Open-Vocabulary 3D Semantic Segmentation with Foundation Models

Li Jiang Shaoshuai Shi Bernt Schiele
Max Planck Institute for Informatics, Saarland Informatics Campus
{lijiang, sshi, schiele}@mpi-inf.mpg.de

Abstract

In dynamic 3D environments, the ability to recognize a diverse range of objects without the constraints of predefined categories is indispensable for real-world applications. In response to this need, we introduce OV3D, an innovative framework designed for open-vocabulary 3D semantic segmentation. OV3D leverages the broad open-world knowledge embedded in vision and language foundation models to establish a fine-grained correspondence between 3D points and textual entity descriptions. These entity descriptions are enriched with contextual information, enabling a more open and comprehensive understanding. By seamlessly aligning 3D point features with entity text features, OV3D empowers open-vocabulary recognition in the 3D domain, achieving state-of-the-art open-vocabulary semantic segmentation performance across multiple datasets, including ScanNet, Matterport3D, and nuScenes.

1. Introduction

For real-world applications like autonomous vehicles [14, 37, 52] and robotics [13, 19], the surrounding 3D environment is dynamic and ever-changing. The objects in the scenes can vary widely, and using fixed categories can limit the system's ability to recognize previously unseen objects. This motivates us to develop open-vocabulary techniques for 3D point cloud understanding to allow the system to handle a broader range of data without relying on pre-defined categories.

In open-vocabulary recognition, a common strategy is to unify visual and language features in the same feature space, leveraging the generalization abilities of language models trained on unbounded open text data. This strategy is commonly employed in 2D open-vocabulary frameworks [15, 28, 32, 46, 66]. To achieve such visual-language alignment, paired image and text data is essential. However, unlike the more plentiful (image, text) pairs available on the Internet, acquiring (point cloud, text) pairs is more difficult due to their relative scarcity and limited availability.

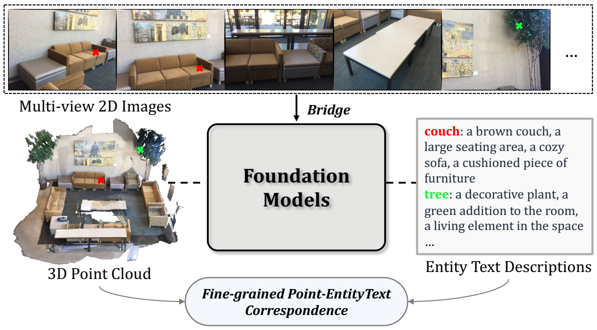

Multi-view 2D Images
...
Bridge
Foundation Models
3D Point Cloud
couch: a brown couch, a large seating area, a cozy sofa, a cushioned piece of furniture
tree: a decorative plant, a green addition to the room, a living element in the space
...
Entity Text Descriptions
Fine-grained Point-EntityText Correspondence

Figure 1. An overview of OV3D. We take multi-view 2D images as a bridge, leveraging the open-world knowledge in vision and language foundation models (*e.g*., large vision-language models and vision foundation models) to build a ***fine-grained correspondence*** between points and entity-level text descriptions.

For 3D open-vocabulary semantic segmentation, current methods [5, 12, 40, 64] employ images as intermediaries to establish the connection between text and 3D modalities, due to the easier acquisition of (point cloud, image) pairs over (point cloud, text) pairs. A typical strategy involves aligning point cloud features with image features that are already aligned with text features. For instance, OpenScene [40] aligns point features with pixel features from 2D open-vocabulary semantic segmentation models [15, 28]. Alternatively, another effective strategy involves generating captions for multi-view images representing the 3D scene, as demonstrated in [12, 64]. By doing so, the extracted 3D scene features can be directly compared and aligned with text features, promoting a more seamless cross-modality interaction. However, establishing a fine-grained point-to-text correspondence can be challenging through this strategy.

In this work, we introduce OV3D for open-vocabulary 3D semantic segmentation. The overview of OV3D is shown in Fig. 1. OV3D also engages in bridging the 3D point cloud and text descriptions with images to enable seamless point-text alignment. Different from previous works, we put our focus on exploiting the broad knowledge of open-world concepts encapsulated within vision and language foundation models [25, 34, 46, 78]. Our approach achieves two critical objectives: **1) Fine-grained Correspondence**: We establish a fine-grained correspondence between entity text descriptions and points; **2) Context-enriched Point-Text Alignment**: We facilitate a semantic-enriched mapping of point features to an open text feature space.

Specifically, our method comprises three components: **1) Mapping Image to EntityText**: Instead of captioning the entire image, we employ the Large Vision-Language Model (LVLM) to generate entity-level text descriptions (*i.e.*, EntityText) for entities in multi-view images. To enhance the openness and contextuality of EntityText, we prompt LVLM to generate various descriptions for the entities. **2) Associating Pixel with EntityText**: To establish a more fine-grained connection between image content and EntityText, we first utilize a vision foundation model for segmentation [25, 78] to produce class-agnostic segments for entities in the image. We then leverage a vision-language model [28, 46, 78] to associate segments with EntityText. Operating in a joint vision-language feature space, vision-language models have the ability to assign an EntityText to a segment (*i.e.* set of pixels) in a zero-shot manner. We thus obtain a fine-grained pixel-EntityText correspondence. **3) Connecting Point and EntityText via Pixel**: In this step, we project 3D points onto multi-view image planes to get the corresponding 2D pixel positions. Based on the pixel-EntityText correspondence, we can then retrieve entity information for each point, creating fine-grained (point, EntityText) pairs.

Integrating these components, our method achieves a seamless and fine-grained alignment of 3D points with entity text features, enabling context-enriched open-vocabulary recognition in 3D. It attains state-of-the-art performance in open-vocabulary semantic segmentation on datasets like ScanNet [11, 48], Matterport3D [4], and nuScenes [3]. Our major contribution is three-fold:

- We propose a framework to leverage vision and language foundation models (*e.g.*, LVLM [34], vision foundation models [25, 78], and vision-language models [15, 46]) to support 3D open-vocabulary recognition, enabling improved understanding of diverse concepts in 3D scenes.
- We introduce a fine-grained point-to-EntityText alignment strategy, enabling a more detailed correspondence between text descriptions and points, which facilitates a more accurate point-level 3D open-world understanding.
- Our work achieves superior zero-shot semantic segmentation performance on several datasets and shows excellent qualitative results on open-world recognition.

