OpenM3D: Open Vocabulary Multi-view Indoor 3D Object Detection without Human Annotations

Peng-Hao Hsu[1]*, Ke Zhang[2]*, Fu-En Wang[2], Tao Tu[3], Ming-Feng Li[4], Yu-Lun Liu[5],
Albert Y. C. Chen[2], Min Sun[1,2]†, Cheng-Hao Kuo[2]†
[1]National Tsing Hua University [2]Amazon [3]Cornell University
[4]Carnegie Mellon University [5]National Yang Ming Chiao Tung University

Abstract

Open-vocabulary (OV) 3D object detection is an emerging field, yet its exploration through image-based methods remains limited compared to 3D point cloud-based methods. We introduce OpenM3D, a novel open-vocabulary multi-view indoor 3D object detector trained without human annotations. In particular, OpenM3D is a single-stage detector adapting the 2D-induced voxel features from the ImGeoNet model. To support OV, it is jointly trained with a class-agnostic 3D localization loss requiring high-quality 3D pseudo boxes and a voxel-semantic alignment loss requiring diverse pre-trained CLIP features. We follow the training setting of OV-3DET where posed RGB-D images are given but no human annotations of 3D boxes or classes are available. We propose a 3D Pseudo Box Generation method using a graph embedding technique that combines 2D segments into coherent 3D structures. Our pseudo-boxes achieve higher precision and recall than other methods, including the method proposed in OV-3DET. We further sample diverse CLIP features from 2D segments associated with each coherent 3D structure to align with the corresponding voxel feature. The key to training a highly accurate single-stage detector requires both losses to be learned toward high-quality targets. At inference, OpenM3D, a highly efficient detector, requires only multi-view images for input and demonstrates superior accuracy and speed (0.3 sec. per scene) on ScanNet200 and ARKitScenes indoor benchmarks compared to existing methods. We outperform a strong two-stage method that leverages our class-agnostic detector with a ViT CLIP-based OV classifier and a baseline incorporating multi-view depth estimator on both accuracy and speed.

1. Introduction

Thanks to the recent breakthrough of Vision-Language Models (VLMs) [9, 17, 38], general representations aligned across the 2D image and free-form-text spaces have become available. These VLMs demonstrate impressive generalization ability to zero-shot object classification tasks. A line of work [10, 11, 54] combines existing class-agnostic 2D object proposals with the zero-shot ability of VLMs to classify 2D object proposals into a large number of object classes. These methods open the door for Open-Vocabulary (OV) 2D object detection and segmentation, handling free-form text descriptions for objects at inference time. For robotics applications, another line of work explores OV 3D indoor scene understanding [13, 16, 29, 33, 46] based on lifting image features from VLMs to 3D. However, all these methods require high-quality 3D point cloud as inputs. This reliance on expensive 3D sensors (*e.g.*, depth cameras, stereo cameras, or laser scanners) is the bottleneck. On the other hand, for fixed-vocabulary, several multi-view image-based methods [42, 50, 53] have achieved significantly improved 3D object detection performance. Unlike point cloud-based methods, image-based methods do not require expensive 3D sensors at inference time.

We propose OpenM3D, a novel OV multi-view indoor 3D object detector that trained without human annotations. To the best of our knowledge, this is the first work generalizing the OV capability to multi-view 3D object detection. OpenM3D is a single-stage 3D detector adapting the 2D-induced voxel features from the ImGeoNet model. The voxel feature is aggregated from multiple RGB features. To support OV, we need to enable it to localize all objects and classify them according to OV descriptions. Hence, it is jointly trained with a class-agnostic 3D localization loss requiring high-quality 3D pseudo boxes and a voxel-semantic alignment loss requiring diverse pretrained CLIP features. We follow the training setting of OV-3DET where posed RGB-D images are given but no human annotations of 3D boxes or classes are available. We proposed a 3D Pseudo Box Generation method using a graph embedding technique that combines 2D segments into coherent 3D structures during training (See Fig. 1). Specifically, we apply SAM [19] on multi-view images to obtain class-agnostic 2D segments.

*Equal contribution.
†Equal advisory contribution.

By treating each segment as a node and computing the relation between nodes according to their connectivity in 3D, we formulate the 3D pseudo bounding boxes generation as a novel graph embedding-based clustering problem so that segments connected in 3D are clustered into a 3D object instance. Our pseudo boxes achieve higher precision and recall than other methods, including those proposed in OV-3DET. We further sample diverse CLIP features from 2D segments associated with each coherent 3D structure to align with the corresponding voxel feature. The key to training a highly accurate single-stage detector requires both losses to be learned toward high-quality targets. While depth is utilized during pseudo-box generation and training, it is not needed for inference. At inference time, OpenM3D is a single-stage OV 3D object detector that requires only multi-view RGB images as input and runs in 0.3 seconds per scene on a V100 GPU. In contrast, most 3D scene understanding methods necessitate point clouds or depth information, as well as the large CLIP ViT model to be applied, leading to significantly higher computational costs. For example, OV-3DET [29] takes 5 seconds per scene, while OpenMask3D [46] requires 5–10 minutes per scene.

We evaluate OpenM3D on ScanNet200 [40] and ARKitScenes [2]. Our 3D pseudo-boxes achieve higher accuracy than those from OV-3DET [29] and SAM3D [58] by jointly considering 2D segments across all views and employing graph embedding-based clustering to mitigate frame-wise errors. OpenM3D also outperforms its counterparts trained with OV-3DET's and SAM3D's 3D boxes in both class-agnostic and multi-class 3D object detection on ScanNet200, demonstrating the effectiveness of our 3D pseudo-boxes. Moreover, OpenM3D with 3D voxel representation surpasses a strong two-stage baseline that classifies objects using 2D CLIP ViT features on both datasets, validating the contribution of Voxel-Semantic feature alignment. We also compare against a multi-view depth estimation baseline, which first estimates depth, applies graph embedding-based clustering for 3D box proposals, and classifies objects using 2D CLIP ViT features. This approach is at least 270 times slower due to depth estimation and CLIP ViT inference, while OpenM3D achieves superior mAP and mAR.

The contributions of our work are the following.

- OpenM3D is the first multi-view open-vocabulary 3D object detector achieving SoTA accuracy on ScanNet200 and ARKitScenes.
- A novel Voxel-Semantic Alignment loss is proposed to align 3D voxel features with multi-view CLIP embeddings. This loss enables open-vocabulary classification by aggregating diverse CLIP features from multiple viewpoints, capturing different object appearances across angles.
- OpenM3D is then trained jointly with both localization and alignment losses as a single-stage detector running 0.3 seconds per scene on V100.
- For localization loss supervision, we propose a novel 3D pseudo box generation pipeline that leverages graph embedding to integrate 2D segments into a coherent 3D structure, surpassing existing methods in experiments.

2. Related Work

3D Object Detection.3D object detection in indoor scenes has gained more research attention due to the availability of datasets with ground truth 3D bounding boxes [2, 5]. When the 3D point cloud is available at inference time, two types of methods are proposed to leverage the 3D geometric information. Point-based methods directly sample based on set abstraction and feature propagation [32, 35–37, 43, 45, 59, 60, 63], while grid-based methods are based on grid representation [7, 12, 21, 30, 41, 44, 55, 56, 66]. Despite the fact that point cloud-based methods perform well on object detection, they rely on costly 3D sensors, which narrows down their use cases.

Multi-View 3D Object Detection. When 3D point clouds are not available at inference time, several other methods can leverage multi-view RGB images for 3D object detection. DETR-based approaches [25, 49, 52] expand upon the capabilities of DETR [4] to tackle the challenge of 3D object detection. Previous studies [15, 23] have established the effectiveness of the bird-eye-view (BEV) representation for object detection in autonomous driving scenarios. Another approach focuses on constructing 3D feature volumes from 2D observations. ImVoxelNet [42] achieves strong indoor 3D object detection using a voxel-based feature volume [31], but struggles to preserve the intrinsic scene geometry. NeRF-Det [53] addresses this by integrating NeRF to estimate 3D geometry while minimizing latency through geometry priors and a shared MLP for a geometry-aware volume. Concurrently, ImGeoNet [50] introduces a geometric-shaping component that predicts surface structure from multiple RGB images in the feature volume and enhances geometric precision. In this work, we build our open-vocabulary multi-view 3D object detector on top of the geometric-shaped 3D feature volume introduced in ImGeoNet.

2D OV Detection. Open-vocabulary object detection (also known as zero-shot detection) is the task of detecting novel classes for which no training labels are provided [11, 39, 61]. Recent methods [11] employ image-text pairs to extract rich semantics from text, thus expanding the number of classes of the detector. However, the detector classes will be fixed after training. Another solution is to replace the classifier with pre-trained vision-language embeddings [39, 61], allowing a detector to utilize an OV classifier and perform OV detection.

