FunGraph: Functionality Aware 3D Scene Graphs Generation from 2D and 3D Data

Anonymous CVPR submission

Paper ID *****

Abstract

The concept of 3D scene graphs (3DSGs) is increasingly recognized as a powerful semantic and hierarchical representation of the environment. Current approaches often address this at a coarse, object-level resolution. In contrast, our goal is to generate a 3DSG representation that takes into account also functional interactive elements (FIEs) of the object through intra-object relationships. The primary challenge lies in the scarcity of data that extends beyond instance-level detection and the inherent difficulty of capturing detailed object features using robotic sensors. We utilize the SceneFun3D dataset to train both 2D and 3D models to extract information about FIE in the scene. We propose two solutions: one using only RGB-D images to generate the 3DSG, and another incorporating the 3D point cloud (PCD) of the scene. Our experiments demonstrate that our approach achieves functional element segmentation comparable to state-of-the-art 3D models and that our augmentation enables task-driven affordance grounding with higher accuracy than the current solutions.

1. Introduction

3D scene graphs (3DSGs) [1, 14] are hierarchical structures that capture a scene's geometry and semantics, where nodes represent objects or spaces, and edges define their relationships. They enable agents to understand and reason about their environment, achieving unprecedented levels of open-world 3D scene understanding [18, 31]. Most 3DSGs focus on object-level granularity, but for tasks like operating household items (e.g., fridges, thermostats), a finer representation is needed. This should include objects and functional interactive elements (FIEs) like knobs and buttons, along with their affordances like pulling or turning. In other words, 3DSGs must account for both inter- and intra-object relationships, a largely unexplored area in robotics and the focus of this paper (see Fig. 1). A major challenge in modeling intra-object relationships is accurately detecting FIEs, which are small, sparsely represented in sensory data, and often missing in public datasets, leading to poor detector performance. To address this, we leverage SceneFun3D [5], a large-scale dataset with sensory data and FIE annotations.

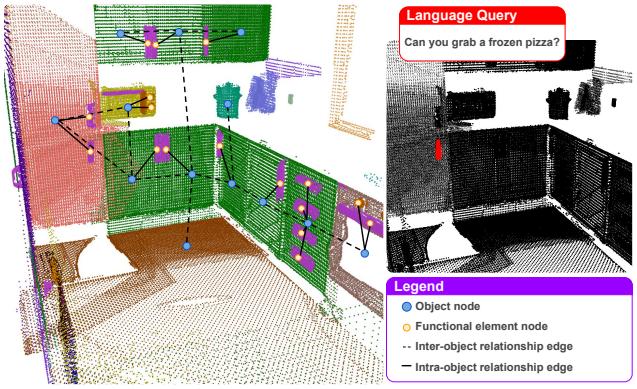

Figure 1. An example of a generated 3D scene graph and its application. The model represents both object and functional element nodes linked through intra- and inter-object relationships.

We generate data from this dataset and use the trained models to extend 3D scene graphs by incorporating intra-object relationships. Specifically, our contributions are:

- A method to detect FIEs in 2D and 3D, predict their affordances, and assign contextualized descriptions.
- A framework, FunGraph, for extending 3D scene graphs to represent FIEs and their affordances.
- A quantitative and qualitative evaluation of how this structure supports FIE segmentation and task-driven affordance grounding.

2. Related Work

3DSG Generation and Prediction. 3DSGs [1, 14] are spatial data representations in the form of hierarchical graphs, with nodes representing different parts of a scene, such as buildings, rooms, and, at the most granular level, objects. These nodes are connected through relationships, such as spatial ones (e.g., "A is next to B"). Different

approaches have explored building 3DSGs from observations, either through incremental creation from raw sensor inputs [9, 18, 19, 31] or post-processing of existing 3D scans [1, 16, 17, 29, 30, 32]. Only CLIO [19] mentions a need for different granularity. However, they only consider object parts and do not group them into objects. They rely on point-grid-prompted SAM [15] to propose object segments, which is not reliably capable of detecting FIEs.

