# OpenMask3D: Open-Vocabulary 3D Instance Segmentation



**Author(s):**
Takmaz, Ayça; Fedele, Elisabetta; Summer, Robert W.; Pollefeys, Marc; Tombari, Federico; Engelmann, Francis

**Publication date:**
2023-12

**Permanent link:**
https://doi.org/10.3929/ethz-b-000637077

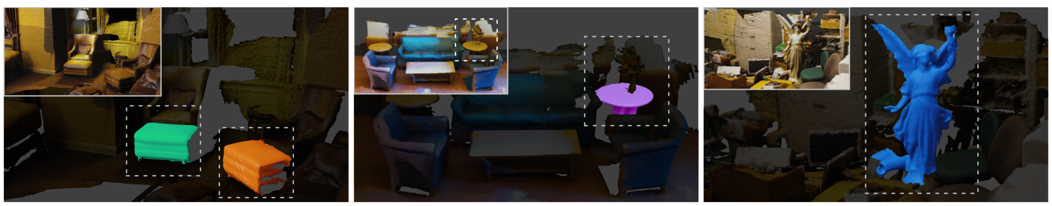
# OpenMask3D: Open-Vocabulary 3D Instance Segmentation

**Ayça Takmaz**[1*] **Elisabetta Fedele**[1*]

**Robert W. Sumner**[1] **Marc Pollefeys**[1] **Federico Tombari**[3] **Francis Engelmann**[1,2,3]

[1]ETH Zurich [2]ETH AI Center [3]Google Zurich [*]equal contribution

openmask3d.github.io



*"footrest"* *"the side table that has a flower vase on it"* *"angel"*

**Figure 1: Open-vocabulary 3D instance segmentation.**

## Abstract

We introduce the task of open-vocabulary 3D *instance* segmentation. Traditional approaches for 3D instance segmentation largely rely on existing 3D annotated datasets, which are restricted to a closed-set of object categories. This is an important limitation for real-life applications where one might need to perform tasks guided by novel, open-vocabulary queries related to objects from a wide variety. Recently, open-vocabulary 3D scene understanding methods have emerged to address this problem by learning queryable features per each point in the scene. While such a representation can be directly employed to perform *semantic* segmentation, existing methods have limitations in their ability to identify object instances. In this work, we address this limitation, and propose OpenMask3D, which is a zero-shot approach for open-vocabulary 3D *instance* segmentation. Guided by predicted class-agnostic 3D instance masks, our model aggregates per-mask features via multi-view fusion of CLIP-based image embeddings. We conduct experiments and ablation studies on the ScanNet200 dataset to evaluate the performance of OpenMask3D, and provide insights about the open-vocabulary 3D instance segmentation task. We show that our approach outperforms other open-vocabulary counterparts, particularly on the long-tail distribution. Furthermore, OpenMask3D goes beyond the limitations of close-vocabulary approaches, and enables the segmentation of object instances based on free-form queries describing object properties such as semantics, geometry, affordances, and material properties.

## 1 Introduction

3D instance segmentation, which is the task of predicting 3D object instance masks along with their corresponding object categories, has many crucial applications in fields such as robotics and augmented reality. Due to its significance, 3D instance segmentation has been receiving a growing amount of attention in recent years. Despite the notable progress made in 3D instance segmentation methods [8, 47, 55, 59], it is noteworthy that all of these methods operate under a closed-set paradigm, in which the set of object categories is limited and closely tied to the datasets used during training.

We argue that there are two key problems with *closed-vocabulary* 3D instance segmentation. First, these approaches are limited in their ability to understand a scene beyond the object categories seen during training. Despite the significant success of 3D instance segmentation approaches from recent years, these closed-vocabulary approaches may fail to recognize novel objects, or potentially incorrectly classify them. One of the main advantages of open-vocabulary approaches, on the other hand, is their capacity to zero-shot learn categories which are not present at all in the training set. This ability has potential benefits for many applications in fields such as robotics, augmented reality, scene understanding and 3D visual search. For example, it is vital for an autonomous robot to have the ability to navigate an unknown environment where novel objects can be present. Furthermore, the robot might need to perform an action based on a free-form query, such as "find the side table with a flower vase on it", which is challenging to perform with the existing closed-vocabulary 3D instance segmentation methods. Hence, the second key problem with closed-vocabulary approaches is their inherent limitation in handling free-form queries.

In an attempt to address and overcome the limitations of a closed-vocabulary setting, there has been a growing interest in open-vocabulary approaches that are able to handle free-form queries. A line of work [18, 39, 41] investigated open-vocabulary 2D image segmentation task. These approaches are driven by the progress in model training at large-scale, and they largely rely on recent foundation models such as CLIP [52] and ALIGN [30] to obtain text-image embeddings. Motivated by the success of these 2D open-vocabulary approaches, another line of work has started exploring 3D open-vocabulary scene understanding task [21, 29, 49], based on the idea of lifting image features from models such as CLIP [52] and OpenSeg [18] to 3D. These approaches aim to obtain a task-agnostic feature representation for each 3D point in the scene, which can be used for querying concepts with open-vocabulary descriptions such as object semantics, affordances or material properties. Their output is typically a heatmap over the points in the scene, which has limited applications in certain aspects, such as handling object instances.

In this work, we propose OpenMask3D, an open-vocabulary 3D instance segmentation method which has the ability to reason beyond a pre-defined set of concepts. Given an RGB-D sequence, and the corresponding 3D reconstructed geometry, OpenMask3D predicts 3D object instance masks, and computes a *mask-feature* representation. Our two-stage pipeline consists of a class-agnostic mask proposal head, and a mask-feature aggregation module. Guided by the predicted class-agnostic 3D instance masks, our mask-feature aggregation module first finds the frames in which the instances are highly visible. Then, in a multi-scale and crop-based manner, it extracts CLIP features from the best images of each mask. These features are then aggregated across multiple views to obtain a feature representation associated with each 3D instance mask. Our approach is intrinsically different from the existing 3D open-vocabulary scene understanding approaches [21, 29, 49] as we propose an instance-based feature computation approach instead of a point-based one. Computing a *mask-feature* per object instance enables us to retrieve object instance masks based on their similarity to any given query, equipping our approach with open-vocabulary 3D instance segmentation capabilities. As feature computation is performed in a zero-shot manner, OpenMask3D is capable of preserving information about novel objects as well as long-tail objects better, compared to trained or fine-tuned counterparts. Furthermore, OpenMask3D goes beyond the limitations of a closed-vocabulary paradigm, and enables segmentation of object instances based on free-form queries describing object properties such as semantics, geometry, affordances, and material properties.

Our contributions are three-fold:

- We introduce the open-vocabulary 3D instance segmentation task in which the object instances that are similar to a given text-query are identified.
- We propose OpenMask3D, which is the first approach that performs open-vocabulary 3D instance segmentation in a zero-shot manner.
- We conduct experiments to provide insights about design choices that are important for developing an open-vocabulary 3D instance segmentation model.

## 2 Related work

**Closed-vocabulary 3D semantic and instance segmentation.** Given a 3D scene as input, 3D semantic segmentation task aims to assign a semantic category to each point in the scene [2–4, 8, 12, 13, 19, 25, 26, 28, 35, 38, 40, 42, 43, 50, 51, 58, 59, 61, 64, 66]. 3D instance segmentation goes a step further by distinguishing multiple objects belonging to the same semantic category, predicting individual masks for each object instance [11, 14, 22, 24, 31, 37, 55, 60, 62, 67]. Current state-of-the-art approach on the ScanNet200 benchmark [9, 54] is Mask3D, proposed by Schult et al.

[55]. Mask3D leverages a transformer architecture to produce 3D mask proposals, together with their corresponding semantic labels. However, similar to existing methods, it assumes a limited set of semantic class labels that can potentially be assigned to an instance. In particular, the number of labels is dictated by the annotations provided in the training datasets, 200 – in the case of ScanNet200 [54]. Given that the English language encompasses numerous nouns, totaling in the range of several hundred thousand [63], it is clear that existing closed-vocabulary approaches have an important limitation in terms of handling object categories and descriptions.

**Foundation models.** Recent text-image foundation models such as CLIP [52], OpenCLIP [7], ALIGN [30], and Flamingo [1] leverage large-scale pretraining to learn image representations guided by natural language descriptions. These models enable zero-shot transfer to various downstream tasks such as object recognition and classification. Driven by the progress in large-scale model pre-training, similar foundation models for images were also explored in another line of work [5, 48], which aim to extract significant class-agnostic features from images. Recently, steps towards a foundation model for the image segmentation task were taken with SAM [33]. SAM has an ability to generate a class-agnostic 2D mask for an object instance, given a set of points that are assumed to belong to that instance. This capability is valuable for our application, especially for recovering high-quality 2D masks from projected 3D instance masks, as further explained in Sec. 3.2.2.

**Open vocabulary 2D segmentation.** As CLIP gained popularity for image classification, numerous new approaches [18, 20, 23, 29, 36, 39, 41, 44, 53, 65, 68, 69] have emerged to tackle open-vocabulary image semantic segmentation. One notable shift was the transition from image-level embeddings to pixel-level embeddings, enhancing models with localization capabilities alongside classification. However, methods with pixel-level embeddings, such as OpenSeg [18] and OV-Seg [41], strongly rely on the accuracy of 2D segmentation masks, and require a certain degree of training. In our work, we rely on CLIP features without finetuning or any additional training, and compute 2D masks using the predicted 3D instance masks.

