlRr}, N_{FlRrOo}) \in \mathcal{E}_{RO}$ is established between the object N_{FlRrOm} and the room node N_{FlRr}.

D. Actionable Navigational Graph

The creation of actionable graphs involves constructing per-floor and cross-floor action graphs. For the floor-level action graph, the approach entails computing the free space map of the floor and creating a Voronoi graph [25] based on it. To obtain the free space map, we first project all camera poses on the floor to the top-down plane and consider areas within a certain radius of each pose as navigable. Subsequently, the entire floor's region is obtained by projecting all points on the floor to the top-down plane, and 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 and δ_1, δ_2 are two thresholds we define. In this paper, we use $\delta_1 = 0.2$ and $\delta_2 = 1.5$. By combining the pose region map with the floor region map and subtracting the obstacle region map, the free space map for the floor is derived. The Voronoi graph of this free map yields the floor action graph. (See Fig. S.2).

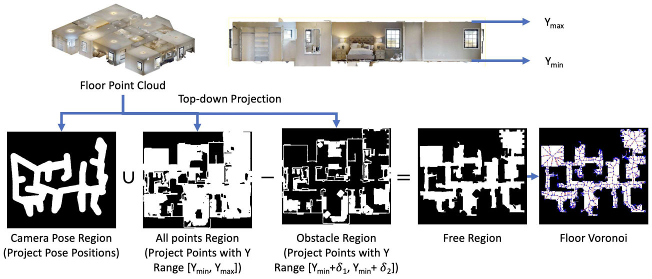

Floor Point Cloud
Y_{max}
Y_{min}
Top-down Projection
∪
−
=
Camera Pose Region (Project Pose Positions)
All points Region (Project Points with Y Range [Y_{min}, Y_{max}])
Obstacle Region (Project Points with Y Range [$Y_{min}+\delta_1$, $Y_{min}+\delta_2$])
Free Region
Floor Voronoi

Fig. S.2. Single-floor Navigational Graphs.

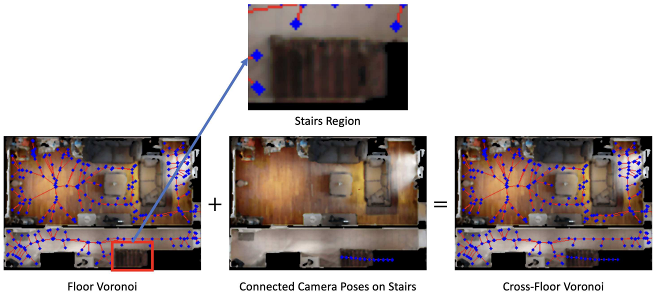


Fig. S.3. Cross-floor Navigational Graphs.

To enable navigation across floors, camera poses on stairs are connected to form a stairs graph. Then, the closest nodes between the stairs graph and the floor graph are selected and connected, see Fig. S.3.

E. Semantic Localization

HOV-SG achieves agent localization within the graph using only RGB images and local odometry using a simple particle filter. The process involves randomly initializing K particles within the free space map, estimated from each floor's point cloud. Subsequently, the global CLIP feature of the RGB image and the CLIP feature of objects within the image

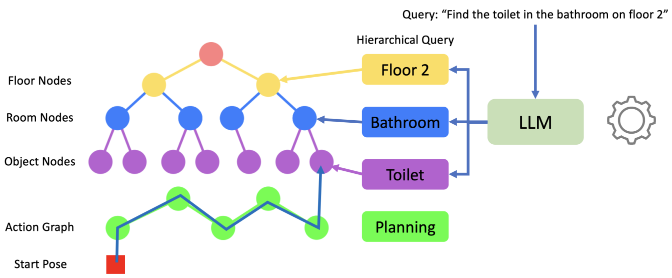


Fig. S.4. The language-grounded navigation module of HOV-SG allows to parse complex queries such as "find the toilet in the bathroom on floor 2" into three queries using a large language model (GPT-3.5) - one each for the floor, room, and object levels. Leveraging HOV-SG's hierarchical structure, we progressively narrow down the search space by querying at each level. Once the target location is identified, the action graph in HOV-SG is used to plan a path from the starting pose to the target.