3D OV Detection. PointCLIP [62] accomplishes OV recognition of point clouds by projecting them into multi-view images and processing these images with CLIP [38]. However, this method cannot be directly applied to point-cloud detection because it does not handle the localization of un-

known objects. Recently, OV-3DET [29] proposed a 3D point cloud-based 3D object detector learning to align point cloud-based feature with pre-trained CLIP [38] feature space. They leverage a large-scale pre-trained external OV-2D detector [65] to generate 3D pseudo boxes for potential novel objects. Since OV-3DET applies OV-2D detector at each view to generate 3D pseudo boxes, there are a large number of overlapping 3D boxes compared to our proposed method. Moreover, CoDA [3] tackles 3D OV detection in a different setting. It assumes a set of base classes are available with ground truth 3D boxes. Then, an iterative novel object discovery and model enhancement procedure is proposed. Hence, we compare with 3D pseudo boxes from OV-3DET rather than CoDA due to differences in training settings. ImOV3D [57] mitigates the scarcity of annotated 3D data in OV 3D object detection by generating pseudo-multimodal representations from 2D images to bridge the modality gap with 3D point clouds. However, all these methods rely on 3D point clouds during inference, whereas our proposed method is a multi-view image-based 3D object detector.

3D OV Scene Understanding. Beyond 3D OV detection, many works in 3D scene understanding have been recently proposed. OpenScene [33] is the seminal work aligning the representation of 3D points with CLIP features in the posed images from back-projection. However, it does not support the output of object 3D box or 3D segment explicitly. OpenMask3D [46] is designed for 3D instance segmentation. It projects 3D instance mask proposals to 2D posed images and refines them with SAM. Label predictions are made by comparing the CLIP features on visual masks and text prompts. However, both OpenScene and OpenMask3D require point clouds and CLIP model computation for inference. LeRF [18] fuses multi-scale CLIP features extracted from 2D multi-view images into a neural radiance field for OV queries. Although no point clouds are required, an extra scene reconstruction step and CLIP model computation are required at inference time. In comparison, OpenM3D is an efficient single-stage OV 3D object detector only requiring multi-view images as input and runs 0.3 seconds per scene during inference.

3. Preprocess: 3D Pseudo Box Generation

Our training follows OV-3DET [29], eliminating manual 3D annotations by leveraging 2D vision and vision-language models to associate class information with posed RGB-D images. To train a class-agnostic detector with high recall and precision, we generate 3D pseudo-boxes from these images by (1) lifting 2D segments into partial 3D segments using a class-agnostic 2D segmenter and (2) merging them across viewpoints via a graph embedding-based method to form complete 3D segments. Fig. 1 illustrates the proposed 3D pseudo-box generation process.

Partial 3D Segments (Fig. 1 (a,b)). Given an RGB image I, we extract the 2D segments n_j^{2D} of the whole scene $S^{\text{2D}} = \{n_j^{\text{2D}}\}$ using an off-the-shelf class-agnostic 2D instance segmentation approach, where each 2D segment contains a set of pixel positions $\{(u, v)_q\}$. Subsequently, we lift each 2D segment into 3D based on the provided camera pose $(\boldsymbol{R}, \boldsymbol{t})$, intrinsic $\boldsymbol{K}$, and depth map $\boldsymbol{D}$ by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \boldsymbol{R}^T \boldsymbol{K}^{-1} \boldsymbol{D}(u, v) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - \boldsymbol{R}^T \boldsymbol{t} \,. \tag{1}$$

We denote the result of Eq. 1 as a partial 3D segment $n_j^{\text{3D}} = \{(x, y, z)_q\}$, comprising numerous 3D points (x, y, z), as it only includes partial observation from a single view. To incorporate information from multiple viewpoints, we explore methods for aggregating 3D partial segments, though this process is complicated by noise from imperfect 2D segments. The goal is to merge these noisy partial segments into a more robust 3D representation. While aggregating partial segments across views is intuitive for completing a 3D object's surface, achieving this without amplifying noise and bias remains challenging.

Complete 3D Segments (Fig. 1 (c,d,e)). A simple sequential aggregation of partial segments accumulates errors due to incomplete object understanding. It relies on limited consecutive frames from a previous time step, resulting in inherent noise. Hence, we propose a graph embedding-based method that considers all viewpoints simultaneously. The scene is represented as a graph, with each node as a partial 3D segment n_j^{3D}, and edges indicating a high likelihood on nodes of the same object.

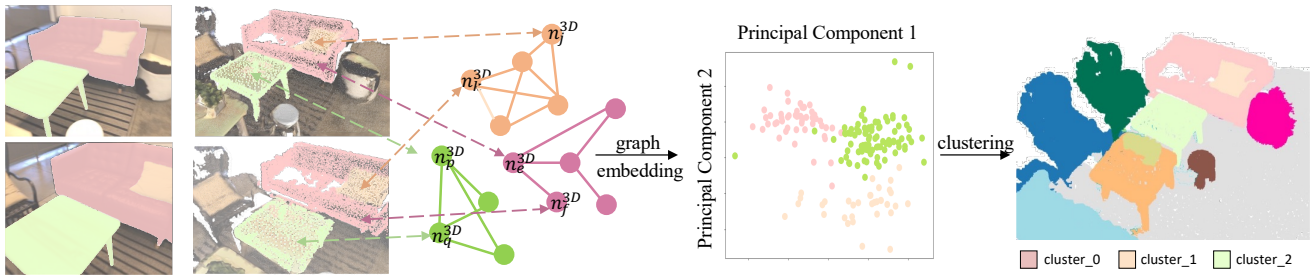
To learn such graph representation, we apply an off-the-shelf graph embedding method on the graph data that considers the entire scene. For a pair of nodes (*i.e.*, partial segments) in this graph data, an edge is established when the overlapping ratio between two nodes exceeds a specific threshold θ as follows:

$$e_{jk} = \text{edge}(n_j^{\text{3D}}, n_k^{\text{3D}}) = \begin{cases} 1, & \text{if } O(n_j^{\text{3D}}, n_k^{\text{3D}}) > \theta, \\ 0, & \text{else} \end{cases} , \tag{2}$$

$$O(n_j^{\text{3D}}, n_k^{\text{3D}}) = \frac{|n_j^{\text{3D}} \bigcap n_k^{\text{3D}}|}{\min(|n_j^{\text{3D}}|, |n_k^{\text{3D}}|)} \tag{3}$$

where $|\cdot|$ counts the number of points in a set, and $\bigcap$ denotes intersection of two sets. We only consider the node pairs within the same voxel for efficiency. The final graph shows the interconnections between nodes (*i.e.*, overlapping partial 3D segments) across the entire scene from multiple views.

After obtaining node features from the off-the-shelf graph embedding method, we generate complete 3D segments that account for partial object segments from all viewpoints by grouping similar nodes (*i.e.*, partial segments) using K-Means. We then form a complete 3D segment by collecting all partial segments in the same cluster q as follows



(a) 2D Segments (b) 3D Partial Segments (c) Build Graph (d) Embedding Feature (e) Complete 3D Segments

Figure 1. **Graph Embedding-Based 3D Pseudo Box Generation.** Given multi-view images, we utilize an off-the-shelf class-agnostic 2D instance segmentation approach to produce 2D segments $S^{2D} = \{n_j^{2D}\}$ (see color-coded 2D masks in (a)). Each 2D segment is then lifted in 3D to form a partial 3D segment n_j^{3D} following Eq. 1 (see (b)). Next, we build a graph in which each partial segment n_j^{3D} is a node and we determine edges based on the overlap of segments in 3D following Eq. 2 (see (c)). The graph embedding feature is computed for each node based on the graph (see (d)). Finally, nodes are clustered with the embedding features to yield complete 3D segments (see (e)). *Best viewed in color.*

$\hat{n}_q^{3D} = \{n^{3D} \in \mathcal{C}_q\}$, where $\mathcal{C}$ is the set of partial 3D segments that share the same segment index of a clustered group q.

Mesh Segmentation Refinement. Besides our complete 3D segments derived from multi-view images, we can further consider the 3D segments $S^{\text{mesh}} = \{n_j^{\text{mesh}}\}$ that are generated by a mesh-based segmentation method. Using ground truth mesh as input, we apply an off-the-shelf graph cut method [8] to generate an additional set of 3D segments S^{mesh}. To fuse these two kinds of 3D segments, that is $\{n_j^{\text{mesh}}\}$ and $\{\hat{n}_q^{3D}\}$, we apply Eq. 3 to determine the overlapping ratio between any pairs of two segments from images and the mesh. For each 3D segment n_j^{mesh} from the mesh, we identify its overlapped 3D segment from our complete 3D segments $\hat{n}_h^{3D}$ with the highest overlapping ratio, and subsequently update the segment index in n_j^{mesh} from j to h. By updating segment indices and combining 3D segments with the same segment index, we fuse the over-segmentation of the mesh to refine the original complete 3D segments back-projected from multi-view.

3D Boxes from Complete 3D Segments. To derive the axis-aligned 3D bounding box that encompasses each complete 3D segment $\hat{n}_q^{3D}$, we calculate the box center (x, y, z) as the mean 3D position of the 3D segment in each axis direction, and the minimum and maximum coordinates in each direction to derive the width w, length l, and height h of the 3D box. Additionally, we apply thresholding on the volume of each 3D box and the number of points contained within each box to remove abnormally small or less visible boxes.