**Open-vocabulary 3D scene understanding.** Recent success of 2D open-vocabulary segmentation models such as LSeg [39], OpenSeg [18], and OV-Seg[41] has motivated researchers in the field of 3D scene understanding to explore the open vocabulary setting [6, 10, 17, 21, 27, 29, 32, 34, 45, 49, 56, 57]. OpenScene [49] uses per-pixel CLIP features extracted from posed images of a given scene and obtains a point-wise semantic representation of the scene as described in Sec. 2. On the other hand, approaches such as LERF [32] and DFF [34] leverage the interpolation capabilities of NERF [46] to extract a semantic field of the scene. However, it is important to note that all of these approaches have a limited understanding of object *instances* and may face challenges when dealing with instance-related tasks.

## 3 Method

**Overview.** Our OpenMask3D model is illustrated in Fig. 2. Given a set of posed RGB-D images captured in a scene, along with its reconstructed point cloud ①, OpenMask3D predicts 3D instance masks along with associated per-mask feature representations, which can be used for querying instances based on open-vocabulary concepts ④. Our OpenMask3D has two main building blocks, a *class agnostic mask proposal head* ②, and a *mask-feature computation module* ③. The class-agnostic mask proposal head predicts binary instance masks over the points in the point cloud. The mask-feature computation module leverages pre-trained CLIP [52] vision-language model in order to compute meaningful and flexible features for each mask. For each proposed instance mask, the mask-feature computation module first selects the views in which the 3D object instance is highly visible. Subsequently, in each selected view, the module computes a 2D segmentation mask guided by the projection of the 3D instance mask, and refined by the SAM model [33]. Next, the CLIP encoder is employed to obtain image-embeddings of multi-scale image-crops bounding the computed 2D masks. These image-level embeddings are then aggregated across the selected frames in order to obtain a mask-feature representation. Sec. 3.1 describes the class agnostic mask proposal head, and Sec. 3.2 describes the mask-feature computation module.

The key novelty of our method is that it follows an *instance-mask oriented* approach, contrary to existing 3D open-vocabulary scene understanding models which typically compute *per-point* features. These point-feature oriented models have limitations in certain aspects, particularly for identifying object *instances*. Our model aims to overcome such limitations by introducing a framework that employs class agnostic instance masks and aggregates informative features for each object *instance*.

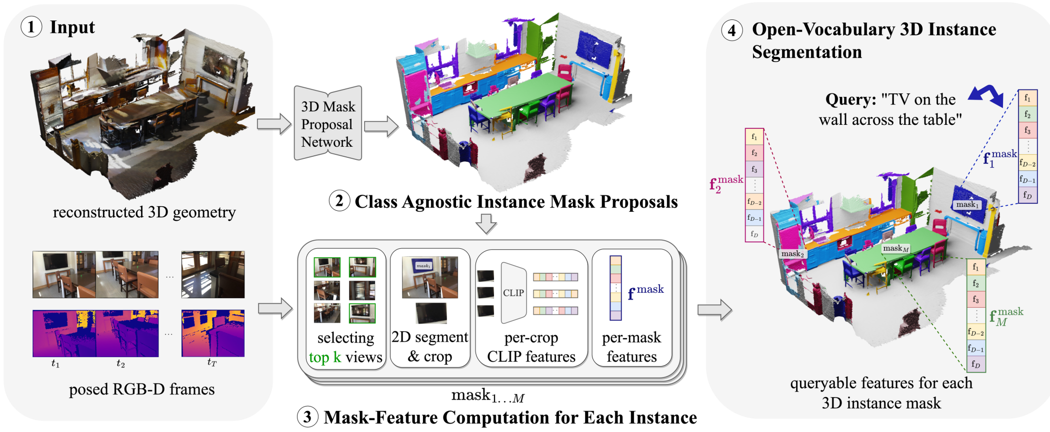



**Figure 2: An overview of our approach.** We propose OpenMask3D, the first open-vocabulary 3D instance segmentation model. Our pipeline consists of four subsequent steps: ① Our approach takes as input posed RGB-D images of a 3D indoor scene along with its reconstructed point cloud. ② Using the point cloud, we compute class-agnostic instance mask proposals. ③ Then, for each mask, we compute a feature representation. ④ Finally, we obtain an open-vocabulary 3D instance segmentation representation, which can be used to retrieve objects related to queried concepts embedded in the CLIP [52] space.

**Input.** Our pipeline takes as input a collection of posed RGB-D images captured in an indoor scene, and the reconstructed point cloud representation of the scene. We assume known camera parameters.

### 3.1 Class agnostic mask proposals

The first step of our approach involves generating $M$ class-agnostic 3D mask proposals $\mathbf{m_1^{3D}}, \ldots, \mathbf{m_M^{3D}}$. Let $\mathbf{P} \in \mathbb{R}^{P \times 3}$ denote the point cloud of the scene, where each 3D point is associated with its corresponding 3D coordinates. Each 3D mask proposal is represented by a binary mask $\mathbf{m_i^{3D}} = (m_{i1}^{3D}, \ldots, m_{iP}^{3D})$ where $m_{ij}^{3D} \in \{0, 1\}$ indicates whether the $j$-th point belongs to $i$-th object instance. To generate these masks, we leverage the transformer-based mask-module of a pre-trained 3D instance segmentation model [55], which is frozen during our computations. The architecture consists of a sparse convolutional backbone based on the MinkowskiUNet [8], and a transformer decoder. Point features obtained from the feature backbone are passed through the transformer decoder, which iteratively refines the instance queries, and predicts an instance heatmap for each query. In the original setup, [55] produces two outputs: a set of $M$ binary instance masks obtained from the predicted heatmaps, along with predicted class labels (from a predefined closed set) for each mask. In our approach, we adapt the model to exclusively utilize the binary instance masks, discarding the predicted class labels and confidence scores entirely. These *class-agnostic* binary instance masks are then utilized in our mask-feature computation module, in order to go beyond semantic class predictions limited to a closed-vocabulary, and obtain open-vocabulary representations instead. Further details about the class-agnostic mask proposal module are provided in Appendix A.1.

### 3.2 Mask-feature computation module

Mask-feature computation module aims to compute a task-agnostic feature representation for each predicted instance mask obtained from the class-agnostic mask proposal module. The purpose of this module is to compute a feature representation that can be used to query open-vocabulary concepts. As we intend to utilize the CLIP text-image embedding space and maximally retain information about long-tail or novel concepts, we solely rely on the CLIP visual encoder to extract image-features on which we build our mask-features.

As illustrated in Fig. 3, the mask-feature computation module consists of several steps. For each instance mask-proposal, we first compute the visibility of the object instance in each frame of the RGB-D sequence, and select top-$k$ views with maximal visibility. In the next step, we compute a 2D object mask in each selected frame, which is then used to obtain multi-scale image-crops in order to extract effective CLIP features. The image-crops are then passed through the CLIP visual encoder to obtain feature vectors that are average-pooled over each crop and each selected view, resulting in the final mask-feature representation. In Sec. 3.2.1, we describe how we select a set of top-$k$ frames for each instance. In Sec. 3.2.2, we describe how we crop the frames, based on the instance we want to embed. In Sec. 3.2.3, we describe how we compute the final mask-features.

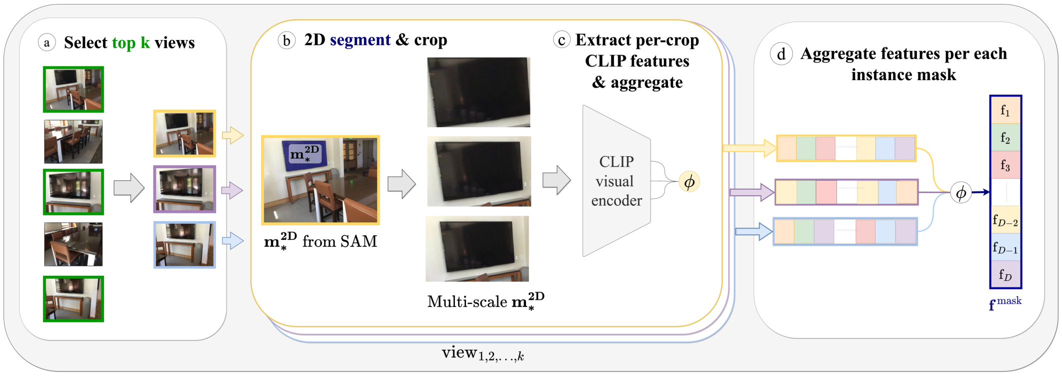



**Figure 3: Mask-Feature Computation Module.** For each instance mask, ⓐ we first compute the visibility of the instance in each frame, and select top-$k$ views with maximal visibility. In ⓑ, we compute a 2D object mask in each selected frame, which is used to obtain multi-scale image-crops in order to extract effective CLIP features. ⓒ The image-crops are then passed through the CLIP visual encoder to obtain feature vectors that are average-pooled over each crop and ⓓ each selected view, resulting in the final mask-feature representation.

### 3.2.1 Frame selection

Obtaining representative *images* of the proposed object instances is crucial for predicting accurate CLIP features (see Appendix A.2.1 for details). To achieve this, we devise a strategy to select, for each of the $M$ predicted instances, a subset of representative frames (Fig. 3, ⓐ) from which we extract CLIP features. In particular, our devised strategy selects frames based on their visibility scores $s_{ij}$ for each mask $i$ in each view $j$. Here, we explain how we compute these visibility scores.

