OpenScene: 3D Scene Understanding with Open Vocabularies

Songyou Peng[1,2,3] Kyle Genova[1] Chiyu "Max" Jiang[4] Andrea Tagliasacchi[1,5]
Marc Pollefeys[2] Thomas Funkhouser[1]

[1] Google Research [2] ETH Zurich [3] MPI for Intelligent Systems, Tübingen [4] Waymo LLC [5] Simon Fraser University

pengsongyou.github.io/openscene

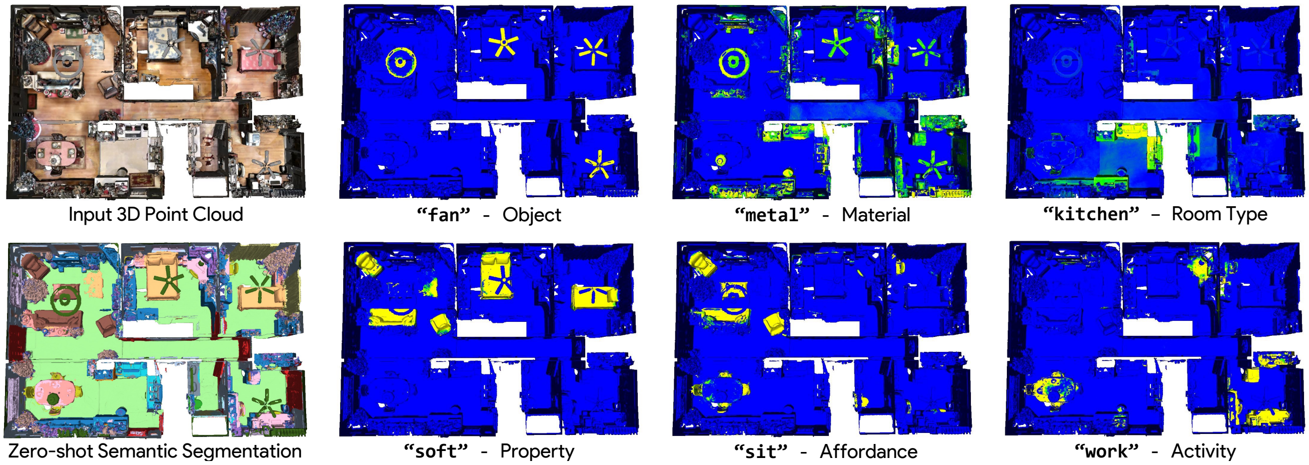


Figure 1. **Open-vocabulary 3D Scene Understanding.** We propose OpenScene, a zero-shot approach to 3D scene understanding that co-embeds dense 3D point features with image pixels and text. The examples above show a 3D scene with surface points colored by how well they match a user-specified query string – yellow is highest, green is middle, blue is low. Because its features are language-based, OpenScene answers a wide variety of example queries, like "soft", "kitchen", or "work", without labeled 3D data.

Abstract

Traditional 3D scene understanding approaches rely on labeled 3D datasets to train a model for a single task with supervision. We propose OpenScene, an alternative approach where a model predicts dense features for 3D scene points that are co-embedded with text and image pixels in CLIP feature space. This zero-shot approach enables task-agnostic training and open-vocabulary queries. For example, to perform SOTA zero-shot 3D semantic segmentation it first infers CLIP features for every 3D point and later classifies them based on similarities to embeddings of arbitrary class labels. More interestingly, it enables a suite of open-vocabulary scene understanding applications that have never been done before. For example, it allows a user to enter an arbitrary text query and then see a heat map indicating which parts of a scene match. Our approach is effective at identifying objects, materials, affordances, activities, and room types in complex 3D scenes, all using a single model trained without any labeled 3D data.

1. Introduction

3D scene understanding is a fundamental task in computer vision. Given a 3D mesh or point cloud with a set of posed RGB images, the goal is to infer the semantics, affordances, functions, and physical properties of every 3D point. For example, given the house shown in Figure 1, we would like to predict which surfaces are part of a fan (semantics), made of metal (materials), within a kitchen (room types), where a person can sit (affordances), where a person can work (functions), and which surfaces are soft (physical properties). Answers to these queries can help a robot interact intelligently with the scene or help a person understand it through interactive query and visualization.

Achieving this broad scene understanding goal is challenging due to the diversity of possible queries. Traditional 3D scene understanding systems are trained with supervision from benchmark datasets designed for specific tasks (e.g., 3D semantic segmentation for a closed set of 20 classes [5, 12]). They are each designed to answer one type of query (is this point on a chair, table, or bed?), but provide little assistance for related queries where training data is scarce (e.g., segmenting rare objects) or other queries with no 3D supervision (e.g., estimating material properties).

In this paper, we investigate how to use pre-trained text-image embedding models (e.g., CLIP [43]) to assist in 3D scene understanding. These models have been trained from

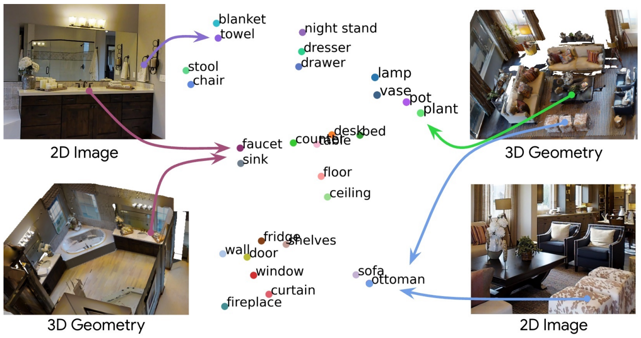


Figure 2. **Key idea.** We co-embed 3D points with text and image pixels in the CLIP feature space (visualized with T-SNE [51]) which has structure learned from large image and text repositories.

large datasets of captioned images to co-embed visual and language concepts in a shared feature space. Recent work has shown that these models can be used to increase the flexibility and generalizability of 2D image semantic segmentation [17, 30, 44, 57, 59, 60]. However, nobody has investigated how to use them to improve the diversity of queries possible for 3D scene understanding.

