Search3D: Hierarchical Open-Vocabulary 3D Segmentation

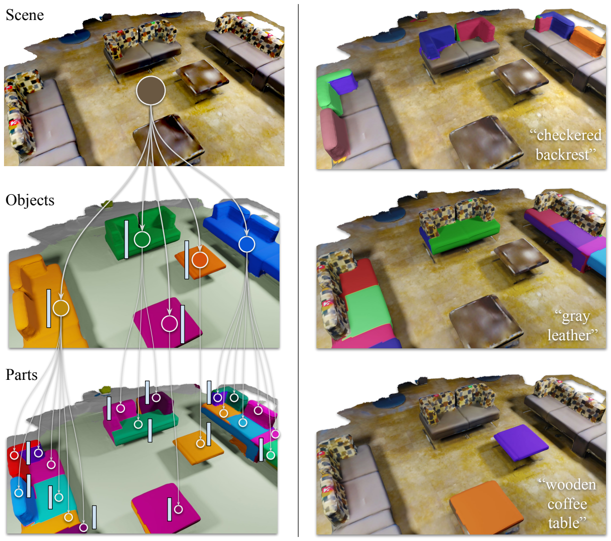
Ayca Takmaz[1,2 †], Alexandros Delitzas[1], Robert W. Sumner[1], Francis Engelmann[1,2],
Johanna Wald[2], Federico Tombari[2]

***Abstract*— Open-vocabulary 3D segmentation enables the exploration of 3D spaces using free-form text descriptions. Existing methods for open-vocabulary 3D instance segmentation primarily focus on identifying *object*-level instances in a scene. However, they face challenges when it comes to understanding more fine-grained scene entities such as *object parts*, or regions described by generic *attributes*. In this work, we introduce Search3D, an approach that builds a hierarchical open-vocabulary 3D scene representation, enabling the search for entities at varying levels of granularity: fine-grained object parts, entire objects, or regions described by attributes like materials. Our method aims to expand the capabilities of open-vocabulary instance-level 3D segmentation by shifting towards a more flexible open-vocabulary 3D search setting less anchored to explicit object-centric queries, compared to prior work. To ensure a systematic evaluation, we also 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++. We verify the effectiveness of Search3D across several tasks, demonstrating that our approach outperforms baselines in scene-scale open-vocabulary 3D part segmentation, while maintaining strong performance in segmenting 3D objects and materials.**

I. Introduction

Extracting semantic meaning from 3D scenes has been traditionally performed by identifying a pre-defined set of classes. For this purpose, most 3D segmentation methods [1]–[3] are trained on annotated datasets, resulting in closed set segmentation capabilities. While these approaches work well on these pre-defined categories, they do not generalize to novel classes. However, personal and assistive robotics systems require the ability to operate in unknown environments and handle tasks of varying complexity.

This necessitates methods to adapt to new tasks and environments, especially in human-centric spaces, which are inherently complex and consist of fine-grained elements defining the interaction landscape of the scene. While identifying novel classes is already a challenging task, for many interactive robotics applications we need to identify not only objects, but also their finer-grained *components* [4], such as elevator buttons, cabinet handles or chair seats. Furthermore, attributes often differ across parts of an object, *e.g.*, the seat of a chair might be made from leather while its legs are wooden. For example, a robot that cleans different materials with selected cleaning agents needs to be able to differentiate the respective materials. A purely object-centric understanding often cannot provide the required information. Ultimately, systems for such interactive applications in the real world must be capable of identifying and segmenting scene entities based on flexible and user-defined descriptions.



Search3D Hierarchical Open-Vocabulary Scene Representation

Open-Vocabulary 3D Search

Fig. 1: Search3D. We propose open-vocabulary 3D search at multiple levels of granularity. Starting from posed RGB-D images and the corresponding reconstructed geometry, we build a hierarchical scene representation with embedded features for the scene objects, and finer-grained parts (left). This allows us to search not only for objects, but also parts or attributes matching a given query (right).

Open-vocabulary 3D segmentation methods [5]–[9] have recently attracted growing interest [10] and demonstrated very 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 [6] and Open3DIS [7], or b) semantics-oriented point-level representations such as OpenScene [5] and ConceptFusion [8].