Alternative Queryable Scene Representations. Implicit representations like NeRFs [21] are known for their high rendering accuracy making them ideal for small FIEs. However, semantic NeRF variants [7, 13, 27, 36] and Gaussian Splatting [24] struggle with small objects and fine details, as they rely on pixel-aligned semantic features like DINO [3] or CLIP [25], which are insufficient for functional elements.

Search3D [28] aims to create a hierarchical, open-vocabulary 3D scene representation for finer-grained entities. However, its purely geometric segmentation fails to capture inter-relationships between entities, making queries like "turn off the light *above* the bin" impossible to answer.

Dataset & Resources. The PartNet dataset [22] consists of dense annotations on 3D CAD models, and datasets like 3D AffordanceNet [6] and PartAfford [34] have been built upon it, focusing on Gibsonian affordances [8], which characterize how humans interact with objects and environments. Their focus is primarily on predicting the affordances of already isolated objects rather than on our investigated problem of identifying functional elements in a larger scene. MultiScan [20] takes a step toward highlighting movable object parts in a room scan; however, it does not provide accurate annotations for functional interactive elements.

The recent SceneFun3D [5] was the first dataset annotating the functional elements themselves in real room-scale scenes. Based on ARKitScenes [2], they select nine Gibsonian-inspired affordance labels to represent interactions with common elements in indoor environments (e.g., *Rotate*, *Hook Pull*, etc.) and annotate the 3D point cloud directly. To the best of our knowledge, no 2D resources with annotated functionally interactive elements currently exist.

3. Method

3.1. Problem Formulation

Our work aims to extend the classical pipeline of 3DSG generation for indoor environments by incorporating intra-object relationships between scene objects and their 3D FIE. For example, we want a cabinet to have a direct relationship with its knobs, enhancing the 3DSG with information about the object's possible interactions. The input to the proposed method, FunGraph, consists of a series of RGB-D observations, $\mathcal{I} = \{I_i\}_{i=1\ldots N}$, and corresponding camera poses, $\mathcal{P} = \{\boldsymbol{P}_i\}_{i=1\ldots N}$. Each image I_i is captured from pose $\boldsymbol{P}_i$ using camera parameters $\boldsymbol{K}_i$. In our variant FunGraph+PTv3, we also take the scene's PCD as input.

The output of the proposed method is a 3D scene graph, specifically a hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. That is, the set of nodes $\mathcal{V}$ can be partitioned into l layers ($\mathcal{V} = \cup_{i=1}^{l} \mathcal{V}_i$). Indeed, by design, introducing intra-object relationships does not interfere with the hierarchical properties of the structure: *single parent*, *locality*, and *disjoint children* [10], which involve nodes $\mathcal{V}$ (the collection of objects of the 3D scene) and edges $\mathcal{E}$ (the *spatial* relationships between objects) of our graph. In the following sections, we outline the process of generating data from SceneFun3D [5] to train models for detecting FIE from 2D and 3D data.

3.2. Generation of 2D Data

For each scene *PCD* in the dataset, along with 3D segment FIE annotations $\mathcal{A} = \{\boldsymbol{A}_j\}_{j=1\ldots M} \subset PCD$, and RGB-D observations $\mathcal{I}$ captured from the scene, the goal is to generate 2D annotated images with bounding boxes for performing 2D object detection of FIE. For each image I_i, each annotation $\boldsymbol{A}_j$ is projected on the 2D image plane as $\boldsymbol{a}_j = (\boldsymbol{x}_j \ \boldsymbol{y}_j \ \boldsymbol{z}_j)^T = \boldsymbol{K}_i \boldsymbol{P}_i^{-1} \boldsymbol{A}_j$ Then, we mask out all points from $\boldsymbol{a}_j$ that are behind ($z_j < 0$), or outside of the current camera image. We also remove all points where the depth $\boldsymbol{d}_i$ in I_i differs from the projected depth $\boldsymbol{z}_j$ by more than a certain threshold θ_{depth}. Finally, if the annotation projected onto the image has a bounding box area larger than a certain threshold θ_{area} and the ratio of pixel points to total points in $\boldsymbol{A}_j$ exceeds a threshold θ_{points}, it is kept; otherwise, it is discarded. At the end, each image I_i retains projected annotations, whose bounding boxes form our 2D FIEs dataset.