We present *OpenScene*, a simple yet effective zero-shot approach for open-vocabulary 3D scene understanding. Our key idea is to compute dense features for 3D points that are co-embedded with text strings and image pixels in the CLIP feature space (Fig. 2). To achieve this, we establish associations between 3D points and pixels from posed images in the 3D scene, and train a 3D network to embed points using CLIP pixel features as supervision. This approach brings 3D points in alignment with pixels in the feature space, which in turn are aligned with text features, and thus enables open vocabulary queries on the 3D points.

Our 3D point embedding algorithm includes both 2D and 3D convolutions. We first back-project the 3D position of the point into every image and aggregate the features from the associated pixels using multi-view fusion. Next, we train a sparse 3D convolutional network to perform feature extraction from only the 3D point cloud geometry with a loss that minimizes differences to the aggregated pixel features. Finally, we ensemble the features produced by the 2D fusion and the 3D network into a single feature for each 3D point. This hybrid 2D-3D feature strategy enables the algorithm to take advantage of salient patterns in both 2D images and 3D geometry, and thus is more robust and descriptive than features from either domain alone.

Once we have computed features for every 3D point, we can perform a variety of 3D scene understanding queries. Since the CLIP model is trained with natural language captions, it captures concepts beyond object class labels, including affordances, materials, attributes, and functions (Fig. 1). For example, computing the similarity of 3D features with the embedding for "soft" produces the result shown in the bottom-left image of Fig. 1, which highlights the couches, beds, and comfy chairs as the best matches.

Since our approach is zero-shot (i.e. no use of labeled data for the target task), it does not perform as well as fully-supervised approaches on the limited set of tasks for which there is sufficient training data in traditional benchmarks (e.g., 3D semantic segmentation with 20 classes). However, it does achieve significantly stronger performance on other tasks. For example, it beats a fully-supervised approach on indoor 3D semantic segmentation with 40, 80, or 160 classes. It also performs better than other zero-shot baselines, and can be used without any retraining on novel datasets even if they have different label sets. It works for indoor RGBD scans as well as outdoor driving captures. Overall, our contributions are summarized as follows:

- We introduce open vocabulary 3D scene understanding tasks where arbitrary text queries are used for semantic segmentation, affordance estimation, room type classification, 3D object search, and 3D scene exploration.
- We propose OpenScene, a zero-shot method for extracting 3D dense features from an open vocabulary embedding space using multi-view fusion and 3D convolution.
- We demonstrate that the extracted features can be used for 3D semantic segmentation with performance better than fully supervised methods for rare classes.

2. Related Work

This paper draws on a large literature of previous work on 3D scene understanding, multi-modal embedding, and zero-shot learning.

Closed-set 3D Scene Understanding. There is a long history of work on 3D scene understanding for vision and robotics applications. Most prior work focuses on training models with ground-truth 3D labels [11, 20, 22, 23, 25, 31, 41, 42, 45, 48, 54]. These works have yielded network architectures and training protocols that have significantly pushed the boundary of several 3D scene understanding benchmarks, including 3D object classification [56], 3D object detection and localization [4, 6, 15, 49], 3D semantic and instance segmentation [2, 3, 5, 12, 24, 34, 40], 3D affordance prediction [14, 32, 53], and so on. The most closely related work to ours of this type is Rozenberszki et al. [46], since they use the CLIP embedding to pre-train a model for 3D semantic segmentation. However, they only use the text embedding for point encoder pretraining, and then train the point decoder with 3D GT annotations afterwards. Their focus is on using the CLIP embedding to achieve better supervised 3D semantic segmentation, rather than open-vocabulary queries.

Another line of research performs 3D scene understanding experiments with only 2D ground truth supervisions [16, 27, 38, 47, 52, 55]. For example, [16] generates pseudo 3D annotation by backprojecting and fusing the 2D predicted labels, from which they learn the 3D segmentation task. However, their 2D network is trained with ground truth 2D labels. A couple of works [36, 47] pretrain the 3D segmentation network using point-pixel pairs via contrastive learning between 2D and 3D features. We also utilize 2D image features as our pseudo-supervision when training the 3D network and no 2D labels are needed.

All these approaches have mainly been applied with small predefined labelsets containing common object categories. They do not work as well when the number of object categories increases, as tail classes have few training examples. In contrast, we are able to segment with arbitrary labelsets without any re-training, and we show strong ability of understanding different contents, ranging from rare object types to even materials or physical properties, which is impossible for previous methods.

Open-Vocabulary 2D Scene Understanding. The recent advances of large visual language models [1, 26, 43] have enabled a remarkable level of robustness in zero-shot 2D scene understanding tasks, including recognizing long-tail objects in images. However, the learned embeddings are often at the image level, thus not applicable for dense prediction tasks requiring pixel-level information. Many recent efforts [17,18,21,28,30,33,37,44,57,59,60] attempt to correlate the dense image features with the embedding from large language models. In this way, given an image at test time, users can define arbitrary text labels to classify, detect, or segment the image.

More recently, Ha and Song [19] take a step forward and perform open-vocabulary partial scene understanding and completion given a single RGB-D frame as input. This method is limited to small partial scenes and requires ground truth training data for supervision. In contrast, in this work, we solely rely on pretrained open-vocabulary 2D models and perform a series of 3D scene-level understanding tasks, without the need for any ground truth training data in 2D or 3D. Moreover,in the absence of 2D images, our method can perform 3D-only open-vocabulary scene understanding tasks based on a 3D point network distilled from an open-vocabulary 2D image model through 3D fusion.

Zero-shot Learning for 3D Point Clouds. While there have been a number of studies on zero-shot learning for 2D images, their application to 3D is still recent and scarce. A handful of works [7–10] attempt to address the 3D point classification and generation tasks. More recently, [35, 39] investigated zero-shot learning for semantic segmentation for 3D point clouds. They train with supervision of 3D ground truth labels for a predefined set of seen classes and then evaluate on new unseen classes. However, these methods are still limited to the closed-set segmentation setting and still require GT training data for the majority of the 3D dataset. Our method *does not* require any labeled 3D data for training, and it handles a broad range of queries supported by a large language model.