4. OpenM3D

OpenM3D, our multi-view 3D object detector, is trained using posed RGB-D images from diverse indoor scenes. It employs a class-agnostic 3D localization loss (Sec.4.1) supervised by 3D pseudo boxes and a voxel-semantic feature alignment loss (Sec.4.2) to enable OV classification by aligning voxel features with pre-trained CLIP features. During inference, only RGB images and their corresponding camera poses are required.

4.1. Class-Agnostic 3D Localization Loss

Given the 3D pseudo boxes with no class labels, we train a class-agnostic multi-view 3D object detector based on the model architecture of ImGeoNet [50]. We recap the model architecture and introduce the learning target below.

The input to the class-agnostic detector consists of a sequence of images $\boldsymbol{I}_t$ along with camera intrinsics $\boldsymbol{K}$ and extrinsics $\boldsymbol{P}_t$. We back-project the 2D features of these images to construct a 3D feature volume $\boldsymbol{V} \in \mathbb{R}^{H_v \times W_v \times D_v \times C}$, where H_v, W_v and D_v represent the height, width, and depth of the 3D volume in terms of the voxel size unit. The channel dimension of each voxel in the 3D volume is denoted as C. Each voxel feature is further weighted according to the probability of that voxel being located on an object's surface to incorporate geometry-shaping information. We define the likelihood of the voxel being on an object surface as the geometry shaping volume $\boldsymbol{S}$, and the geometry shaping network $geo(\cdot)$ to generate $\boldsymbol{S} = geo(\boldsymbol{V}^{'})$. Here we concatenate feature variance with the feature volume $\boldsymbol{V}$ to obtain $\boldsymbol{V}^{'}$. Note that $\boldsymbol{S}$ shares the same volume size as the original feature volume $\boldsymbol{V}$. As such, the geometry-aware feature volume is obtained by directly applying the geometry shaping weights from $\boldsymbol{S}$ to the original feature volume ($\boldsymbol{V}_{geo} = \boldsymbol{S} \odot \boldsymbol{V}$) . Furthermore, we add dense 3D convolution layers to the geometry-aware volume to harvest volumes at different scales, which has proven helpful in detecting objects at different sizes [42, 50]: $\{\boldsymbol{V}_h^{(i)} = \text{Conv3D}^{(i)}(\boldsymbol{V}_{geo}) \mid i \in \{0, 1, ..., L-1\}\}$. In total we have L volumes at different scales. A single-stage anchor-free detector is deployed as the detection head, and takes the multiscale feature volume $\boldsymbol{V}_h^{(i)}$ as input. We train each voxel cell to predict the 3D pseudo box location using rotated 3D IoU loss [64] for aligning box center, size, and yaw; centerness using cross-entropy loss [48], which reflects the proximity of the voxel to object centers; and binary

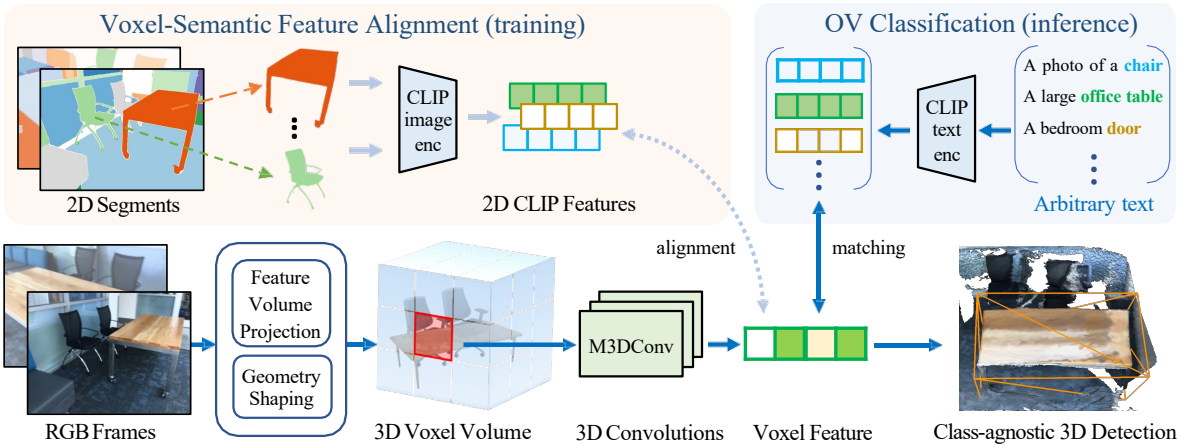


Figure 2. **Overview.** OpenM3D learns class-agnostic 3D box prediction and open-vocabulary (OV) assignments during training and only needs multi-view RGB images to infer OV 3D boxes. The bottom branch is our class-agnostic 3D object detector (Sec. 4.1), where we build the 3D voxel features based on ImGeoNet. During training, given a set of RGBD images and their corresponding poses, we back-project 2D features from images to form the initial 3D voxel volume and perform geometry-shaping on the volume for 3D object localization. The top-left panel is our Voxel-Semantic feature alignment branch (only required in training) to empower OV classification on the 3D voxel features (Sec. 4.2). In the training phase, we use a depth map to match semantic features from 2D segments extracted by CLIP encoder to corresponding 3D voxels and align the semantic features and voxel features. In the inference phase, OpenM3D only requires a set of RGB images and their corresponding camera poses to predict 3D boxes and perform OV classification, as shown in the top-right panel.

foreground probability using focal loss [24] to address the foreground-background imbalance. The complete process of the class-agnostic detector is depicted as the bottom branch in Fig. 2.

4.2. Voxel-Semantic Feature Alignment Loss

The 3D feature volume $\boldsymbol{V}_h$ acquired through class-agnostic 3D object detection encodes rich geometric information for 3D object localization. It still lacks the capability of open-vocabulary classification on 3D objects. To address this limitation, we introduce an OV classification branch atop the class-agnostic 3D object detection branch during training. We propose a novel training loss to minimize the difference between voxel features from $\boldsymbol{V}_h$ and pre-trained CLIP features extracted from the projected multi-view 2D segments (See Fig. 2, Top-Left Panel). We refer to this as the Voxel-Semantic Feature Alignment Loss, leveraging CLIP's ability to capture image-text semantics and bridge the gap between 2D visual concepts and 3D voxel representations. In detail, we first deploy the image encoder of CLIP to extract the embedding $\boldsymbol{f}_j^{2D}$ for each 2D segment n_j^{2D}. Following a similar strategy of Eq.1 in Sec. 3, we lift each 2D segment n_j^{2D} to a partial 3D segment $n_j^{3D} = \{(x, y, z)_q\}$. Each 3D point (x, y, z) is then mapped to a voxel indexed by $H_v \times W_v \times D_v$ in $\boldsymbol{V}_h$. The set of voxel indices for n_j^{3D} is denoted as $V_j^{3D} = \{(x_k, y_k, z_k)_k\}$, and each corresponding voxel feature is denoted as $\boldsymbol{f}_k^{3D} = \boldsymbol{V}_h(x_k, y_k, z_k, :) \in \mathbb{R}^C$. The training objective is to minimize the cosine distance between CLIP embeddings from 2D segment and voxel features from 3D volume for alignment: $\mathcal{L}_{align}(\boldsymbol{f}_\mathbf{j}^\mathbf{2D}, \{\boldsymbol{f}_k^{3D}\}) = \sum_{k \in V_j^{3D}} (1 - \frac{\boldsymbol{f}_\mathbf{j}^\mathbf{2D} \cdot \boldsymbol{f}_\mathbf{k}^\mathbf{3D}}{\|\boldsymbol{f}_\mathbf{j}^\mathbf{2D}\| \|\boldsymbol{f}_\mathbf{k}^\mathbf{3D}\|})$, where the j^{th} 2D segment is aligned with each voxel feature indexed by $k \in V_j^{3D}$.

Inference. OpenM3D simply gets 3D boxes from the class-agnostic 3D detector, and computes softmax over cosine similarity (i.e., matching) between average voxel features within the box and the CLIP text embeddings of text prompts to perform OV classification (See Fig. 2-Top-Right Panel). Since CLIP text embeddings are precomputed, OpenM3D is a single-stage detector not requiring heavy computation of CLIP visual or text features during inference. Contrastingly, most 3D scene understanding methods necessitate the computation of the large CLIP ViT model during inference.

5. Experiments

We first introduce the setup including datasets, evaluation metrics, and baseline methods (Sec. 5.1), and implementation details (Sec. 5.2). In Sec. 5.3, we compare 3D pseudo boxes generated by our proposed OpenM3D to ones by the current state-of-the-art approaches, OV-3DET [29] and SAM3D [58]. For 3D object detection, we show our experimental results in Sec. 5.4. We further conduct ablation studies of our proposed OpenM3D in Sec. 5.5.