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 [11]. 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 [5] or ConceptFusion [8] build a per-point representation and aggregate per-point features to obtain a finer feature granularity. Such models have a

[1] Ayca Takmaz, Alexandros Delitzas, Robert W. Sumner and Francis Engelmann are with ETH Zurich (Rämistrasse 101, 8092 Zurich, Switzerland).

[2] Francis Engelmann and Federico Tombari are with Google Zurich (Brandschenkestrasse 110, 8002 Zürich, Switzerland), and Johanna Wald is with Google Munich (Erika-Mann-Straße 33, 80636 Munich, Germany).

[†] Work done during the time Ayca Takmaz spent at Google Zurich as an intern. Correspondence to ayca.takmaz@inf.ethz.ch

few limitations: Storing per-point features is costly; these features are inherently noisy and lack instance information, which requires additional post-processing steps to extract individual 3D masks from the cluttered feature representation. 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 mask queries that describe areas spanning multiple regions of the scene, *e.g.*, material segmentation.

In light of these limitations, we argue that fine-grained open-vocabulary 3D segmentation should evolve to encompass a wider array of scene elements. Ideally, an open-vocabulary 3D segmentation method should be able to robustly segment not only long-tail objects ("Nerf gun"), but also object parts ("chair backrest") and queries spanning multiple regions ("wooden"), while separating instances when necessary. This goes beyond the capabilities of existing methods. We aim to develop a method less anchored on the explicit level of *object-centric* queries, and move closer towards a setting with more flexible open-vocabulary 3D search capabilities.

Inspired by this vision, we propose Search3D, a hierarchical open-vocabulary 3D instance segmentation method that uses an underlying hierarchical scene graph representation in the form of a tree. Our method can segment the 3D scene entity that corresponds to an arbitrary textual query which could describe either an object instance (level 1) or parts of an object (level 2), see Figure 1. For this purpose, we first build a tree representation that consists of a scene, object and part-entity layers. For each object and part node in our representation, we compute open-vocabulary features which enable 3D segmentation at each level.

To evaluate our method, we introduce a novel evaluation suite for open-vocabulary scene-scale 3D part segmentation based on MultiScan [12]. Additionally, we perform experiments using hierarchical annotations for selected ScanNet++ [13] 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 [14]). 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 [12] dataset for *open-vocabulary* 3D part segmentation.
- We contribute open-vocabulary hierarchical part annotations for a selection of ScanNet++ [13] 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 3D scene understanding*

Existing open-vocabulary 3D scene understanding methods typically focus on either object-level segmentation or point-level semantic segmentation. For instance, methods such as OpenMask3D [6] and Open3DIS [7] are object-centric and hence unable to segment scene entities at varying granularities. In contrast, point-level methods such as OpenScene [5] and ConceptFusion [8], along with open-vocabulary implicit methods such as LeRF [15] and OpenNeRF [16] offer open-vocabulary features that are insufficiently detailed to segment fine-grained scene elements. Hierarchical open-vocabulary querying is supported by N2F2 [17], a recent 3D Gaussian splatting-based method which embeds hierarchical features within the neural scene representation. However, these features do not enable *explicit* querying at the part-*instance* level. GARField [18] builds a representation with an affinity field allowing to group scene elements at various granularities, but it does not provide any language-guided querying abilities.

A few recent works [19]–[22] have addressed 3D part segmentation in an open-vocabulary setting. However, these methods do not tackle 3D open-vocabulary part segmentation at the *scene* level. Instead, they focus on segmenting parts within single-object representations, *e.g.* they only support open-vocabulary segmentation within object point clouds directly, rather than taking a scene scale input. Additionally, approaches which benefit from language-guided segmentation methods such as GLIP [23] require queries to be defined *while* building the representation to be able to perform open-vocabulary segmentation. This means that for each new query, these methods need to be run again using the set of available images. Our method differs from those by building an intermediate hierarchical feature representation, neither requiring prior knowledge of queries nor storing the input images. It allows for querying the scene during inference, enabling efficient use for real-world applications.

There are also a few methods such as HOV-SG [24] and CLIO [25] employing hierarchical open-vocabulary 3D scene graphs for robotic navigation. However, their methods operate at the floor, room, region or object levels, whereas our approach further breaks down objects into their smaller parts within the hierarchical representation.