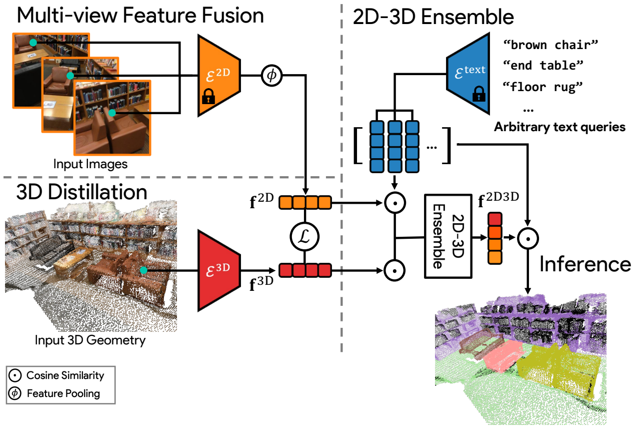


Figure 3. **Method Overview.** Given a 3D model (mesh or point cloud) and a set of posed images, we train a 3D network $\mathcal{E}^{3D}$ to produce dense features for 3D points $\mathbf{f}^{3D}$ with a distillation loss $\mathcal{L}$ to multi-view fused features $\mathbf{f}^{2D}$ for projected pixels. We ensemble $\mathbf{f}^{2D}$ and $\mathbf{f}^{3D}$ based on cosine similarities to CLIP embeddings for an arbitrary set of queries to form $\mathbf{f}^{2D3D}$. During inference, we can use the similarity scores between per-point features and given CLIP features to perform open-vocabulary 3D scene understanding tasks.

3. Method

An overview of our approach is illustrated in Fig. 3. We first compute per-pixel features for every image using a model pre-trained for open-vocabulary 2D semantic segmentation. We then aggregate the pixel features from multiple views onto every 3D point to form a per-point fused feature vector Sec. 3.1. We next distill a 3D network to reproduce the fused features using only the 3D point cloud as input Sec. 3.2. Next, we ensemble the fused 2D features and distilled 3D features into a single per-point feature Sec. 3.3 and use it to answer open-vocabulary queries Sec. 3.4.

3.1. Image Feature Fusion

The first step in our approach is to extract dense per-pixel embeddings for each RGB image from a 2D visual-language segmentation model, and then back-project them onto the 3D surface points of a scene.

Image Feature Extraction. Given RGB images with a resolution of $H \times W$, we can simply compute the per-pixel embeddings from the (frozen) segmentation model $\mathcal{E}^{2D}$, denoted as $\mathbf{I}_i \in \mathbb{R}^{H \times W \times C}$, where C is the feature dimension, and i is the index spanning the total number of images. For $\mathcal{E}^{2D}$, we experiment with two pretrained image segmentation models OpenSeg [17] and LSeg [30].

2D-3D Pairing. Given a 3D surface point $\mathbf{p} \in \mathbb{R}^3$ in the point clouds $\mathbf{P} \in \mathbb{R}^{M \times 3}$ of a scene with M points, we calculate its corresponding pixel $\mathbf{u} = (u, v)$ when the intrinsic matrix I_i and world-to-camera extrinsic matrix E_i of that frame i are provided. We only consider the pinhole camera model in this paper so the projection can be represented as $\tilde{\mathbf{u}} = I_i \cdot E_i \cdot \tilde{\mathbf{p}}$, where $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{p}}$ are the homogeneous coordinates of $\mathbf{u}$ and $\mathbf{p}$, respectively. Note that for indoor datasets like ScanNet and Matterport3D where the depth images are provided, we also conduct occlusion tests to guarantee the pixel $\mathbf{u}$ are only paired with a visible surface point $\mathbf{p}$.

Fusing Per-Pixel Features. With the 2D-3D pairing, the corresponding 2D features in frame i for point $\mathbf{p}$ can be written as $\mathbf{f}_i = \mathbf{I}_i(\mathbf{u}) \in \mathbb{R}^C$. Now, assume a total number of K views can be associated with point $\mathbf{p}$, we can then fuse such 2D pixel embeddings to obtain a single feature vector for this point $\mathbf{f}^{\text{2D}} = \phi(\mathbf{f}_1, \cdots, \mathbf{f}_K)$, where $\phi : \mathbb{R}^{K \times C} \mapsto \mathbb{R}^C$ is an average pooling operator for multi-view features. An ablation study on different fusion strategies are discussed in supplemental. After repeating the fusion process for each point, we can build a feature point cloud $\mathbf{F}^{\text{2D}} = \{\mathbf{f}_1^{\text{2D}}, \cdots, \mathbf{f}_M^{\text{2D}}\} \in \mathbb{R}^{M \times C}$.

3.2. 3D Distillation

The feature cloud $\mathbf{F}^{\text{2D}}$ can be directly used for language-driven 3D scene understanding *when* images are present. Nevertheless, such fused features could lead to noisy segmentation due to potentially inconsistent 2D predictions. Moreover, some tasks only provide 3D point clouds or meshes. Therefore, we can distill such 2D visual-language knowledge into a 3D point network that only takes 3D point positions as input.

Specifically, given an input point cloud $\mathbf{P}$, we seek to learn an encoder that outputs per-point embeddings:

$$\mathbf{F}^{\text{3D}} = \mathcal{E}^{\text{3D}}(\mathbf{P}), \quad \mathcal{E}^{\text{3D}} : \mathbb{R}^{M \times 3} \mapsto \mathbb{R}^{M \times C} \tag{1}$$

where $\mathbf{F}^{\text{3D}} = \{\mathbf{f}_1^{\text{3D}}, \cdots, \mathbf{f}_M^{\text{3D}}\}$. To enforce the output of the network $\mathbf{F}^{\text{3D}}$ to be consistent with the fused features $\mathbf{F}^{\text{2D}}$, we use a cosine similarity loss:

$$\mathcal{L} = 1 - \cos(\mathbf{F}^{\text{2D}}, \mathbf{F}^{\text{3D}}) \tag{2}$$

We use MinkowskiNet18A [11] as our 3D backbone $\mathcal{E}^{\text{3D}}$, and change the dimension of outputs to C.

