IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Search3D: Hierarchical Open-Vocabulary 3D Segmentation

Ayca Takmaz[1,2 †], Alexandros Delitzas[1], Robert W. Sumner[1], Francis Engelmann[1,2,3*]
Johanna Wald[2*], Federico Tombari[2]

***Abstract*—Open-vocabulary 3D segmentation enables exploration of 3D spaces using free-form text descriptions. Existing methods for open-vocabulary 3D instance segmentation primarily focus on identifying *object*-level instances but struggle with finer-grained scene entities such as *object parts*, or regions described by generic *attributes*. In this work, we introduce Search3D, an approach to construct hierarchical open-vocabulary 3D scene representations, enabling 3D search at multiple levels of granularity: fine-grained object parts, entire objects, or regions described by attributes like materials. Unlike prior methods, Search3D shifts towards a more flexible open-vocabulary 3D search paradigm, moving beyond explicit object-centric queries. For systematic evaluation, we further contribute a scene-scale open-vocabulary 3D part segmentation benchmark based on MultiScan, along with a set of open-vocabulary fine-grained part annotations on ScanNet++. Search3D outperforms baselines in scene-scale open-vocabulary 3D part segmentation, while maintaining strong performance in segmenting 3D objects and materials. Our project page is *search3d-segmentation.github.io*.**

***Index Terms*—Semantic scene understanding, object detection, segmentation and categorization, RGB-D perception**

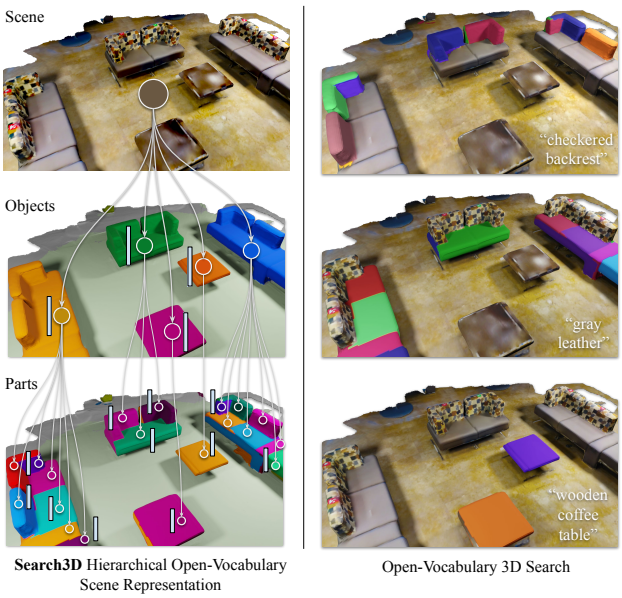


Fig. 1: We propose Search3D, a method for open-vocabulary 3D search at multiple levels of granularity. From posed RGB-D images and reconstructed geometry, we build a hierarchical scene representation with embedded features for objects, and finer-grained parts *(left)*. This enables searching across objects, parts, and attributes matching any given user query *(right)*.

I. Introduction

EXTRACTING semantic meaning from 3D scenes has traditionally relied on identifying a fixed set of pre-defined classes. For this purpose, most 3D segmentation methods [1]–[3] are trained on annotated datasets, limiting their capabilities to closed-set segmentation. While effective for these pre-defined classes, such approaches struggle to generalize to novel classes. However, personal and assistive robotics systems must operate in diverse, unknown environments and handle tasks of varying complexity, requiring the ability to handle unseen classes. This calls for methods that can adapt to new tasks and environments, especially in human-centric spaces, which are inherently complex and composed of fine-grained elements critical to scene interaction. While identifying novel classes is already challenging, many interactive robotics applications [4] require identifying not only objects, but also their finer-grained *components* [5], such as elevator buttons, light switches or chair seats. Further, attributes may vary across parts of an object, *e.g.*, the seat of a chair may be leather while its legs are wooden. Such distinctions are crucial for tasks like cleaning, where robots must handle different materials with appropriate cleaning agents. A purely object-centric understanding fails to provide this level of detail. Ultimately, systems designed for such real-world interactions must be able to identify scene entities based on flexible and user-defined descriptions.

Open-vocabulary 3D segmentation methods [6]–[12] have recently attracted growing interest [13] and demonstrated promising results. These open-vocabulary methods can be grouped based on the underlying scene representation used to aggregate the features: a) instance-level *object-centric* representations such as OpenMask3D [7] or Open3DIS [8], or b) semantics-oriented point-level representations such as OpenScene [6], OpenNeRF [11] or ConceptFusion [9].

Manuscript received: September 26, 2024; Revised January 8, 2025; Accepted January 13, 2025.

This paper was recommended for publication by Editor M. Vincze upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by an SNSF PostDoc.Mobility Fellowship during Francis Engelmann's stay at Stanford University, in part by an Innosuisse grant (48727.1 IP-ICT), and in part by the Swiss National Science Foundation Advanced Grant 216260: "Beyond Frozen Worlds: Capturing Functional 3D Digital Twins from the Real World". Alexandros Delitzas is supported by the Max Planck ETH Center for Learning Systems (CLS).

[1]Ayca Takmaz, Alexandros Delitzas, Robert W. Sumner and Francis Engelmann are with ETH Zurich, Switzerland.

[2]Ayca Takmaz, Francis Engelmann and Federico Tombari are with Google Zurich, Switzerland, and Johanna Wald is with Google Munich, Germany.

[3]Francis Engelmann is with Stanford University, USA.

† Work done at Google Zurich as an intern. * equal supervision.

Correspondence to {ayca.takmaz@inf.ethz.ch}

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Object-centric open-vocabulary 3D segmentation methods typically first extract a set of class-agnostic 3D object instance masks and then compute a feature representation per object, represented in the joint vision-language embedding space of models such as CLIP [14]. These methods are characterized by compact scene representations and are well-suited for directly segmenting object instances that match a given open-ended query. They are however not designed to identify scene entities of varying levels of granularity, *e.g.*, "seat of a chair".