5.1. Datasets, Evaluation Metrics, and Baselines

ScanNet200. The ScanNet dataset [5] is a widely used RGB-D video dataset for benchmarking various 3D tasks. There are in total 1,513 room-level scenes, with 2.5 million views. We use 1,201 scenes for training and 312 scenes for testing, which adhere to the public train-val split proposed in ScanNet. Following [37, 50], we generate axis-aligned

bounding boxes based on semantic labels assigned to the 3D mesh of each scene. For the training and evaluation of image-based methods, a uniform sampling of 20 views per scene is performed, guided by frame indices. The images are then standardized to a resolution of 480 $\times$ 640. Rozenberszki *et al*. [40] further extend ScanNet from 18 to 200 object categories, denoted as **ScanNet200**. The 200 categories are further split into 3 subsets, based on the frequency of the number of labeled surface points in the train set, namely head (66 classes), common (68 classes), and tail (66 classes) groups. Our experiment focuses primarily on the ScanNet200 dataset due to the diversity of its categories, aligning well with real-world OV scenarios. Note that there are 189 classes available on the validation set for our evaluation.

ARKitScenes. ARKitScenes [2] provides 5,048 scans collected from 1,661 scenes using Apple LiDAR sensors. These scans contain RGB-D frames along with 3D object bounding box annotations of 17 categories. Due to the limited categories compared to ones in ScanNet200, ARKitScenes is used to evaluate recall rather than precision.

Besides, it is noteworthy that for the 3D object detection task, the point clouds in ARKitScenes are of lower quality compared to ScanNet200, as the depth maps in ARKitScenes are low-resolution (192×256) from iPad Pro.

Evaluation Metrics. We employ typical precision and recall for evaluating 3D pseudo box performance on ScanNet200 training set. Moreover, the average precision (AP) and the average recall (AR) metrics are also applied to measure detection performance. For each class, AP is calculated by computing the area under the precision-recall curve, and AR is computed as the average recall across all intersection over union (IoU) thresholds. More precisely, we utilize AP_{25} and AR_{25}, where the numerical values denote the 3D IoU threshold as 0.25, the minimum IoU required to classify a detection as a positive match to a ground truth box. Consequently, we report mean AP and mean AR across all classes, denoted as mAP_{25} and mAR_{25}.

Baseline - Pseudo 3D Boxes. We generate pseudo-3D boxes from the segmentation results of SAM3D [58], which requires frames from multiple viewpoints as input. We also use OV-3DET [29] generated boxes independently from each frame, which results in a large number of near-duplicated pseudo boxes. Hence, it has low precision and longer training times compared to our approach. To mitigate this issue, we randomly sample the same number of pseudo boxes generated by our method.

Baseline - Strong Two-stage Detector (S2D). Unlike OpenM3D, a single-stage detector simultaneously localizing 3D objects and classifying open-set descriptions of objects, S2D uses OpenM3D to localize candidate 3D boxes. In the second stage, each candidate 3D box is projected back to multi-view 2D images, and the CLIP embeddings are extracted. The averaged embedding over all projected regions is used to match text prompts of each class. Note that this baseline uses CLIP ViT/B-32 during the inference stage which is 7 times slower than our method.

Baseline - S2D using Depth estimated 3D Boxes. We employ a well-trained multi-view depth estimator [1] to extract depth from testing images, and generate 3D bounding boxes by pseudo box generation using estimated depth. The second stage of the Strong Two-Stage approach classifies these boxes, providing an effective solution for open-vocabulary multi-view 3D object detection. However, an inference time of 81 seconds per scene for depth estimation on a V100 GPU is prohibitively long, making it impractical for real-world applications.

5.2. Implementation Details

Class-agnostic 2D Segments and CLIP Embeddings. Given multi-view RGB images, we generate class-agnostic 2D instance segments by SAM [19]. Besides, to mitigate the impact of background elements such as the floor, walls, and ceiling, which can potentially introduce errors in graph embedding due to their substantial spatial presence, we use [22] to filter out such backgrounds in training. We then extract each segment with CLIP image encoder for the embedding. Unless otherwise specified, we utilized ViT/L-14 in our experiments, with minimal impact observed from alternative image encoders like ViT/B-32 and ViT/B-16. For more details, please refer to our supplementary material.

Coordinates Standardization. To address minor variations in 3D point coordinates caused by depth map noise, we standardize coordinates across all 3D partial segments. We employ voxelization and K-nearest neighbors (KNN) to fuse 3D points to vertices extracted from the ground truth mesh. This procedure, involving the fusion of point sets within voxel grids to the nearest extracted vertex through KNN, ensures a unified representation of coordinates in 3D partial segments from diverse viewpoints.

Complete 3D segments and boxes. We construct a graph from partial 3D segments and apply DeepWalk [34] to generate graph embeddings. We then cluster these embeddings using K-means (K=100 for all scenes), grouping similar nodes into clusters that form complete 3D segments. To assign a segment label to each point, we first map every node's cluster index to its associated 3D points. For points shared across multiple nodes, we apply majority voting to determine the final segment assignment. Additionally, a connected-component algorithm is used to separate spatially distant point sets within the same cluster. To ensure completeness, we discard boxes derived from $\hat{n}_q^{3D}$ that contain fewer than 300 and 500 points for ScanNet200 and ARKitScenes, respectively, or have volumes exceeding $8.5, \mathrm{m}^3$, as such boxes are unlikely to represent entire objects.

Model Training. We follow the general configuration of

Table 1. **3D Pseudo Box Evaluation on ScanNet200 and ARKitScenes.** Our boxes, with and without Mesh Segmentation Refinement (MSR), exceed OV-3DET and SAM3D in precision at both IoU thresholds at 0.25(@25) and 0.5(@50) across both datasets. Our bounding boxes outperform OV-3DET in recall significantly and demonstrate competitive performance to SAM3D in most settings. (a) Please refer to the supplementary material for detailed evaluations in different subsets (head, common, tail) on ScanNet200. (b) The * indicates that precision is expected to be low since only 17 classes are labeled in ARKitScenes. Many pseudo boxes are associated with unlabeled objects and counted as false positives.

Method	(a) ScanNet200				(b) ARKitScenes			
	Precision (%)		Recall (%)		Precision* (%)		Recall (%)	
	@25	@50	@25	@50	@25	@50	@25	@50
OV-3DET [29]	11.62	4.40	21.13	7.99	3.74	0.91	32.43	7.93
SAM3D [58]	14.48	9.05	57.70	**36.07**	6.01	1.49	43.78	10.87
Ours w/o MSR	27.09	11.98	52.43	23.18	**6.06**	1.34	51.40	11.41
Ours	**32.07**	**18.14**	**58.30**	32.99	5.97	**1.58**	**51.92**	**13.74**

[50] to train OpenM3D. The 2D feature encoder of the input images I_t is a ResNet-50 [14] pretrained on ImageNet [6]. In Voxel-Semantic feature alignment, we add an MLP layer atop the voxel feature to match the CLIP feature dimension. Our network is trained using AdamW [26] optimizer with an initial learning rate as $1e^{-3}$. Learning rate decay is applied at the 18^{th} and 45^{th} epochs with a decay rate of 0.1, and the network undergoes 50 training epochs.

5.3. 3D Pseudo Boxes

We evaluate our class-agnostic pseudo boxes by comparing them with ground truth boxes using various IoU thresholds. Additionally, we investigate the impact of Mesh Segmentation Refinement (MSR) on our method. The evaluation results for ScanNet200 and ARKitScenes are shown in Table 1. Our bounding boxes demonstrate higher quality compared to OV-3DET and SAM3D in terms of precision at IoU@0.25 and IoU@0.5. For training the detector, pseudo box precision is relatively more important than recall. Our bounding boxes also outperform OV-3DET in recall by a significant margin and remain comparable to SAM3D in most of the settings. This suggests that our pseudo boxes can more effectively capture objects from various viewpoints. OV-3DET generates pseudo boxes by back-projecting results from Detic [65], an OV-2Ddet, into 3D for each 2D frame. This inherent difference between 2D and 3D domains causes OV-3DET to fall short in overall precision and recall compared to our multi-view-aware 3D pseudo box generation method. Unlike SAM3D, which can be affected by 2D segmentation errors due to its local adjacent frame merging, OpenM3D utilizes graph embedding-based clustering to consider frames from all viewpoints simultaneously. This approach reduces the impact of segmentation errors from individual frames, resulting in better pseudo box quality. In conclusion, MSR improves performance on both ScanNet200 and ARKitScenes, though the gains are linked to mesh quality, with a less pronounced effect on ARKitScenes due to its lower mesh quality. These findings indicate that OpenM3D consistently surpasses OV-3DET and SAM3D in pseudo box quality, irrespective of the mesh quality.

Table 2. **Class-agnostic 3D Object Detection on ScanNet200.** Our proposed 3D pseudo boxes enable OpenM3D to achieve higher AP and AR than boxes from OV-3DET and SAM3D.

Method	Trained Box	AP@25(%)	AR@25(%)
OpenM3D	OV-3DET [29]	19.53	35.19
	SAM3D [58]	23.77	47.82
	Ours (w/o MSR)	25.95	48.14
	Ours	**26.92**	**51.19**

Table 3. **3D Object Detection on ScanNet200.** For OpenM3D, different "Trained Boxes" are used in training, while for S2D, different "Candidate Boxes" are applied. Specifically, "S2D+Ours" and "S2D+Depth Estimated" correspond to "Baseline-S2D" and "Baseline-S2D using Depth Estimated 3D Boxes", respectively.