3.3. Generation of 3D Data

Our goal is to generate groups of objects with associated FIEs to train a 3D model for segmenting FIEs from an input object PCD. Since SceneFun3D provides FIE annotations only for the scene PCD, we combine these with object annotations from the same scenes in ARKit LabelMaker [11].

We first align the ARKit LabelMaker 3D annotation PCD with the SceneFun3D PCD using the transformation provided by SceneFun3D. Then, we transfer annotations by matching points between the two datasets, using a nearest-neighbor (NN) approach to assign each ARKit LabelMaker annotation to the closest θ_p points in the SceneFun3D PCD.

Since ARKit LabelMaker provides only semantic annotations and not instance segmentations, we apply DBSCAN clustering to extract object instances.

Finally, we associate FIEs with specific objects by matching the center of mass of FIE annotations to the closest object instance, ensuring that each FIE is correctly linked to its corresponding object in our 3D FIEs dataset. We do not consider this association if a FIE is more than 20 [cm] away from the closest object.

3.4. FunGraph

Recently, a clear methodology has emerged for generating 3D scene graphs from RGB-D observations [9, 12, 18, 31]. The process consists of three phases as shown in Fig. 2:

- **Detection:** Image instance segmentation of entities.
- **Node creation:** Multi-view merging.
- **Edge creation:** Relationship generation.

Each of these phases requires particular attention when dealing with objects the size of FIEs.

Detection. For each $I_i \in \mathcal{I}$, the q classes and bounding boxes of objects and FIEs are detected independently using YOLO-World [4] for objects and RT-DETR [35] for FIEs.

After filtering out detections with confidence below θ_{bbox} and relative-to-image area ratio lower than θ_{rarea}, we prompt SAM2 [15] with each remaining bounding box. The segments are then reprojected into 3D using the depth information of I_i and de-noised using DBSCAN clustering, resulting in a set of 3D object segments (or PCDs): $\mathcal{O}^{[i]} = \{\mathcal{O}_m^{[i]}, \mathcal{O}_a^{[i]}\} = \{\boldsymbol{o}_{m1}^{[i]}, \ldots, \boldsymbol{o}_{mp}^{[i]}, \boldsymbol{o}_{ap+1}^{[i]}, \ldots, \boldsymbol{o}_{aq}^{[i]}\}$ where the subscript m denotes objects and a denotes FIE.

Each object has a label $c_j^{[i]}$, and semantic features $\boldsymbol{f}_j^{[i]}$ from CLIP-extracted bounding-box crops. We discard FIE that do not overlap with any object's box by at least θ_{rel}.

Context-Based Label Refinement. With the detector trained using the resource in Sec. 3.2, FIEs are classified into a closed set of Gibsonian-inspired affordances based on the SceneFun3D annotations. To ensure the method's open vocabulary ability and obtain more concrete descriptions of the FIEs, we prompt GPT-4o [23] for each group of object and its associated FIEs annotated using their bounding box and current label name. For example, this allows us to obtain the names "Refrigerator Handles" and "Freezer Handle" from an image of a refrigerator, which is associated with two "Hook Pull" FIEs through bounding box overlap.

Node Creation. For each image, each $\boldsymbol{o}_{mj}^{[i]} \in \mathcal{O}_m^{[i]}$ is compared to PCDs of object nodes in the graph. To merge, two PCDs must exceed θ_{geo} according to the geometric similarity score of [31] and the cosine similarity between the semantic features of node and the object should be higher than θ_{sem}. The object segment is merged into the node with the highest combined similarity score.

After each merge, the node's PCD is denoised and downsampled, and semantic features are updated as in [18]. If no similarity check is passed, $\boldsymbol{o}_{mj}^{[i]}$ is merged into a new empty node with a zero-feature vector.