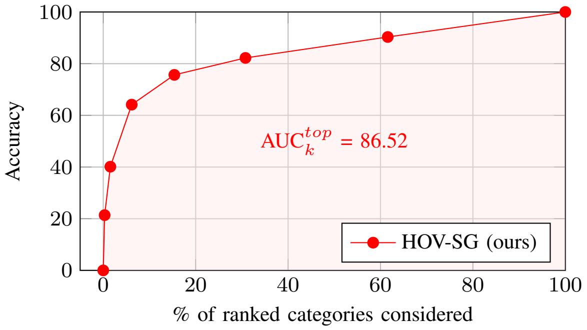


Fig. S.5. We visualize the AUC_k^{top} curve for different evaluation thresholds k, which this plot measures in terms of percent out of the total number of categories (HM3DSem: 1624). The shown curve represents the results of our method HOV-SG on the HM3DSem scene *00824*.

are extracted using the same pipeline as used for the graph creation. In the prediction step, the particle poses are updated based on robot odometry. Thus, we assign each particle a floor and room based on its updated coordinates. In the update step, we calculate cosine similarity scores between the current RGB image's global CLIP feature and the graph's room features for each particle. Additionally, scores are computed between object features in the RGB image and observed objects in front of each particle. Then, particle weights are adjusted based on these similarity scores. This integrated approach allows HOV-SG to semantically localize the agent within the graph at the floor and room level within a short span of 10 observed frames.

S.2. Experimental Evaluation

A. Open-Vocabulary Similarity Metric (AUC_k^{top})

In this section, we present a visualization of the open-vocabulary similarity metric AUC_k^{top} introduced in the paper. As shown in Fig. S.5, the AUC_k^{top} metric represents the area under the top-k accuracy curve. The closer this curve is to the upper left point, the higher the open-vocabulary similarity. Instead of showing the accuracy at distinct values of k as in the main paper, we normalize k over the extent of the label category 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. We envision the future use of this metric in a number of open-vocabulary tasks.

B. Open-Vocabulary Semantics Evaluation

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 [7] does not predict masks by design, it takes the masks predicted by HOV-SG and evaluates wrt. those.

C. Room Classification

We quantitatively support our proposed view embedding-based room category labeling method by comparing it against two strong baselines across the set of 8 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. Thus, objects are assigned to rooms based on ground-truth room layouts. This is different from the general HOV-SG method, which estimates room segments. Please refer to the main manuscript for room segmentation results.

In this evaluation, we utilize a closed set of room categories. To do so, we manually labeled the regions of the 8 scenes as given in Tab. S.1. The HM3DSem dataset does not provide annotated room categories but merely educated votes, which are often not sufficient. We will make the used room labels available as part of this work. The first and privileged baseline operates on ground-truth maps. This means that ground-truth rooms, objects (masks), and object categories are taken. In the next step, the baseline prompts GPT3.5 to provide a room category guess (out of the closed set of room categories) based on the objects contained in each room. The second and unprivileged baseline applies the same principle of prompting GPT3.5 but relies on the solutions obtained by HOV-SG. This means that object masks are not perfect and the top-1 predicted object category is taken to label objects. In general, we expect that the number of objects will be different from the privileged baseline because of under- and over-segmentation of the produced solutions. In comparison, our view embedding method relies on 10 distinct view embeddings which are scored against the defined set of room categories. The final predicted room category is defined by the room category that showed the highest similarity across all view embeddings, which is further described in the main manuscript.

We apply two different evaluation criteria: The first accuracy called $\mathrm{Acc}_{\mathrm{GT}}$ fosters replicability by evaluating whether the predicted and the ground-truth room category are text-wise equal. Different from that, the performance regarding the $\mathrm{Acc}_{\mathrm{valid}}$ metric is produced via human evaluation. This is crucial as room categories are not always fully determinable when labeling such as combined kitchen and living room areas. On top of that, the answers provided by GPT3.5 do not always state definitive categories because of frequent hallucinations. This is exacerbated by a high number of objects per room.