In contrast, other 3D open-vocabulary segmentation methods such as OpenScene [6], OpenNeRF [11] and ConceptFusion [9] build per-point representations that aggregate features for each 3D point, resulting in a more fine-grained understanding of the scene. However, storing these per-point features is memory-intensive, they are inherently noisy, and they lack instance-level information – a critical requirement for real-world applications in which a robot must identify the specific object to interact with among multiple instances [4], [15]. Finally, the least obvious limitation is derived from the way these models compute the point-level features: Although the projected open-vocabulary features are fine-grained at the level of the geometrical scene representation, the intermediate 2D feature backbones these methods use lack the detailed level of semantic meaning and are biased towards an object-level understanding. Consequently, these methods often cannot robustly identify object parts and fine-grained elements, or address queries that describe areas spanning multiple regions of the scene, *e.g.*, material segmentation.

In light of these limitations, we advocate for fine-grained open-vocabulary 3D segmentation to encompass a broader array of scene elements. An ideal open-vocabulary 3D segmentation method should robustly segment not only long-tail objects ("Nerf gun"), but also object parts ("chair backrest") and queries that span multiple regions ("wooden"), while separating instances when necessary. This goes beyond the capabilities of existing methods. Our goal is to develop a method that moves beyond a strictly *object-centric* query paradigm, and to move closer towards more flexible open-vocabulary 3D search capabilities.

Inspired by this vision, we propose Search3D, a hierarchical open-vocabulary 3D instance segmentation method based on a tree-structured hierarchical scene graph. Search3D segments 3D scene entities from an arbitrary textual query, whether targeting object instances (level 1) or object parts (level 2), as shown in Figure 1. To achieve this, we construct a tree representation where nodes represent scenes, objects and part-entities. For each object and part node, we compute open-vocabulary features enabling 3D segmentation across all levels.

To evaluate our method, we introduce a novel evaluation suite for open-vocabulary scene-scale 3D part segmentation based on MultiScan [16]. Additionally, we perform experiments using hierarchical annotations for selected ScanNet++ [17] scenes. Our method outperforms baselines for 3D open-vocabulary segmentation of object instances, as well as object parts (MultiScan, ScanNet++), and is able to segment the scene beyond instances, *e.g.*, material segmentation (3RScan [18]). To summarize our key contributions:

- We propose a hierarchical open-vocabulary 3D segmentation method capable of segmenting both entire objects and their parts given arbitrary textual queries, by aggregating features anchored to different granularity levels in a hierarchical tree structure.
- We introduce a benchmark for *open-vocabulary scene-scale 3D part segmentation* by adapting MultiScan [16] dataset for *open-vocabulary* 3D part segmentation. This data will be released publicly through our project page.
- We contribute open-vocabulary hierarchical part annotations for a selection of ScanNet++ [17] scenes.
- Our approach outperforms baselines on open-vocabulary 3D segmentation of object instances, part-level tasks, and scene-scale tasks such as material segmentation.

II. Related work

A. Open-Vocabulary & Hierarchical 3D Scene Understanding

Existing open-vocabulary 3D scene understanding methods typically focus on either object-level segmentation or point-level semantic segmentation. Object-centric methods like OpenMask3D [7] and Open3DIS [8] are limited to object-level segmentation and cannot handle varying levels of granularity. In contrast, point-level methods such as OpenScene [6] and ConceptFusion [9], along with open-vocabulary implicit methods such as LeRF [19] and OpenNeRF [11] provide sufficiently detailed features for fine-grained segmentation but lack structured, hierarchical representations. Hierarchical querying is partially supported by N2F2 [20], a 3D Gaussian splatting-based method embedding hierarchical features within a neural scene representation. However, it does not enable *explicit* part-*instance* queries. Human3D [21] provides a hierarchical decomposition of humans into body parts but is limited to human-specific segmentation and lacks open-vocabulary capabilities. GARField [22] represents scene elements with an affinity field for multi-granularity grouping but lacks language-guided querying capabilities. Segment3D [23] supports segmenting 3D scenes at varying granularity, akin to SAM for 2D image segmentation. SceneFun3D [5] focuses on segmenting functional, interactable sub-parts in 3D environments given task-oriented language queries. AGILE3D [24] takes an entirely different approach, allowing users to segment arbitrary objects, parts, or elements through click-based interactions. Nevertheless, these works [5], [23], [24] do not provide an explicit hierarchical structure.

Recent works [25]–[28] have explored 3D part segmentation in an open-vocabulary setting. However, these methods are limited to single-object representations, focusing on part segmentation within individual object point clouds rather than handling scene-scale inputs. Additionally, language-guided segmentation methods relying on GLIP [29] require queries to be defined *during* the construction of the representation, necessitating reprocessing with the original images for each new query. In contrast, our method builds an intermediate hierarchical feature representation, enabling open-vocabulary segmentation without prior query knowledge or storing the input images. This allows for efficient querying during inference, making it well suited for real-world applications.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

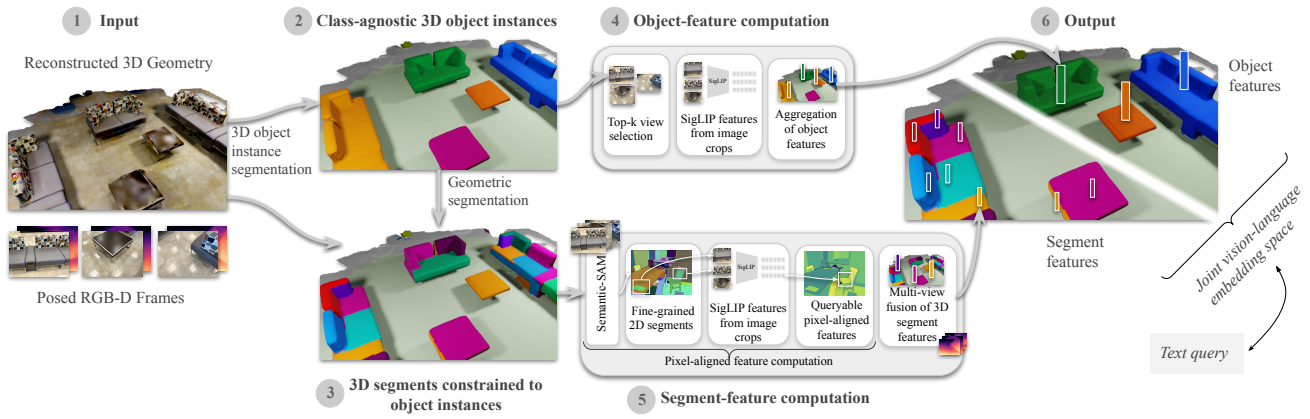


Fig. 2: Search3D overview: ① The inputs of our approach are posed RGB-D images of a 3D indoor scene along with its reconstructed 3D geometry. ② computes class-agnostic 3D instances which are passed to a geometric segmentation method ③, yielding a hierarchical 3D scene representation. In steps ④ and ⑤, feature vectors are obtained for each object and segment. The hierarchical output representation ⑥ is queryable with open-vocabulary features for objects and their corresponding parts enabling search in 3D via arbitrary text queries.