2. Related Work

3D Point Cloud Understanding has made great progress in recent years, covering a range of tasks including segmentation [17, 21–23, 26, 43, 56, 60, 65, 72, 73], detection [24, 35, 44, 49–51, 71, 74, 75], and classification [42, 55, 58, 61]. For point cloud feature extraction, existing approaches operate on raw points [42, 73] or sparse voxels [10, 16]. Though high performance is achieved, traditional methods typically reply on predefined categories.

Foundation Models that are trained on vast data have significantly impacted the fields of vision [25, 78] and language [9, 54]. Large language models [9, 54] have shown strong ability in open-world comprehension and reasoning. Recent large vision-language models [29, 34, 41, 70] further bridge the domains of image and language understanding. Vision-language models like CLIP [46] and ALIGN [20] align image and text features, excelling in zero-shot tasks due to extensive paired training data. Vision foundation models [25, 78] demonstrate robust zero-shot performance in segmentation. Our method leverages these foundation models to enable open-world 3D scene understanding.

2D Open-Vocabulary Learning has gained attention for its capacity to recognize objects beyond a constrained set of categories. Vision-language foundation models like CLIP [46] and ALIGN [20], trained on Internet-scale (image, text) pairs, excel in open-world scenarios, especially in novel class recognition. As these foundation models focus on image-level recognition, recent research has increasingly concentrated on establishing fine-grained object-level or pixel-level alignment between visual and language features, enabling open-vocabulary detection [30, 67] and segmentation [15, 28, 32, 45]. The availability of paired (image, text) data is crucial for training these models, ensuring their ability to interpret a wide range of concepts.

3D Open-Vocabulary Learning. Developing open-vocabulary techniques for 3D presents more challenges than in 2D, primarily due to the scarcity of (point cloud, text) pairs. Early zero-shot approaches [7, 8, 38] lack the connection between 3D data and open text descriptions, limiting openness. Recent methods typically bridge 3D and text modalities using images, given that 3D point clouds are commonly paired with multi-view 2D images. In this context, one stream of works [5, 18, 19, 40, 53, 57, 68] align point cloud features with image features extracted by vision-language models [15, 28, 46], using their inherent image-text feature alignment to implicitly align point clouds with text. For example, OpenScene [40] introduces point-pixel alignment, using pixel features from 2D open-vocabulary semantic segmentation models [15, 28]. Another stream of works [12, 64] uses image caption models [47, 57, 63] to generate text descriptions for images, thus enabling an alignment of point cloud features with open text features. However, this alignment is at the scene or region level, lacking the fine-grained guidance required for dense prediction tasks like segmentation. In our work, we leverage the extensive open-world knowledge embedded within vision and language foundation models [25, 34, 46, 78] to build a dense correspondence between individual points and entity text descriptions, effectively boosting the open-vocabulary ability of the 3D semantic segmentation model.

3. Method

In this work, we propose OV3D, a novel approach that enables open-vocabulary 3D point cloud semantic segmenta-

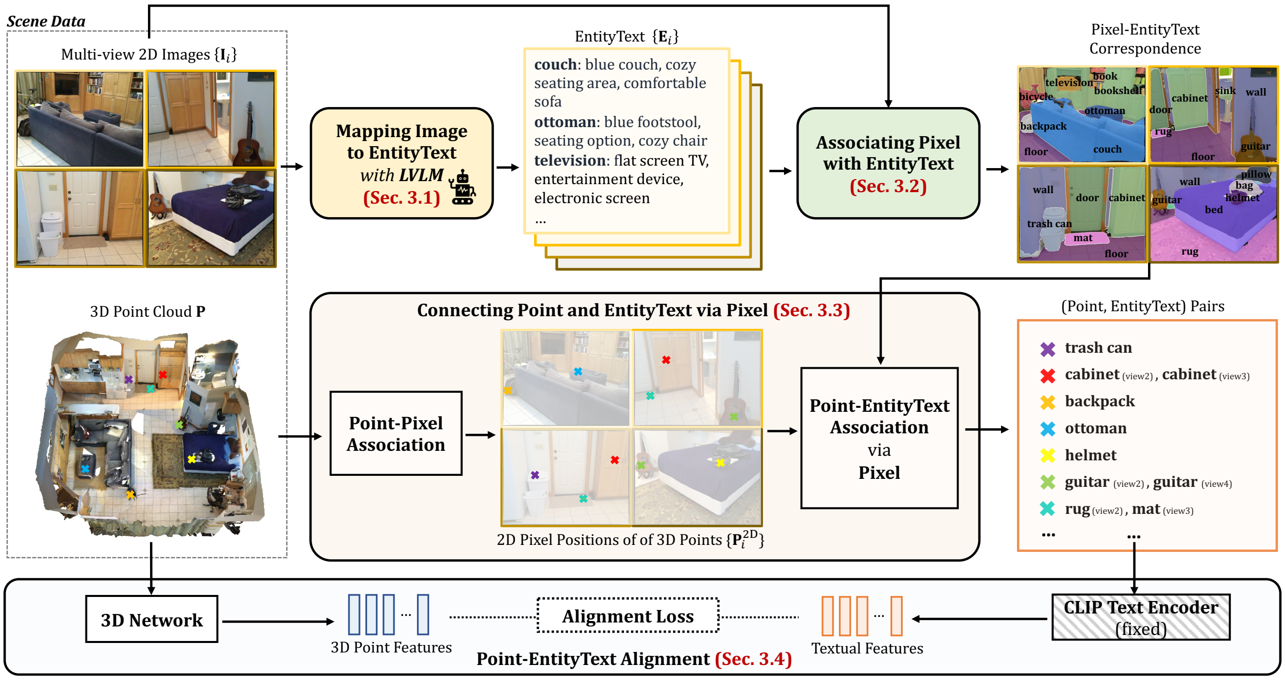

Scene Data
Multi-view 2D Images $\{\mathbf{I}_i\}$
3D Point Cloud $\mathbf{P}$
Mapping Image to EntityText with LVLM (Sec. 3.1)
EntityText $\{\mathbf{E}_i\}$
couch: blue couch, cozy seating area, comfortable sofa
ottoman: blue footstool, seating option, cozy chair
television: flat screen TV, entertainment device, electronic screen
...
Associating Pixel with EntityText (Sec. 3.2)
Pixel-EntityText Correspondence
television
book
bookshelf
bicycle
ottoman
backpack
floor
couch
cabinet
sink
wall
door
rug
guitar
floor
wall
door
cabinet
trash can
mat
floor
wall
pillow
bag
helmet
guitar
bed
rug
Connecting Point and EntityText via Pixel (Sec. 3.3)
Point-Pixel Association
2D Pixel Positions of of 3D Points $\{\mathbf{P}_i^{2D}\}$
Point-EntityText Association via Pixel
(Point, EntityText) Pairs
trash can
cabinet (view2), cabinet (view3)
backpack
ottoman
helmet
guitar (view2), guitar (view4)
rug (view2), mat (view3)
... ...
3D Network
3D Point Features
Alignment Loss
Textual Features
CLIP Text Encoder (fixed)
Point-EntityText Alignment (Sec. 3.4)