TABLE S.1

Semantic Room Classification Results (HM3DSem).

Room Identification Method	Scene	Acc_{GT} [%]	$\text{Acc}_{\text{valid}}$ [%]
GPT3.5 w/ ground-truth object categories (privileged)	*00824*	70.00	90.00
	00829	71.43	100.0
	00842	61.54	69.23
	00861	58.33	70.83
	00862	50.00	72.22
	00873	81.82	90.91
	00877	69.23	76.92
	00877	72.73	81.82
	Overall	66.89	81.49
GPT3.5 w/ predicted object categories (unprivileged)	*00824*	30.00	40.00
	00829	42.86	57.14
	00842	38.46	38.46
	00861	16.67	25.00
	00862	19.44	25.00
	00873	45.45	63.64
	00877	07.69	30.77
	00877	27.27	63.64
	Overall	28.48	42.95
View embeddings (ours)	*00824*	80.00	90.00
	00829	85.71	100.0
	00842	69.23	76.92
	00861	54.17	79.17
	00862	63.89	83.33
	00873	90.91	90.91
	00877	61.54	61.54
	00877	81.82	90.91
	Overall	**73.93**	**84.10**

The table shows the room classification performance of our method (view embeddings) and two baselines (at the top) on HM3DSem. The baselines utilize GPT3.5 for labeling the rooms based on either ground-truth objects (masks) and categories or on predicted masks and categories. We consider two different evaluation criteria: Acc_{GT} measures whether the exact text-wise room category was predicted while $\text{Acc}_{\text{valid}}$ measures correct room labels based on qualitative human evaluation.

TABLE S.2

Object Retrieval from Language Queries (HM3DSem).

Query Type	Method	Scene	# Floors	# Regions	# Trials	SR_{10}[%]
`(o, r, f)`	ConceptGraphs	*00824*	1	10	33	33.33
		00829	1	7	20	**65.00**
		00843	2	13	26	03.85
		00861	2	24	55	01.82
		00862	3	36	90	**21.11**
		00873	2	11	28	10.71
		00877	2	13	32	09.38
		00890	2	11	41	04.88
		Overall	-	-	40.63	16.31
	HOV-SG (ours) w/ OVSeg	*00824*	1	10	33	**57.57**
		00829	1	7	20	45.00
		00843	2	13	26	**34.62**
		00861	2	24	55	**25.45**
		00862	3	36	90	**21.11**
		00873	2	11	28	**14.29**
		00877	2	13	32	**25.00**
		00890	2	11	41	**21.95**
		Overall			40.63	**28.00**
`(o, r)`	ConceptGraphs	*00824*	1	10	33	33.33
		00829	1	7	20	**65.00**
		00843	2	13	23	34.78
		00861	2	24	46	19.57
		00862	3	36	67	**26.98**
		00873	2	11	25	**30.00**
		00877	2	13	24	25.00
		00890	2	11	41	21.95
		Overall	-	-	34.88	29.26
	HOV-SG (ours)	*00824*	1	10	33	**57.58**
		00829	1	7	20	45.00
		00843	2	13	23	**39.13**
		00861	2	24	46	**30.43**
		00862	3	36	67	20.63
		00873	2	11	25	20.00
		00877	2	13	24	**33.33**
		00890	2	11	41	**21.95**
		Overall	-	-	34.88	**31.48**

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 floor and room counts refer to the ground-truth labels. The number of trials is lower for `(o,r)` compared to `(o,r,f)` because we observe a higher number of query duplicates whenever we drop the floor specification. The 20 categories evaluated are: *picture, pillow, door, lamp, cabinet, book, chair, table, towel, plant, sink, stairs, bed, toilet, tv, desk, couch, flowerpot, nightstand, faucet*.