Other methods, such as HOV-SG [30] and CLIO [31] use hierarchical open-vocabulary 3D scene graphs for robotic navigation. However, these approaches operate at higher levels of abstraction—floor, room, region, or object—while our method extends the hierarchy to include finer-grained decomposition of objects into their smaller parts.

B. VLMs and Open-Vocabulary Image Segmentation

Large-scale vision-language models (VLMs) like CLIP [14], SigLIP [32], and SILC [33] provide a joint embedding space for image and text encoders. While effective for tasks like image classification, their global per-image embeddings are not suited for pixel-aligned segmentation tasks. To address this, methods OpenSeg [34], LSeg [35] and others [36]–[40] offer pixel-aligned representations, associating each pixel with an embedding vector. However, their training on full-object masks limits their ability to segment fine-grained entities like object parts. Recent works [41]–[43] tackle open-vocabulary part segmentation but rely on text queries as inputs to the segmentation network, lacking an explicit intermediate feature representation. This makes them unsuitable for building 3D open-vocabulary representations with the desired part-segmentation capabilities.

III. Method

We introduce a novel hierarchical 3D scene representation enabling open-vocabulary segmentation for scene entities at multiple granularities, including objects and their parts. This representation is built upon 3D scenes reconstructed using posed RGB-D image sequences, as shown in Fig. 2 ①, and addresses two key challenges:

1) Representing scene entities at both object and part levels, Fig. 2 ② and ③, discussed in Sec. III-A.
2) Computing open-vocabulary features for the scene representation, Fig. 2 ④ and ⑤ described in Sec. III-B.

A. Hierarchical 3D Scene Representation

To capture both whole objects and their finer components, we construct a hierarchical scene representation as a tree structure (Fig. 1). The root node represents the *scene*, comprising class-agnostic *object* instances, which are further subdivided into smaller object components, *e.g.*, object *parts*.

Our approach starts with an object-level mask proposal module, $\mathcal{F}_{obj}$, built on a transformer-based Mask3D backbone [1] pretrained on ScanNet200 [44]. This module extracts class-agnostic object-level instances from the reconstructed 3D scene geometry. Given the 3D scene $P_{scene} \in \mathbb{R}^{N\times 3}$ where N is the number of points, it outputs M binary instance masks $\mathbf{M} = \mathcal{F}_{obj}(P_{scene}) = \{\mathbf{m}_1^{3D}, \mathbf{m}_2^{3D} ..., \mathbf{m}_M^{3D}\}$. These masks represent the object nodes at the first level of our hierarchical scene representation.

The second stage of our method is the part-level segmentation module, $\mathcal{F}_{seg}$, which refines object instances into more granular segments $\mathbf{S}$. For each object m, we apply an instance-aware geometric over-segmentation technique which computes a set of segments $\mathbf{S}_m$ such that $\mathbf{S}_m = \mathcal{F}_{part}(P_{obj,m}) = \{\mathbf{s}_1^{3D}, \mathbf{s}_2^{3D} ..., \mathbf{s}_S^{3D}\}$, where $P_{obj,m} \in \mathbb{R}^{N_m\times 3}$ represents the 3D points that correspond to the predicted object mask $\mathbf{m}_m^{3D}$. The segmentation module is a 3D adaptation of the graph-cut segmentation algorithm used in [45], originally proposed by [46]. Instead of segmenting the entire scene using this geometric segmentation approach, we use the previously computed object masks $\mathbf{M}$ to segment each instance individually. This ensures that the resulting segments remain within the boundaries of a single object, preserving the hierarchical tree structure. Further, it guarantees that each segment contains points from only one object preventing overlap across masks.

For geometric over-segmentation, we use a segmentation clustering threshold of 0.05, which adaptively controls the merging of regions based on edge weights, and we require at least 100 vertices per segment to prevent over-fragmentation.

So far, we have computed scene entities hierarchically using a geometric representation of 3D object instances and their segments. While these masks capture spatial structures effectively, they lack the semantic information needed for open-vocabulary 3D search. Next, we detail how we enrich these scene entities with open-vocabulary features, enabling flexible 3D segmentation from free-form text queries.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Fig. 3: Pixel-level features. OpenSeg [34], used in OpenScene, has a limited understanding of finer-grained object parts in the scene. We propose to obtain pixel-aligned features by combining Semantic-SAM segments [47] and SigLIP [32], enabling fine-grained localization of concepts such as object parts and materials. Bright yellow means higher similarity to the text query.

B. Bringing Semantic Meaning to 3D Scenes

To enable querying of scene entities across hierarchical levels, both object and part-level features are co-embedded in a shared embedding space using the SigLIP [32] VLM. Building on the hierarchical 3D scene representation from Sec. III-A, semantic features are explicitly computed at two levels: *objects* and *part segments* as illustrated in Fig. 2 (4) and (5).

Object-features (4) are extracted using a method inspired by [7] and [8], leveraging class-agnostic object masks to identify optimal views for semantic feature extraction. These views are selected based on the projection characteristics of the object masks $\mathbf{M}$ initially generated by the object predictor (4). For each 3D object proposal and camera pose, the visibility ratio is computed by projecting the object's points onto the camera image. These visibility scores are then ranked in descending order, and the top-K views ($K = 5$) with the highest visibility ratios and therefore with minimal occlusion are selected. To streamline processing, we subsample the RGB sequence by selecting every 5th frame, following [7].