B. *Vision-language models and open-vocabulary image segmentation*

Recent success of large-scale model training has resulted in a series of vision-language models (VLMs). Models such as CLIP [11], SigLIP [26], and SILC [27] provide a joint embedding space for their image and text encoders. These image encoders typically take a single image as input and provide a global image embedding. While this enables tasks such as image classification, these representations fall short when localization abilities are required. To address localization based open-vocabulary detection and segmentation tasks, a series of methods [28]–[34] were developed. Models

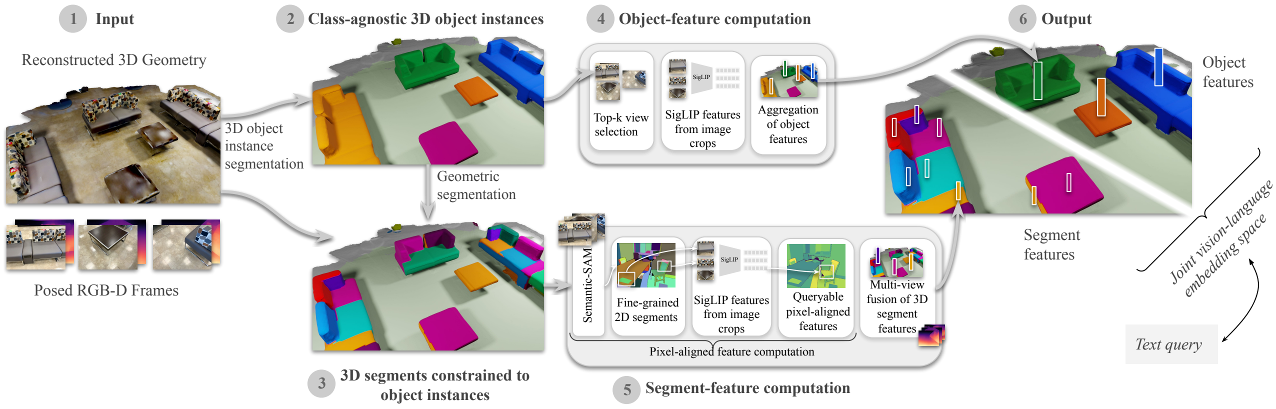

1 Input
Reconstructed 3D Geometry
3D object instance segmentation
Posed RGB-D Frames
2 Class-agnostic 3D object instances
Geometric segmentation
3 3D segments constrained to object instances
4 Object-feature computation
Top-k view selection
SigLIP
SigLIP features from image crops
Aggregation of object features
Semantic-SAM
Fine-grained 2D segments
SigLIP features from image crops
Queryable pixel-aligned features
Multi-view fusion of 3D segment features
Pixel-aligned feature computation
5 Segment-feature computation
6 Output
Object features
Segment features
Joint vision-language embedding space
Text query

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.

such as OpenSeg [29] and LSeg [28] have a pixel-aligned feature representation where each pixel is associated with an embedding vector in the joint vision-language embedding space. As these models are generally trained to ensure a feature alignment with full object masks, they have a limited ability to identify fine-grained scene entities such as object parts. To address this, a few recent methods have addressed open-vocabulary part segmentation [35]–[37]. A critical limitation of these methods is that they lack an explicit intermediate feature representation, and instead they require the text query to be given as an input to the segmentation network. Therefore, these methods are unfortunately not suitable for directly aggregating meaningful features while building a 3D open-vocabulary representation with part-segmentation capabilities we desire.

III. Method

We introduce a novel hierarchical 3D scene representation that enables open-vocabulary segmentation for scene entities at multiple granularities, including objects and their parts. The representation is built upon 3D scenes reconstructed from a sequence of posed RGB-D images, shown in Fig. 2 ①, and requires us to solve the following two challenges:

1) Building a 3D scene representation that accurately captures scene entities at both object and part levels, Fig. 2 ② and ③, discussed in section III-A.
2) Computing open-vocabulary features for the scene representation, Fig. 2 ④ and ⑤ described in section III-B.

A. Hierarchical 3D scene representation

To capture both whole objects and their finer-level components, we construct a hierarchical scene representation. For this purpose, we propose to build a tree structure (Fig. 1) that represents the *scene* as the root node, which consists of class-agnostic *object* instances which are further subdivided into smaller object components, e.g. object *parts*.