Figure 2. Illustration of our OV3D framework. OV3D leverages multi-view 2D images $\{\mathbf{I}_i\}$ to connect 3D point cloud $\mathbf{P}$ with language descriptions. Initially, entity-level text descriptions (*i.e*., EntityText) are extracted from the images using a Large Vision-Language Model (LVLM) (Sec. 3.1). These EntityText are then assigned to pixels to form a fine-grained pixel-EntityText correspondence (Sec. 3.2). Subsequently, 3D points are associated with EntityText based on the point-pixel correspondence (Sec. 3.3), which are then utilized to train the 3D network in an open-vocabulary setting, aligning point features with textual features (Sec. 3.4).

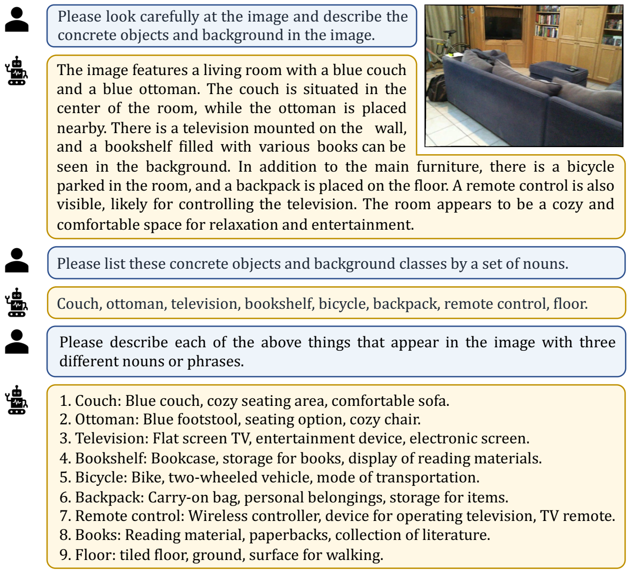
tion. The overall architecture of OV3D is shown in Fig. 2. For each scene, we denote the 3D point cloud as $\mathbf{P}$ and the corresponding set of multi-view 2D images as $\{\mathbf{I}_i\}_{i=1,\dots,M}$. Our initial step involves extracting names and descriptions for entities (*i.e*., EntityText) within the images, with the assistance of a Large Vision-Language Model (LVLM) (Sec. 3.1). Subsequently, we employ a vision foundation model to generate segments for entities present in each image. These segments are matched with the EntityText set based on their embedding distance within a shared image-language feature space, establishing fine-grained correspondence between pixel and EntityText. (Sec. 3.2). We then connect each point with the corresponding EntityText via point-pixel association (Sec. 3.3). The established (point, EntityText) pairs are then used to facilitate the 3D open-vocabulary training process, aligning the distribution of point features with that of open text features (Sec. 3.4). This alignment empowers the 3D model with open-vocabulary abilities, enhancing its capacity to handle diverse semantic concepts.

3.1. Mapping Image to EntityText

In this section, our objective is to map the visual content of images to linguistic semantics, identifying the entities depicted within the images. Recent advancements in the domain of Large Language Model (LLM) [54] have showcased their remarkable capacity for open-world understanding and reasoning. Additionally, the advent of LLM-powered LVLMs [34, 41] has empowered us to decode the visual information contained in images through textual descriptions. We employ LVLM in our framework to build a mapping from image contents to context-enriched entity-level text representations. Instead of directly requesting the LVLM to describe and list all the entities, we empirically find that adopting a conversation mode based on the chain-of-thought [59] process leads to enhanced and more stable outputs.

Fig. 3 shows an example of the conversation process using an up-to-date open-source LVLM, LLaVA-1.5 [33, 34]. We initiate the process by requesting the LVLM to provide an overall language description of the concrete entities present in the image. After that, we engage in a conversation with LVLM, asking it to list the names of all these entities. Specifically, we denote the name of the j-th entity in image $\mathbf{I}_i$ as $t^n_{i,j}$. To obtain more open and context-enriched descriptions for each entity, we further interact with LVLM to elicit diverse nouns or phrases depicting these entities. This yields an entity description set $\mathbf{t}^d_{i,j} = \{t^d_{i,j,k}\}_{k=1,\dots,K}$ for each entity, with K being the number of descriptions for a given entity. In this conversation manner, we step-by-step guide the LVLM to generate textual semantics for each entity in a desired format. It is noted that the conversation process is automated by using the same prompts for all images.

We use the term ***EntityText*** to represent the textual semantics, including both the name and diverse descriptions, for each entity. For the j-th entity in the i-th image, we



Please look carefully at the image and describe the concrete objects and background in the image.

The image features a living room with a blue couch and a blue ottoman. The couch is situated in the center of the room, while the ottoman is placed nearby. There is a television mounted on the wall, and a bookshelf filled with various books can be seen in the background. In addition to the main furniture, there is a bicycle parked in the room, and a backpack is placed on the floor. A remote control is also visible, likely for controlling the television. The room appears to be a cozy and comfortable space for relaxation and entertainment.

Please list these concrete objects and background classes by a set of nouns.

Couch, ottoman, television, bookshelf, bicycle, backpack, remote control, floor.

Please describe each of the above things that appear in the image with three different nouns or phrases.

1. Couch: Blue couch, cozy seating area, comfortable sofa.
2. Ottoman: Blue footstool, seating option, cozy chair.
3. Television: Flat screen TV, entertainment device, electronic screen.
4. Bookshelf: Bookcase, storage for books, display of reading materials.
5. Bicycle: Bike, two-wheeled vehicle, mode of transportation.
6. Backpack: Carry-on bag, personal belongings, storage for items.
7. Remote control: Wireless controller, device for operating television, TV remote.
8. Books: Reading material, paperbacks, collection of literature.
9. Floor: tiled floor, ground, surface for walking.

Figure 3. An example of the conversation process with LLaVA [34] for mapping image content to entity-level text descriptions.

represent EntityText as $\mathbf{e}_{i,j} = (t^n_{i,j}, \mathbf{t}^d_{i,j})$. The EntityText set for image $\mathbf{I}_i$ is $\mathbf{E}_i = \{\mathbf{e}_{i,j}\}_{j=1,\dots,N_{\mathbf{E}_i}}$, with $N_{\mathbf{E}_i}$ being the entity number in image $\mathbf{I}_i$. We thus build an open and comprehensive mapping from image $\mathbf{I}_i$ to EntityText $\mathbf{E}_i$.