For each selected view of an object, we first crop the image around a tight 2D bounding box that encapsulates the projected object points. To gradually incorporate more scene context, we perform multi-scale cropping by extending the bounding box by a ratio of $k_{exp} = 0.2$ for L steps, producing L crops per view ($L = 3$). This results in $K \cdot L$ image crops per object. These crops are encoded into image embedding vectors of dimension $D = 1152$ using the SigLIP [32] image encoder (So-400m). The final feature vector (6) is obtained by average pooling embeddings across all multi-view crops.

Segment-features (5), particularly for smaller entities such as object parts, are more challenging to extract. While technically feasible to adapt the object-feature computation – selecting optimal views for image crops – our experiments reveal that this approach yields less informative features for segments, which are typically much smaller in scale. Pixel-aligned VLMs like OpenSeg [34] offer potential for capturing pixel-level details but, as illustrated in Fig. 3, they are biased towards object-level understanding and often fail to represent the fine granularity required for smaller object parts.

To address this challenge, we propose a method to extract pixel-aligned features capable of representing finer-grained scene entities. Using pixel-to-3D point mapping, features are directly aggregated for each predicted 3D part-segment (from (3)) individually, as illustrated in (5) of Fig. 2. First, we apply the automatic mask generator from Semantic-SAM [47] to *all* images in our RGB sequence, specifying the three highest granularity levels to consistently generate 2D segments representing smaller object parts. Following a cropping strategy similar to the one for object-feature computation, we expand the tightest fitting 2D bounding box by a factor of $k_{exp} = 0.1$ to obtain image crops of the fine-grained segments. These 2D segment crops are then passed through the SigLIP [32] image encoder, producing feature vectors of dimension D for each segment. Since our 2D segments are non-overlapping, the computed segment feature vectors are assigned to all pixels within each segment, producing a queryable pixel-aligned feature representation with shape $H \times W \times D$, where H and W represent the height and width of the image and D is the feature dimensionality. Finally, these multi-view features are fused at the 3D segment-level through average pooling. This step leverages the camera poses and scene geometry associated with each RGB frame to align the pixel-level features with the corresponding 3D segments (as extracted in (3)).

Since the initial 3D segmentation is a geometric over-segmentation of each object (3), some parts may be split into multiple segments, even if they belong to the same component (*e.g.*, the front and back parts of a chair's backrest). To address this, we perform a semantically-informed merging of part-segments at the final stage. For each 3D segment, neighboring segments within the same object that exhibit similar features are identified and merged based on two constraints:

1) Proximity: The closest distance between points in the segments is below a threshold thr_{dist}.

2) Feature similarity: The cosine similarity of their feature vectors exceeds a threshold thr_{feat}.

Pairs of segments satisfying both conditions are iteratively merged until no more candidates meet the criteria. We set $thr_{dist} = 0.07$ and $thr_{feat} = 0.13$. Once all candidate segment pairs are merged, the new 3D segment becomes the union of the merged components, and its feature vector is the average of all contributing features. This refinement updates the 3D segments in the hierarchical scene representation, reflecting the semantic merging.

Hierarchical open-vocabulary 3D search. Our hierarchical feature representation enables 3D search across multiple levels of granularity, enhancing open-vocabulary 3D segmentation. When querying for a specific part of an object, we leverage both part- and object-level semantic information encoded in the hierarchical representation. Given an input query such as "seat of a chair", we first encode the query using the SigLIP text encoder to obtain an embedding vector $e_{txt} \in \mathcal{R}^D$. In our hierarchical representation, each object and segment is associated with feature vectors: e_{obj} for the object node and e_{seg} for the part segment node. For any pair of features (e_{obj}, e_{seg}) – where e_{obj} represents the parent object feature and e_{seg} represents the child part feature – we compute the overall similarity with the query text embedding e_{txt} as follows: $sim_{query} = avg(cos\_sim(e_{txt}, e_{obj}), cos\_sim(e_{txt}, e_{seg}))$. Here, cos_sim denotes the cosine similarity between L2-normalized embedding vectors. This average similarity score captures the combined relevance of object-level and segment-level features to the query.

By leveraging this hierarchical approach, we can effectively

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

capture the desired object parts, even when the query targets a specific segment of a larger object. This method enables accurate identification and retrieval of both individual parts and their broader contextual components within 3D scenes. Such flexibility ensures more precise and contextually relevant results for fine-grained 3D segmentation tasks.

IV. Data

In this work, we aim to extend open-vocabulary 3D segmentation capabilities across different granularity levels. To this end, we evaluate our method's ability to perform scene-scale 3D part-level instance segmentation guided by open-vocabulary descriptions. Comprehensive evaluation requires a dataset with scene-scale annotations that ideally captures the object-part hierarchy.

A. Open-Vocabulary 3D Part Segmentation on MultiScan

As mentioned earlier, the MultiScan [16] dataset is the only available resource that includes both scene-scale *object* and *part* instance annotations. It provides essential assets such as RGB-D sequences, calibrated camera data, and 3D surface meshes. Crucially, it includes part- and object-level semantic labels that preserve the scene-object-part hierarchy, making it highly suitable for our work.

The MultiScan dataset was originally annotated with 419 fine-grained categories, later grouped into coarser category sets. For 3D object-level instance segmentation, the original benchmark focuses on 17 common object categories. However, for 3D part-instance segmentation, it only includes 5 part-semantic categories: *static, door, drawer, window, lid*. While these 5 categories is meaningful for MultiScan's original focus on articulated part segmentation, they are insufficient for the *open-vocabulary* scenarios we aim to address. To enable a broader evaluation, we analyzed the existing MultiScan annotations and identified a larger set of object and part categories suitable for open-vocabulary evaluation. The adapted dataset we release, based on existing fine-grained annotations from MultiScan, includes 155 object and 15 part categories.

Another key consideration for open-vocabulary part segmentation is the meaningfulness of part names at the scene scale. Existing part annotations in MultiScan specify only the semantic category of the part, such as "door". However, this can be problematic when performing and evaluating open-vocabulary part segmentation based solely on the part category name. For example, a "desk" object may have a part labeled as "door," which, without additional context, could cause confusion about the intended scene entity. Even humans might incorrectly associate it with the typical meaning of a "door." To mitigate this, we recognize that *(object, part)* pairs are generally more informative for identifying part-level entities in the scene, such as the "seat" of a "chair", or the "door" of the "cabinet". Based on this insight, we extracted a set of 47 joint object-part labels from the MultiScan dataset, consisting of these more informative (object, part) pairs.