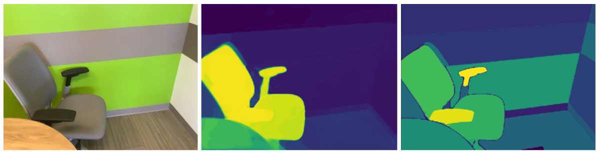
Our approach begins with an object-level mask proposal module, $\mathcal{F}_{obj}$. This module leverages a transformer-based Mask3D backbone [1] pretrained on ScanNet200 [38], and extracts class-agnostic object-level instances from the reconstructed 3D scene geometry. Given the 3D scene $P_{scene} \in \mathbb{R}^{N\times 3}$ where N is the number of points, the module outputs a set of M binary instance masks, *i.e.*, $\mathbf{M} = \mathcal{F}_{obj}(P_{scene}) = \{\mathbf{m}_1^{3D}, \mathbf{m}_2^{3D}..., \mathbf{m}_M^{3D}\}$. 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, $\mathcal{F}_{seg}$. This module first breaks down object instances into more granular segments $\mathbf{S}$ by applying an instance-aware geometric over-segmentation technique which computes a set of segments $\mathbf{S}_m$ for each object 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 [39], originally proposed by [40]. Instead of applying geometric segmentation to the entire scene at once, we leverage the previously computed 3D object instance masks $\mathbf{M}$. By computing the part segments within each instance separately, we ensure that the resulting segments remain within the boundaries of a single object, preserving the hierarchical tree structure. This also implies that each segment contains points from only one object entity and therefore segments do not span across multiple object masks. For geometric over-segmentation, we use a clustering threshold of 0.05 and a minimum number of vertices of 100 per segment.

So far, we have computed the scene entities hierarchically using a geometric representation of the 3D object instances and their constituent segments. While these 3D masks effectively capture the spatial structures, they inherently lack the semantic information required for open-vocabulary 3D search. In the next section, we describe how we enrich these scene entities with open vocabulary features, enabling flexible 3D segmentation based on free-form text queries.

B. Bringing semantic meaning to 3D scenes

To enable querying scene entities across different hierarchical levels within our scene representation, both object and part-



“black armrests of the chair” OpenSeg [29] Search3D (Ours)

Fig. 3: Pixel-level features. OpenSeg [29], 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 [41] and SigLIP [26], enabling fine-grained localization of concepts such as object parts and materials (right). Bright yellow means higher similarity to the text query.

level features are co-embedded in a joint embedding space using the same vision-language model. For this, we build the semantic 3D feature representation using SigLIP [26]. Building on the hierarchical 3D scene representation discussed in Sec. III-A, features are computed explicitly at two levels: *objects* and *part segments* as shown in Fig. 2 ④ and ⑤.

Object-features ④ are obtained using a method similar to [7] and [6] leveraging class-agnostic object masks to identify the optimal views of objects for extracting semantic features. These views are selected based on the projection characteristics of the object masks $\mathbf{M}$ initially provided by the object-predictor ④. Given the 3D object proposals and the set of camera poses, the visibility ratio of each object is determined by projecting the object's points onto the camera image. These scores are then ranked in descending order, allowing us to select the top-K views ($K = 5$) with the highest visibility ratio and therefore lowest levels of occlusion. We also subsample the RGB sequence by processing only 1 out of every 5 frames for view selection, similar to [6].

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, extending the bounding box by a ratio of $k_{exp} = 0.2$ for L steps, obtaining L crops per view, yielding a total of $K \cdot L$ image crops representing the object ($L = 3$). These crops are then encoded into image embedding vectors of dimension $D = 1152$ using the SigLIP [26] image encoder (So-400m). The final feature vector for each object in our output representation ⑥ is obtained by average pooling the multi-view embeddings.

Segment-features ⑤, particularly for smaller scene entities such as object parts, are more challenging to obtain. While it is technically possible to use the approach from object-feature computation – selecting best views for image crops – our experiments show that this approach leads to less informative features for segments, which are typically much smaller in scale. Pixel-aligned VLMs like OpenSeg [29] may seem like a promising alternative due to their ability to capture pixel-level details, but as illustrated in Fig. 3, they remain biased towards an object-level understanding and often fail to capture the fine granularity required to represent smaller object parts.