Given a mask proposal $\mathbf{m}_\mathbf{i}^{\mathbf{3D}}$ corresponding to the $i$-th instance, we compute the visibility score $s_{ij} \in [0,1]$ for the $j$-th frame using the following formula:

$$s_{ij} = \frac{vis(i,j)}{\max_{j'}(vis(i,j'))}$$

Here, $vis(i,j)$ represents the number of points from mask $i$ that are visible in frame $j$. Note that we assume that each mask is visible in at least one frame. We compute $vis(i,j)$ using the following approach: First, for each 3D point from the $i$-th instance mask, we determine whether the point appears in the camera's field of view (FOV) in the $j$-th frame. To achieve this, we use intrinsic ($\mathbf{I}$) and extrinsic ($\mathbf{R}|\mathbf{t}$) matrices of the camera from that frame to project each 3D point in the point cloud to the image plane. We obtain corresponding 2D homogeneous coordinates $\mathbf{p_{2D}} = (u,v,w)^\top = \mathbf{I}\cdot(\mathbf{R}|\mathbf{t})\cdot\mathbf{x}$, where $\mathbf{x} = (x,y,z,1)^\top$ is a point represented in homogeneous coordinates. We consider a point to be in the camera's FOV in the $j$-th frame if $w \neq 0$, and the $\frac{u}{w}$ value falls within the interval $[0, W-1]$ while $\frac{v}{w}$ falls within $[0, H-1]$, where $W$ and $H$ represent the width and height of the image, respectively. Next, it is important to note that 3D points that are in the camera's FOV are not necessarily visible from a given camera pose, as they might be occluded by other parts of the scene. Hence, we need to check whether the points are in fact *visible*. In order to examine whether a point is occluded from a given camera viewpoint, we compare the depth value of the 3D point deriving from the geometric computation (i.e. $w$) and the measured depth, i.e. depth value of the projected point in the depth image. We consider points that satisfy the inequality $w - d > k_{\text{threshold}}$ as occluded, where $k_{\text{threshold}}$ is a hyper-parameter of our method. If the projection of a point from mask $i$ is not occluded in the $j-$th view, it contributes to the visible point count, $vis(i,j)$.

Once we compute mask-visibility scores for each frame, we select the top $k_{\text{view}}$ views with the highest scores $s_{ij}$ for each mask $i$. Here, $k_{\text{view}}$ represents another hyperparameter.

### 3.2.2 2D mask computation and multi-scale crops

In this section, we explain our approach for computing CLIP features based on the selected frames from the previous step. Given a selected frame for a mask, our objective is to find the optimal image crops from which to extract features, as illustrated in (Fig. 3, ⓑ). Simply considering all of the projected points of the mask often results in imprecise and noisy bounding boxes, largely affected by the outliers (see Appendix A.2.2 for details). To address this, we employ a class-agnostic 2D segmentation model, SAM [33], which predicts a 2D mask conditioned on a set of input points, along with a mask confidence score.

SAM is sensitive to the set of input points (please see Appendix A.2.3, A.2.4). Hence, in order to obtain a high quality mask from SAM with a high confidence score, we draw inspiration from the RANSAC algorithm [16] and proceed as outlined in Algorithm 1. We sample $k_{sample}$ points from the projected points, and run SAM using these points. The output of SAM at a specific iteration $r$ is a 2D mask ($\mathbf{m_r^{2D}}$) and a confidence score ($\texttt{score}_r$). This process is repeated for $k_{rounds}$ iterations, and the 2D mask with the highest confidence score is selected.

**Algorithm 1** - 2D mask selection algorithm

$\texttt{score}_* \leftarrow 0$, $\mathbf{m_*^{2D}} \leftarrow \mathbf{0}$, $r \leftarrow 0$
**while** $r < k_{rounds}$ **do**
    Sample $k_{sample}$ points among the projected points at random
    Compute the mask $\mathbf{m_r^{2D}}$ and the score $\texttt{score}_r$ based on the sampled points using SAM
    **if** $\texttt{score}_r > \texttt{score}_*$ **then**
        $\texttt{score}_* \leftarrow \texttt{score}_r$, $\mathbf{m_*^{2D}} \leftarrow \mathbf{m_r^{2D}}$
    **end if**
    $r \leftarrow r + 1$
**end while**

Next, we use the resulting mask $\mathbf{m_*^{2D}}$ to generate $L = 3$ multi-level crops of the selected image. This allows us to enrich the features by infusing more context information from the surrounding environment. Specifically, the first bounding box $\mathbf{b^1} = (x_1^1, y_1^1, x_2^1, y_2^1)$ with $0 \le x_1^1 < x_2^1 < W$ and $0 \le y_1^1 < y_2^1 < H$ is a tight bounding box derived from the 2D mask. The other bounding boxes $\mathbf{b^2}$ and $\mathbf{b^3}$ are incrementally larger, and their 2D coordinates are obtained as follows:

$$
\begin{aligned}
x_1^l &= \max(0, x_1^1 - (x_2^1 - x_1^1) \cdot k_{exp} \cdot l) \\
y_1^l &= \max(0, y_1^1 - (y_2^1 - y_1^1) \cdot k_{exp} \cdot l) \\
x_2^l &= \min(x_2^1 + (x_2^1 - x_1^1) \cdot k_{exp} \cdot l, W - 1) \\
y_2^l &= \min(y_2^1 + (y_2^1 - y_1^1) \cdot k_{exp} \cdot l, H - 1)
\end{aligned}
$$

$l = 2, 3$ represents the level of the features under consideration, $k_{exp} = 0.2$ is a predefined constant.

#### 3.2.3 CLIP feature extraction and mask-feature aggregation

For each instance mask, we collect $k \cdot L$ images by selecting top-$k$ views and obtaining $L$ multi-level crops as described in Sec. 3.2.1 and Sec. 3.2.2. Collected image-crops are then passed through the CLIP visual encoder in order to extract image features in the CLIP embedding space, as illustrated in (Fig. 3, c). We then aggregate the features obtained from each crop that correspond to a given instance mask in order to get an average per-mask CLIP feature (Fig. 3, d). The computed features are task-agnostic, and can be used for various instance-based tasks by encoding a given text or image-based query, using the same CLIP model we employed to encode the image-crops.

## 4 Experiments

In this section, we present quantitative and qualitative results from our method. To quantitatively evaluate our approach, we compare OpenMask3D with supervised 3D instance segmentation approaches as well as existing open-vocabulary 3D scene understanding models we adapted for the 3D instance segmentation task. Furthermore, we provide an ablation study for OpenMask3D. In addition, we provide qualitative results from our method on the open-vocabulary 3D instance segmentation task, demonstrating potential applications. Additional results are provided in Appendix C.

### 4.1 Quantitative results: closed-vocabulary 3D instance segmentation evaluation

We quantitatively evaluate our approach on the closed-vocabulary 3D instance segmentation task on the ScanNet200 [9, 54] dataset. Results are presented in Tab. 1.

#### 4.1.1 Experimental setting

**Data.** We perform our experiments using the 312 validation scenes from the ScanNet200 dataset [54], and evaluate for 3D instance segmentation task using the closed vocabulary of 200 categories from the ScanNet200 [54] annotations. Rozenberszki et al. [54] also provide a grouping of ScanNet200 categories based on the frequency of the number of labeled surface points in the training set, resulting

| Model | Supervision | Image Features | AP | $AP_{50}$ | $AP_{25}$ | head (AP) | common (AP) | tail (AP) |
|---|---|---|---|---|---|---|---|---|
| *Closed-vocabulary* | | | | | | | | |
| Mask3D [55] | fully | - | 26.9 | 36.2 | 41.4 | 39.8 | 21.7 | 17.9 |
| *Open-vocabulary* | | | | | | | | |
| OpenScene [49] (2D Fusion) | - | OpenSeg [18] | 11.7 | 15.2 | 17.8 | 13.4 | **11.6** | 9.9 |
| OpenScene [49] (3D Distill) | - | OpenSeg [18] | 4.8 | 6.2 | 7.2 | 10.6 | 2.6 | 0.7 |
| OpenScene [49] (2D/3D Ensemble) | - | OpenSeg [18] | 5.3 | 6.7 | 8.1 | 11.0 | 3.2 | 1.1 |
| OpenScene [49] (2D Fusion) | - | LSeg [39] | 6.0 | 7.7 | 8.5 | **14.5** | 2.5 | 1.1 |
| OpenMask3D (Ours) | - | CLIP [52] | **12.8** | **16.8** | **19.0** | 14.0 | **11.6** | **13.0** |

**Table 1: 3D instance segmentation results on the ScanNet200 validation set.** Metrics are AP evaluated at 50%, 25% and averaged over an overlap range. We also report AP scores on the head, common, tail subsets of the ScanNet200 dataset [54]. Note that Mask3D [55] relies on fully-supervised training on 3D scenes, while OpenScene [49] is built upon on 2D models (i.e. LSeg [39] and OpenSeg [18]), both trained on labeled datasets for 2D semantic segmentation. Our approach OpenMask3D outperforms other open-vocabulary counterparts, particularly on the long-tail classes.

in 3 subsets: *head* (66 categories), *common* (68 categories), *tail* (66 categories). This grouping enables us to evaluate the performance of our method on the long-tail distribution, making ScanNet200 a natural choice as the evaluation dataset.

**Metrics.** We employ a commonly used 3D instance segmentation metric, average precision (AP). AP scores are evaluated at mask overlap thresholds of $50\%$ and $25\%$, and averaged over the overlap range of $[0.5 : 0.95 : 0.05]$ following the evaluation scheme from ScanNet [9]. Computation of the metrics requires each mask to be assigned a prediction confidence score. We assign a prediction confidence score of $1.0$ for each predicted mask in our experiments.