Since the open-vocabulary image embeddings from [17, 30] are co-embedded with CLIP features, the output of our distilled 3D model naturally lives in the same embedding space as CLIP. Therefore, even without any 2D observations, such text-3D co-embeddings $\mathbf{F}^{\text{3D}}$ allow 3D scene-level understanding given arbitrary text prompts. We show such results in the ablation study in Sec. 4.2.

3.3. 2D-3D Feature Ensemble

Although one can already perform open-vocabulary queries with the 2D fused features $\mathbf{F}^{\text{2D}}$ or 3D distilled features $\mathbf{F}^{\text{3D}}$, here we introduce a 2D-3D ensemble method to obtain a hybrid feature to yield better performance.

The inspiration comes from the observation that 2D fused features specialize in predicting small objects (e.g. a mug on the table) or ones with ambiguous geometry (e.g., a painting on the wall), while 3D features yield good predictions for objects with distinctive shapes (e.g. walls and floors). We aim to combine the best of both.

Our ensemble method leverages a set of text prompts, either provided at inference or offline (e.g. predefined classes from public benchmarks like ScanNet, or arbitrary classes defined by users). We first compute the embeddings for all the text prompts using the CLIP [43] text encoder $\mathcal{E}^{\text{text}}$, denoted as $\mathbf{T} = \{\mathbf{t}_1, \cdots, \mathbf{t}_N\} \in \mathbb{R}^{N \times C}$, where N is the number of text prompts and C the feature dimension. Next, for each 3D point, we obtain its 2D fused and 3D distilled embeddings $\mathbf{f}^{\text{2D}}$ and $\mathbf{f}^{\text{3D}}$ (dropping the subscript for simplicity). We can now correlate text features with these two sets of features via cosine similarity, respectively:

$$\mathbf{s}_n^{\text{2D}} = \cos(\mathbf{f}^{\text{2D}}, \mathbf{t}_n), \qquad \mathbf{s}_n^{\text{3D}} = \cos(\mathbf{f}^{\text{3D}}, \mathbf{t}_n) \tag{3}$$

Once having the similarity scores wrt. every text prompt $\mathbf{t}_n$, we can use the max value $\mathbf{s}^{\text{2D}} = \max_n(\mathbf{s}_n^{\text{2D}})$ and $\mathbf{s}^{\text{3D}} = \max_n(\mathbf{s}_n^{\text{3D}})$ among all N prompts as the *ensemble scores* for both features. Our final 2D-3D ensemble feature $\mathbf{f}^{\text{2D3D}}$ is simply the feature with the highest ensemble score.

3.4. Inference

With any per-point feature described in the previous subsections ($\mathbf{f}^{\text{2D}}$, $\mathbf{f}^{\text{3D}}$, or $\mathbf{f}^{\text{2D3D}}$) and CLIP features from an arbitrary set of text prompts, we can estimate their similarities by simply calculating the cosine similarity score between them. We use this similarity score for all of our scene understanding tasks. For example, for the zero-shot 3D semantic segmentation using 2D-3D ensemble features, the final segmentation for each 3D point is computed point-wise by $\operatorname{argmax}_n\{\cos(\mathbf{f}^{\text{2D3D}}, \mathbf{t}_n)\}$.

4. Experiments

We ran a series of experiments to test how well the proposed methods work for a variety of 3D scene understanding tasks. We start by evaluating on traditional closed-set 3D semantic segmentation benchmarks (in order to be able to compare to previous work), and later demonstrate the more novel and exciting open-vocabulary applications in the next section.

Datasets. To test our method in a variety of settings, we evaluate on three popular public benchmarks: ScanNet [12, 46], Matterport3D [5], and nuScenes Lidarseg [4]. These three datasets span a broad gamut of situations – the first two provide RGBD images and 3D meshes of indoor scenes, and the last provides Lidar scans of outdoor scenes. We use all three datasets to compare to alternative methods. Moreover, Matterport3D is a complex dataset with highly

	mIoU					mAcc				
	Bookshelf	Desk	Sofa	Toilet	Mean	Bookshelf	Desk	Sofa	Toilet	Mean
3DGenZ [39]	6.3	3.3	13.1	8.1	7.7	13.4	5.9	49.6	26.3	23.8
MSeg Voting	47.8	40.3	56.5	68.8	53.4	50.1	67.7	69.8	81.0	67.2
Ours - LSeg	**67.1**	**46.4**	**60.2**	**77.5**	**62.8**	**85.5**	69.5	79.0	90.0	81.0
Ours - OpenSeg	64.1	27.4	49.6	63.7	51.2	73.7	**73.4**	**92.5**	**95.3**	**83.7**

Table 1. **Comparison on Zero-shot 3D Semantic Segmentation.** We show quantitative comparison between our method and the most recent zero-shot 3D segmentation approach [39] and a multi-view fusion baseline utilizing MSeg [29]. Following [39], we take 4 classes (bookself, desk, sofa, toilet) out of 20 classes from ScanNet validation set for evaluation. Unlike [39], which requires training on 16 seen classes, our approach does not train with any 2D or 3D ground labels on any classes. Still, both of our variants show significantly better performance in both mIoU and mAcc.

detailed scenes, and thus provides the opportunity to stress open-vocabulary queries.

4.1. Comparisons

Comparison on zero-shot 3D semantic segmentation. We first compare our approach to the most closely related work on zero-shot 3D semantic segmentation: MSeg [29] Voting and 3DGenz [39]. MSeg Voting predicts a semantic segmentation for each posed image using MSeg [29] with mapping to the corresponding label sets. For each 3D point, we perform majority voting of the *logits* from multi-view images. 3DGenZ [39] divides the 20 classes of the ScanNet dataset into 16 seen and 4 unseen classes, and trains a network utilizing the ground truth supervision on the seen classes to generate features for both sets.

Following the experimental setup in [39], we report the mIoU and mAcc values on their 4 unseen classes in Table 1. Our results on those classes is significantly better than [39] (7.7% vs 62.8% mIoU), even though 3DGenz [39] utilizes ground truth data for 16 seen classes and ours does not. We also outperform MSeg Voting. In this case, the difference is mainly because our method (regress CLIP features and then classify) naturally models the similarities and differences between classes, where as the MSeg Voting approach (classify and then vote) treats every class as equally distinct from all other classes (a couch and a love seat are just as different as a couch and an airplane in their model).