To address this issue, we propose a method for obtaining pixel-aligned features which have the representative power to also capture finer-grained scene entities. Using pixel-to-3D point mapping, we can directly aggregate these features for each predicted 3D part-segment (from ③) individually. High-level illustration of this process is provided in ⑤ of Fig. 2. As the first step, we apply the automatic mask generator from Semantic-SAM [41] on *all* images in our RGB sequence, explicitly specifying the three highest granularity levels to consistently generate 2D segments representing smaller object parts. Following a cropping strategy similar to that of the object-feature computation pipeline, 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 [26] image encoder, resulting in feature vectors of dimension D for each segment.

Since our 2D segments are non-overlapping, we assign the computed segment feature vector 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 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 our 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. We assess two key constraints to determine whether two segments should be merged: 1) are the segments close enough (*i.e.* is the closest distance between points of the two segments below a threshold thr_{dist}), and 2) are the feature vectors similar enough (*i.e.* is the cosine similarity between two vectors greater than a threshold thr_{feat})? Pairs of segments meeting both conditions are iteratively merged until no more candidates satisfy the criteria. We employ $thr_{dist} = 0.07$ 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 process refines the set of 3D segments, which are then updated in the hierarchical scene representation to reflect the semantic merging.

Hierarchical open-vocabulary 3D search. Our hierarchical feature representation allows us to search at multiple levels of granularity, enhancing the open-vocabulary 3D segmentation capability. When querying for a specific part of an object, we leverage both the part-level and object-level semantic information encoded in our hierarchical representation.

Given an input query such as “seat of a chair”, we first embed the full query text 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 segment feature – we compute the overall similarity with the query text embedding e_{txt} using the following formula: $sim_{query} = avg(cos\_sim(e_{txt}, e_{obj}), cos\_sim(e_{txt}, e_{seg}))$. Here, cos_sim denotes the cosine similarity between L2-normalized embedding vectors. This average similarity score combines the relevance of both the object-level and segment-level features with respect to the query.

By leveraging this hierarchical approach, we can effectively capture the desired object parts even when the query pertains to a specific segment of a larger object. This method enables us to accurately identify and retrieve both individual parts and their broader contextual components within 3D scenes. This flexibility allows for more accurate and contextually relevant results in fine-grained 3D segmentation tasks.

IV. Data

In this work, we aim to extend open-vocabulary 3D segmentation capabilities to different granularity levels. In that direction, we explore our method's ability to perform scene-scale 3D part-level instance segmentation guided by open-vocabulary descriptions. To comprehensively evaluate our method, we need to identify a dataset with scene-scale annotations, ideally capturing the object-part hierarchy.

A. Open-vocabulary 3D part segmentation on MultiScan

As previously mentioned, the MultiScan [12] dataset is the only available resource with both scene-scale *object* and *part* instance annotations. MultiScan provides key assets such as RGB-D sequences, calibrated camera data and corresponding 3D surface meshes. Most importantly, it offers part-level and object-level semantic labels, which maintain the scene-object-part hierarchy, which is crucial for our work.

Originally, MultiScan dataset was 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 features 5 part-semantic categories: *static, door, drawer, window, lid*. While the choice of these 5 categories is quite meaningful for MultiScan's original focus on articulated part segmentation, for the *open-vocabulary* scenarios we would like to address, we need an evaluation suite that covers a much broader variety. To this end, we analyze the MultiScan annotations, and identify a larger set of object and part categories suitable for evaluating open-vocabulary performance. 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 how meaningful part names are at the scene-scale. For instance, existing part annotations only specify the part's semantic category, such as "door". Upon closer inspection, we found this could be problematic when performing and evaluating open-vocabulary part segmentation based solely on the part category name. An example from the dataset is the following: a "desk" object has a part labeled as "door", which, without additional context, could lead to confusion about what scene entity is being referenced. Even humans might mistakenly 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". Following this insight, we extract a set of 47 joint object-part labels from the MultiScan dataset, consisting of these more informative (object, part) pairs.

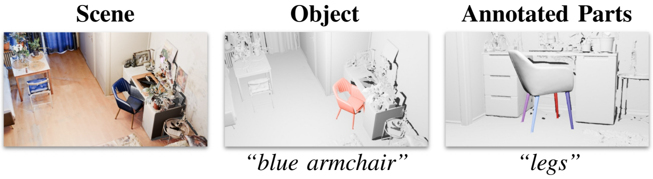


Fig. 4: An example from our hierarchical object and part annotations on a selection of ScanNet++ [13] scenes.