**OpenMask3D implementation details**. We use posed RGB-depth pairs from the validation set of the ScanNet dataset, and we process the RGB-D sequences at a frame rate of $3Hz$. In order to compute image features on the mask-crops, we use CLIP [52] visual encoder from the ViT-L/14 model pre-trained at a 336 pixel resolution, which has a feature dimensionality of 768. For the visibility score computation, we use $k_{threshold}$ = $0.2$, and for top-view selection we use $k_{view}$ = $5$. In all experiments with multi-scale crops, we use $L$ = $3$ levels. In the 2D mask selection algorithm based on SAM [33], we repeat the process for $k_{rounds}$ = $10$ rounds, and sample $k_{sample}$ = $5$ points at each iteration. For the class-agnostic mask proposal, we use the Mask3D [55] model trained on ScanNet200 instances, and exclusively use the instance mask heatmaps. We do not filter any instance mask proposals, and run DBSCAN [15] to obtain spatially contiguous clusters, breaking down masks into smaller new masks when necessary. This process results in a varying number of mask proposals for each scene. For further implementation details about OpenMask3D, please refer to Appendix A. Computation of the mask-features of a ScanNet scene on a single GPU takes on average 2-5 minutes depending on the number of mask proposals and number of frames in the RGB-D sequence.

**Methods in comparison.** We compare with Mask3D [55], which is the current state-of-the-art on the ScanNet200 3D instance segmentation benchmark. We additionally compare against recent open-vocabulary 3D scene understanding model variants (2D-fusion, 3D distill, 2D/3D ensemble) from OpenScene [49]. As the output from OpenScene is a per-point task-agnostic feature representation, we have to adapt it for the instance-segmentation task using our class-agnostic masks. We aggregate per-mask features from OpenScene by average-pooling per-point features for each point in a given instance mask (please refer to Appendix B for further details).

**Class assignment.** Our open-vocabulary approach does not predict semantic category labels per each instance mask, but it instead computes a task-agnostic feature vector for each instance, which can be used for performing a semantic label assignment. In order to evaluate our model on the closed-vocabulary 3D instance segmentation task, we need to assign each object instance to a semantic category. Similar to OpenScene [49], we compute cosine similarity between mask-features and the text embedding of a given query in order to perform class assignments. Following Peng et al. [49], we use prompts in the form of "a {} in a scene", and compute text-embeddings using CLIP model ViT-L/14(336px) [52] for each semantic class in the ScanNet200 dataset. This way, we compute a similarity score between each instance and each object category, and assign instances to the category with the closest text embedding.

#### 4.1.2 Results

**3D closed-vocabulary instance segmentation results.** We quantitatively evaluate our approach on the closed-vocabulary instance segmentation task on the ScanNet200 [9, 54] dataset, and qualitatively

| Top-k | 2D Mask | Multi-Scale | AP | $AP_{50}$ | $AP_{25}$ | head (AP) | common (AP) | tail (AP) |
|---|---|---|---|---|---|---|---|---|
| *Top-k analysis* | | | | | | | | |
| 1 | ✓ | ✓ | 11.1 | 14.2 | 16.9 | 12.6 | 10.5 | 9.9 |
| 5 | ✓ | ✓ | **12.8** | **16.8** | **19.0** | 14.0 | 11.6 | **13.0** |
| 10 | ✓ | ✓ | 12.7 | 16.4 | 18.9 | **14.3** | **11.8** | 11.8 |
| all | ✓ | ✓ | 11.9 | 15.2 | 17.6 | 12.9 | 10.5 | 12.4 |
| *Component analysis* | | | | | | | | |
| 5 | ✗ | ✗ | 9.0 | 12.0 | 13.9 | 11.2 | 7.6 | 8.0 |
| 5 | ✗ | ✓ | 11.5 | 14.9 | 17.0 | 13.1 | 10.7 | 10.4 |
| 5 | ✓ | ✗ | 11.4 | 15.0 | 17.5 | 13.6 | 9.6 | 11.0 |
| 5 | ✓ | ✓ | **12.8** | **16.8** | **19.0** | **14.0** | **11.6** | **13.0** |

**Table 2: OpenMask3D Ablation Study.** *Top-k frame selection*, *2D mask* and *multi-scale crop* components. *2D mask* refers to whether SAM was employed for computing 2D masks. Results on the ScanNet200 validation set.

| Model | Supervision | Image Features | AP | head (AP) | common (AP) | tail (AP) |
|---|---|---|---|---|---|---|
| *Closed-vocabulary* | | | | | | |
| Mask3D [55] | fully | – | 26.9 | 39.8 | 21.7 | 17.9 |
| *Open-vocabulary* | | | | | | |
| OpenScene [49] (2D Fusion, oracle masks) | – | OpenSeg [18] | 22.9 | 26.2 | **22.0** | 20.2 |
| OpenScene [49] (2D Fusion, oracle masks) | – | LSeg [39] | 11.8 | **26.9** | 5.2 | 1.7 |
| OpenMask3D (Ours, oracle masks) | – | CLIP [52] | **23.4** | 24.6 | 19.3 | **27.0** |

**Table 3: 3D instance segmentation results on the ScanNet200 validation set, using oracle masks.** All open-vocabulary results were computed using ground-truth masks as an oracle. All open-vocabulary approaches use a query of the form: 'a { } in a scene'.

investigate the performance on open-vocabulary queries. Closed-vocabulary instance segmentation results on the ScanNet200 dataset are provided in Tab. 1.

State-of-the-art fully supervised approach Mask3D [55] demonstrates superior performance compared to open-vocabulary counterparts. While this gap is more prominent for the *head* and *common* categories, the difference is less apparent for the *tail* categories. This outcome is expected, as Mask3D benefits from full-supervision using the class labels from the ScanNet200 dataset. Furthermore, as there are more training samples for categories within the *head* and *common* subsets (please refer to [54] for statistics), the fully-supervised approach is more frequently exposed to these categories - resulting in a stronger performance. When we compare OpenMask3D with other open-vocabulary approaches, we observe that our method performs better on 5 out of 6 metrics. Notably, OpenMask3D surpasses the performance of other open-vocabulary counterparts, particularly on the *tail* categories by a significant margin. These results indicate that our instance-centric method OpenMask3D specifically designed for the open-vocabulary 3D instance segmentation task shows stronger performance compared to other open-vocabulary approaches which adopt a point-centric feature representation.

**Ablation Study.** In Tab. 2, we analyze design choices for OpenMask3D, *i.e.*, *top-k view selection*, *multi-scale cropping* and *2D mask segmentation*. 2D mask segmentation refers to whether we use SAM [33] for refining the 2D mask from which we obtain a 2D bounding box. When SAM is not used, we simply crop the tightest bounding box around the projected 3D points. The ablation study shows that multi-scale cropping and segmenting a 2D mask to obtain a more accurate image crop for a given instance both positively affect the performance. Our analysis on the components of OpenMask3D shows that the effect of 2D mask segmentation is less significant than the effect of multi-scale cropping. Based on our analysis on the top-$k$ view selection, we suspect that using too many frames (in which an object instance is visible) could detriment the quality of the mask-features.

**How well would our approach perform if we had *perfect* masks?** Another analysis we conduct is related to the class-agnostic masks. As the quality of the masks plays a key role in our process, we aim to quantify how much the performance could improve if we had "perfect" oracle masks. For this purpose, we run OpenMask3D using ground truth instance masks from the ScanNet200 dataset instead of using our predicted class-agnostic instance masks. In a similar fashion, we re-compute mask-features for the OpenScene 2D-fusion variant (best-performing variant of OpenScene for our task), using the oracle masks. In Tab. 3, it is evident that the quality of the masks plays an important role for our task. Remarkably, our mask-feature computation approach that is *not* trained on any labeled data, when provided with oracle masks, even surpasses the performance of the fully supervised method Mask3D, on the *long-tail* categories by +9.1% AP. While open-vocabulary counterparts perform well on *head* and *common* subsets, this is again expected, as OpenSeg [18] and LSeg [39] models involve a certain degree of training with class labels. Overall, this analysis indicates that our approach has indeed promising results which can be further improved by higher quality class-agnostic masks, without requiring any additional training or finetuning with labeled data.

### 4.2 Qualitative results

In Fig. 4, we share qualitative results from our approach for the open-vocabulary 3D instance segmentation task. With its zero-shot learning capabilities, OpenMask3D is able to segment a given query object that might not be present in common segmentation datasets. Furthermore, object properties such as colors, textures, situational context and affordances are successfully recognized by OpenMask3D. Additional qualitative results in this direction are provided in Appendix C.