Then, for each $\boldsymbol{o}_{ak}^{[i]}$ associated with $\boldsymbol{o}_{mj}^{[i]}$, merged into node n, we search for the best matching node among the associated FIE nodes of the 3DSG related to n. Only the geometric score is used, with a different threshold θ_{geo2}. The same merging process is followed, with two key differences: redundant points are removed without downsampling, and the FIE segment is merged with all nodes that pass the similarity check. This approach is beneficial because FIE-sized objects are often only partially observed, and these gaps could lead to a final merge later. If no similarity check is passed, the FIE is merged into a newly initialized empty node and associated with node n.

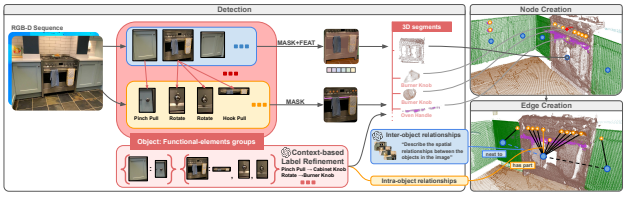

Figure 2. Overview of our 3DSG generation pipeline.

Periodically, nodes are processed in batches, merging object nodes first, followed by related FIE nodes if they pass the similarity checks discussed.

Edge Creation. For *inter-object relationships*, during node creation, we track image sources of object segments that merge into different nodes and then prompt GPT-4o with images where all detected objects are highlighted to extract binary relationships between objects (i, j). The most common relationship is added as an edge between the nodes of i and j, encoding this spatial knowledge.

For *intra-object relationships*, we use the bounding box association described in the detection phase to establish a "has-part" directed relationship, linking the object to its associated FIE. An attribute of this relationship is the affordance label extracted by the 2D detector.

3.5. FunGraph+PTv3

When a high-resolution PCD K of the scene is available, we refine our FunGraph reconstruction by selecting, for each reconstructed object along with its associated FIEs, the θ_p closest points within 2 [cm] of K, thereby obtaining a PCD N. To mitigate noise from the 2D mask reconstruction of FIEs, we apply PTv3 [33] trained on 3D instance segmentation using the resource in Sec. 3.3 to N. The result is a list of FIE PCDs, which we then use to replace those in the nodes of our 3DSG by assigning the attributes and labels of each FunGraph FIE to the closest 3D predicted FIE.

4. Experiments

We validate the accuracy of FIEs segmentation with our 3DSG generation solutions and investigate the usefulness of our 3DSGs in responding to task-driven affordance queries, such as "open the left window above the radiator".

FIEs 3D Segmentation. Among the nine classes of SceneFun3D, we retained only the seven most appropriate for describing FIE, discarding "unplug" and "plug_in" along with their respective annotations. Before validating the 3D, we train RT-DETR on 2D detec-

Affordance	AP	AP_{50}	AP_{25}
FunGraph	5.9	16.0	30.3
FunGraph+PTv3	**8.1**	**21.2**	**37.7**
Mask3D-F [5, 26]	[7.9]	[18.3]	[26.6]

Table 1. Results for 3D FIEs segmentation. Mask3D-F is evaluated on a larger dataset and with all the classes. The full model is not yet available.

tion using an 80/20 train-validation split of the dataset Sec. 3.2, ensuring that train and val images come from different scenes. To benchmark the AP metrics [5] of the 3D reconstruction and segmentation of the FIE 2D detected, we select 10 scenes from our validation dataset: `423070`, `423306`, `423738`, `434892`, `435357`, `435715`, `435724`, `442392`, `464754`, `467330`, and associate the PCD of our FIE node with the eight nearest points within 5 [mm] from the original laser scan, for which the ground truth segmentation has been annotated. Note that we retain all FIE detections, even without object associations, to avoid penalizing scores when parent objects are undetected. For the evaluation of FunGraph+PTv3 we also associated the detected FIE to the closest segmented object in [11] if they have no parent object. Given that the measured performance (Tab. 1) on the different splits of the same datasets are in a similar range, we carefully conclude that FunGraph achieves similar results to SOTA approaches [26] that directly predicts the class for the points in 3D. We also comment that the information provided by FIE's parent objects is a valuable direction for improving task results.