Comparison on 3D semantic segmentation benchmarks. In Table 2 we compare our approach with both fully-supervised and zero-shot methods on all classes of the nuScenes [4] validation set, ScanNet [12] validation set, and Matterport3D [5] test set. Again, we outperform the zero-shot baseline (MSeg Voting) on both mIoU and mAcc metrics all three datasets. Although we have noticeable gap to the state-of-the-art fully-supervised approaches, our zero-shot method is surprisingly competitive with fully-supervised approaches from a few years ago [13, 25, 50]. Among all 3 datasets our approach has the smallest gap (only -11.6 mIoU and -8.0 mAcc) to the SOTA fully-supervised approach on Matterport3D. We conjecture the reason is that Matterport3D is more diverse, which makes the fully-supervised training harder.

	nuScenes [4]		ScanNet [12]		Matterport [5]	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
Fully-supervised methods						
TangentConv [50]	-	-	40.9	-	-	46.8
TextureNet [25]	-	-	54.8	-	-	63.0
ScanComplete [13]	-	-	56.6	-	-	44.9
DCM-Net [48]	-	-	65.8	-	-	66.2
Mix3D [41]	-	-	**73.6**	-	-	-
VMNet [23]	-	-	73.2	-	-	**67.2**
LidarMultiNet [58]	**82.0**	-	-	-	-	-
MinkowskiNet [11]	78.0	83.7	69.0	77.5	54.2	64.6
Zero-shot methods						
MSeg [29] Voting	31.0	36.9	45.6	54.4	33.4	39.0
Ours - LSeg	36.7	42.7	**54.2**	66.6	**43.4**	53.5
Ours - OpenSeg	**42.1**	**61.8**	47.5	**70.7**	42.6	**59.2**

Table 2. **Comparisons on Indoor and Outdoor Benchmarks.** We compare our method with both zero-shot and fully-supervised baselines for semantic segmentation of one outdoor dataset (nuScenes) and two indoor datasets (ScanNet and Matterport). Note that our zero-shot method performs worse than SOTA approaches trained on this data, but comparable to supervised approaches from a few years ago, and better than the previous SOTA zero-shot approach. Except for [11], the numbers for fully-supervised methods (in gray) are taken from previous papers.

Visual comparisons of semantic segmentations are shown in Fig. 4. They show that some of the predictions marked wrong in our results are actually either incorrect or ambiguous ground truth annotations. For example, in the first row in Fig. 4, we successfully segment the picture on the wall, while the GT misses it. And in the nuScenes results, the truck composed of a trailer and the truck head is segmented correctly, but the annotation is not fine-grained enough to separate the parts.

Impact of increasing the number of object classes. Besides the standard benchmarks with only a small set of classes, we also show comparisons as the number of object classes increases. We evaluate on the most frequent K classes[1] of Matterport3D, where $K = 21, 40, 80, 160$. For the baseline, we train a separate MinkowskiNet for each K.

[1] $K = 21$ was from original Matterport3D benchmark. For $K = 40, 80, 160$ we use most frequent K classes of the NYU label set provided with the benchmark.

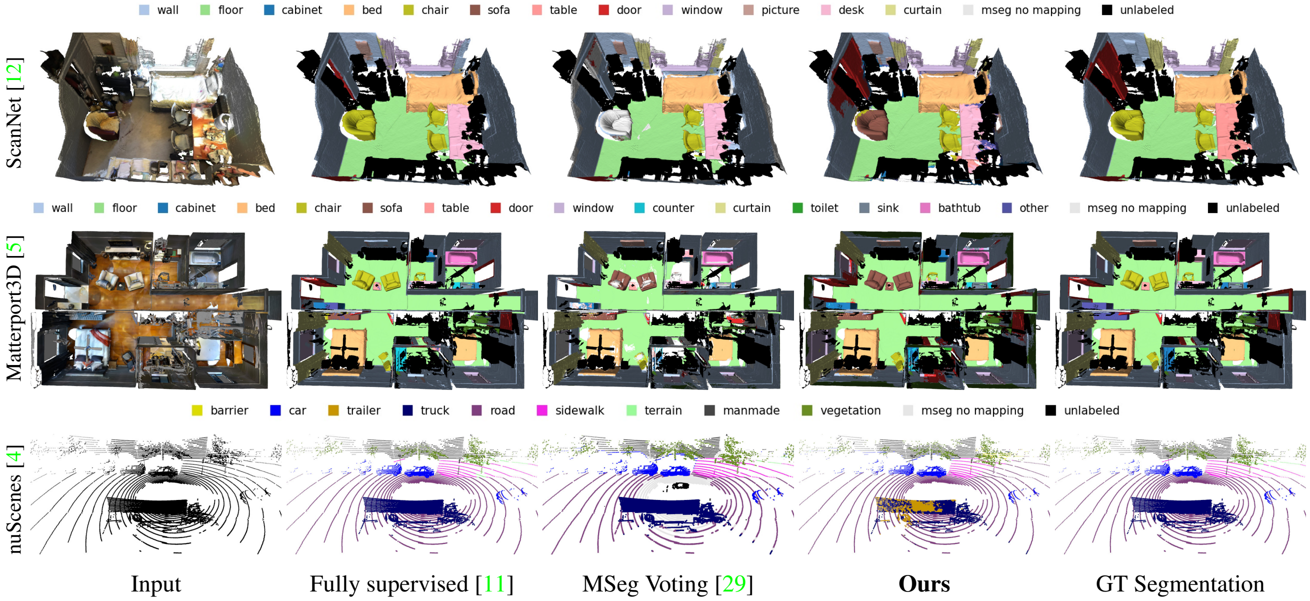
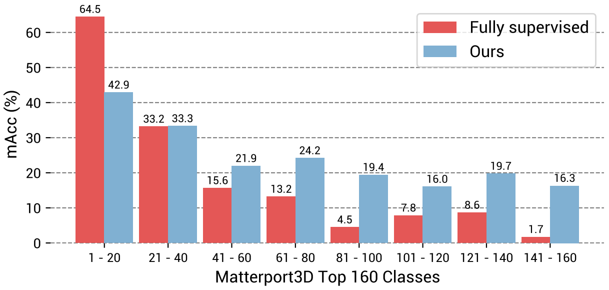


Figure 4. **Qualitative comparisons.** Images of 3D semantic segmentation results on public indoor and outdoor benchmarks.