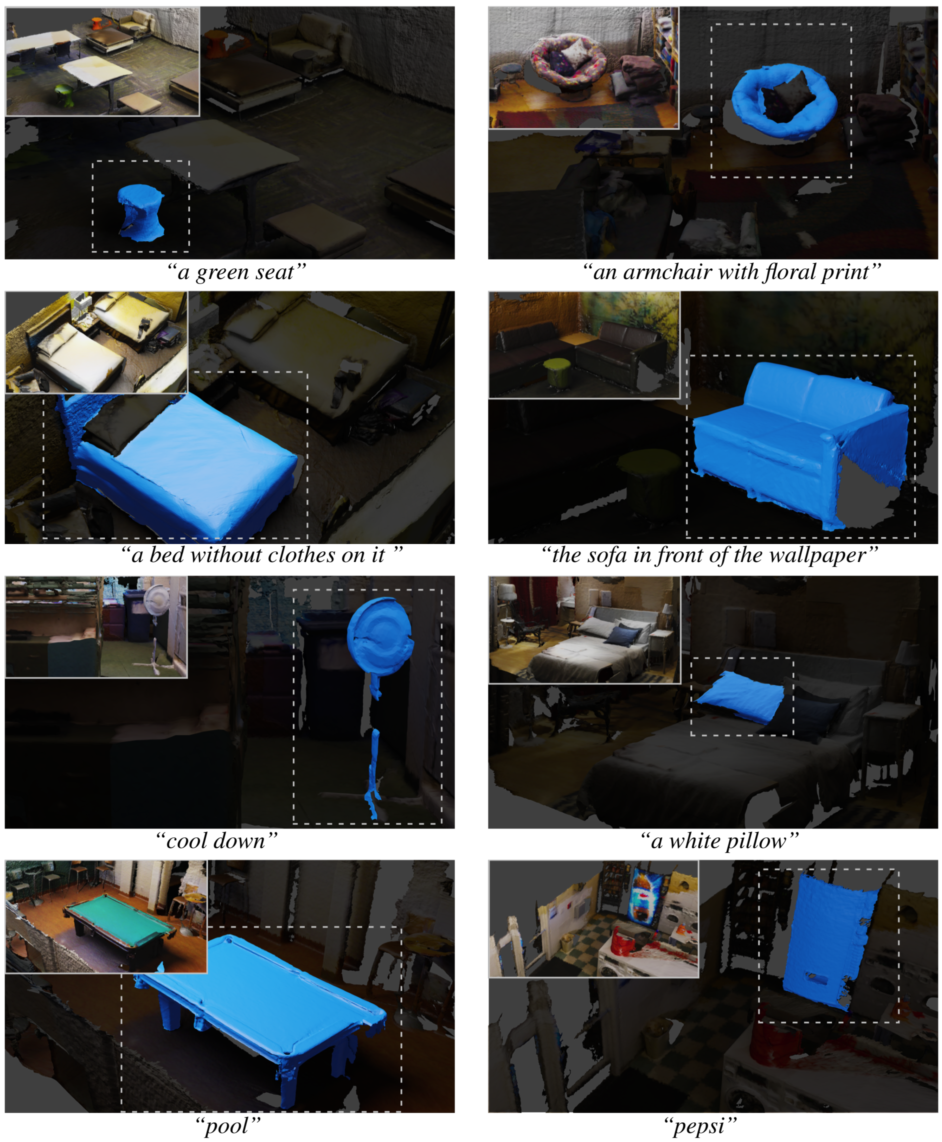



**Figure 4: Qualitative results from OpenMask3D.** Our open-vocabulary instance segmentation approach is capable of handling different types of queries. Novel objects as well as objects described by colors, textures, situational context and affordances are successfully retrieved by OpenMask3D.

In Fig. 5, we provide qualitative comparisons between our instance-based open-vocabulary representation and the point-based representation provided by OpenScene [49]. Our method OpenMask3D computes the similarity between the query embedding and each per-mask feature vector for each object *instance*, which results in crisp instance boundaries. This is particularly suitable for the use cases in which one needs to handle object instances. Additional analysis and further details about the visualization are provided in Appendix B.

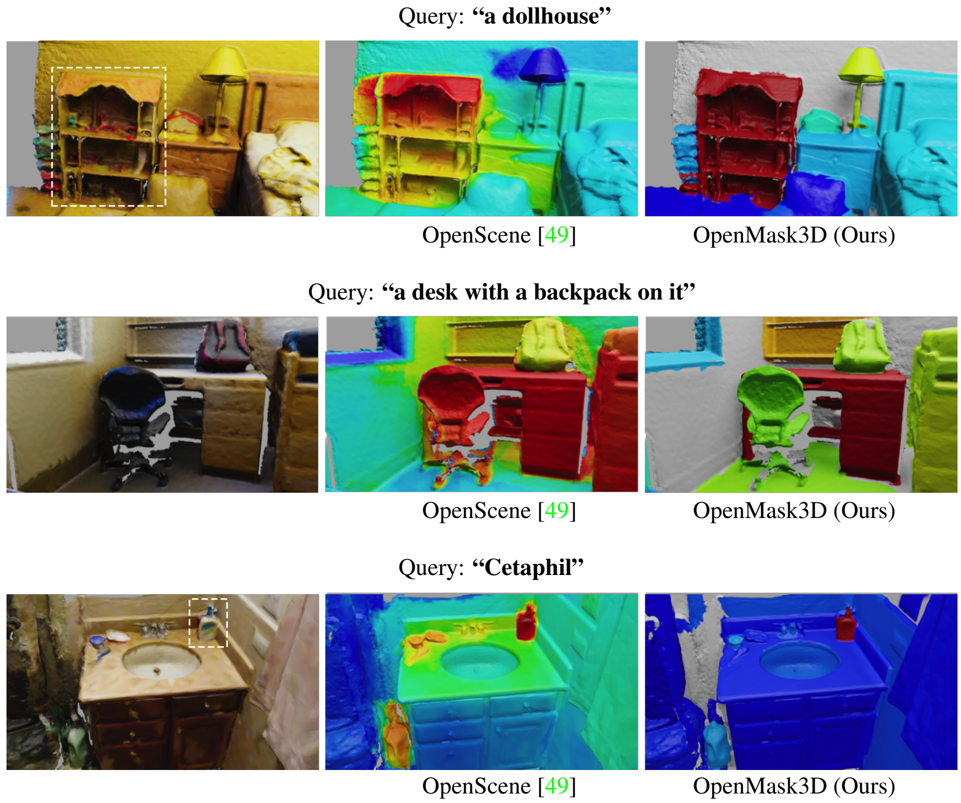



**Figure 5: Qualitative comparisons.** Heatmaps illustrating the similarity between the CLIP embeddings of a specific query and the open-vocabulary representation of the scene. A comparison is made between the point-based OpenScene approach (left) and our mask-based approach OpenMask3D (right). Dark red means high similarity, and dark blue means low similarity with the query text.

## 5 Conclusion

We propose OpenMask3D, the first open-vocabulary 3D instance segmentation model that can identify objects instances in a 3D scene, given arbitrary text queries. This is beyond the capabilities of existing 3D semantic instance segmentation approaches, which are typically trained to predict categories from a closed vocabulary. With OpenMask3D, we push the boundaries of 3D instance segmentation. Our method is capable of segmenting object instances in a given 3D scene, guided by open-vocabulary queries describing object properties such as semantics, geometry, affordances, material properties and situational context. Thanks to its zero-shot learning capabilities, OpenMask3D is able to segment multiple instances of a given query object that might not be present in common segmentation datasets on which closed-vocabulary instance segmentation approaches are trained. This opens up new possibilities for understanding and interacting with 3D scenes in a more comprehensive and flexible manner. We encourage the research community to explore open-vocabulary approaches, where knowledge from different modalities can be seamlessly integrated into a unified and coherent space.

**Limitations** The experiments conducted with oracle masks indicate that there is room for improvement in terms of the quality of 3D mask proposals. In addition, designing an evaluation scheme for systematically assessing open-vocabulary capabilities still remains a challenge. Closed-vocabulary evaluation, while valuable for initial assessment, falls short in revealing the true extent of open-vocabulary potentials of proposed models.

**Acknowledgments and disclosure of funding.** Francis Engelmann is a postdoctoral research fellow at the ETH AI Center. This project is partially funded by the ETH Career Seed Award "Towards Open-World 3D Scene Understanding", and Innosuisse grant (48727.1 IP-ICT). We sincerely thank Jonas Schult for helpful discussions.

## References


[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[2] Abhishek Anand, Hema Swetha Koppula, Thorsten Joachims, and Ashutosh Saxena. Contextually Guided Semantic Labeling and Search for 3D Point Clouds. In *International Journal on Robotics Research (IJRR)*, 2011.

[3] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point Convolutional Neural Networks by Extension Operators. In *ACM Transactions On Graphics (TOG)*, 2018.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *International Conference on Computer Vision (ICCV)*, 2021.

[6] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S. Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary Queryable Scene Representations for Real World Planning. *arXiv*, 2022.

[7] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *ArXiv*, abs/2212.07143, 2022.

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. PLA: Language-Driven Open-Vocabulary 3D Scene Understanding. In *CVPR*, 2023.

[11] Cathrin Elich, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. 3D-BEVIS: Birds-Eye-View Instance Segmentation. In *German Conference on Pattern Recognition (GCPR)*, 2019.

[12] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. In *International Conference on Computer Vision (ICCV) Workshops*, 2017.

[13] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. In *European Conference on Computer Vision (ECCV) Workshops*, 2018.

[14] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi Proposal Aggregation for 3D Semantic Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Knowledge Discovery and Data Mining*, 1996.

[16] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981.

[17] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv*, 2022.

[18] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling Open-Vocabulary Image Segmentation with Image-Level Labels. In *European Conference on Computer Vision (ECCV)*, 2021.

[19] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In *ICLR*, 2022.

[21] Huy Ha and Shuran Song. Semantic Abstraction: Open-World 3D Scene Understanding from 2D Vision-Language Models. In *CORL*, 2022.

[22] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. OccuSeg: Occupancy-aware 3D Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] Sunan He, Taian Guo, Tao Dai, Ruizhi Qiao, Bo Ren, and Shu-Tao Xia. Open-Vocabulary Multi-Label Classification via Multi-modal Knowledge Transfer. In *AAAI*, 2023.

[24] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] Zeyu Hu, Xuyang Bai, Jiaxiang Shang, Runze Zhang, Jiayu Dong, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew Lan Tai. VMNet: Voxel-Mesh Network for Geodesic-Aware 3D Semantic Segmentation. In *International Conference on Computer Vision (ICCV)*, 2021.

[26] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Point-wise Convolutional Neural Network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual Language Maps for Robot Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[28] Jing Huang and Suya You. Point Cloud Labeling Using 3D Convolutional Neural Network. In *International Conference on Pattern Recognition (ICPR)*, 2016.

[29] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. ConceptFusion: Open-Set Multimodal 3D Mapping. *arXiv*, 2023.

[30] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[31] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[32] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language Embedded Radiance Fields. *arXiv*, 2023.

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment Anything. *arXiv*, 2023.

[34] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing NeRF for Editing via Feature Field Distillation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume abs/2205.15585, 2022.

[35] Hema Koppula, Abhishek Anand, Thorsten Joachims, and Ashutosh Saxena. Semantic Labeling of 3D Point Clouds for Indoor Scenes. In *Neural Information Processing Systems (NeurIPS)*, 2011.