Affordance Grounding To answer task-driven affordance grounding queries, we convert our 3DSG representation into a JSON format, retaining information about each node's ID, 3D center of mass, 3D bounding box extension, label, relationships with the environment, and functionality affordance if it is a FIE.

We then instruct GPT-4o to find in the JSON the ID(s) of the node(s) that solve the query. This highlights the general advantage of 3DSG representations as they can be easily parsed by LLMs. On the same set of scenes, and in the same manner discussed in Sec. 4, we retrieve the closest points to our prediction in the original PCD and compute the 3D PCD intersection over union (IoU) between our prediction and the ground truth answer elements. We count a query as passed if the IoU is at least 25% (AP_{25}).

In Tab. 2, we report per-scene results and compare them to the SOTA ConceptGraphs [18] that can answer unconstrained language queries on the map. As is evident from the numbers, ConceptGraphs does not account for the possibility of providing FIEs as answers to queries. Instead, it returns whole object PCDs, which results in low IoU with the ground truth. Therefore, we further report $AP_{>0}$, where we count queries successfully if there is even a single overlap point between the response and the ground truth. The results, however, show that the return of ConceptGraphs is still less accurate, indicating that including FIEs and object-part relations in the 3DSG improves retrieval localization and generally allows for answering more queries correctly. Interestingly, one of the main advantages of storing segmented FIEs in the 3DSG, is that the scores between 3D FIE segmentation and affordance grounding do not differ much because all the information needed is stored and only needs to be identified.

Scene	#Queries	ConceptGraphs		FunGraph (ours)	
		AP_{25}	$AP_{>0}$	AP_{25}	$AP_{>0}$
423070	8	0.0	25.0	**50.0**	**50.0**
423306	3	0.0	0.0	**33.3**	**66.7**
423738	21	0.0	57.1	**33.3**	**85.7**
434892	5	0.0	**40.0**	**40.0**	**40.0**
435357	10	0.0	50.0	**30.0**	**60.0**
435715	12	0.0	8.3	**33.3**	**75.0**
435724	10	0.0	10.0	**10.0**	**20.0**
442392	8	0.0	25.0	**37.5**	**37.5**
464754	18	0.0	22.2	**27.8**	**44.4**
467330	4	0.0	50.0	**100**	**100**
Total	99	0.0	31.3	**33.3**	**58.6**

Table 2. Results for task-driven affordance grounding. For each method, the percentage of success (IoU at least 25% and $> 0\%$) in task-driven affordance grounding is shown for the scenes in our validation sample. The number of queries for each scene is reported.

5. Conclusions

In this work, we presented FunGraph, the first 3DSG solution that captures intra-object relationships, focusing on FIEs to enable tasks requiring interaction with objects in a scene. Through our experiments, we demonstrated that our finer-grained representation achieves performance comparable to SOTA 3D detectors and highlight the method's superiority to direct point cloud affordance grounding. However, our standard approach is fundamentally rooted in the 2D domain. It does not rely on segmenting a pre-existing high-quality PCD, which makes it also suitable for robotics applications with affordable RGB-D sensing. We are able to detect and store information about the FIEs of objects while extending the general 3DSG generation pipeline and preserving the graph's hierarchical property. In the future, we will augment 3D scene graphs with even more fine-grained representations by introducing intermediate object parts before linking the objects themselves to functional elements. Moreover, we will embed all necessary manipulation details into FIE nodes, enabling robots to interact with objects and ultimately achieve an end-to-end solution.

References

[1] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. *ICCV*, 2019. 1, 2

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. *NeurIPS*, 2021. 2

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *ICCV*, 2021. 2

[4] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. *CVPR*, 2024. 3

[5] Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes. *CVPR*, 2024. 1, 2, 4

[6] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. *CVPR*, 2021. 2

[7] Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, Marc Pollefeys, and Federico Tombari. OpenNeRF: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered Novel Views. *ICLR*, 2024. 2

[8] James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979. 2

[9] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. *RSS*, 2022. 2, 3

[10] Nathan Hughes, Yun Chang, Siyi Hu, Rajat Talak, Rumaisa Abdulhai, Jared Strader, and Luca Carlone. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems. *The Int. J. of Robotics Research*, 2024. 2