This particularly applies to the unprivileged baselines when facing over-segmentation. In order to circumvent this, we manually filter all outputs across the set of 8 scenes and check whether the LLM *leaned* towards the correct answer, which boosts results in favor of the LLM-based methods.

As presented in Tab. S.1, the view embedding method outperforms even the privileged baseline that relies on ground-truth object categories by $\sim$ 7% wrt. the strict accuracy evaluation (Acc_{GT}). We also observe a significant performance gap in terms of human evaluation, which is at 2.6%. There is only a single scene in which the privileged baseline outperforms our view embedding method (*00877*). The naïve baseline operating on predicted object categories is significantly outperformed, which is mostly due to over-segmentation and wrongly predicted top-1 object categories. Thus, we conclude that our method is robust and even outperforms privileged methods by a significant margin.

D. Language-Grounded Navigation with Long Queries

In order to support our proposed hierarchical segregation of the environment, we present another comparison with ConceptGraphs [22]. To do so, we compare the object retrieval from language queries performance to demonstrate the efficacy of hierarchically decomposing scenes. We draw this comparison by augmenting ConceptGraphs to also work with room and floor queries. For both HOV-SG as well as ConceptGraphs, we decompose the original query via GPT3.5 parsing as before. Using this, we obtain text variables stating the requested floor name, room name, and object name. Since the floor segmentation of HOV-SG consistently showed 100% accuracy, we directly provide ConceptGraphs with that information. Our augmentation of ConceptGraphs allows us to implicitly identify potential target rooms and objects: We compute the cosine similarity between the set of all object embeddings and the queried room text. Similarly, we compute the cosine similarity between the set of all objects and the queried object name. We combine these two similarities by taking the product of those scores per object to identify the most probable objects. This allows ConceptGraphs to draw answers at the floor level and room level. The remaining details of this evaluation are detailed in the main manuscript.

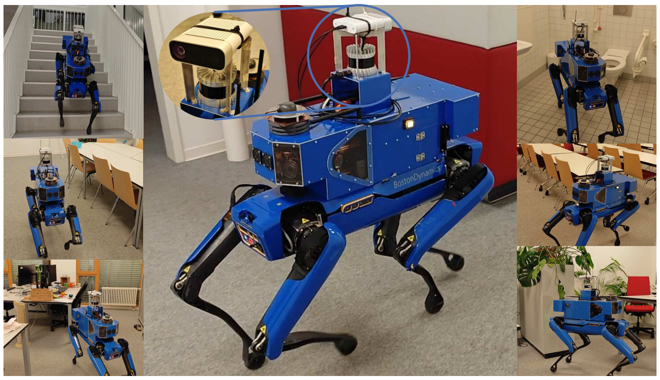


Fig. S.6. Boston Dynamics *Spot* robot traversing a two-story office building with multiple types of rooms. The quadruped is equipped with an Azure Kinect RGB-D camera and a 3D LiDAR. We obtain accurate pose estimates from LiDAR-based odometry estimation.

The results in Tab. S.2 regarding object-room-floor queries demonstrate a significant performance improvement of 11.69% when using HOV-SG compared to ConceptGraphs. We observe that ConceptGraphs struggles with larger scenes and under-segmentation of the produced maps, which often makes finding the object in question hard. Regarding the object-room queries, the drawbacks of ConceptGraphs are not as apparent because the search domain is significantly larger. Still, HOV-SG shows a 2.2% advantage over ConceptGraphs. In general, erroneous room segmentations produced by HOV-SG make finding the object in question hard, which remains subject to future work.