[36] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. Open-Vocabulary Object Detection upon Frozen Vision and Language Models. In *ICLR*, 2023.

[37] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R. Oswald. 3D Instance Segmentation via Multi-task Metric Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[38] Loic Landrieu and Martin Simonovsky. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven Semantic Segmentation. In *International Conference on Learning Representations (ICLR)*, 2022.

[40] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on X-transformed Points. In *Neural Information Processing Systems (NeurIPS)*, 2018.

[41] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-Vocabulary Semantic Segmentation with Mask-adapted CLIP. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[42] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[43] Yan Lu and Christopher Rasmussen. Simplified Markov Random Fields for Efficient Semantic Labeling of 3D Point Clouds. In *International Conference on Intelligent Robots and Systems (ICIRS)*, 2012.

[44] Chao Ma, Yu-Hao Yang, Yanfeng Wang, Ya Zhang, and Weidi Xie. Open-vocabulary Semantic Segmentation with Frozen Vision-Language Models. In *British Machine Vision Conference (BMVC)*, 2022.

[45] Kirill Mazur, Edgar Sucar, and Andrew Davison. Feature-Realistic Neural Fusion for Real-Time, Open Set Scene Understanding. In *International Conference on Robotics and Automation (ICRA)*, 2023.

[46] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume abs/2003.08934, 2020.

[47] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3D: Out-of-Context Data Augmentation for 3D Scenes. In *International Conference on 3D Vision (3DV)*, 2021.

[48] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *arXiv*, 2023.

[49] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D Scene Understanding with Open Vocabularies. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[50] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[51] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Neural Information Processing Systems (NeurIPS)*, 2017.

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[53] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. DenseCLIP: Language-Guided Dense Prediction with Context-Aware Prompting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[54] David Rozenberszki, Or Litany, and Angela Dai. Language-Grounded Indoor 3D Semantic Segmentation in the Wild. In *European Conference on Computer Vision (ECCV)*, 2022.

[55] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D for 3D Semantic Instance Segmentation. In *International Conference on Robotics and Automation (ICRA)*, 2023.

[56] Nur Muhammad Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur D. Szlam. CLIP-Fields: Weakly Supervised Semantic Fields for Robotic Memory. *arxiv*, 2022.

[57] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action. *arXiv*, 2022.

[58] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. SEGCloud: Semantic Segmentation of 3D Point Clouds. In *International Conference on 3D Vision (3DV)*, 2017.

[59] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *International Conference on Computer Vision (ICCV)*, 2019.

[60] Thang Vu, Kookhoi Kim, Tung M. Luu, Xuan Thanh Nguyen, and Chang D. Yoo. SoftGroup for 3D Instance Segmentation on 3D Point Clouds. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[61] Tianyi Wang, Jian Li, and Xiangjing An. An Efficient Scene Semantic Labeling Approach for 3D Point Cloud. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[62] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[63] Tianhan Wei, Xiang Li, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. FSS-1000: A 1000-Class Dataset for Few-Shot Segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2866–2875, 2019.

[64] Daniel Wolf, Johann Prankl, and Markus Vincze. Fast Semantic Segmentation of 3D Point Clouds using a Dense CRF with Learned Parameters. In *International Conference on Robotics and Automation (ICRA)*, 2015.

[65] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. GroupViT: Semantic Segmentation Emerges from Text Supervision. In *CVPR*, 2022.

[66] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters. In *European Conference on Computer Vision (ECCV)*, 2018.

[67] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[68] Nir Zabari and Yedid Hoshen. Semantic Segmentation In-the-Wild Without Seeing Any Segmentation Examples. *arXiv*, 2021.

[69] Chong Zhou, Chen Change Loy, and Bo Dai. Extract Free Dense Labels from CLIP. In *European Conference on Computer Vision (ECCV)*, 2022.

# Appendix

The Appendix section is structured as follows: In Sec. A, we provide further details about our OpenMask3D model, and we elaborate on certain design choices. In Sec. B, we provide further details on the baseline experiments . In Sec. C, we show additional qualitative results. Our code and model will be made publicly available.

## A OpenMask3D details

### A.1 Class-agnostic mask proposal module

As OpenMask3D is an instance-centric approach, we first need to obtain instance mask proposals. In the open-vocabulary setting, these instance masks are not associated with any class label; they are class agnostic.

We use the mask module of the transformer-based Mask3D [55] architecture as a basis for our class-agnostic mask proposal module. Specifically, we train the proposal module on the training set of ScanNet200 [54] for instance segmentation (without class labels). For the open-set class assignments, we keep the weights of the mask proposal module frozen. Unlike Mask3D, our mask proposal module discards class predictions and mask confidence scores which are based on class likelihoods, and we only retain the binary instance mask proposals.

Our mask proposal architecture consists of a sparse convolutional backbone based on the MinkowskiUNet [8], and a transformer decoder. Point features obtained from the feature backbone are passed through the transformer decoder, which iteratively refines the instance queries, and predicts an instance heatmap for each query. The query parameter specifies the desired number of mask proposals from the transformer-based architecture. We set the number of queries to $150$, following the original implementation of Mask3D. This choice enables us to obtain a sufficient number of mask proposals for our open-vocabulary setting. Output from the final mask-module in the transformer decoder is a set of binary instance masks.

The original implementation of the Mask3D [55] model first ranks the proposed instance masks based on their confidence scores, and retains the top $k$ masks based on this ranking. As the confidence scores are guided by class likelihoods, we do not utilize such scores to rank the masks, and do not filter out any of the proposed instance masks. Furthermore, since we aim to retain as many mask proposals as possible, we do not perform mask-filtering based on object sizes, or pairwise mask overlaps. In other words, we keep all masks proposed by the mask module as each of these masks might correspond to a potential open-vocabulary query.

As the model may occasionally result in mask proposals that are not spatially contiguous, the original implementation of Mask3D employs the DBSCAN clustering algorithm [15] to break down such masks into smaller, spatially contiguous clusters. We follow this practice, and perform DBSCAN clustering on the instance masks, setting the epsilon parameter, $\epsilon$, to $0.95$. This procedure often increases the final number of instance masks. As such, our OpenMask3D model generates class-agnostic instance masks, for which we compute open-set mask-features as described next.

### A.2 Mask-feature computation module details

In the main paper, we outlined the steps to extract per-mask CLIP [52] features. In this section, we will delve into the motivations behind specific design choices made throughout the process.

#### A.2.1 Per mask view ranking: assumptions and visualizations

Once we obtain the class-agnostic masks, our primary objective is to determine the best CLIP features for each of them. We operate under the assumption that CLIP is capable of extracting meaningful features from instances when provided with a favorable viewpoint, and in our feature computation process, we prioritize selecting views that offer a higher number of visible points. In practice, to assess the quality of a particular viewpoint for each mask, we compute the proportion of points of the 3D instance mask that are visible from that view. Next, we rank the views based on their visibility scores, and we select the top $k$ views with the highest scores, where $k$ is a hyper parameter.

Fig. 6 illustrates this concept, where two views of a given instance are compared based on the number of visible points. The view that has a higher number of visible points from the instance is selected (outlined in green).

#### A.2.2 Why do we need SAM?

In order to get the CLIP image features of a particular instance in the selected views, we need to crop the image part containing the instance. This involves obtaining an initial tight crop of the instance projected in 2D and then incrementally expanding the crop to include some visual context around it.

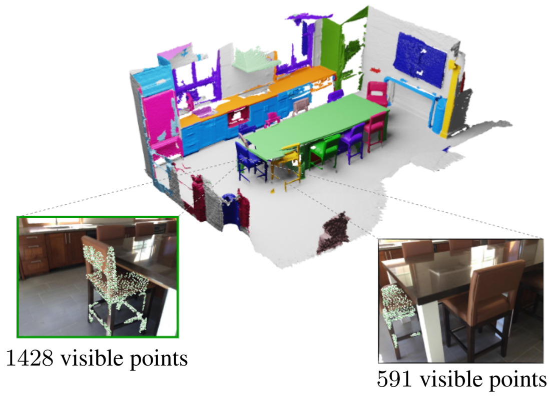



**Figure 6:** Visual comparison between a selected view (marked in green outline) for the corresponding chair instance and a discarded view. The green points in the images represent the 3D instance points projected onto the 2D image. In the selected view the instance is fully visible, and mostly occluded in the discarded one.

A straightforward approach to perform the cropping would be to project all of the visible 3D points belonging to the mask onto the 2D image, and fit a bounding box around these points. However, as we do not discard any mask proposals, the masks can potentially include outliers, or the masks might be too small as an outcome of the DBSCAN algorithm described in Sec. A.1. As depicted in Fig. 7, this can result in bounding boxes that are either too large or too small. Using CLIP-image features obtained from crops based solely on projected points can lead to inferior instance mask-features.

To address this challenge, and to improve the robustness of our model against noisy instance masks, we propose using a 2D segmentation method that takes a set of points from the mask as input, and produces the corresponding 2D mask, together with its confidence score. The Segment Anything Model (SAM) [33] precisely fulfills this task, allowing us to generate accurate 2D masks in a class-agnostic manner.

**Figure 7:** Difference between the bounding boxes obtained by tightly cropping around the projected points from the 3D instance mask (left), and the bounding box obtained from the 2D mask generated by SAM (right).

### A.2.3 Which points should we give as input to SAM?

When using SAM for predicting a 2D mask, we need to determine a set of points which the mask generation process will be conditioned on. Initially, we attempted to input all the visible points projected from the 3D mask. However, this approach resulted in poor quality masks due to the inaccuracies of the projections and the noise present in the 3D masks, as illustrated in Figure 8.

To address this issue, we explored an alternative approach. Instead of using all of the visible points as input for SAM, we randomly sample a subset of points from the projected 3D mask. Interestingly, this method produces much cleaner masks, as it can be seen in Fig. 9.