Method	Trained Box / Candidate box	mAP@25(%)	mAR@25(%)
OpenM3D	OV-3DET [29]	3.13	10.83
	SAM3D [58]	3.92	13.33
	Ours (w/o MSR)	4.04	13.77
	Ours	**4.23**	**15.12**
S2D	Depth estimated	3.80	8.60
	Ours	4.17	10.05

Table 4. **3D Object Detection on ScanNetv2.** OpenM3D performs comparably to point-cloud-based methods. This table serves as a reference for comparing different input modalities in inference, including point clouds (pc) and images (im). † indicates methods evaluated with OV-3DET's pseudo-boxes, while our evaluation uses ground-truth 3D boxes from ScanNetv2 in our multi-view setting.

Method	Training Data	Input	Detector	AP@25(%)
OV-3DET† [29]	ScanNet	pc + im	Two-Stage	18.02
CoDA† [3]	ScanNet	pc	One-Stage	19.32
ImOV3D† [57]	ScanNet, LVIS	pc	One-Stage	21.45
Ours	ScanNet	im	One-Stage	19.76

5.4. OV 3D Object Detection Results

Class-Agnostic Scenario. This scenario validates the class-agnostic 3D object detector of OpenM3D in Sec. 4.1. No class information is utilized during inference, focusing solely on accurately predicting foreground bounding boxes. We show the evaluation results of ScanNet200 in Table 2. Compared to OV-3DET [29], our proposed framework can improve AP@25 by 37% (19.53%→26.92%), underscoring the superiority of our pseudo boxes. Given that OV-3DET generates 3D boxes solely based on a single-view RGB image and depth map, there is a risk that the resulting 3D box may deviate significantly from the actual object, thereby influencing the class-agnostic training. Additionally, we compare our framework to SAM3D, which requires multi-view frames as input. Our proposed approach consistently outperforms SAM3D by 13% in AP@25. Furthermore, significant improvements are observed in mAR@25, with our framework

surpassing SAM3D by 3.3% and OV-3DET by 16%.

Open-Vocabulary Scenario. This scenario validates OpenM3D, covering both Sec. 4.1 and Sec. 4.2. We evaluated OV 3D detection on ScanNet200 and report in Table 3. Similar to the trend in Table 2, OpenM3D outperforms our method trained with OV-3DET boxes on both mAP@25 and mAR@25, for 1.1% and 4.3%, respectively, with relative improvements exceeding 30% on the challenging ScanNet200 dataset. Moreover, OpenM3D surpasses models trained with SAM3D pseudo boxes, thereby highlighting the effectiveness of our graph-embedding-based pseudo boxes. Notably, using better segmentation models, *e.g.*, CropFormer [27], OpenM3D achieves a significant 12.5% improvement in mAP@25, from 4.23% to 4.76%. For more details, please refer to the supplementary materials.

To highlight the contribution of our Voxel-Semantic feature alignment, we compare OpenM3D to *S2D*, both utilizing the same class-agnostic foreground detector. As a single-stage detector, our method achieves comparable mAP@25 to *S2D*. However, *S2D* shows a significant drop in mAR@25, from 0.15 to 0.10, indicating that 3D voxel features recall object classes better than multi-view 2D CLIP features. Furthermore, *S2D* requires the CLIP image encoder during inference, introducing high computational costs and a sixfold increase in inference time compared to our framework. On ARKitScenes, our method achieves 42.77 mAR@25, outperforming *S2D* at 19.58 mAR@25, with mAP not reported for the same reason as in 3D pseudo box evaluation.

Table 4 presents the results of OpenM3D on ScanNetv2, demonstrating performance comparable to other point-cloud-based methods. This indicates that OpenM3D, using only 2D images at inference, achieves results on par with methods that rely on 3D data. For additional baseline results on ScanNetv2, please refer to the supplementary material.

Detection results on ScanNet200 and ARKitScenes are shown in Fig. 3. To showcase OV detection ability, we visualized detection results using a subset of text prompts by CLIP on ImageNet, and specific prompts in Fig. 3. OpenM3D consistently detects 3D objects across various classes using general prompts like 'a photo of a large {}', and accurately locates specific objects, such as chairs and small desks, demonstrating its strength in OV 3D object detection.

5.5. Ablation study

Influence of Class Number. To investigate the impact of class number, we evaluate OpenM3D across various numbers of classes (18 to 189), following the "head," "common," and "tail" splits from ScanNet200. As shown in Fig. 4, our method consistently achieves higher mAP and mAR than the "Strong Two-stage" method as the class number increases.

Different Prompts. CLIP [38] indicated that multiple prompts benefit more comprehensive understanding of the desired context, and provided a list of prompts for various

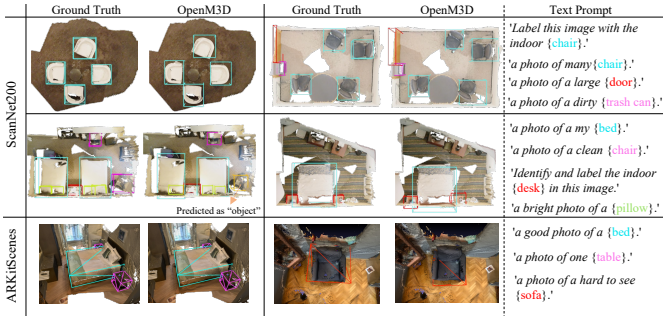


Figure 3. **Qualitative Results of OpenM3D** on ScanNet200 and ARKitScenes. Given multi-view images and corresponding camera poses, OpenM3D can detect objects by arbitrary text prompts towards open-vocabulary detection. The color-coded boxes correspond to different object classes. We show a subset of text prompts used in the ImageNet dataset and *specific prompts*.

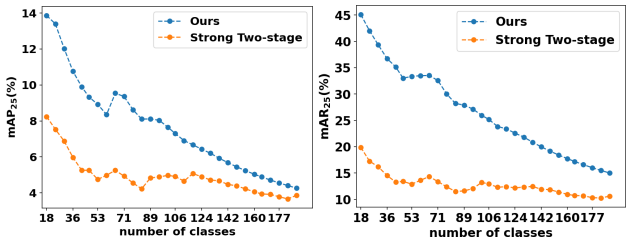


Figure 4. **mAP_{25} and mAR_{25} in various class numbers from 18 to 189 classes.** As the class number gets larger, OpenM3D scores consistently higher mAP (*Left*) and mAR (*Right*) compared with the "Strong Two-stage" method.

datasets depending on the dataset domains. OpenM3D predicts classes by matching the 3D voxel feature with the text embeddings of class names wrapped in multiple prompts, such as "`A photo of a {}`", and selects the category with the highest cosine similarity. We apply the same here and mostly used the prompts of ImageNet [6] dataset. We primarily used ImageNet [6] prompts but also evaluated our model with Cifar100 [20] prompts. The Cifar100 prompts resulted in mAP@25 and mAR@25 values of 4.14 and 14.79, respectively, showing minimal difference in performance . This implies that adjusting the prompts might not significantly improve performance.

6. Conclusion

We introduced OpenM3D, a novel single-stage open-vocabulary multi-view 3D object detector trained without human annotations. It leverages class-agnostic 3D localization and voxel-semantic alignment losses, guided by high-quality 3D pseudo-boxes and diverse CLIP features. We introduce a graph-based 3D pseudo-box generation method achieving superior precision and recall in pseudo-box quality than OV-3DET and SAM3D. At inference, requiring only multi-view images, OpenM3D outperforms a strong two-stage approach and models trained with OV-3DET and SAM3D boxes on ScanNet200 and ARKitScenes, while excelling over an estimated multi-view depth baseline in accuracy and speed.

References

[1] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Multi-view depth estimation by fusing single-view depth probability with multi-view geometry. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 6

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. Arkitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 6

[3] Yang Cao, Yihan Zeng, Hang Xu, and Dan Xu. Coda: Collaborative novel box discovery and cross-modal alignment for open-vocabulary 3d object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3, 7, 16

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 5, 12, 16

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 7, 8

[7] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 2

[8] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 59, 2004. 4

[9] 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)*, 2022. 1

[10] 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)*, 2022. 1

[11] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2

[12] JunYoung Gwak, Christopher Choy, and Silvio Savarese. Generative sparse detection networks for 3d single-shot object detection. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[13] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In *Conference on Robot Learning (CoRL)*, 2022. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[15] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2

[16] 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. In *Robotics: Science and Systems (RSS)*, 2023. 1

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc 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. 1

[18] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 3

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 1, 6, 12, 15, 16

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 8

[21] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[22] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *IEEE Con-*

ference on Computer Vision and Pattern Recognition (CVPR), 2023. 6

[23] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 5

[25] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 7

[27] Qi Lu, Jason Kuen, Shen Tiancheng, Gu Jiuxiang, Guo Weidong, Jia Jiaya, Lin Zhe, and Yang Ming-Hsuan. High-quality entity segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 8, 15, 16

[28] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *7th Annual Conference on Robot Learning*, 2023. 15

[29] Yuheng Lu, Chenfeng Xu, Xiaobao Wei, Xiaodong Xie, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. Open-vocabulary point-cloud object detection without 3d annotation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 3, 5, 6, 7, 12, 15, 16

[30] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[31] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[32] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[33] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 3

[34] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014. 6

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[36] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[37] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 5