	$K = 21$	$K = 40$	$K = 80$	$K = 160$
Fully-supervision [11]	**64.5**	50.8	33.4	18.4
Ours	59.2	**50.9**	**34.6**	**23.1**

(a) Results on different number of classes in mAcc



(b) Window evaluation on groups of 20 classes

Table 3. **Impact of Increasing the Number of Object Classes.** Here we show (a) mAcc on Matterport3D [5] with different numbers of classes K, and (b) mAcc within a window of 20 classes ranked by decreasing numbers of examples in training set, i.e. the right-most bars represent average of the 20 classes with fewest examples (e.g., only 5 instances). Even though the fully-supervised approach [11] is trained on each labelset separately, while our model is fixed for all label sets, we can handle the less-common / long-tail classes much better.

		ScanNet [12]		Matterport3D [5]	
		mIoU	mAcc	mIoU	mAcc
Ours LSeg	2D Fusion	50.0	62.7	32.3	40.0
	3D Distill	52.9	63.2	41.9	51.2
	2D-3D Ens.	**54.2**	66.6	**43.4**	53.5
Ours OpenSeg	2D Fusion	41.4	63.6	32.4	45.0
	3D Distill	46.0	66.3	41.3	55.1
	2D-3D Ens.	47.5	**70.7**	42.6	**59.2**

Table 4. **Ablation Study.** Comparison of semantic segmentation performance of different 3D features computed by our method.

However, for ours we use the *same* model for all K, since it is class agnostic.

As shown in Table 3 (a), when trained on only 21 classes, the fully-supervised method performs much better due to the rich 3D labels in the most common classes (wall, floor, chair, etc.). However, with the increase of the number of classes, our zero-shot approach overtakes the fully-supervised approach, especially when K gets large. The reason is demonstrated in Table 3 (b), where we show the mean accuracy for groups of 20 classes ranked by frequency. Fully-supervised suffers severely in segmenting tail classes because there are only a few instances available in the training dataset. In contrast, we are more robust to such rare objects since we do not rely upon any 3D labeled data.

4.2. Ablation Studies & Analysis

Does it matter which 2D features are used? We tested our method with features extracted from both OpenSeg [17] and LSeg [30]. In most experiments, we found the accuracy and generalizability of OpenSeg features to be better than LSeg (Table 1, Table 2, and Table 4), so we use OpenSeg for all experiments unless explicitly stated otherwise.

Is our 2D-3D ensemble method effective? In Table 4, we ablate the performance for predicting features on 3D points including only image feature fusion (Sec. 3.1), only running the distilled MinkowskiNet (Sec. 3.2), and our full 2D-3D ensemble model (Sec. 3.3). As can be seen, on all scenarios (different datasets, metrics, and 2D features), our proposed 2D-3D ensemble model performs the best. This suggests that leveraging patterns in both 2D and 3D domains makes the ensemble features more robust and descriptive.

	$K = 21$	$K = 40$	$K = 80$	$K = 160$
2D Features	28.56%	29.96%	31.58%	32.46%
3D Features	71.44%	70.04%	68.42%	67.54%

Table 5. **Behavior of Ensemble Model.** Each entry indicates the percentage of points for which the Ensemble Model selects 2D or 3D features for semantic segmentation on Matterport3D for different numbers of classes K in the labelset.

What features does our 2D-3D ensemble method use most? Here we study how our ensemble model selects among the 2D and 3D features, and investigate how it changes with increasing numbers of classes in the label set. As shown in Table 5, we find that the majority of predictions ($\sim 70\%$) select the 3D features, corroborating the value of our 3D distillation model. However, the percentage of predictions coming from 2D features increases with the number of classes, suggesting that the 2D features are more important for long-tailed classes, which tend to be smaller in both size and number of training examples.

5. Applications

This section investigates new 3D scene understanding applications enabled by our approach. Since the feature vectors estimated for every 3D point are co-embedded with text and images, it becomes possible to extract information about a scene using arbitrary text and image queries. The following are just a few example applications.[2]

Open-vocabulary 3D object search. We first investigate whether it is possible to query a 3D scene database to find examples based on their names – e.g., "find a teddy bear in the Matterport3D test set." To do so, we ask a user to enter an arbitrary text string as a query, encode the CLIP embedding vector for the query, and then compute the cosine-similarity of that query vector with the features of every 3D point in the Matteport3D test set (containing 18 buildings with 406 indoor and outdoor regions) to discover the best matches. In our implementation, we return at most one match per region (i.e., room, as defined in the dataset) to ensure diversity of the retrieval results.

Fig. 5 shows a few example top-1 results. Most other specific text queries yield nearly perfect results. To evaluate that observation quantitatively, we chose a sampling of 10 raw categories from the ground truth set of Matterport3D, retrieved the best matching 3D points from the test set, and then visually verified the correctness of the top matches. For each query, Table 6 reports the numbers of instances in the test set (# Test) along with the number of instances found with 100% precision before the first mistake in the ranked list. The results are very encouraging. In all of these queries, only two ground truth instances were missed (two telephones). On the other hand, 26 instances were found among these top matches that were not correctly labeled in the ground truth, including 13 telephones. Overall, these results suggest that our open-vocabulary retrieval application identifies these relatively rare classes at least as well as the manually labeling process did. See the supplemental material for the full set of results.

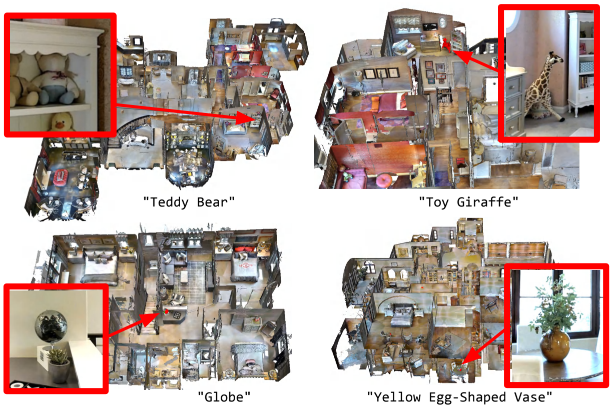


Figure 5. **Open-vocabulary 3D Search.** These images show the 3D point within a database of 3D house models that best matches a text query. The inset image shows a zoomed view of the match.