Discussion: Image-level vs. Entity-level Text Description. In PLA [12] and RegionPLC [64], caption models [31, 57, 77] are employed to generate language descriptions for entire images or image crops. However, this strategy can pose challenges in achieving robust vision-semantic alignments, particularly when multiple objects coexist in the same image or crop. Relying on such a coarse (image, text) correspondence for vision and language alignment can hinder the model's ability to distinguish objects. Rather than generating captions at the image or crop level, we employ LVLM to generate entity-specific descriptions for individual entities in the image, which enables a finer alignment between vision and text in our subsequent steps.

Discussion: Concrete Entity Identification. Entity names can be generated by extracting nouns from image descriptions, as applied in [12]. However, direct noun extraction can introduce noise by inadvertently including abstract nouns like "relaxation", or location indicators like "center". In contrast, our approach employs LVLM to selectively extract only the concrete entities, thereby providing a more accurate set of entities. Table 6 shows a quantitative comparison of different entity identification strategies.

3.2. Associating Pixel with EntityText

In point cloud segmentation, a dense prediction task, the goal is to produce a point-level semantic map correlating each point to its class. Achieving such dense recognition necessitates fine-grained guidance. Therefore, in this section, we focus on establishing a finer-grained correspondence between image contents and EntityText at the pixel level.

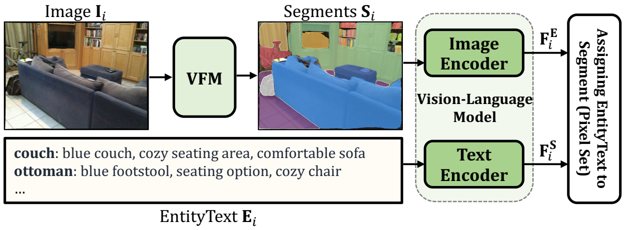


Image $\mathbf{I}_i$
VFM
Segments $\mathbf{S}_i$
Image Encoder
Vision-Language Model
Text Encoder
$\mathbf{F}^{\mathbf{E}}_i$
$\mathbf{F}^{\mathbf{S}}_i$
Assigning EntityText to Segment (Pixel Set)
couch: blue couch, cozy seating area, comfortable sofa
ottoman: blue footstool, seating option, cozy chair
...
EntityText $\mathbf{E}_i$

Figure 4. Process of associating pixel with EntityText. The Vision Foundation Model (VFM) is used to produce class-agnostic segments of the input image. The vision-language model is then applied to map the image segments and EntityTexts to a joint vision-language feature space. EntityTexts are assigned to segments (*i.e.*, pixel sets) based on the feature similarity.

To achieve this goal, we begin by exploiting the capabilities of the vision foundation models, such as SAM [25] and SEEM [78]. These models provide a universal interface for segmentation, allowing the generation of class-agnostic entity segments. As shown in Fig. 4, for each 2D image $\mathbf{I}_i$, we feed it into the vision foundation model to generate segments for the entities in the image. We denote these entity segments as $\mathbf{S}_i = \{\mathbf{s}_{i,j}\}_{j=1,\dots,N_{\mathbf{S}_i}}$, with $N_{\mathbf{S}_i}$ signifying the number of segments identified. Each entity segment represents a set of pixels belonging to this entity.

The next step is to assign each segment an EntityText to establish a detailed pixel-EntityText correspondence. To achieve this, we employ vision-language models [28, 46, 78]. These models consist of text and image encoders that map language and vision data into a unified embedding space, where corresponding text and image features are well aligned. Taking advantage of this property of vision-language models, we use the text encoder to encode the EntityText $\mathbf{E}_i$ and obtain textual embeddings $\mathbf{F}^{\mathbf{E}}_i = \{f^{\mathbf{e}}_{i,j}\}_{j=1,..,N_{\mathbf{E}_i}}$. Meanwhile, we use the corresponding image encoder to extract visual embeddings $\mathbf{F}^{\mathbf{S}}_i = \{f^{\mathbf{s}}_{i,j}\}_{j=1,\dots,N_{\mathbf{S}_i}}$ for the entity segments. We then compute cosine similarity between the segment visual embeddings $\mathbf{F}^{\mathbf{S}}_i$ and the textual embeddings $\mathbf{F}^{\mathbf{E}}_i$. Using these similarity scores, we assign to each segment the EntityText with the highest correspondence. Formally, for the j-th segment in the i-th image, the index of the assigned EntityText is calculated as

$$\mathcal{A}(i,j) = \arg\max_k(\cos(f^{\mathbf{s}}_{i,j}, f^{\mathbf{e}}_{i,k})), \quad (1)$$

where cos represents cosine similarity and $\mathcal{A}$ is the assignment function. The resulting correspondence between segment (set of pixels) and EntityText is $(\mathbf{s}_{i,j}, \mathbf{e}_{i,\mathcal{A}(i,j)})$, which associates the pixels with entity text descriptions.

Context-enriched Text Embedding Generation. As introduced in Sec. 3.1, we produce multiple descriptions for each entity within the image. In order to generate more open and context-enriched textual embeddings for these entities and thus enhance their ability to be accurately matched with the corresponding visual embeddings, we integrate both entity names $t^n_{i,j}$ and diverse entity descriptions $\mathbf{t}^d_{i,j} = \{t^d_{i,j,k}\}_{k=1,\ldots,K}$ in text embedding generation as

$$f^{\mathbf{e}}_{i,j} = \frac{1}{1+K}\left(\mathrm{TE}(t^n_{i,j}) + \sum_{k=1}^{K}\mathrm{TE}(t^d_{i,j,k})\right), \qquad (2)$$

where TE represents the text encoder.

Visual Embedding Generation. We use different ways to extract segment visual embeddings $f^{\mathbf{s}}$, depending on the specific vision-language model [15, 28, 46, 78] utilized in our framework. For instance, in the case of CLIP [46], which offers image-level vision-text alignment, we crop the segment area and input it to the CLIP image encoder to obtain $f^{\mathbf{s}}$. For models like OpenSeg [15] or LSeg [28], which are open-vocabulary segmentation models achieving pixel-level vision-text alignment, we utilize average pooling on the pixel features within each segment to create $f^{\mathbf{s}}$. When using SEEM [78], a Mask2Former [6]-style vision foundation model that provides a joint representation space where mask embeddings and text embeddings are aligned, we employ the mask embeddings as $f^{\mathbf{s}}$. In Table 8, we show the effects of different vision-language models in our framework.