B. Fine-grained part annotations on ScanNet++

To evaluate our method on fine-grained object and part segmentation, we provide an additional evaluation dataset containing annotations on laser scans (Fig. 4). As discussed in [4] and [13], laser scans capture finer 3D geometry details of object parts within indoor environments. These details are often absent in datasets captured with commodity devices (*e.g.*, iPhone), such as MultiScan [12]. To address this, we provide an evaluation dataset with 14 object and 20 part annotations across 8 ScanNet++ [13] scenes along with open-vocabulary text descriptions. To do this, we utilize the SceneFun3D annotation tool [4] which enables the fine-grained semantic annotation of high-resolution point clouds, and we extend it to incorporate object-part hierarchy information.

V. Experiments

To evaluate our method's capability to search and segment in 3D via arbitrary open-vocabulary queries, we evaluate it on diverse tasks including a) 3D part segmentation (Sec. V-A), b) 3D object instance segmentation (Sec. V-B) and c) 3D material segmentation (Sec. V-C). Furthermore, method design choices are supported by respective ablation studies.

A. 3D part segmentation

To evaluate our method's ability to handle queries beyond object-level descriptions, we formulate the task of scene-level 3D open-vocabulary part segmentation. For this analysis, we use our adapted MultiScan [12] scene-level part segmentation data (see Sec. IV-A), as well as the set of annotations we provide on the ScanNet++ [13] dataset (see Sec. IV-B). In our experiments for part-level instance segmentation, we use the Average Precision metric evaluated at 50% (AP$_{50}$), 25% (AP$_{25}$) overlap thresholds, as well as averaged over the range of [0.5 : 0.95 : 0.05] (AP) following previous works.

First, we assess the quality of our segment features for identifying object parts using an oracle mask experiment, isolating feature quality from 3D geometric part segmentation quality. For this analysis, we use ground-truth (GT) part segments for all methods: OpenScene [5], OpenMask3D [6] and Search3D (Ours). For OpenScene, we aggregate per-point features for each GT 3D part segment to obtain segment

Methods	Segments	AP
OpenScene [5]	Oracle	31.4
OpenMask3D [6]	Oracle	35.7
Search3D (Ours)	Oracle	**49.5** (+13.8)

TABLE I: 3D part feature quality evaluation on the MultiScan [12] dataset using GT part segments. We conduct an oracle experiment using annotated GT part segments to aggregate features for OpenScene [5], OpenMask3D [6] and Search3D (Ours) to measure the quality of the features computed from each method, when isolated from geometric segmentation performance.

Methods	Aggregation	AP	AP$_{50}$	AP$_{25}$
(1) OpenScene [5]	segments	3.2	5.5	13.7
(2) OpenMask3D [6]	objects	3.3	6.1	11.3
(3) OpenMask3D [6]	segments	3.1	6.2	18.2
(4) GARField [18] + Search3D	segments	3.5	8.9	20.5
(5) GARField [18] + Search3D	seg. + hierarchy	3.2	8.4	15.3
(6) Search3D (Ours)	seg. + hierarchy	**7.9**	**14.5**	**31.5**
		(+4.6)	(+8.3)	(+13.3)

TABLE II: 3D part segmentation on MultiScan [12]. We formulate joint queries combining object and part descriptions to perform open-vocabulary part retrieval. (1) uses 2D fused OpenSeg [29] feats., and per-point feats. are aggregated over part segments. (2) uses the orig. object-level masks from OpenMask3D. (3) is our adaptation of (2) as a stronger baseline using segment-level aggregation. (4) and (5) use object and part masks from GARField [18] at scale levels 0.1 and 0.35, but use Search3D for feature computation. Our method (6) uses all components, including hierarchical search.

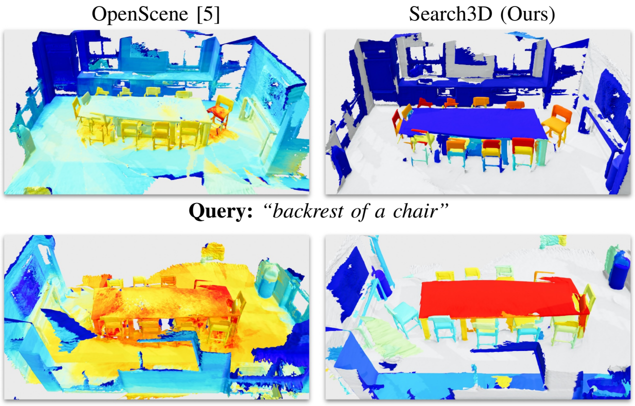
features, and for OpenMask3D we simply aggregate per-mask features for each GT part segment. Results from this oracle experiment on the MultiScan dataset are presented in Tab. I, illustrating strong open-vocabulary part-segmentation performance using our segment-level features, showing at least +13.8 AP improvement over baseline methods.