B. Fine-grained Part Annotations on ScanNet++

To evaluate our method on fine-grained object and part segmentation, we provide an additional evaluation dataset with annotations on laser scans (Fig. 4). As highlighted in SceneFun3D [5] and ScanNet++ [17], laser scans capture finer 3D geometry details of object parts in indoor environments – details often missing in datasets captured with commodity devices (*e.g.*, iPhone), such as MultiScan [16]. To address this gap, our dataset includes 14 object and 20 part annotations across 8 ScanNet++ [17] scenes, along with open-vocabulary text descriptions. Using the SceneFun3D annotation tool [5], we performed fine-grained semantic annotation on high-resolution point clouds, and extended it to incorporate object-part hierarchy information.

Fig. 4: An example from our hierarchical object and part annotations on a selection of ScanNet++ [17] scenes.

Methods	Segments	AP
OpenScene [6]	Oracle	31.4
OpenMask3D [7]	Oracle	35.7
Search3D (Ours)	Oracle	**49.5** (+13.8)

TABLE I: 3D part feature quality evaluation on the MultiScan [16] dataset using GT part segments. We conduct an oracle experiment using annotated GT part segments to aggregate features for OpenScene [6], OpenMask3D [7] and Search3D (Ours) to measure the quality of the features computed from each method, when isolated from geometric segmentation performance.

V. Experiments

To assess our method's ability to search and segment in 3D via arbitrary open-vocabulary queries, we evaluate it on three diverse tasks: 1) 3D part segmentation (Sec. V-A), 2) 3D object instance segmentation (Sec. V-B), and 3) 3D material segmentation (Sec. V-C). Additionally, we validate our design choices through corresponding ablation studies.

A. 3D Part Segmentation

To evaluate our method's ability to handle queries beyond object-level descriptions, we introduce the task of scene-level 3D open-vocabulary part segmentation. For this analysis, we use our adapted MultiScan [16] scene-level part segmentation data (see Sec. IV-A), and the annotations we provide on the ScanNet++ [17] dataset (see Sec. IV-B). In the part-level instance segmentation experiments, we report the Average Precision metric evaluated at 50% (AP_{50}), 25% (AP_{25}) overlap thresholds, and the average over the range of [0.5 : 0.95 : 0.05] (AP) following established benchmarks [45].

First, we evaluate the quality of our segment features for identifying object parts using an oracle mask experiment, isolating feature quality from the effect of 3D geometric part segmentation quality. In this analysis, all methods use ground-truth (GT) part segments: For OpenScene, we aggregate per-point features for each GT 3D part segment, and for OpenMask3D we aggregate per-mask features for each GT part

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Methods	Aggregation	AP	AP_{50}	AP_{25}
(1) OpenScene [6]	segments	3.2	5.5	13.7
(2) OpenMask3D [7]	objects	3.3	6.1	11.3
(3) OpenMask3D [7]	segments	3.1	6.2	18.2
(4) GARField [22] + Search3D	segments	3.5	8.9	20.5
(5) GARField [22] + Search3D	seg. + hierarchy	3.2	8.4	15.3
(6) Search3D (Ours)	seg. + hierarchy	**7.9** (+4.6)	**14.5** (+8.3)	**31.5** (+13.3)

TABLE II: 3D Part Segmentation on MultiScan [16]. The queries combine object and part descriptions to perform open-vocabulary part retrieval. (1) uses 2D fused OpenSeg [34] feats., and per-point feats. are aggregated over part segments. (2) uses the orig. object-level masks from OpenMask3D. (3) is a stronger baseline adapted from (2) using segment-level aggregation. (4) and (5) use object and part masks from GARField [22] at scales 0.35 and 0.1 respectively and employ Search3D for feature computation using these masks. Search3D (6) uses all components, incl. hierarchical search.

Methods	AP	AP_{50}	AP_{25}
OpenMask3D [7]	5.2	15.0	18.1
Search3D (Ours)	**17.0**	**32.4**	**38.3**

TABLE III: 3D part instance segmentation results on the set of annotations we provide on a selection of ScanNet++ [17] scenes.

Methods	Seg. Aggr.	Merging	Hier. search	AP	AP_{50}	AP_{25}
(1) Ours	✓			4.7	8.2	17.6
(2) Ours	✓	✓		6.6	11.4	23.7
(3) Ours	✓	✓	✓ (max.)	7.5	13.5	28.4
(4) Ours	✓	✓	✓ (avg.)	**7.9** (+3.2)	**14.5** (+6.3)	**31.5** (+13.9)

TABLE IV: Ablation study on Search3D components for 3D part segmentation on MultiScan [16]. *Merging* refers to post-processing and merging 3D segments based on their feature similarities. *Hier. search* refers to measuring the overall similarity between text query and each segment using both object and part features.

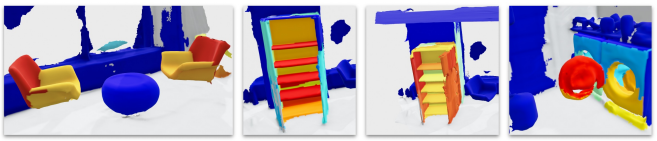
segment. Tab. I shows the results from this oracle experiment. It demonstrates the strong open-vocabulary part-segmentation performance of our segment-level features, with at least +13.8 AP improvement over baseline methods.

After analyzing feature informativeness using *ground-truth part* masks, we evaluate part segmentation performance using *predicted* part masks on the adapted MultiScan dataset. The results, presented in Tab. II, validate Search3D's strong 3D part segmentation ability. Fig. 5 and 6 further show the improved part localization compared to methods like OpenScene, which relies on per-point feature representations. Additionally, in Tab. III, we provide part-segmentation results on our ScanNet++ annotations (Sec. IV-B).