E. Graph Representation on HM3DSem

In the following, we also show the produced hierarchical 3D scene graphs on the set of 8 scenes we evaluated in Fig. S.7. Each distinct object is colored with a different color and the ground truth floor surface is underlayed for easier visibility. The blue nodes denote rooms and its links to the objects denote the object-room associations. The edges among the yellow nodes and the blue nodes show the association between rooms and floors. For clear visualization, we do not visualize the root node that connects multiple floors. We reject certain objects for visualization based on their top-1 predicted object category (out of 1624 categories). Any categories containing sub-strings of the following have not been visualized: wall, floor, ceiling, paneling, banner, overhang. All other predicted object categories are shown. Remarkably, this procedure removed the fair majority of ceilings, walls, etc., which confirms the accuracy of the top-1 predicted open-vocabulary object labels. Nonetheless, future work should address the problem of over- and under-segmentation in these maps. Coping with multiple overlapping masks produced during iterative mask merging is still an open question. While having multiple overlapping masks per point drives the recall in semantic retrieval, this does not produce visually appealing maps. In general, one could argue that depending on the language query at hand different concepts are requested. In case of a query such as *"Find the sofa"*, one would like to obtain the mask that encloses the whole sofa. On the other hand, if the query comes in the form of *"Find the cushion"* (on the sofa), we would want to singulate the cushion in question. This however is difficult when the sofa is masked as one, which would then be considered under-segmentation. Thus, we envision maps that can hold multiple overlapping object masks that could represent various sub-concepts. Essentially, this translates to an additional object hierarchy layer that decomposes objects into their parts.

TABLE S.3
REPRESENTATION SIZE (HM3DSEM)

Scene	Floor Number	Size (MB)		
		VLMaps [7]	ConceptGraphs [22]	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**
Sum	-	6068	1638	**1493**

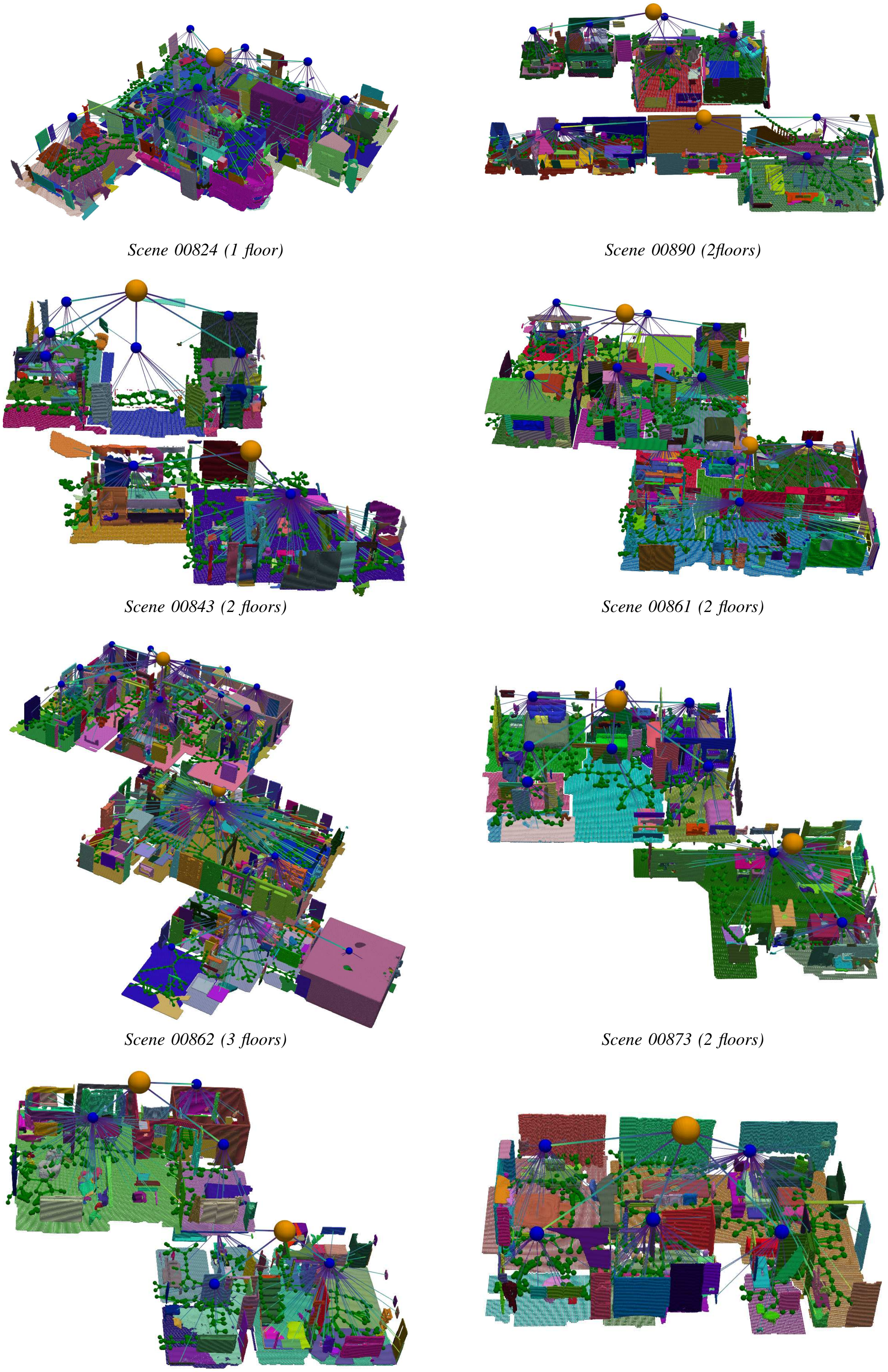
Evaluation of representation size of HOV-SG compared to VLMaps and ConceptGraphs.