[38] 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. 1, 2, 3, 8

[39] Shafin Rahman, Salman Khan, and Nick Barnes. Improved visual-semantic alignment for zero-shot object detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 2

[40] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 6

[41] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: fully convolutional anchor-free 3d object detection. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[42] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *Winter Conference on Applications of Computer Vision (WACV)*, 2022. 1, 2, 4

[43] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[44] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 2

[45] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[46] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2, 3

[47] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. *arXiv preprint arXiv:2412.06974*, 2024. 15

[48] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: A simple and strong anchor-free object detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 4

[49] Ching-Yu Tseng, Yi-Rong Chen, Hsin-Ying Lee, Tsung-Han Wu, Wen-Chin Chen, and Winston Hsu. Crossdtr: Cross-view and depth-guided transformers for 3d object detection. *arXiv preprint arXiv:2209.13507*, 2022. 2

[50] Tao Tu, Shun-Po Chuang, Yu-Lun Liu, Cheng Sun, Ke Zhang, Donna Roy, Cheng-Hao Kuo, and Min Sun. Imgeonet: Image-induced geometry-aware voxel representation for multi-view 3d object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 4, 5, 7

[51] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 15

[52] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning (CoRL)*, 2022. 2

[53] Chenfeng Xu, Bichen Wu, Ji Hou, Sam Tsai, Ruilong Li, Jialiang Wang, Wei Zhan, Zijian He, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Nerf-det: Learning geometry-aware volumetric representation for multi-view 3d object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 1, 2

[54] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[55] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 2

[56] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[57] Timing Yang, Yuanliang Ju, and Li Yi. Imov3d: Learning open-vocabulary point clouds 3d object detection from only 2d images. *NeurIPS 2024*, 2024. 3, 7

[58] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023. 2, 5, 6, 7, 12, 15, 16

[59] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[60] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[61] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[62] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. *arXiv preprint arXiv:2112.02413*, 2021. 2

[63] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[64] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *International Conference on 3D Vision (3DV)*, 2019. 4

[65] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision (ECCV)*, 2022. 3, 7, 15, 16

[66] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

A. Visualization

This section showcases visualizations of 3D pseudo boxes generated by our method, along with additional qualitative results from OpenM3D.

Visualize 3D Pseudo Boxes. The localization capability of our pseudo boxes has been validated in Table 1, 2 of the main

paper. In Fig. 5, we show some examples of our 3D pseudo boxes and their corresponding 3D segmentations. To clearly present our pseudo boxes, we organize them based on two distinct ranges—small and medium—using the volumes of the boxes. Moreover, our 3D pseudo boxes can accurately locate novel objects, as illustrated in Fig. 6, in addition to those annotated in the ground truth. These results validate the localization capability of our generated class-agnostic pseudo boxes for various potential objects in the scene, paving the way for open-vocabulary 3D object detection.

More Qualitative Results. We present more qualitative results of open-vocabulary 3D object detection obtained by OpenM3D in Fig. 7. Some detection results for tail and novel objects are also shown in Fig. 8. With general prompts used in CLIP, OpenM3D demonstrates consistent 3D detections across multiple classes. This strongly showcases OpenM3D's capability in open-vocabulary 3D object detection.

B. Experiment Details

B.1. Implementation Details

Frame Selection for Generating 2D Segments. To achieve fine-grained SAM [19] results for each frame, we aim to automatically choose frames with distinct outlines as SAM inputs for improving segmentation quality. As a result, we utilize Laplacian calculations to determine sharpness as the basis for selecting frames. For every scene in ScanNet200 and ARKitScenes, we divide all frames into intervals based on chronological order. Within each interval, we select the frame with the highest sharpness value. This process was repeated until 300 frames were chosen, iterating through the remaining frames in each round.

2D Segments Filtering and Refinement. When generating 2D segments from an image, we noticed that SAM may generate excessively small segments. Such problematic segments confuse the CLIP image encoder, resulting in poor embedding quality. Multiple small segments may map to the same voxel and further worsen the open-vocabulary classification of OpenM3D. We thus add preprocessing steps to exclude such patches: We set a minimum bounding-box size of 30 pixels, and a 0.02 ratio threshold of observed 3D points within each segment's bounding box.

Given that CLIP is trained using real-world images, our approach involves incorporating the surrounding regions of the bounding box when calculating the CLIP embedding for each 2D segment, to provide a scenario similar to real-world images. In addition to the image patch tightly cropped by the bounding box around each segment, we include patches from areas surrounding the segment with dimensions of 110% and 120% relative to the size of the bounding box. In OV-3DET, Lu *et al*. use a predefined vocabulary with 364 categories for pseudo box generation, we follow the same and use the vocabulary to improve on CLIP segment embeddings. Specifically, for each segment, we compare the embedding of the segment to the text embedding of all categories, and use the embedding of the closest category. Please refer to Alg. 1 for the pseudo code of 3D pseudo box generation.

Algorithm 1: 3D Pseudo Box Generation

Input : RGB images, corresponding pose $(\mathbf{R}, \mathbf{t})$, intrinsic $\mathbf{K}$, and depth map $\mathbf{D}$
Output : 3D Pseudo Boxes b^{3D}

for *each RGB image* I **do**
  $n_j^{2D} \leftarrow$ Segment2D(I)
  $n_j^{3D} \leftarrow$ Backproject(n_j^{2D}, $(\mathbf{R}, \mathbf{t})$, $\mathbf{K}$, $\mathbf{D}$) ; // Eq.1
end
$Nodes := \{n_j^{3D}\}$
$V \leftarrow$ Voxelize($Nodes$) ; // Voxelize based on 3D coordinates of each node
for *voxel in* V **do**
  for *any pair* (n_j^{3D}, n_k^{3D}) *in voxel* **do**
    $e_{jk} := \text{edge}(n_j^{3D}, n_k^{3D})$; // Eq.2
  end
end
$Edges := \{e_{jk}\}$
$Embedding \leftarrow$ GraphEmbed(GenGraph($Nodes$, $Edges$))
$C \leftarrow$ Clustering($Embedding$) ; // Give each node a clustered group
for $\mathcal{C}_q$ *in* C **do**
  $\hat{n}_q^{3D} := \{n^{3D} \in \mathcal{C}_q\}$; // Collect partial segments in the same cluster q
  $b_q^{3D} :=$ AxisAlignedBox($\hat{n}_q^{3D}$)
end
$b^{3D} := \{b_q^{3D}\}$

Voxel and 3D Volume. The feature volume measures 6.4 × 6.4 × 2.56 meters, with a voxel size of 0.16 meters in all three dimensions.

B.2. 3D Pseudo Box

3D Pseudo Box on ScanNetv2. The evaluation result for ScanNetv2 [5] is presented in Table 5. Similar to the performance on ScanNet200, our method consistently outperforms OV-3DET [29] and SAM3D [58] in terms of precision at IoU@0.25 and IoU@0.50, while maintaining a comparable recall with SAM3D. This validates the contribution of the graph embedding-based clustering strategy, which simultaneously considers the 2D segmentation results across all frames. This approach helps mitigate the impact of segmen-