Ablation study. We analyze the impact of semantically informed segment merging and hierarchical search. The results in Tab. IV emphasize the importance of those components. Semantically informed segment merging contributes +3.2 AP_{50}, while leveraging the scene hierarchy for part-level entity search adds +3.1 AP_{50}. Additionally, averaging the object-level and part-level similarity scores yields slightly better results than using the maximum of these scores. Overall, these enhancements lead to a combined improvement of +6.3 AP_{50}.

B. 3D Instance Segmentation

Another key question is whether our method maintains strong open-vocabulary 3D instance segmentation performance



"back of the armchair" *"shelf of the cupboard"* *"sides of the cupboard"* *"door of the washing machine"*

Fig. 5: Heatmaps showing response to text queries of Search3D. Dark red means high similarity and dark blue means low similarity.

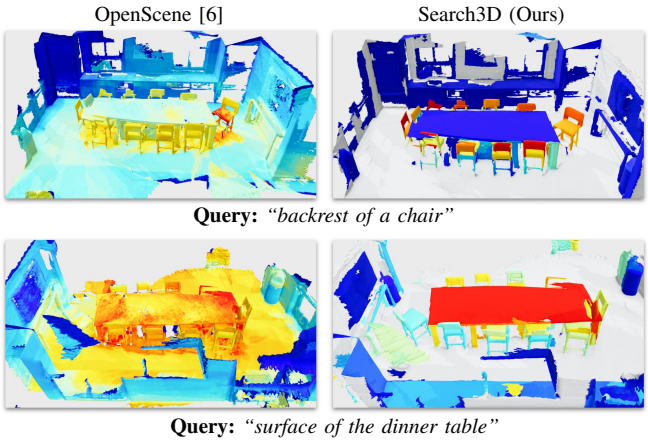


Fig. 6: Similarity heatmaps between text queries and scene features. We compare OpenScene [6] per-point features with the segment features from our method. Dark red means high similarity and dark blue means low similarity. Our method shows highly localized understanding of object parts.

while also being capable of segmenting part-level scene entities. To evaluate this, we compare our method with existing open-vocabulary 3D instance segmentation methods using the standard benchmark on ScanNet200 [44] in Tab. V, and additionally on MultiScan [16] in Tab. VI, using the AP metrics as defined earlier in Sec. V-A. As shown in Tab. V, our method has very strong 3D instance segmentation performance, outperforming other counterparts that rely solely on 3D masks for identifying object-level instances. These results also validate the effectiveness of using SigLIP [32] as a VLM for our method, resulting in strong gains compared

Model	Masks	Img. Feat.	AP	Head (AP)	Common (AP)	Tail (AP)
Closed-vocabulary, full sup.						
Mask3D [1]	Mask3D	–	26.9	39.8	21.7	17.9
Open-vocabulary (using 2D and 3D object mask predictors)						
Open3DIS [8] (2D&3D)	SAM+ISBNet	CLIP	23.7	27.8	21.2	21.8
Open3DIS [8] (2D&3D)	SAM+Mask3D	CLIP	23.7	26.4	22.5	21.9
Open-vocabulary (using only 3D object mask predictors)						
OpenScene [6] (2D F.)	Mask3D	OpenSeg	11.7	13.4	11.6	9.9
OpenMask3D [7]	Mask3D	CLIP	15.4	17.1	14.1	14.9
Open3DIS [8] (3D M.)	ISBNet	CLIP	18.6	24.7	16.9	13.3
Open3DIS [8] (3D M.)	Mask3D	CLIP	18.9	23.9	17.4	15.3
Ours (Search3D)	Mask3D	CLIP	14.3	16.1	13.6	12.9
Ours (Search3D)	Mask3D	SigLIP	**23.0**	**26.3**	**21.2**	**21.4**

TABLE V: 3D Instance Segmentation on ScanNet200 val. Head, Common, and Tail are subsets of ScanNet200 classes, ordered by descending frequency [44]. Our method, while capable of segmenting fine-grained scene entities such as object parts, thanks to its hierarchical representation also preserves strong open-vocabulary 3D object segmentation performance.

Methods	AP	AP_{50}	AP_{25}
GARField [22]	2.4	5.6	9.6
OpenScene [6]	9.0	12.6	16.7
OpenMask3D [7]	10.7	15.7	20.8
Search3D (Ours)	**18.1**	**26.3**	**33.5**

TABLE VI: 3D object instance segmentation scores on MultiScan [16]. We compare with object-level features from our method.

Methods	mIoU	Acc
(1) MinkowskiNet [3] *(fully supervised)*	23.5	30.6
Open-vocabulary, 3D distillation of features		
(2) OpenScene (3D distill) [6]	15.3	26.4
(3) OpenScene (2D/3D ensemble) [6]	20.1	35.6
Open-vocabulary, multi-view fusion		
(4) OpenScene (2D fusion) [6]	18.6	31.9
(5) Search3D (Ours)	**20.2**	**38.4**

TABLE VII: 3D material segmentation scores on 3RScan [18] using object-level material annotations from 3DSSG [48]. To assess the capabilities of our method on open-vocabulary segmentation with a focus on concepts other than object or part semantic categories, we present material segmentation results.

to using CLIP [14] to compute open-vocabulary features. Additionally, while GARField [22] provides a flexible grouping mechanism for segmenting 3D scenes at arbitrary granularities, the lack of an explicit objectness prior leads to inconsistent object segmentation. In contrast, our mask module can identify objects, which enables strong performance for both instance segmentation and part segmentation through the hierarchical search mechanism.

C. 3D Material Segmentation

Next, we perform an analysis on 3D material segmentation task using the object-level material annotations from the 3RScan dataset [18]. We use Intersection-over-Union (mIoU) and mean accuracy (Acc) to evaluate material class predictions obtained using query-similarity based assignments similar to the instance segmentation task. The results in Tab. VII highlight our method's ability to go beyond object semantics.

D. Runtime Analysis

In Tab. VIII, we present a runtime analysis of our method. The construction of the open-vocabulary hierarchical representation ①-⑤ is performed offline. Once this representation is built, inference ⑥, *i.e.*, 3D search based on user input queries can be performed at around 1-2 FPS.

E. Discussion and Limitations

One limitation of our work is the reliance on a simple geometrical over-segmentation method for identifying object *parts*. This is evident from the comparison between Tab. I and Tab. II, where oracle mask experiment yields much higher AP scores than those with predicted part masks, indicating room for improvement in 3D part mask quality.