Having analyzed 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 our method's strong 3D part segmentation ability. Fig. 5 further demonstrates the improved part localization of our approach compared to methods such as OpenScene with per-point feature representations. Additionally in Tab. III, we provide part-segmentation results on our ScanNet++ annotations (Sec. IV-B).

Ablation study. We also analyze the impact of semantically informed segment merging and hierarchical search capabilities. The results shown in Tab. IV emphasize that semantically informed segment merging is crucial, contributing to +3.2 AP$_{50}$. Leveraging the scene-hierarchy when searching for part-level entities adds another +3.1 AP$_{50}$. When applying hierarchical search for open-vocabulary part segmentation, averaging the object-level and part-level similarity scores yields slightly better results than using the maximum of these scores. Overall, these additional components 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 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 [38] in Tab. V,



Query: *"backrest of a chair"*

Query: *"surface of the dinner table"*

Fig. 5: Heatmap visualizations demonstrating the similarity between the text query and scene features. We compare OpenScene [5] 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.

Methods	AP	AP$_{50}$	AP$_{25}$
OpenMask3D [6]	5.2	15.0	18.1
Search3D (Ours)	**17.0**	**32.4**	**38.3**

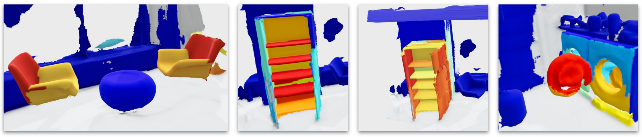
TABLE III: 3D part instance segmentation results on the set of annotations we provide on a selection of ScanNet++ [13] scenes.

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**	**14.5**	**31.5**
				(+3.2)	(+6.3)	(+13.9)

TABLE IV: Ablation study on components of Search3D for 3D part segmentation evaluated on MultiScan [12]. *Merging* refers to post-processing and merging 3D segments based on their feature similarities. *Hier. search* refers to the process of measuring the overall similarity between text query and each segment using both object and part features.

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 [7] (2D&3D)	SAM+ISBNet	CLIP	23.7	27.8	21.2	21.8
Open3DIS [7] (2D&3D)	SAM+Mask3D	CLIP	23.7	26.4	22.5	21.9
Open-vocabulary (using only 3D object mask predictors)						
OpenScene [5] (2D F.)	Mask3D	OpenSeg	11.7	13.4	11.6	9.9
OpenMask3D [6]	Mask3D	CLIP	15.4	17.1	14.1	14.9
Open3DIS [7] (3D M.)	ISBNet	CLIP	18.6	24.7	16.9	13.3
Open3DIS [7] (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 results on the ScanNet200 [38] validation set. Head, Common and Tail AP represent 3 subsets of the ScanNet200 classes based on class frequency [38], in decreasing order of frequency. 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. Among open-vocabulary methods which only use 3D object mask predictors for obtaining 3D object instances, our method has stronger results.



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

Fig. 6: Heatmap visualizations for the similarity between the text query and the segment features from our method Search3D. Dark red means high similarity and dark blue means low similarity.

Methods	AP	AP$_{50}$	AP$_{25}$
GARField [18]	2.4	5.6	9.6
OpenScene [5]	9.0	12.6	16.7
OpenMask3D [6]	10.7	15.7	20.8
Search3D (Ours)	**18.1**	**26.3**	**33.5**

TABLE VI: 3D object instance segmentation scores on MultiScan [12]. 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) [5]	15.3	26.4
(3) OpenScene (2D/3D ensemble) [5]	20.1	35.6
Open-vocabulary, multi-view fusion		
(4) OpenScene (2D fusion) [5]	18.6	31.9
(5) Search3D (Ours)	**20.2**	**38.4**

TABLE VII: 3D material segmentation scores on 3RScan [14] using object-level material annotations from 3DSSG [42]. 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.