3.3. Connecting Point and EntityText via Pixel

Given the fine-grained correspondence between pixels and EntityText, our next step is to connect the points and EntityText using pixels as a bridge. For this purpose, we first associate 3D points with 2D pixels by projecting the 3D points $\mathbf{P} \in \mathbb{R}^{N\times 3}$ onto multi-view image planes. Formally, the 2D pixel positions of 3D points in the i-th view, denoted as $\mathbf{P}^{2D}_i \in \mathbb{R}^{N\times 2}$, are calculated as

$$[\,\mathbf{P}^{2D}_i \mid \mathbf{1}\,] = \frac{1}{\mathbf{P}^{C}_i[:,2]}\mathbf{K}\mathbf{P}^{C}_i, \quad \mathbf{P}^{C}_i = [\,\mathbf{R}_i \mid \mathbf{t}_i\,]\,\mathbf{P}, \qquad (3)$$

where $[\cdot|\cdot]$ denotes the block matrix, and $\mathbf{P}^{C}_i \in \mathbb{R}^{N\times 3}$ represents the point positions in the camera coordinate system. The camera intrinsic matrix is denoted as $\mathbf{K} \in \mathbb{R}^{3\times 3}$, and $[\,\mathbf{R}_i \mid \mathbf{t}_i\,]$ is the extrinsic matrix, which combines the rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$ and the translation vector $\mathbf{t} \in \mathbb{R}^3$.

To verify the validity of points $\mathbf{P}$ in the i-th view, we first ensure their 2D pixel positions $\mathbf{P}^{2D}_i$ are within the image bounds. In cases where depth data is accessible, we also check whether the points are occluded in the i-th view by assessing the consistency between the Z-coordinate of $\mathbf{P}^{C}_i$ and the depth values at corresponding pixel positions $\mathbf{P}^{2D}_i$, as applied in [17, 40]. A point is valid if it is in the image bounds and remains unoccluded in the view. We denote the 2D pixel positions of valid points in the i-th view as $\hat{\mathbf{P}}^{2D}_i$.

Subsequently, by examining whether 2D pixel positions $\hat{\mathbf{P}}^{2D}_i$ fall within a segment, we identify valid 3D points for that segment. We denote the valid points for segment $\mathbf{s}_{i,j}$ as $\hat{\mathbf{P}}^{\mathbf{s}}_{i,j}$. Leveraging the pre-established segment-EntityText relation $(\mathbf{s}_{i,j}, \mathbf{e}_{i,\mathcal{A}(i,j)})$, we then associate $\hat{\mathbf{P}}^{\mathbf{s}}_{i,j}$ with the EntityText $\mathbf{e}_{i,\mathcal{A}(i,j)}$, where the pixel segment $\mathbf{s}_{i,j}$ serves as a bridge. Considering that each point may be valid in multiple views, it will be accordingly linked to a variety of EntityText. For each point, we aggregate the EntityText from all applicable views and denote the obtained (point, EntityText) pairs as $(p, \mathbf{E}^p)$, where $p \in \mathbf{P}$ and $\mathbf{E}^p$ represents the set of EntityText describing the entity to which point p belongs.

By connecting points to EntityText, our method enables a fine-grained and seamless alignment of point features with textual features, which directly leverages text models trained on unbounded open text data, thus offering improved openness and robustness compared to methods that rely on intermediary image features [40, 68].

3.4. Training and Inference

Training: Point-EntityText Alignment. We use (point, EntityText) pairs $(p, \mathbf{E}^p)$ to guide the open-vocabulary 3D network training. For each point p, we align its feature f^p from the 3D network with textual features $\{f^{\mathbf{e}}\}_{\mathbf{e}\in\mathbf{E}_p}$ of the corresponding entity descriptions. We use CLIP [46] text encoder for textual feature extraction, selected for its extensive training on large-scale, open-world data and its exceptional zero-shot capabilities with novel objects. Mathematically, the alignment is formulated as

$$L^p = \frac{1}{|\mathbf{E}_p|}\sum_{\mathbf{e}\in\mathbf{E}_p}(1 - \cos(f^p, f^{\mathbf{e}})). \qquad (4)$$

The total point-EntityText alignment loss is then calculated as the mean loss across all points, *i.e.*, $L = \sum_{p\in\mathbf{P}} {L^p}/{|\mathbf{P}|}$. Note that less-confident pairs are removed in the alignment.

By this means, we embed the point features into the CLIP text embedding space, formulating a joint point-text space with aligned point and text features, thus empowering the point network with the ability to generalize to novel classes.

Zero-shot Inference. During training, our model uses 2D images to link 3D points with language. However, at inference, it operates exclusively on 3D point clouds, without needing extra data compared to a standard 3D point cloud semantic segmentation network. Specifically, we employ the CLIP text encoder to extract textual features for an arbitrary open set of classes and compute the cosine similarities between point and class text features. Each point is then assigned to the class with the highest similarity.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct experiments across four widely recognized 3D semantic segmentation benchmarks: ScanNet [11], Matterport3D [4], nuScenes [3], and ScanNet200 [48]. ScanNet includes 1,613 indoor 3D scenes from 2.5 million RGB-D video views, with evaluations on standard 20 and more challenging 200 class sets. Matterport3D provides

Method	Training Overhead	Testing requires Images	ScanNet [11]				Matterport3D [4]			
			mIoU	mAcc	f-mIoU	f-mAcc	mIoU	mAcc	f-mIoU	f-mAcc
MSeg Voting [27]	-	✓	45.6	54.4	-	-	33.4	39.0	-	-
PointCLIP-Seg† [69]	-	✓	-	-	2.1	5.5	-	-	-	-
MaskCLIP† [76]	-	✓	-	-	23.1	40.9	-	-	-	-
OpenScene-2D [40]	-	✓	50.0	62.7	-	-	32.4	45.0	-	-
OpenScene-3D [40]	High		52.9	63.2	-	-	41.3	55.1	-	-
OpenScene [40]	High	✓	54.2	66.6	-	-	42.6	59.2	-	-
PLA† [12]	Low		-	-	17.7	33.5	-	-	-	-
RegionPLC [64]	Low		-	-	43.8	65.6	-	-	-	-
RegionPLC w/ OpenScene-3D [64]	High		-	-	60.0	75.8	-	-	-	-
OpenScene-3D‡ [40]	High		53.9	65.2	59.1	71.8	42.0	59.4	49.7	64.0
OV3D (Ours)	Low		57.3	72.9	64.0	76.3	45.8	62.4	50.4	65.7
OV3D w/ OpenScene-3D (Ours)	High		**59.6**	**74.5**	**67.6**	**79.1**	**48.2**	**64.0**	**53.8**	**68.0**
Fully-Sup.‡	-		72.0	80.7	79.5	86.8	55.7	67.4	60.9	72.2

Table 1. Performance comparison for annotation-free 3D semantic segmentation on ScanNet and Matterport3D. "†" denotes results reproduced by RegionPLC [64], while "‡" indicates results reproduced by us. The "High" training overhead implies that the approaches require extensive high-dimensional image feature processing during training. This demands considerable latency for online image model inferencing during training or substantial additional storage for offline image feature storage.

detailed 3D building environments from 194k RGB-D images. NuScenes focuses on urban driving scenarios with 34k LiDAR point clouds.