One might reasonably suggest fusing 2D Semantic-SAM masks instead of obtaining 3D segments directly. While a few

Method component	Runtime	Proportional
② ***3D Object Instance Segmentation***	(per-scene avg.)	
Forward-pass of 3D instance seg. model	0.55 s	-
Post-processing & I/O	18.43 s	$T \propto M$
Total (per-scene)	18.98 s	-
③ ***Geometric Segmentation for Part Segmentation***		
Normal-based geometric segmentation	4.33 s	-
Hierarchical tree formation & I/O cost	17.52 s	$T \propto (M \cdot S)$
Total (per-scene)	21.85 s	-
④ ***Object-Feature Computation***	(per-scene avg.)	
Top-k view selection	1.51 s	$T \propto (n_{\text{frames}} \cdot M)$
Pre-computation of point projections	32.00 s	$T \propto (n_{\text{frames}} \cdot M)$
Multi-level image crops	2.46 s	$T \propto M$
SigLIP features from image crops	215.62 s	$T \propto M$
Aggregation of object-features	3 ms	-
I/O overhead	15.76 s	-
Total (per-scene)	($\sim$ 4-5 min)	-
⑤ ***Segment-Feature Computation***	(per-frame avg.)	
Fine-grained 2D segments	1.99 s	$T \propto n_{\text{frames}}$
Pixel-aligned feature computation	5.72 s	$T \propto n_{\text{frames}}$
Multi-view fusion of segment-features	0.04 s	$T \propto S$
Total (per-scene)	($\sim$ 10-15 min)	(for 75-150 frames)
⑥ ***Inference***		
New text query / vocab. embedding	0.61 s	-
Search in 3D (similarity computation)	1.57 ms	-

TABLE VIII: Runtime and Computational Complexity. Reported times are averaged over MultiScan test scenes. The rightmost column shows whether there is a direct proportionality relationship between the total time per *scene*, vs. other parameters such as the total number of predicted object masks (M), total number of predicted part-segments (S), and RGB frames in the image sequence n_{frames}.

methods such as SAM3D [49] proposed to perform multi-view fusion of 2D masks from SAM [50] to obtain segments in 3D, and presented promising results for *object-level* 3D instance segmentation, our empirical analysis has shown that such methods struggle with fusing inconsistent and small *part-level* masks from multiple views. We repeatedly observed that the multi-view fusion of high-granularity Semantic-SAM [47] masks directly in 3D yields noisy segments, and concluded that using a geometrical over-segmentation method is more effective for part segmentation. Nevertheless, there are limitations to the geometrical segmentation method we employ for *part* segmentation, as it relies on surface normals. When multiple part segments share similar surface normals, such as drawers of a wardrobe, this approach struggles to distinguish these scene entities from each other. Additionally, our hierarchical segmentation assumes a reasonably well-reconstructed scene. Highly incomplete point clouds or severe reconstruction noise can degrade segmentation quality, though our semantic merging step helps mitigate over-segmentation issues.

Lastly, our approach is limited to two explicit granularity levels (objects and parts), reflecting the lack of evaluation benchmarks for finer-grained segmentation. We hope future work will address these challenges to enable more granular and flexible hierarchical segmentation.

VI. Conclusion

We presented a novel open-vocabulary 3D scene understanding approach that extends beyond traditional object-centric queries to enable fine-grained search in 3D environments. By

introducing a hierarchical scene representation, our method segments not only object instances but also object parts and generic attributes. We validated our approach through extensive experiments and introduced new benchmarks for scene-scale open-vocabulary 3D part segmentation, achieving significant improvements over existing methods. We hope this work inspires future advancements in 3D open-vocabulary segmentation, enabling more flexible handling of scene entities across different levels of granularity.

REFERENCES

[1] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3D: Mask Transformer for 3D Semantic Instance Segmentation," in *ICRA*, 2023.

[2] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner, "3D-MPA: Multi Proposal Aggregation for 3D Semantic Instance Segmentation," in *CVPR*, 2020.

[3] C. Choy, J. Gwak, and S. Savarese, "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks," in *CVPR*, 2019.

[4] O. Lemke, Z. Bauer, R. Zurbrügg, M. Pollefeys, F. Engelmann, and H. Blum, "Spot-Compose: A Framework for Open-Vocabulary Object Retrieval and Drawer Manipulation in Point Clouds," in *ICRAW*, 2024.

[5] A. Delitzas, A. Takmaz, F. Tombari, R. Sumner, M. Pollefeys, and F. Engelmann, "SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes," in *CVPR*, 2024.

[6] S. Peng, K. Genova, C. M. Jiang, A. Tagliasacchi, M. Pollefeys, and T. Funkhouser, "OpenScene: 3D Scene Understanding with Open Vocabularies," in *CVPR*, 2023.

[7] A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann, "OpenMask3D: Open-Vocabulary 3D Instance Segmentation," in *NeurIPS*, 2023.

[8] P. D. A. Nguyen, T. D. Ngo, E. Kalogerakis, C. Gan, A. Tran, C. Pham, and K. Nguyen, "Open3DIS: Open-Vocabulary 3D Instance Segmentation with 2D Mask Guidance," in *CVPR*, 2024.

[9] K. Jatavallabhula, A. Kuwajerwala, *et al.*, "ConceptFusion: Open-Set Multimodal 3D Mapping," in *RSS*, 2023.

[10] Q. Gu, A. Kuwajerwala, S. Morin, K. Jatavallabhula, B. Sen, *et al.*, "ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning," in *ICRA*, 2023.

[11] F. Engelmann, F. Manhardt, M. Niemeyer, K. Tateno, M. Pollefeys, and F. Tombari, "OpenNeRF: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered Novel Views," in *ICLR*, 2024.

[12] V. Bieri, M. Zamboni, N. S. Blumer, Q. Chen, and F. Engelmann, "OpenCity3D: 3D Urban Scene Understanding with Vision-Language Models," in *WACV*, 2025.