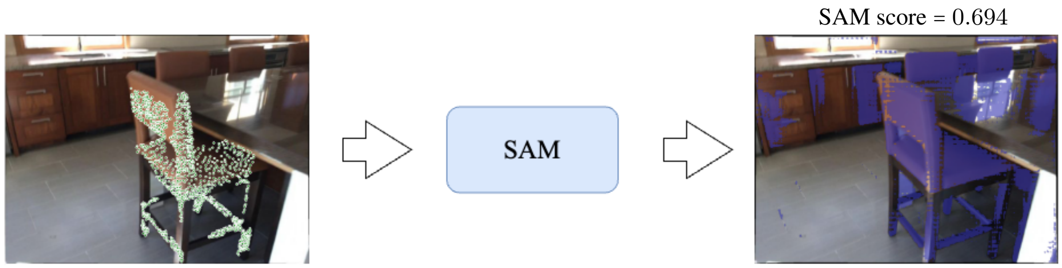



**Figure 8:** Output of SAM, using all of the visible points from the projected 3D mask as input.

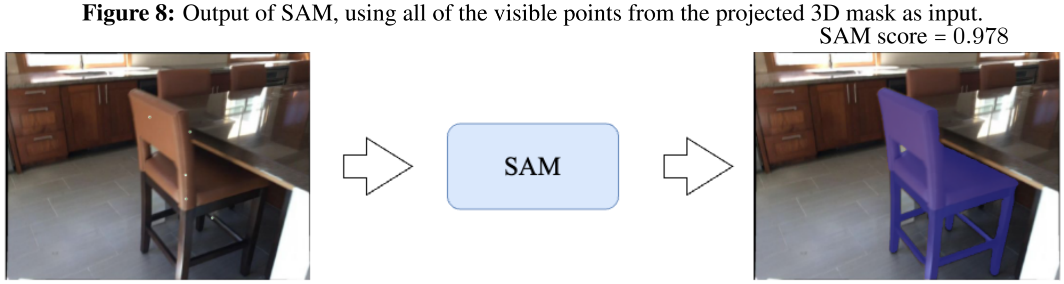



**Figure 9:** Output of SAM, using only 5 randomly sampled points (visualized as green dots) of the projected 3D mask as input.

### A.2.4 Why do we need to run SAM for multiple rounds?

Relying solely on a small set of random points as input for SAM might be unreliable, particularly, when the sampled points are outliers, or too concentrated in a particular area of the instance we want to segment.

To address this limitation, we implemented a sampling algorithm (as outlined in the main paper) inspired by RANSAC [16]. In this approach, we perform $k_{round}$ = 10 sampling rounds, and in each round we randomly sample $k_{sample}$ = 5 points from the projected 3D instance mask. SAM returns the predicted 2D mask along with the corresponding mask confidence score for each set of sampled points. Based on the confidence scores returned by SAM, we select the mask with the highest score. Fig. 10 illustrates an example where the points sampled in one round are concentrated in a small spatial range, resulting in an incorrect mask prediction by SAM.

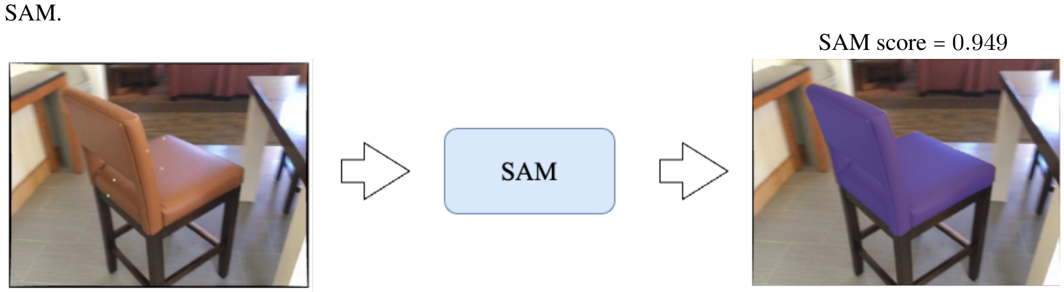



**Figure 10:** Output of SAM, using only 5 randomly sampled points of the mask as input. Here the sampled points (the green points visualized in the image) are concentrated in a small spatial range and cause SAM to predict a wrong mask, which does not include the legs of the chair.

However, by employing our sampling algorithm and running SAM for multiple rounds, we achieve improved results. The iterative process allows us to select the mask with the highest score among the rounds, as shown in Fig. 11. In this particular example, the selected mask, i.e. the one with the highest SAM confidence score, accurately segments the chair.

## B Details on baseline experiments

### Quantitative baseline experiments

As mentioned in the main paper, to compare our OpenMask3D method with the 3D open-vocabulary scene understanding approach OpenScene [49], we adapted OpenScene for the 3D instance segmentation task. As

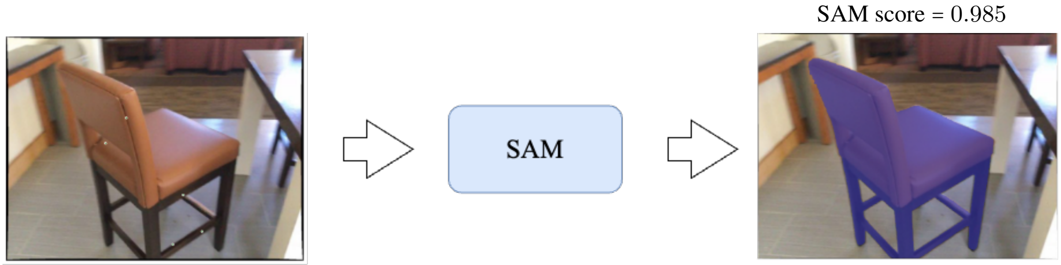



**Figure 11:** Illustration of the results from the round which gives the highest confidence score. On the left, we visualize the 5 sampled points which are given as input to SAM. On the right, we visualize the SAM mask prediction, showing a perfect segmentation of the chair.

OpenScene computes a per-point feature representation for each point in the point cloud, we use our proposed class-agnostic instance masks to aggregate OpenScene features for each mask. This way, we can associate each object instance mask with a "per-mask" feature computed from OpenScene features.

OpenScene provides 3 different model variants, namely 2D fusion, 3D distill, and 2D/3D ensemble. We primarily compare with the OpenScene models using OpenSeg [18] features, whereas we also experiment with the 2D fusion model using LSeg [39] features. The main reason why we primarily compare against the models using OpenSeg features is that the features have a dimensionality of 768, relying on the same CLIP architecture we employ in our work – ViT-L/14 (336px) [52] –, whereas the LSeg features have a dimensionality of 512. For our closed-vocabulary experiments with the OpenScene model using LSeg features, we embed each text category using ViT-B/32 CLIP text encoder, and we assign object categories using these embeddings. In all other experiments, we embed text queries using the CLIP architecture ViT-L/14 (336px).

### Qualitative baseline experiments

In addition to the quantitative results presented in the main paper, we conducted qualitative comparisons to further evaluate our mask-based open-world representation in comparison to the point-based representation provided by OpenScene [49].

In Fig. 12, we provide these qualitative comparisons. We show results from the OpenScene [49] 2D fusion model using OpenSeg [18] features, and from OpenMask3D. To visualize results from OpenScene, for a given query, we compute the similarity score between each point-feature and the query text embedding, and visualize the similarity scores for each point as a heatmap over the points. On the other hand, OpenMask3D results are based on mask-features, i.e., each mask has an associated feature vector, for which we compute a similarity score. Hence, we visualize the similarity scores for each mask as a heatmap over the instance masks, and each point in a given instance mask is assigned to the same similarity score.

As evident from Fig. 12, while similarity computation between the query embedding and each OpenScene per-point feature vector results in a reasonable heatmap, it is challenging to extract object instance information from the heatmap representation. Our method OpenMask3D, on the other hand, computes the similarity between the query embedding and each per-mask feature vector. This results in crisp instance boundaries, particularly suitable for the use cases in which one needs to handle object instances.

## C Qualitative results

In this section, we provide additional qualitative results.

In Fig. 13, we show results from object categories that are not present in the ScanNet200 label set. Objects from these novel categories are successfully segmented using our OpenMask3D approach. Thanks to its zero-shot learning capabilities, OpenMask3D is able to segment a given query object that might not be present in common segmentation datasets on which closed-vocabulary instance segmentation approaches are trained.

In Fig. 14, we show open-vocabulary 3D instance segmentation results using queries describing various object properties such as affordances, color, geometry, material type and object state. This highlights that our model is able to preserve information about such object properties and go beyond object semantics, contrary to the capabilities of existing closed-vocabulary 3D instance segmentation approaches.