[11] Guangda Ji, Silvan Weder, Francis Engelmann, Marc Pollefeys, and Hermann Blum. Arkit labelmaker: A new scale for indoor 3d scene understanding. *CVPR*, 2025. 2, 4

[12] Christina Kassab, Matías Mattamala, Sacha Morin, Martin Büchner, Abhinav Valada, Liam Paull, and Maurice Fallon. The bare necessities: Designing simple, effective open-vocabulary scene graphs. *arXiv preprint arXiv:2412.01539*, 2024. 3

[13] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. *ICCV*, 2023. 2

[14] Ue-Hwan Kim, Jin-Man Park, Taek-jin Song, and Jong-Hwan Kim. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE Trans. on Cybernetics*, 2020. 1

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *ICCV*, 2023. 2, 3

[16] Sebastian Koch, Pedro Hermosilla, Narunas Vaskevicius, Mirco Colosi, and Timo Ropinski. Sgrec3d: Self-supervised 3d scene graph learning via object-level scene reconstruction. *Winter Conference on Applications of Computer Vision (WACV)*, 2023. 2

[17] Sebastian Koch, Narunas Vaskevicius, Mirco Colosi, Pedro Hermosilla, and Timo Ropinski. Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships. *CVPR*, 2024. 2

[18] Alihusein Kuwajerwala, Qiao Gu, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *ICRA*, 2024. 1, 2, 3, 4

[19] Dominic Maggio, Yun Chang, Nathan Hughes, Matthew Trang, Dan Griffith, Carlyn Dougherty, Eric Cristofalo, Lukas Schmid, and Luca Carlone. Clio: Real-time task-driven open-set 3d scene graphs. *RA-L*, 2024. 2

[20] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel X Chang, and Manolis Savva. Multiscan: Scalable rgbd scanning for 3d environments with articulated objects. *NeurIPS*, 2022. 2

[21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020. 2

[22] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. *CVPR*, 2019. 2

[23] OpenAI. GPT-4 technical report. *CoRR*, 2023. 3

[24] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. *CVPR*, 2024. 2

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *ICML*, 139, 2021. 2

[26] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. *ICRA*, 2023. 4

[27] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. *CVPR*, 2023. 2

[28] Ayca Takmaz, Alexandros Delitzas, Robert W. Sumner, Francis Engelmann, Johanna Wald, and Federico Tombari. Search3D: Hierarchical Open-Vocabulary 3D Segmentation. *RA-L*, 2025. 2

[29] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions. *CVPR*, 2020. 2

[30] Ziqin Wang, Bowen Cheng, Lichen Zhao, Dong Xu, Yang Tang, and Lu Sheng. Vl-sat: Visual-linguistic semantics assisted training for 3d semantic scene graph prediction in point cloud. *CVPR*, 2023. 2

[31] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. *RSS*, 2024. 1, 2, 3

[32] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. *CVPR*, 2021. 2

[33] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024. 3

[34] Chao Xu, Yixin Chen, He Wang, Song-Chun Zhu, Yixin Zhu, and Siyuan Huang. Partafford: Part-level affordance discovery from 3d objects. *arXiv preprint arXiv:2202.13519*, 2022. 2

[35] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. *CVPR*, 2024. 3

[36] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. *ICCV*, 2021. 2

A. Implementation details

For the 2D dataset generation, we use $\theta_{depth} = 0.1$ [m], $\theta_{depth} = 800$ [pixels^2], and $\theta_{points} = 0.6$ as thresholds. The thresholds used in the 3DSGs generation pipeline are $\theta_{bbox} = 0.4$, $\theta_{rarea} = 0.7$, $\theta_{rel} = 1$, $\theta_{geo} = 0.5$, $\theta_{sem} = 0.6$, $\theta_{geo2} = 0.6$, $\theta_{num} = 3$ and $\theta_{p} = 8$. All computations are performed on a machine with an NVIDIA 4090 GPU, 64GB RAM + 64GB SWAP, and an AMD Ryzen 9 7950X Processor.