[13] F. Engelmann, A. Takmaz, J. Schult, E. Fedele, J. Wald, S. Peng, X. Wang, O. Litany, S. Tang, F. Tombari, M. Pollefeys, L. Guibas, *et al.*, "OpenSUN3D: 1st Workshop Challenge on Open-Vocabulary 3D Scene Understanding," *arXiv preprint arXiv:2402.15321*, 2024.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *ICML*, 2021.

[15] R. Zurbrügg, Y. Liu, F. Engelmann, S. Kumar, M. Hutter, V. Patil, and F. Yu, "ICGNet: A Unified Approach for Instance-Centric Grasping," in *ICRA*, 2024.

[16] Y. Mao, Y. Zhang, H. Jiang, A. X. Chang, and M. Savva, "MultiScan: Scalable RGBD Scanning for 3D Environments with Articulated Objects," in *NeurIPS*, 2022.

[17] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes," in *ICCV*, 2023.

[18] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Nießner, "RIO: 3D Object Instance Re-Localization in Changing Indoor Environments," in *ICCV*, 2019.

[19] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "LERF: Language Embedded Radiance Fields," in *ICCV*, 2023.

[20] Y. Bhalgat, I. Laina, J. F. Henriques, A. Zisserman, and A. Vedaldi, "N2F2: Hierarchical Scene Understanding with Nested Neural Feature Fields," in *ECCV*, 2024.

[21] A. Takmaz, J. Schult, I. Kaftan, M. Akçay, B. Leibe, R. Sumner, F. Engelmann, and S. Tang, "3D Segmentation of Humans in Point Clouds with Synthetic Data," in *ICCV*, 2023.

[22] C. M. Kim, M. Wu, J. Kerr, M. Tancik, K. Goldberg, and A. Kanazawa, "GARField: Group Anything with Radiance Fields," in *CVPR*, 2024.

[23] R. Huang, S. Peng, A. Takmaz, F. Tombari, M. Pollefeys, S. Song, G. Huang, and F. Engelmann, "Segment3D: Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels," in *ECCV*, 2024.

[24] Y. Yue, S. Mahadevan, J. Schult, F. Engelmann, B. Leibe, K. Schindler, and T. Kontogianni, "AGILE3D: Attention Guided Interactive Multi-object 3D Segmentation," in *ICLR*, 2024.

[25] T. Chen, C. Yu, J. Li, Z. Jianqi, D. Ji, J. Ye, and J. Liu, "Reasoning3D - Grounding and Reasoning in 3D: Fine-Grained Zero-Shot Open-Vocabulary 3D Reasoning Part Segmentation via Large Vision-Language Models," *arXiv:2405.19326*, 2024.

[26] M. Liu, Y. Zhu, H. Cai, S. Han, Z. Ling, F. Porikli, and H. Su, "Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models," in *CVPR*, 2023.

[27] A. Abdelreheem, I. Skorokhodov, M. Ovsjanikov, and P. Wonka, "SATR: Zero-Shot Semantic Segmentation of 3D Shapes," in *ICCV*, 2023.

[28] Z. Ma, Y. Yue, and G. Gkioxari, "Find Any Part in 3D," *arXiv preprint arXiv:2411.13550*, 2024.

[29] L. H. Li*, P. Zhang*, H. Zhang*, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao, "Grounded Language-Image Pre-training," in *CVPR*, 2022.

[30] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation," *Robotics: Science and Systems*, 2024.

[31] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time Task-Driven Open-Set 3D Scene Graphs," *Robotics and Automation Letters*, 2024.

[32] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid Loss for Language Image Pre-Training," in *ICCV*, 2023.

[33] M. F. Naeem, Y. Xian, X. Zhai, L. Hoyer, L. V. Gool, and F. Tombari, "SILC: Improving Vision Language Pretraining with Self-Distillation," *ArXiv*, vol. abs/2310.13355, 2023.

[34] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, "Scaling Open-Vocabulary Image Segmentation with Image-Level Labels," in *ECCV*, 2021.

[35] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven Semantic Segmentation," in *ICLR*, 2022.

[36] 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 *CVPR*, 2023.

[37] C. Zhou, C. C. Loy, and B. Dai, "Extract Free Dense Labels from CLIP," in *ECCV*, 2022.

[38] X. Zou, Z.-Y. Dou, J. Yang, Z. Gan, L. Li, C. Li, X. Dai, H. Behl, J. Wang, L. Yuan, N. Peng, L. Wang, Y. J. Lee, and J. Gao, "Generalized Decoding for Pixel, Image and Language," in *CVPR*, 2023.

[39] S. Cho, H. Shin, S. Hong, S. An, S. Lee, A. Arnab, P. H. Seo, and S. Kim, "CAT-Seg: Cost Aggregation for Open-Vocabulary Semantic Segmentation," in *CVPR*, 2023.

[40] T. Lüddecke and A. Ecker, "Image Segmentation Using Text and Image Prompts," in *CVPR*, 2022.

[41] X. Wang, S. Li, K. Kallidromitis, Y. Kato, K. Kozuka, and T. Darrell, "Hierarchical Open-vocabulary Universal Image Segmentation," *arXiv 2307.00764*, 2023.

[42] M. Wei, X. Yue, W. Zhang, S. Kong, X. Liu, and J. Pang, "OV-PARTS: Towards Open-Vocabulary Part Segmentation," in *NeurIPS*, 2023.

[43] P. Sun, S. Chen, C. Zhu, F. Xiao, P. Luo, S. Xie, and Z. Yan, "Going Denser with Open-Vocabulary Part Segmentation," in *ICCV*, 2023.

[44] D. Rozenberszki, O. Litany, and A. Dai, "Language-Grounded Indoor 3D Semantic Segmentation in the Wild," in *ECCV*, 2022.

[45] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes," in *CVPR*, 2017.

[46] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-based Image Segmentation," in *IJCV*, 2004.

[47] F. Li, H. Zhang, P. Sun, X. Zou, S. Liu, J. Yang, C. Li, L. Zhang, and J. Gao, "Semantic-SAM: Segment and Recognize Anything at Any Granularity," *arXiv preprint arXiv:2307.04767*, 2023.

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

[49] Y. Yang, X. Wu, T. He, H. Zhao, and X. Liu, "SAM3D: Segment Anything in 3D Scenes," *arxiv*, vol. abs/2306.03908, 2023.

[50] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *ICCV*, 2023.