F. Real-World Environment

In Fig. S.8, we present three real-world trials that were executed with a Boston Dynamics Spot quadrupedal robot, which allowed us to traverse multi-floor environments safely, see Fig.S.6. The trials are performed based on complex hierarchical language queries that specify the floor, the room, and the object to find. All hierarchical concepts relied on in these experiments are identified using our open-vocabulary HOV-SG pipeline. The top row in Fig. S.8 shows the taken path (blue) from the start position (red) to the goal location (green). The following rows show the time-wise progression of the trial from top to bottom. The unique difficulty in these experiments is the typical office/lab environment with many similar rooms, which often produced similar room names. Having semantically varied rooms instead drastically simplifies these tasks. Nonetheless, as reported in the main manuscript, we reach real-world success rates of around 55%.

G. Representation Storage Overhead Evaluation

A key advantage of HOV-SG is the compactness of the representation. We compare the sizes of VLMaps [7], ConceptGraphs [22], and HOV-SG created for the eight scenes in the Habitat Matterport 3D Semantic dataset and show the results in Table S.3. We adapt VLMaps to store LSeg features at 3D voxel locations. 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 a 1024-dimension feature 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% of storage on average compared to VLMaps.



Scene 00824 (1 floor)

Scene 00890 (2floors)

Scene 00843 (2 floors)

Scene 00861 (2 floors)