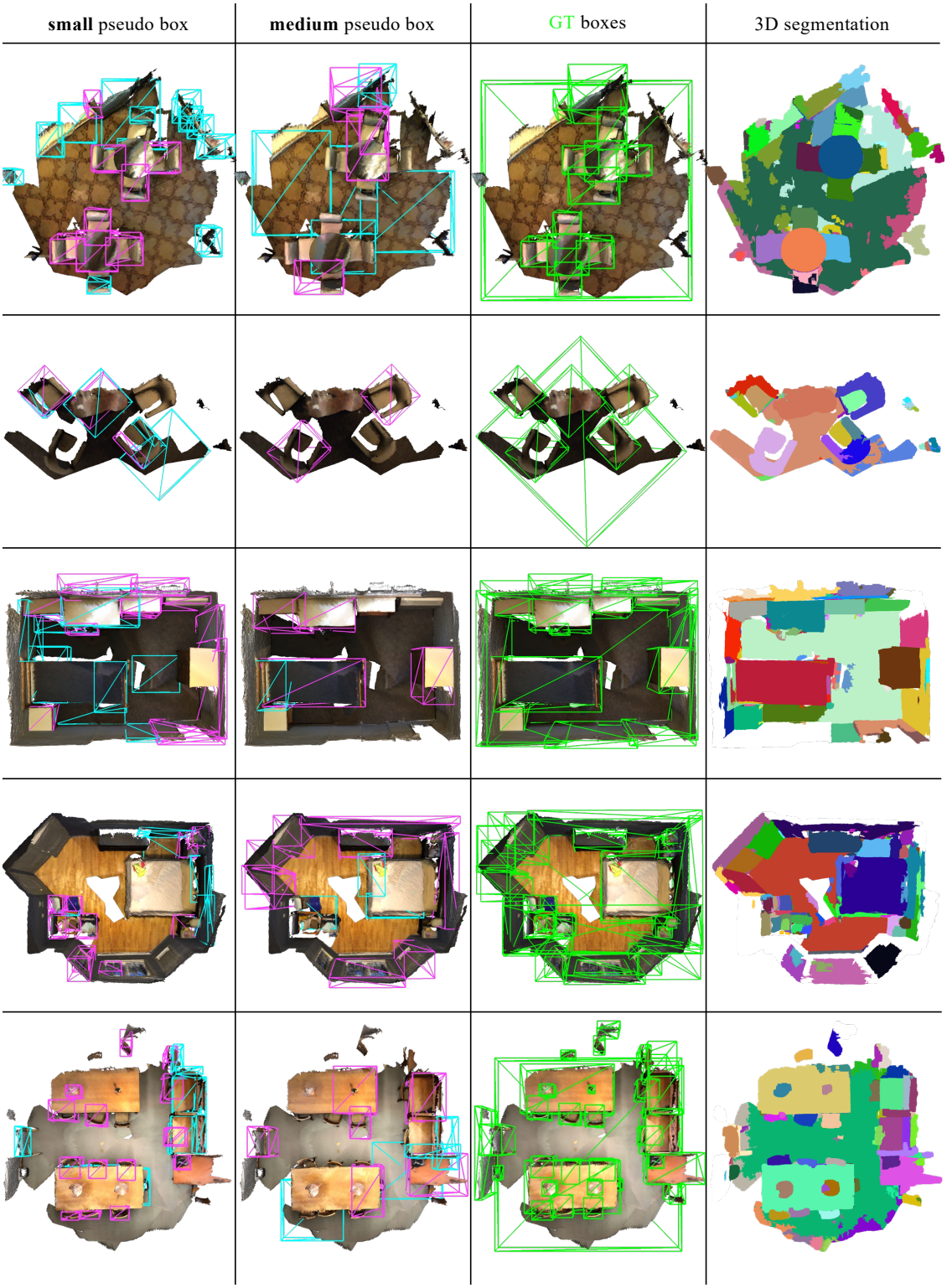


Figure 5. **Visualize Pseudo Boxes of OpenM3D** on ScanNet200. We visualize our 3D pseudo boxes using two different volume sizes (small and medium). In this visualization, cyan represents false positives, while magenta represents true positives matching the GT boxes at IoU@0.25.

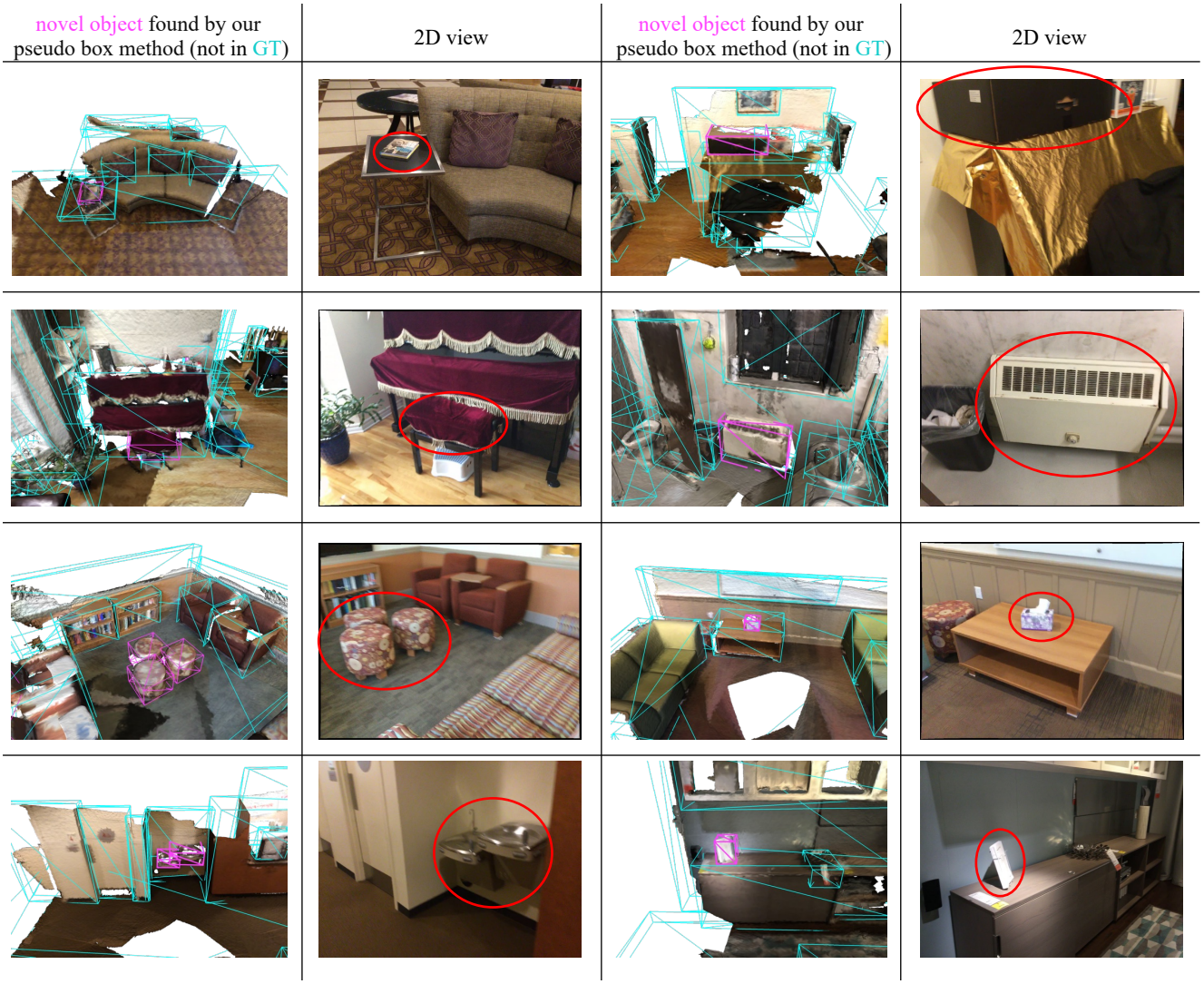


Figure 6. **Localizing Novel Object with Pseudo Box** on ScanNet200.

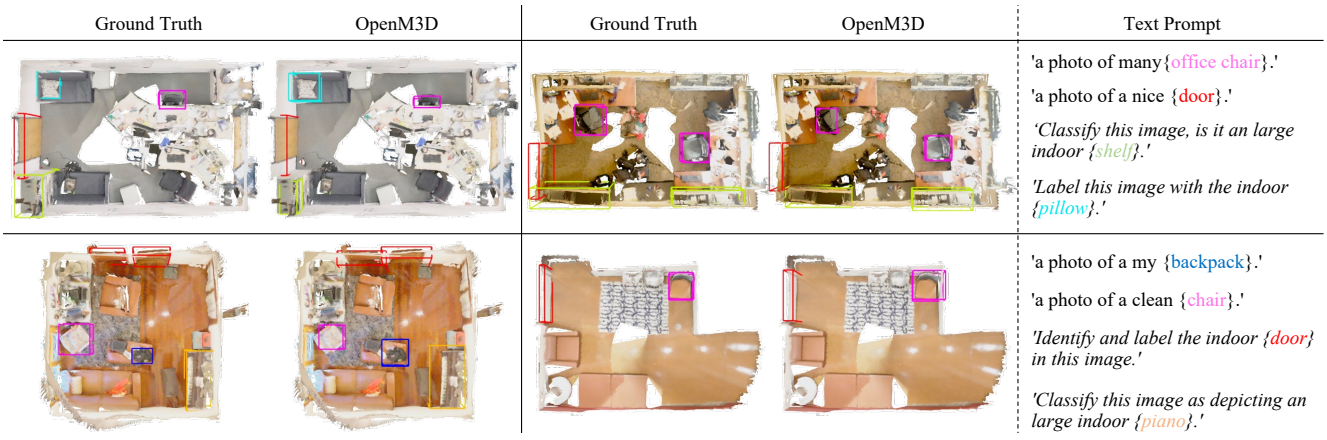


Figure 7. **More Qualitative Results of OpenM3D** on ScanNet200. We show general text prompts used in the ImageNet dataset, as well as prompts from *specific text*.

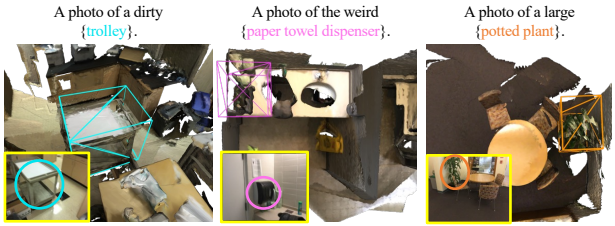


Figure 8. **Novel and tail predictions in OpenM3D**.

Table 5. **3D Pseudo Box Evaluation** on ScanNetv2. Our 3D pseudo boxes demonstrate higher quality compared to OV-3DET and SAM3D in terms of precision at IoU@0.25 and IoU@0.50.