Evaluation Settings and Metrics. Our approach's effectiveness and flexibility in 3D open-world scenarios are demonstrated through a two-tiered experiment. Initially, we test in an annotation-free environment, without category annotations (Sec. 4.2). Then, we extend the analysis to a base-annotated context with predefined base and novel categories, as used in previous studies [12, 64] (Sec. 4.3). We primarily use mean intersection over union (mIoU) for evaluating 3D semantic segmentation. For indoor scenes, foreground mIoU (f-mIoU), excluding walls, floors, and ceilings, as well as mean accuracy (mAcc) and foreground mean accuracy (f-mAcc), are also measured. In base-annotated scenarios, mIoU is calculated separately for base and novel classes, with harmonic mean IoU (hIoU) included for comprehensive analysis, as in [12, 62, 64].

Implementation Details. Our framework utilizes SparseConvNet [10, 16] as the 3D backbone for point-wise feature extraction, coupled with the frozen CLIP [46] text encoder for text embeddings used in point-wise semantic classification. LLaVA-1.5 [33] is adopted as the LVLM in OV3D for EntityText generation. SEEM [78] serves as both the vision foundation model and vision-language model for the pixel-EntityText association. Training is conducted using the AdamW optimizer [36], with a batch size of 8.

4.2. Understanding Annotation-Free 3D Worlds

In this section, we present an evaluation of our proposed OV3D within the context of annotation-free, open-world 3D semantic segmentation. This setting involves training all models without any annotations and employing a zero-shot approach during testing. We follow [40] to test all approaches on the ScanNet [11] validation set, Matterport3D [4] test set, and nuScenes [3] validation set.

Method	mIoU
OpenScene-LSeg [40]	36.7
OpenScene-OpenSeg [40]	42.1
OpenScene-3D‡ [40]	41.3
OV3D (Ours)	44.6
OV3D w/ OpenScene-3D (Ours)	**45.5**
Fully-Sup.‡	76.4

Table 2. Performance comparison for annotation-free 3D semantic segmentation on nuScenes [3] dataset with outdoor driving scenarios. "‡" indicates results reproduced by us.

Method	mIoU
PLA [12]	1.8
OpenScene-3D‡ [40]	7.3
RegionPLC [64]	6.5
OV3D (Ours)	8.7
OV3D w/ OpenScene-3D (Ours)	**9.8**
Fully-Sup.‡	21.3

Table 3. Performance comparison for annotation-free 3D semantic segmentation on ScanNet200 [48] dataset with 200 categories. "‡" indicates results reproduced by us.

Comparative Analysis with Zero-Shot Methods. Our approach OV3D showcases superior performance in zero-shot 3D semantic segmentation, as shown in Table 1. Remarkably, OV3D outperforms all previous methods, including the advanced OpenScene [40] and the recent RegionPLC [64]. On the ScanNet dataset, OV3D shows a +3.1% improvement in mIoU and +6.3% in mAcc. Similarly, on Matterport3D, it achieves +3.2% mIoU and +3.0% mAcc increases.

Besides that, OV3D is significantly more efficient than OpenScene setup, requiring far less storage and training I/O costs. OpenScene needs over 300GB for image feature generation [64], while OV3D only requires about 6GB for ScanNet and 4GB for Matterport3D to store (point, EntityText) pairs. This efficiency allows OV3D to be easily integrated with OpenScene-3D by simply applying the complementary alignment objectives from the two methods concurrently.

As shown in Table 1, combining OV3D with OpenScene-3D yields notable performance boosts over OpenScene-3D: +5.7% mIoU and +9.3% mAcc on ScanNet, and +6.2% mIoU and +4.6% mAcc on Matterport3D.

Moreover, OV3D notably reduces the performance gap between zero-shot methods and fully-supervised performance, especially in complex environments like Matterport3D. It reduces the gap in mIoU and mAcc to just -7.5% and -3.4%, showcasing its robustness in complex scenarios where fully supervised methods may be less effective.

Table 2 shows that OV3D also excels in outdoor driving scenarios, highlighting its universality and versatility.

Assessment in Long-Tail Scenarios. We further evaluate OV3D's open-vocabulary capability and generalizability on the ScanNet200 benchmark, which presents a significant challenge with its 200 categories - ten times more than previous benchmarks. According to Table 3, OV3D outperforms the best existing model with a +2.5% increase in mIoU across these categories. This success is attributed to OV3D's approach of learning directly from language foundation models [46], unlike OpenScene's [40] reliance on knowledge distillation from more limited 2D segmentation models. In contrast to methods like PLA [12] and RegionPLC [64], which align 3D features with text on an image or regional basis, OV3D leverages vision foundation models to establish a more precise link between 3D points and entity texts, leading to its enhanced performance.

Qualitative Results. To illustrate the effectiveness of our OV3D in understanding open-vocabulary 3D scenes, we present a series of detailed qualitative results in Fig. 5 and Fig. 6. It includes a variety of examples that show OV3D's robust performance in diverse open-world scenarios.

4.3. Interpreting Base-Annotated 3D Worlds

In this section, we evaluate our method on base-annotated open-world 3D semantic segmentation. Following [12, 64], we perform experiments on ScanNet [11], categorizing it into three settings: B15/N4, B12/N7, and B10/N9, where "B" and "N" represent base (annotated) and novel (unannotated) categories, respectively. All models are trained on the official training split and assessed on the validation split.

Extending OV3D to Base-Annotated Setting. Our OV3D model, originally designed for training without annotations, can be seamlessly adapted to base-annotated settings. As described in Sec. 3.4, OV3D uses generated (point, EntityText) pairs for training. For base categories, we replace the EntityText with the actual category names in these pairs, while pairs for novel categories remain unchanged. This allows OV3D to be effectively trained using a combination of accurate point annotations for base categories and unsupervised (point, EntityText) pairs, enhancing its applicability and performance in varied scenarios.