Class Name	# All	# Test	# Found	# Missed	# New
Fire Extinguisher	25	3	3	0	0
Telephone	21	4	15	2	13
Exit Sign	15	5*	8	0	3
Piano	15	1	2	0	1
Ball	15	1*	4	0	3
Hat	15	1*	1	0	0
Bulletin Board	6	0	1	0	1
Globe	5	2	5	0	3
Teddy Bear	2	0	1	0	1
Toy Giraffe	1	1	1	0	0
Yellow Egg-Shaped Vase	1	0	1	0	1

Table 6. **Open-vocabulary 3D Search Results.** Each row depicts a search of the Matterport3D test set for a class given as a text query. The columns list the # of instances in the ground truth for the whole dataset (# All), the # in the test set (# Test, counting clusters of nearby objects as one when marked with a '*'), the # of top matches found with 100% precision (# Found), the # of GT instances missed amongst those top matches (# Missed), and the # newly discovered that were not in the GT (# New).

Image-based 3D object detection. We next investigate whether it is possible to query a 3D scene database to retrieve examples based on similarities to a given input image – e.g., "find points in a Matterport3D building that match this image." Given a set of query images, we encode them with CLIP image encoder, compute cosine-similarities to 2D-3D ensemble features for 3D points, and then threshold to produce a 3D object detection and mask, see Fig. 7. Note that the pool table and dining table are identified correctly, even though both are types of "tables."

Open-vocabulary 3D scene understanding and exploration. Finally, we investigate whether it is possible to

[2] Please note that all of these applications are zero-shot – i.e., none of them leverage any labeled data from any 3D scene understanding dataset.

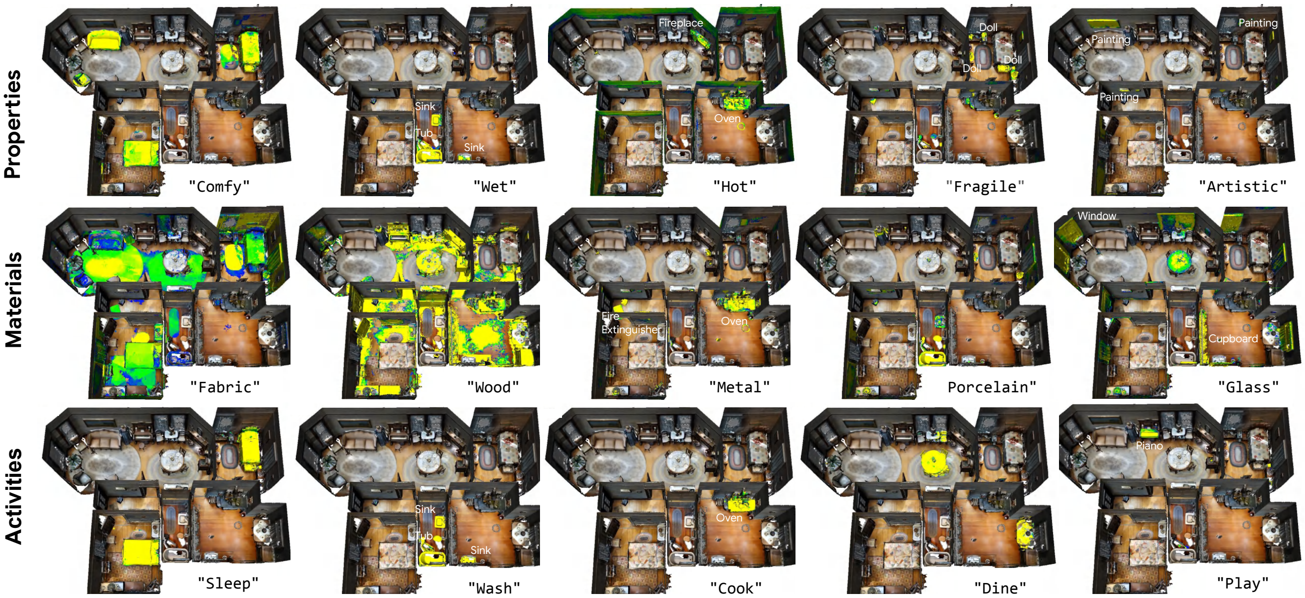


Figure 6. **Open-vocabulary 3D Scene Exploration**. Examples of discovering properties, surface materials, and activity sites within a scene using open-vocabulary queries. For each example, the query text is listed below (e.g., "Comfy"), and the 3D points are colored based on their cosine similarity to the clip embedding for the query text – yellow is highest, green is middle, blue is low, and uncolored is lowest.

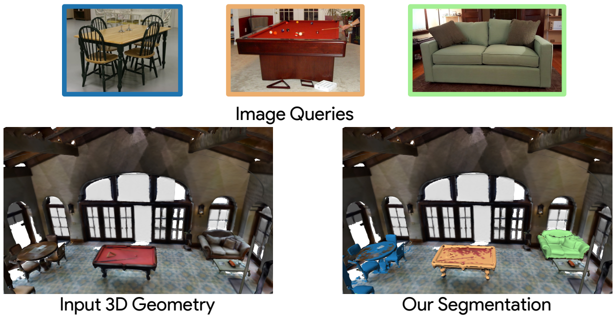


Figure 7. **Image-based 3D Object Detection.** A 3D scene (bottom left) can be queried with images from Internet (top) to find matching 3D points (bottom right). The colors of the image query outlines indicate the corresponding matches in the 3D point cloud. All 3 images are under Creative Commons licenses.

query a 3D scene to understand properties that extend beyond category labels. Since the CLIP embedding space is trained with a massive corpus of text, it can represent far more than category labels – it can encode physical properties, surface materials, human affordances, potential functions, room types, and so on. We hypothesize that we can use the co-embedding our 3D points with the CLIP features to *discover* these types of information about a scene.