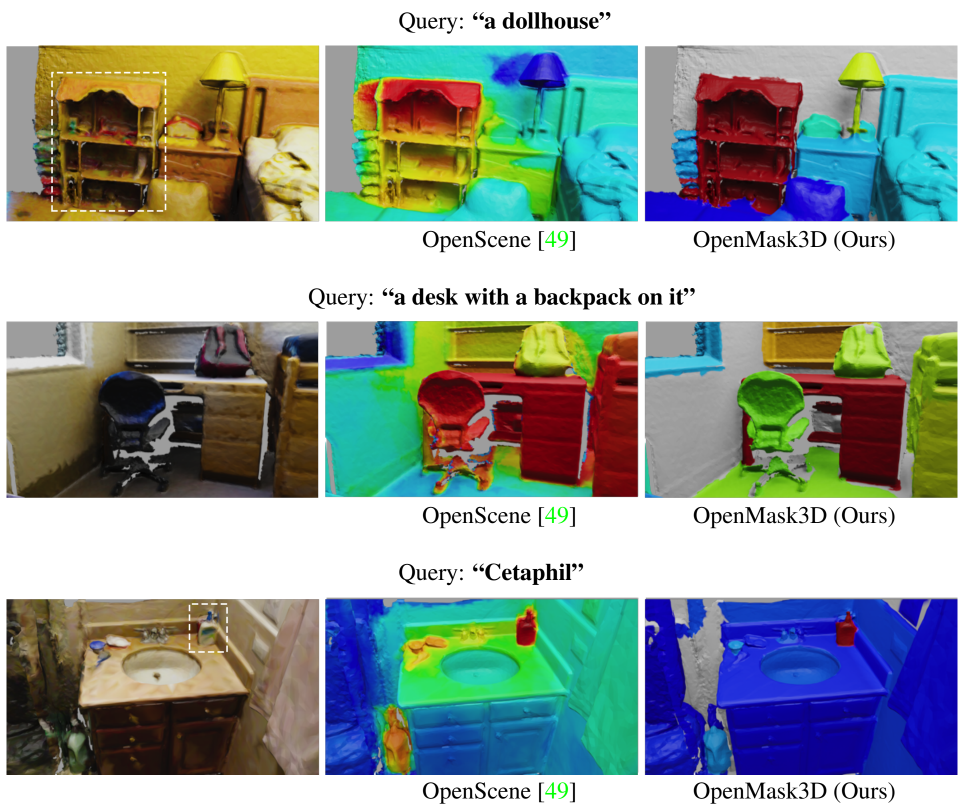



**Figure 12: Qualitative comparisons.** Heatmaps illustrating the similarity between the CLIP embeddings of a specific query and the open-vocabulary representation of the scene. A comparison is made between the point-based OpenScene approach proposed by Peng et al. [49] (left) and our mask-based approach OpenMask3D (right). Dark red means high similarity, and dark blue means low similarity with the query.

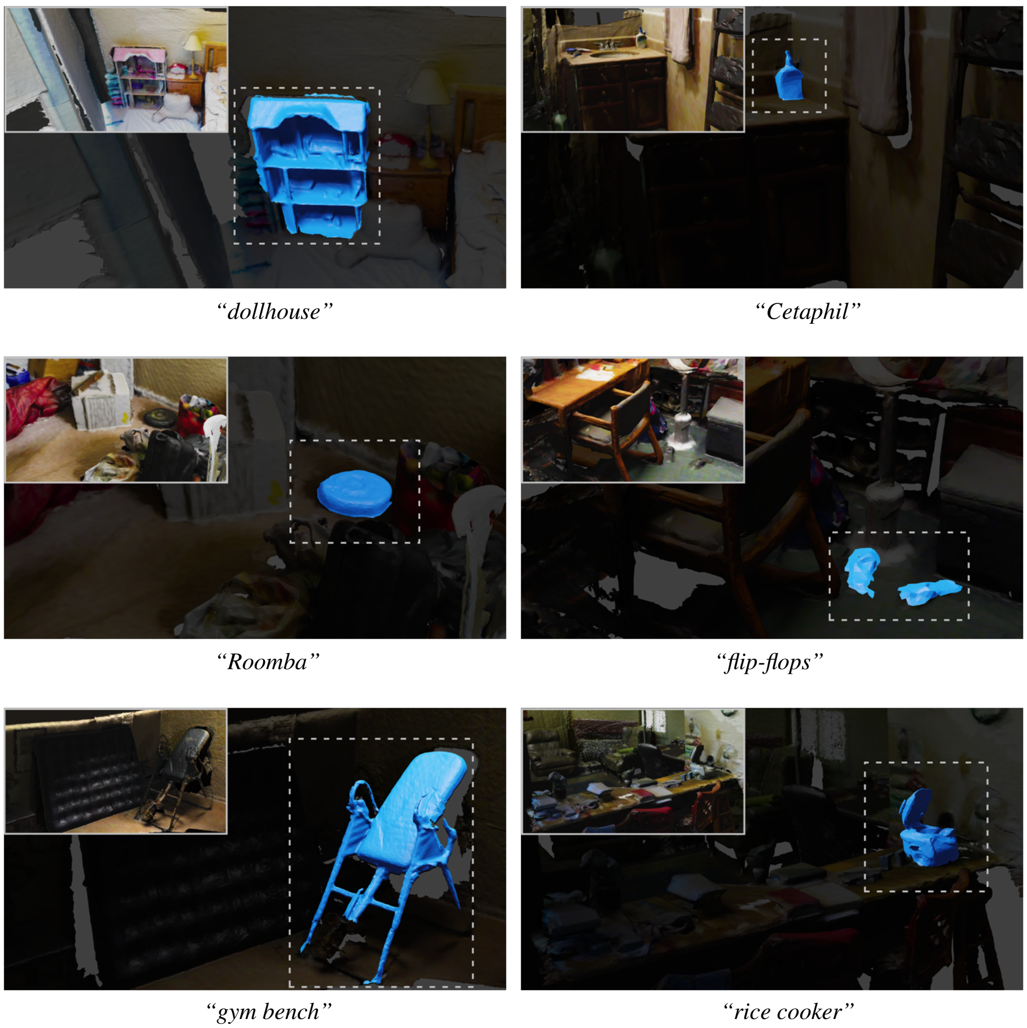



**Figure 13: Qualitative results from OpenMask3D.** We show open-vocabulary instance segmentation results using arbitrary queries involving object categories that are not present in the ScanNet200 dataset labels. In each scene, we visualize the instance with the highest similarity score for the given query embedding. These predictions show the zero-shot learning ability of our model, highlighting the open-vocabulary capabilities.

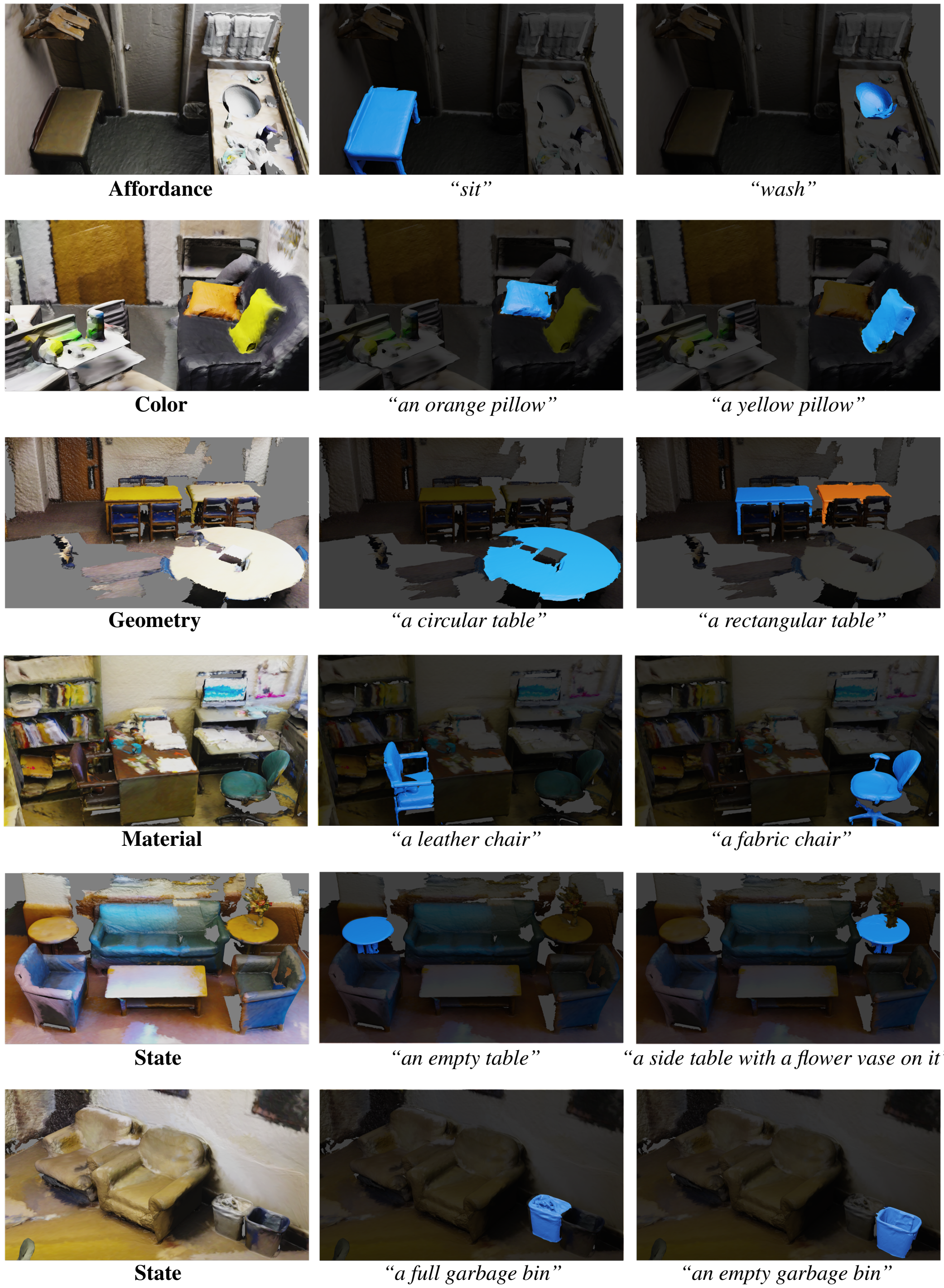



**Figure 14: Qualitative results from OpenMask3D, queries related to object properties.** We show open-vocabulary 3D instance segmentation results using queries describing various object properties such as affordances, color, geometry, material type, and object state. In each row, we show two different queries per category, for which our method successfully segments different object instances. These results highlight the strong open-vocabulary 3D instance segmentation capabilities of our model, going beyond object semantics.