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	$T \propto (n_{frames} \cdot M)$
Pre-computation of point projections	32.00 s	$T \propto (n_{frames} \cdot M)$
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 milisec.	-
I/O overhead	15.76 s	-
Total (per-scene)	(∼ 4-5 min)	-
⑤ ***Segment-Feature Computation***	(per-frame avg.)	
Fine-grained 2D segments	1.99 s	$T \propto n_{frames}$
Pixel-aligned feature computation	5.72 s	$T \propto n_{frames}$
Multi-view fusion of segment-features	0.04 s	$T \propto S$
Total (per-scene)	(∼ 10-15 min)	(for 75-150 frames)
⑥ ***Inference***		
New text query / vocab. embedding	0.61 s	-
Search in 3D (similarity computation)	1.57 milisec.	-

TABLE VIII: Runtime and computational complexity of system components. Reported times are averaged over test scenes from MultiScan. In the rightmost column, we also depict 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 the number of RGB frames in the image sequence n_{frames}.

and additionally on MultiScan [12] 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 [26] as a VLM for our method, resulting in strong gains compared to using CLIP [11] to compute open-vocabulary features.

C. 3D material segmentation

Next, we perform an analysis on 3D material segmentation task using the object-level material annotations from the 3RScan dataset [14]. 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 of the main limitations of our work is that we rely on a simple geometrical over-segmentation method to identify the object *parts*. This is also evident from a comparison between Tab. I and Tab. II: we observe that the AP scores in the oracle mask experiment are much higher than the scores obtained 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 methods such as SAM3D [43] proposed to perform multi-view fusion of 2D masks from SAM [44] 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 [41] 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 a few limitations of the geometrical segmentation method we employ for *part* segmentation, which relies on surface normals. Particularly when several part segments share the same surface normal, *e.g.*, drawers of a wardrobe, this approach struggles with separating these scene entities from each other. The hierarchical scene segmentation module also expects a relatively well-reconstructed scene as input. Another limitation is that currently the feature computation needs to take place offline. Finally, our approach is limited to two explicit granularity levels (objects and parts). One key reason for this design choice was the lack of existing evaluation benchmarks tailored for even finer-grained segments. We hope that future work will address this to go beyond these explicit hierarchical levels.

VI. Conclusion

In this work, we presented a novel open-vocabulary 3D scene understanding approach, extending beyond traditional object-centric queries to enable fine-grained search capabilities in 3D environments. Introducing a hierarchical scene representation, our method can effectively identify and segment not only object instances but also object parts and generic attributes. We validate our approach through various experiments and introduce new benchmarks for scene-scale open-vocabulary 3D part instance segmentation, showing improvements over existing methods. We hope this work will pave the way for 3D open-vocabulary segmentation methods that are less anchored at the object-level, and can flexibly handle scene entities of varying 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] 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.

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

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

[7] 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.

[8] K. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, J. Tenenbaum, C. de Melo, M. Krishna, L. Paull, F. Shkurti, and A. Torralba, "ConceptFusion: Open-Set Multimodal 3D Mapping," in *RSS*, 2023.

[9] 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," in *ICRA*, 2023.

[10] F. Engelmann, A. Takmaz, J. Schult, E. Fedele, J. Wald, S. Peng, X. Wang, O. Litany, S. Tang, F. Tombari, M. Pollefeys, L. Guibas, H. Tian, C. Wang, X. Yan, B. Wang, X. Zhang, X. Liu, P. Nguyen, K. Nguyen, A. Tran, C. Pham, Z. Huang, X. Wu, X. Chen, H. Zhao, L. Zhu, and J. Lasenby, "OpenSUN3D: 1st Workshop Challenge on Open-Vocabulary 3D Scene Understanding," *arXiv preprint arXiv:2402.15321*, 2024.

[11] 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.

[12] 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.

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

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

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

[16] 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.

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

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

[19] 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.

[20] 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.

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

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

[23] 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.

[24] 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.

[25] 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*, vol. 9, no. 10, pp. 8921–8928, 2024.

[26] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid Loss for Language Image Pre-Training," in *ICCV*, 2023, pp. 11 941–11 952.

[27] 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.

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

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

[30] 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.

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

[32] 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.

[33] 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.

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

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

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

[37] 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.

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

[39] 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.

[40] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-based Image Segmentation," in *International Journal on Computer Vision (IJCV)*, 2004.

[41] 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.

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

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

[44] 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.