Method	Precision (%)		Recall (%)	
	@0.25	@0.50	@0.25	@0.50
OV-3DET [29]	4.28	0.20	53.14	25.90
SAM3D [58]	7.39	4.94	70.02	**46.76**
Ours w/o MSR	15.81	7.52	72.62	34.56
Ours	**17.11**	**9.91**	**73.84**	42.80

Table 6. **Detailed 3D Pseudo Box Evaluation with different 2D segmentation** on ScanNet200. We perform a comprehensive evaluation across different subsets of ScanNet200. Additionally, we leverage various 2D segmentation sources to generate pseudo boxes. The use of different 2D segmentation sources in our method results in 3D pseudo boxes of varying quality. For example, when CropFormer is applied, these boxes outperform all other methods in terms of precision and recall at IoU@0.25 and IoU@0.50.

Method	2DSeg	Classes	Precision (%)		Recall (%)	
			@0.25	@0.50	@0.25	@0.50
OV-3DET [29]	Detic [65]	overall	11.62	4.40	21.13	7.99
		head	9.59	3.68	20.39	7.82
		common	1.95	0.74	26.12	9.95
		tail	1.06	0.29	24.22	6.77
SAM3D [58]	SAM [19]	overall	14.48	9.05	57.70	36.07
		head	12.54	7.68	58.44	35.81
		common	1.86	1.33	**56.97**	**40.67**
		tail	0.87	0.59	**44.72**	**30.27**
Ours w/o MSR	SAM [19]	overall	27.09	11.98	52.43	23.18
		head	24.26	10.67	54.90	24.14
		common	2.99	1.40	45.15	21.23
		tail	0.86	0.36	20.65	8.68
Ours	SAM [19]	overall	32.07	18.14	58.30	32.99
		head	28.55	16.00	60.68	34.01
		common	3.66	2.20	51.88	31.68
		tail	1.15	0.69	26.30	22.68
Ours	CropFormer [27]	overall	**35.58**	**22.72**	**62.60**	**39.97**
		head	**31.67**	**19.97**	**65.14**	**41.08**
		common	**3.94**	**2.75**	55.07	38.53
		tail	**1.31**	**0.94**	29.77	21.33

tation errors from individual frames.

3D Pseudo Box in Different Subset on ScanNet200. In Table 6, we showcase the detailed 3D pseudo box evaluation in ScanNet200 for different subsets (head, common, tail). The evaluation computed overall precision without considering classes, given our pseudo boxes lack class information. While calculating precision in a certain subset, such as "head," only ground truth boxes in head classes are considered. This may result in a lower head precision than the overall precision, as pseudo boxes overlapping with ground truth common/tail classes contribute to false positives in the head precision calculation.

Moreover, we utilize an advanced image segmentation method, CropFormer [27], to acquire more accurate 3D pseudo boxes. CropFormer's improved object-wise understanding reduces the risk of over-segmentation, enhancing the consistency in 2D views. This improvement benefits our 3D pseudo box generation method, resulting in less noisy 3D segments and more precise refinements. Our method prioritizes pseudo box precision over recall for detector training, resulting in higher precision at IoU@0.25 and IoU@0.5 compared to OV-3DET and SAM3D in each subset. This superior quality is evident in our boxes generated based on both SAM and CropFormer. They also achieve significantly better recall than OV-3DET and remain comparable to SAM3D in most settings.

B.3. Baseline Using *3R methods

Recent 3R methods such as MV-Dust3R [47] and VGGT [51] enable 3D scene reconstruction from RGB images and camera poses without requiring depth, aligning well with OpenM3D's inference setting. To establish a baseline, we implemented a multi-stage pipeline that combines VGGT for 3D reconstruction and OVIR-3D [28] for open-vocabulary instance segmentation on ScanNet200. All components were executed using official implementations and default settings, with 3D boxes computed from axis-aligned segment bounds.

This pipeline incurs substantial computational overhead—particularly during 2D-3D fusion—resulting in an inference time of 300 seconds per scene, compared to 0.3 seconds for OpenM3D. In terms of accuracy, it achieved only 5.97% AP@25 (class-agnostic), significantly lower than OpenM3D's 26.92%. We also observed that VGGT often fails to reconstruct fine-grained indoor geometry (see Fig. 9), which is crucial for accurate 2D-3D matching in instance segmentation—a limitation also noted in the OVIR-3D paper.

Overall, this reconstruction-based pipeline is substantially less effective than OpenM3D in both accuracy and efficiency for open-vocabulary 3D object detection.

B.4. Inference Efficiency

As shown in Table 7. OpenM3D achieves the fastest inference time of 0.3 seconds per scene, using only multi-view RGB images, and significantly outperforms baselines such as OV-3DET (5 s), S2D (2.1 s), and S2D with depth estima-

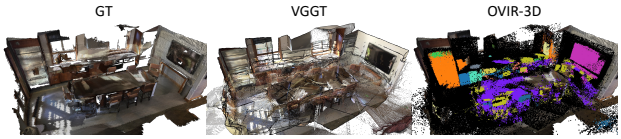


Figure 9. **3R baseline qualitative result.** Comparison between (left) ground-truth ScanNet scene, (middle) VGGT 3D reconstruction using only RGB images and poses, and (right) OVIR-3D segmentation result on the VGGT output. The reconstruction lacks fine-grained indoor geometry, resulting in inaccurate 2D–3D matching and degraded segmentation quality.

tion (81 s). Unlike others, it avoids costly CLIP inference and depth prediction, making it highly suitable for real-time 3D detection.

Table 7. **Inference time comparison** on ScanNet200 on a V100 GPU. OpenM3D is over 16× faster than OV-3DET and 270× faster than the depth-estimated S2D baseline.

Method	**OV-3DET**	**S2D**	**S2D Depth Est.**	**Ours**
Inference time (s)	5	2.1	81	**0.3**

B.5. Transferability of Pretrained Model

OpenM3D does not rely on predefined 'seen' categories or 3D annotations during training, making it naturally OV - all categories are essentially novel. OpenM3D demonstrates strong performance across head, common, and tail classes in ScanNet200 (see Fig. 8), highlighting its ability to handle rare or unseen classes.

B.6. Ablation Study

CLIP Visual Encoders. We aligned our voxel feature to the pre-trained CLIP feature extracted by the ViT-L/14 image encoder during training. Furthermore, we showcase alternative results employing various other CLIP image encoders in this section. As outlined in Table 8, the use of different CLIP image encoders exhibited negligible impact on both evaluation metrics, namely mAP@25 and mAR@25. This observation emphasizes the robust open-vocabulary classification capability of our method, OpenM3D.

3D Detection with Pseudo Box using CropFormer. When deploying better segmentation models, e.g., CropFormer [27], we can generate more accurate pseudo boxes as detailed in Table 6. The improvement on 2D segmentation benefits our 3D pseudo box generation method on 3D segments refinements. Trained with these boxes, OpenM3D demonstrates a notable improvement of 12.5% in mAP@25, rising from 4.23% to 4.76% on ScanNet200, as shown in Table 9. This highlights the potential of our 3D pseudo boxes on better 2D segmentation. Note that in ARKitScenes, given the sparse point cloud, improving 2D segmentation using CropFormer alone has not significantly improved 3D box metric performance.

Table 8. **Results of OpenM3D trained with different CLIP encoders** on ScanNet200.

CLIP Encoder	mAP@25 (%)	mAR@25 (%)
ViT-L/14	4.23	15.12
ViT-B/16	4.16	15.50
ViT-B/32	4.02	14.74

Table 9. **3D Object Detection** on ScanNet200. Our pseudo boxes with CropFormer improve upon SAM.

Method	Trained Box (2DSeg)	mAP@25(%)	mAR@25(%)
OV-3DET [29]	Detic [65]	3.13	10.83
SAM3D [58]	SAM [19]	3.92	13.33
Ours	SAM [19]	4.23	**15.12**
	CropFormer [27]	**4.76**	14.62

Table 10. **3D Object Detection** on ScanNetv2. OpenM3D outperforms our method trained on the boxes from OV-3DET and SAM3D.

Method	Trained Box	AP@25 (%)	AR@25 (%)	AP@50 (%)	AR@50 (%)
OpenM3D	OV-3DET [29]	17.65	40.37	2.87	9.27
	SAM3D [58]	16.69	49.39	5.18	19.33
	Ours	**19.76**	**50.40**	**7.34**	**20.94**

3D Detection on ScanNetv2. We reported the results of our model evaluated on the common 18 classes in ScanNetv2 [5] in Table 10. OpenM3D trained with our pseudo boxes consistently outperforms the models trained with SAM3D and OV-3DET on all metrics, including AP@25, AP@50, AR@25, and AR@50. Notably, OpenM3D achieved over 12% and 20% improvements in AP@25 and AR@50, respectively, compared to OV-3DET. Larger gaps were observed, with 7.34% vs. 2.87% in AP@50 and 20.94% vs. 9.27% in AR@50. The substantial improvements brought by our method on AP@50 and AR@50 underscore the limitations associated with solely relying on single-view depth maps and images for bounding box generation. The notable improvements of our pseudo boxes over SAM3D in AP@25 and AP@50 metrics showcase the efficacy of our graph embedding-based pseudo boxes. Note that the train/evaluate split applied in [3, 29] differs from the official split by ScanNetv2 [5], making direct comparisons with their reported results challenging.

C. Limitation

The gap between class-agnostic and OV 3D detection implies that the pre-trained CLIP feature can be improved in classifying many semantically similar household objects. We leave this as a future direction.