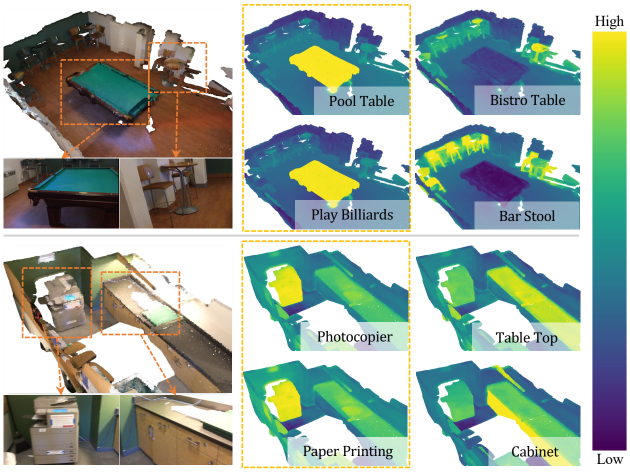


Figure 5. Qualitative results of our OV3D on open-vocabulary 3D scene understanding without using any annotations. We explore 3D scenes by using different query texts (*e.g.*, "pool table" and "photocopier") and color 3D points based on their feature similarity, with brighter yellow color indicating higher similarity. Key findings include: 1) OV3D is able to accurately locate 3D regions relevant to a broad spectrum of query texts. (2) OV3D has the ability to recognize the same object described by different texts, exemplified by identifying a "pool table" both as an object and by its function ("play billiards") (Note: similar concepts are marked with yellow rectangles in the middle part). (3) OV3D is proficient in distinguishing subcategories within a concept, such as differentiating between a "pool table" and a "bistro table".

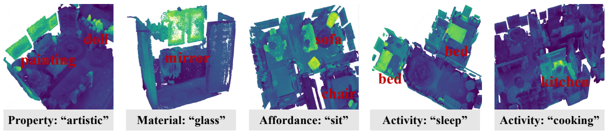


Figure 6. More qualitative results (zoom in for better viewing).

Comparative Performance Analysis. OV3D excels in all metrics across three different settings, as shown in Table 4. It surpasses the state-of-the-art model RegionPLC [64] in harmonic mIoU, recording increases of +2.5%, +3.4%, and +6.0% in the B15/N4, B12/N7, and B10/N9 settings, respectively. Notably, OV3D's performance enhancement grows with the increase in the number of novel categories. For example, in the B10/N9 setting, it improves mIoU for novel categories by +7.9% over RegionPLC, demonstrating its strong adaptability and effectiveness in recognizing and classifying unseen categories.

4.4. Ablation Study

In this section, we conduct ablation studies on ScanNet validation set in an annotation-free setting.

Entity Identification Strategy. We investigate various entity identification strategies, as presented in Table 6. For "Noun Extraction", we use the Natural Language Toolkit (NLTK) [2] library to extract nouns from the image descriptions generated by LVLM, as in [12]. In an enhanced version, "Noun Extraction†", we further verify whether the

Method	B15/N4			B12/N7			B10/N9		
	hIoU	mIoU$^{\mathcal{B}}$	mIoU$^{\mathcal{N}}$	hIoU	mIoU$^{\mathcal{B}}$	mIoU$^{\mathcal{N}}$	hIoU	mIoU$^{\mathcal{B}}$	mIoU$^{\mathcal{N}}$
3DGenZ [38]	20.6	56.0	12.6	19.8	35.5	13.3	12.0	63.6	06.6
3DTZSL [7]	10.5	36.7	06.1	03.8	36.6	02.0	07.8	55.5	04.2
LSeg-3D [28]	00.0	64.4	00.0	00.9	55.7	00.1	01.8	68.4	00.9
PLA [12]	65.3	68.3	62.4	55.3	69.5	45.9	53.1	76.2	40.8
RegionPLC [64]	69.9	68.4	71.5	65.1	69.6	61.1	58.8	76.6	47.7
OV3D (Ours)	**72.4**	**70.2**	**74.7**	**68.5**	**74.1**	**63.7**	**64.8**	**77.6**	**55.6**
Fully-Sup.	74.6	70.2	79.6	72.1	72.2	72.0	71.5	77.0	66.7

Table 4. Performance comparison for base-annotated 3D semantic segmentation on ScanNet [11]. mIoU$^{\mathcal{B}}$ indicates the mIoU of base categories while mIoU$^{\mathcal{N}}$ indicates the mIoU of novel categories.

LSeg-3D [12]	PLA [12]	RegionPLC [64]	OV3D (Ours)
1.7 (11.2)	13.4 (25.1)	36.9 (53.6)	41.3 (64.4)

Table 5. Performance comparison for zero-shot domain transfer from ScanNet [11] to S3DIS [1]. The model is trained on ScanNet (annotation-free) and evaluated on S3DIS using f-mIoU (f-mAcc).

Approach for Entity Identification	Generated Entities		mIoU(%)
	Concrete	Context-Aware	
Caption + Noun Extraction			55.3
Caption + Noun Extraction†	✓		57.2
Instruct LVLM	✓	✓	**59.6**

Table 6. Effects of different strategies for entity identification. "Noun Extraction†" indicates that concreteness check using WordNet [39] is conducted after extracting nouns. "Context-Aware" refers to the process of identifying concrete entities by conditioning on input images, rather than relying solely on noun analysis.

	Entity Name-only	Context-Enriched
mIoU(%)	57.6	**59.6**

Table 7. Effects of various text embedding generation strategies.

VFM	VLM	mIoU(%)
SAM [25]	CLIP [46]	51.0
SAM [25]	LSeg [28]	54.5
SAM [25]	OpenSeg [15]	55.1
SEEM [78]	SEEM [78]	**59.6**

Table 8. Effects of different strategies for associating pixels with EntityText. "VFM" denotes the vision foundation model used to generate entity segments. "VLM" denotes the vision-language model for assigning EntityText to segments.

extracted nouns correspond to concrete entities by leveraging the WordNet [39] database for concreteness information retrieval. In our primary method, "Instruct LVLM", we directly harness the LVLM's ability to identify concrete entities. Specifically, we instruct LLaVA-1.5 [33] to enumerate all the concrete entities based on a given image. As indicated by our results in Table 6, LVLM's performance in recognizing concrete entities, informed by visual context, is superior to traditional methods like NLTK noun extraction in our task.