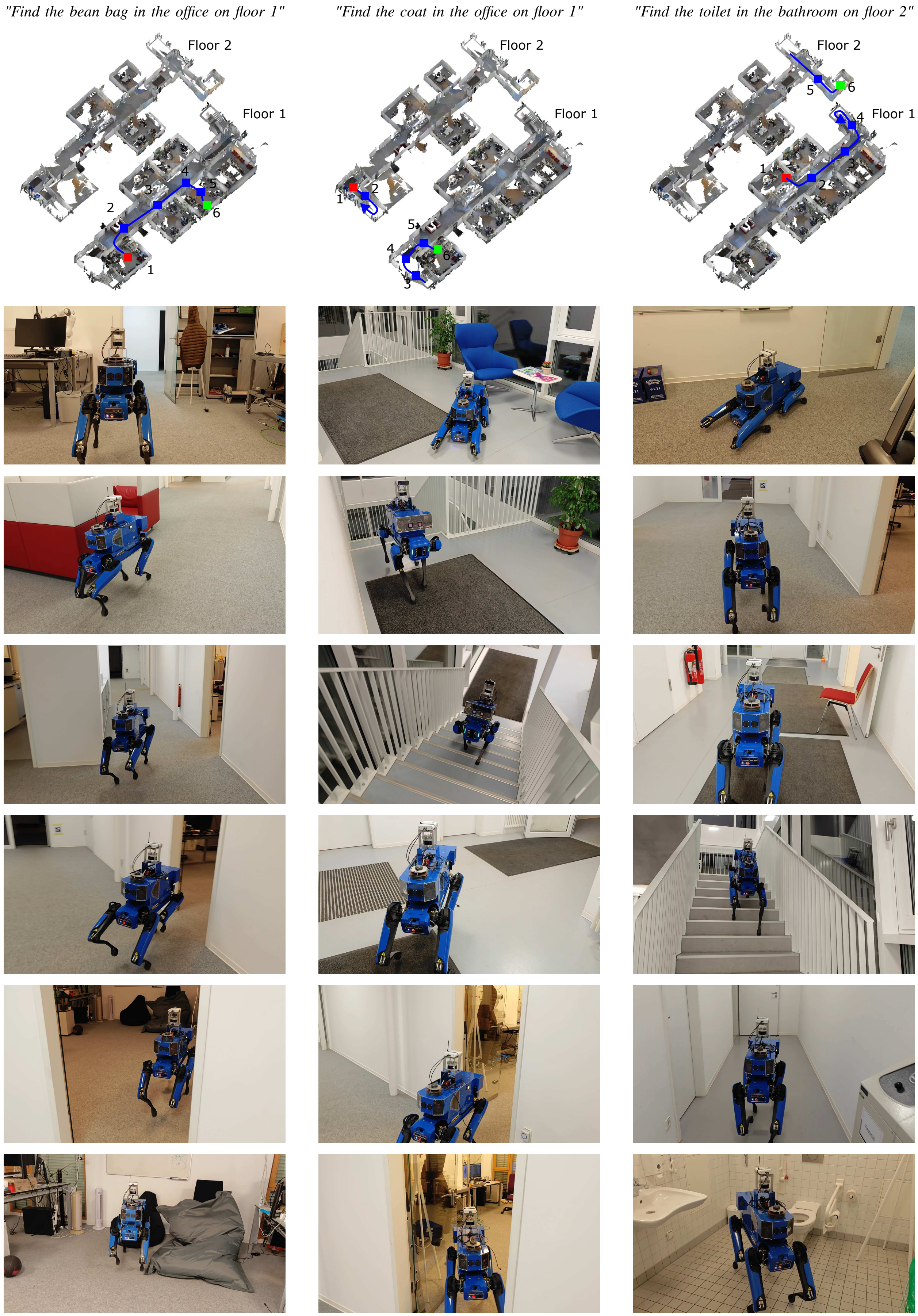
Scene 00862 (3 floors)

Scene 00873 (2 floors)

Scene 00877 (2 floors)

Scene 00829 (1 floor)

Fig. S.7. We show a visualization of the hierarchical open-vocabulary scene graphs produced on HM3Dsem. To make the visualization more clear we do not show the root node connecting (multiple) floors. In addition, we underlay the ground-truth floor surface for easier visibility. We reject certain objects for visualization based on their top-1 predicted object category (out of 1624 categories). Any categories containing sub-strings of the following have not been visualized: `wall`, `floor`, `ceiling`, `paneling`, `banner`, `overhang`. All other predicted object categories are shown. Remarkably, this procedure removed the fair majority of ceilings, walls, etc., which confirms the accuracy of the top-1 predicted open-vocabulary object labels. Best viewed zoomed in.


"Find the bean bag in the office on floor 1"

"Find the coat in the office on floor 1"

"Find the toilet in the bathroom on floor 2"

Fig. S.8. Real-World Object Navigation from Language Queries: We show a set of qualitative results of the real-world demonstration trials, which uses a Boston Dynamics Spot to allow for multi-floor traversals. The first row displays the observed scene and the taken path from the start (red) to the goal location (green). The following rows detail the time-wise progression (top-to-bottom).