Fig. 6 shows some example results for querying about physical properties, surface materials, and potential sites of activities. From these examples, we find that the OpenScene is indeed able to relate words to relevant areas of the scene – e.g., the beds, couches, and stuffed chairs match "Comfy," the oven and fireplace match "Hot," and the piano keyboard matches "Play." This diversity of 3D scene understanding would be difficult to achieve with fully supervised methods without massive 3D labeling efforts. In the authors' opinion, this is the most interesting result of the paper.

6. Limitations and Future Work

This paper introduces a task-agnostic method to co-embed 3D points in a feature space with text and image pixels and demonstrates its utility for zero-shot, open-vocabulary 3D scene understanding. It achieves state-of-the-art for zero-shot 3D semantic segmentation on standard benchmarks, outperforms supervised approaches in 3D semantic segmentation with many class labels, and enables new open-vocabulary applications where arbitrary text and image queries can be used to query 3D scenes, all without using any labeled 3D data. These results suggest a new direction for 3D scene understanding, where foundation models trained from massive multi-modal datasets guide 3D scene understanding systems rather than training them only with small labeled 3D datasets.

There are several limitations of our work and still much to do to realize the full potential of the proposed approach. First, the inference algorithm could probably take better advantage of pixel features when images are present at test time using earlier fusion (we tried this with limited success). Second, the experiments could be expanded to investigate the limits of open-vocabulary 3D scene understanding on a wider variety of tasks. We evaluated extensively on closed-set 3D semantic segmentation, but provide only qualitative results for other tasks since 3D benchmarks with ground truth are scarce. In future work, it will be interesting to design experiments to quantify the success of open vocabulary queries for tasks where ground truth is not available.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 3

[2] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 2

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 2

[4] 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, 4, 5, 6

[5] 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. In *3DV*, 2017. 1, 2, 4, 5, 6

[6] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *ECCV*, 2020. 2

[7] Ali Cheraghian, Shafin Rahman, Dylan Campbell, and Lars Petersson. Mitigating the hubness problem for zero-shot learning of 3d objects. In *BMVC*, 2019. 3

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

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

[10] Ali Cheraghian, Shafin Rahman, and Lars Petersson. Zero-shot learning of 3d point cloud objects. In *MVA*, 2019. 3

[11] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 2, 4, 5, 6

[12] 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. 1, 2, 4, 5, 6

[13] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *CVPR*, 2018. 5

[14] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *CVPR*, 2021. 2

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2

[16] Kyle Genova, Xiaoqi Yin, Abhijit Kundu, Caroline Pantofaru, Forrester Cole, Avneesh Sud, Brian Brewington, Brian Shucker, and Thomas Funkhouser. Learning 3d semantic segmentation with only 2d image supervision. In *3DV*, 2021. 2

[17] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. In *ECCV*, 2022. 2, 3, 4, 6

[18] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022. 3

[19] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In *CoRL*, 2022. 3

[20] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *CVPR*, 2020. 2

[21] Sunan He, Taian Guo, Tao Dai, Ruizhi Qiao, Bo Ren, and Shu-Tao Xia. Open-vocabulary multi-label classification via multi-modal knowledge transfer. In *AAAI*, 2023. 3

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

[23] Zeyu Hu, Xuyang Bai, Jiaxiang Shang, Runze Zhang, Jiayu Dong, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew-Lan Tai. Vmnet: Voxel-mesh network for geodesic-aware 3d semantic segmentation. In *ICCV*, 2021. 2, 5

[24] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *3DV*, 2016. 2

[25] Jingwei Huang, Haotian Zhang, Li Yi, Thomas Funkhouser, Matthias Nießner, and Leonidas J Guibas. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. In *CVPR*, 2019. 2, 5

[26] 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. 3

[27] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *ECCV*, 2020. 2

[28] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. In *ICLR*, 2023. 3

[29] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *CVPR*, 2020. 5, 6

[30] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 2, 3, 4, 6

[31] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In *CVPR*, 2022. 2

[32] Xueting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Putting humans in a scene: Learning affordance in 3d indoor environments. In *CVPR*, 2019. 2

[33] 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. 3

[34] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE TPAMI*, 2022. 2

[35] Bo Liu, Shuang Deng, Qiulei Dong, and Zhanyi Hu. Language-level semantics conditioned 3d point cloud segmentation. *arXiv preprint arXiv:2107.00430*, 2021. 3

[36] Yunze Liu, Qingnan Fan, Shanghang Zhang, Hao Dong, Thomas Funkhouser, and Li Yi. Contrastive multimodal fusion with tupleinfonce. In *ICCV*, 2021. 3

[37] Chaofan Ma, Yuhuan Yang, Yanfeng Wang, Ya Zhang, and Weidi Xie. Open-vocabulary semantic segmentation with frozen vision-language models. In *BMVC*, 2022. 3

[38] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *ICRA*, 2017. 2

[39] 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. 3, 5

[40] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *CVPR*, 2019. 2

[41] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. In *3DV*, 2021. 2, 5

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

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 3, 4

[44] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, 2022. 2, 3

[45] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In *CVPR*, 2022. 2

[46] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*, 2022. 2, 4

[47] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *CVPR*, 2022. 2, 3

[48] Jonas Schult, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dualconvmesh-net: Joint geodesic and euclidean convolutions on 3d meshes. In *CVPR*, 2020. 2, 5

[49] 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. 2

[50] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *CVPR*, 2018. 5

[51] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 2

[52] Vibhav Vineet, Ondrej Miksik, Morten Lidegaard, Matthias Nießner, Stuart Golodetz, Victor A Prisacariu, Olaf Kähler, David W Murray, Shahram Izadi, Patrick Pérez, et al. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *ICRA*, 2015. 2

[53] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *CVPR*, 2021. 2

[54] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM TOG*, 2017. 2

[55] 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 *CoRL*, 2022. 2

[56] 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

[57] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022. 2, 3

[58] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. *arXiv preprint arXiv:2209.09385*, 2022. 5

[59] Nir Zabari and Yedid Hoshen. Open-vocabulary semantic segmentation using test-time distillation. In *ECCVW*, 2022. 2, 3

[60] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 2, 3