Context-enriched Text Embedding. We experiment with different strategies for generating the entity text embedding, as detailed in Table 7. We find that integrating both entity names and diverse entity descriptions as in Eq. (2) effectively enriches the contextuality and openness of the text embeddings. This approach achieves better performance in open-vocabulary semantic segmentation compared to using solely entity names for text embedding generation.

Vision Foundation Model & Vision-Language Model for Associating Pixel with EntityText. Table 8 compares different strategies for associating pixels with EntityText. When using SAM as the vision foundation model to generate entity segments, we experiment with three vision-language models: CLIP [46], LSeg [28], and OpenSeg [15], to assign EntityText to these segments. With CLIP, the entity segment area is cropped and processed through the CLIP image encoder, but this might result in misalignment between the entity's visual and textual features due to insufficient context or multiple objects in the crop, potentially reducing the accuracy of the pixel-EntityText association. With LSeg or OpenSeg, both 2D open-vocabulary semantic segmentation models, we aggregate the pixel features within the segment region to derive the segment's visual feature. This method ensures a better alignment of visual features with the segment region, leading to improved performance compared to using CLIP. When SEEM [78], a universal segmentation interface that enables both visual and text prompts, serves as the vision foundation model, we directly utilize its joint vision-language feature space for assigning EntityText to segments. SEEM allows us to directly leverage the mask embeddings as the segment's visual feature, ensuring precise segment depiction. As shown in Table 8, using SEEM for pixel-EntityText association performs best in our experiments.

5. Conclusion

We present OV3D, advancing the development of 3D open-vocabulary semantic segmentation. By utilizing vision and language foundation models, OV3D establishes a fine-grained correspondence between individual points and entity text descriptions, achieving a seamless and dense alignment of point features with open and context-enriched text features. OV3D's excellent performance on both indoor and outdoor datasets attests to its effectiveness and adaptability. With the advancement of foundation models, we believe OV3D's performance will be further unleashed, serving as a powerful 3D open-vocabulary solution in real-world applications.

References

[1] 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 *CVPR*, 2016. 8

[2] Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009. 7

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 5, 6

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *3DV*, 2017. 2, 5, 6

[5] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *CVPR*, 2023. 1, 2

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 5

[7] Ali Cheraghian, Shafin Rahman, Dylan Campbell, and Lars Petersson. Transductive zero-shot learning for 3d point cloud classification. In *WACV*, 2020. 2, 8

[8] Ali Cheraghian, Shafin Rahman, Townim F Chowdhury, Dylan Campbell, and Lars Petersson. Zero-shot learning on 3d point cloud objects and beyond. *IJCV*, 2022. 2

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 2

[10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 2, 6

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 5, 6, 7, 8

[12] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *CVPR*, 2023. 1, 2, 4, 6, 7, 8

[13] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 1

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 1

[15] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. 1, 2, 5, 8

[16] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018. 2, 6

[17] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimensional scene understanding. In *CVPR*, 2021. 2, 5

[18] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *ICCV*, 2023. 2

[19] 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 *RSS*, 2023. 1, 2

[20] 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 *ICML*, 2021. 2

[21] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. In *ICCV*, 2019. 2

[22] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *CVPR*, 2020.

[23] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In *ICCV*, 2021. 2

[24] Li Jiang, Zetong Yang, Shaoshuai Shi, Vladislav Golyanik, Dengxin Dai, and Bernt Schiele. Self-supervised pre-training with masked shape prediction for 3d scene understanding. In *CVPR*, 2023. 2

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 1, 2, 4, 8

[26] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *CVPR*, 2022. 2

[27] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *CVPR*, 2020. 6

[28] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 1, 2, 4, 5, 8

[29] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 2

[30] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 2

[31] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 4

[32] 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 *CVPR*, 2023. 1, 2

[33] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 3, 6, 8

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 3, 4

[35] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. *ICCV*, 2021. 2

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 6

[37] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3d object detection for autonomous driving: A comprehensive survey. *IJCV*, 2023. 1

[38] Björn Michele, Alexandre Boulch, Gilles Puy, Maxime Bucher, and Renaud Marlet. Generative zero-shot learning for semantic segmentation of 3d point clouds. In *3DV*, 2021. 2, 8

[39] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995. 8

[40] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, 2023. 1, 2, 5, 6, 7

[41] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 2, 3

[42] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2

[43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017. 2

[44] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 2

[45] Jie Qin, Jie Wu, Pengxiang Yan, Ming Li, Ren Yuxi, Xuefeng Xiao, Yitong Wang, Rui Wang, Shilei Wen, Xin Pan, et al. Freeseg: Unified, universal and open-vocabulary image segmentation. In *CVPR*, 2023. 2

[46] 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 *ICML*, 2021. 1, 2, 4, 5, 6, 7, 8

[47] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. 2

[48] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision*, 2022. 2, 5, 6

[49] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 2

[50] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 2020.

[51] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *IJCV*, 2022. 2

[52] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1

[53] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. In *NeurIPS*, 2023. 2

[54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 3

[55] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *ICCV*, 2019. 2

[56] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *CVPR*, 2022. 2

[57] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*, 2022. 2, 4

[58] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 2019. 2

[59] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022. 3

[60] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022. 2

[61] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 2

[62] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network

for zero-and few-label semantic segmentation. In *CVPR*, 2019. 6

[63] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[64] Jihan Yang, Runyu Ding, Zhe Wang, and Xiaojuan Qi. Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. *arXiv preprint arXiv:2304.00962*, 2023. 1, 2, 4, 6, 7, 8

[65] Zetong Yang, Li Jiang, Yanan Sun, Bernt Schiele, and Jiaya Jia. A unified query-based paradigm for point cloud understanding. In *CVPR*, 2022. 2

[66] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, 2021. 1

[67] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[68] Junbo Zhang, Runpei Dong, and Kaisheng Ma. Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. *arXiv preprint arXiv:2303.04748*, 2023. 2, 5

[69] 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. In *CVPR*, 2022. 6

[70] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 2

[71] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *ECCV*, 2020. 2

[72] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, 2019. 2

[73] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 2

[74] Wu Zheng, Weiliang Tang, Sijin Chen, Li Jiang, and Chi-Wing Fu. CIA-SSD: Confident iou-aware single-stage object detector from point cloud. In *AAAI*, 2021. 2

[75] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. SE-SSD: Self-ensembling single-stage object detector from point cloud. In *CVPR*, 2021. 2

[76] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 6

[77] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 4

[78] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *NeurIPS*, 2023. 1, 2, 4, 5, 6, 8