CLIP-FO3D: Learning Free Open-world 3D Scene Representations from 2D Dense CLIP

Junbo Zhang\(^1\)  Runpei Dong\(^2\)  Kaisheng Ma\(^1\)*
\(^1\)Tsinghua University  \(^2\)Xi’an Jiaotong University
zhangjb21@mails.tsinghua.edu.cn, runpei.dong@stu.xjtu.edu.cn, kaisheng@mail.tsinghua.edu.cn

Abstract

Training a 3D scene understanding model requires complicated human annotations, which are laborious to collect and result in a model only encoding close-set object semantics. In contrast, vision-language pre-training models (e.g., CLIP) have shown remarkable open-world reasoning properties. To this end, we propose directly transferring CLIP’s vision feature space to 3D scene understanding model without any form of supervision. We first modify CLIP’s input and forwarding process so that it can be adapted to extract dense pixel features for 3D scene contents. We then project multi-view image features to the point cloud and train a 3D scene understanding model with feature distillation. Without any annotations or additional training, our model achieves promising annotation-free semantic segmentation results on open-vocabulary semantics and long-tailed concepts. Besides, serving as a cross-modal pre-training framework, our method can be used to improve data efficiency during fine-tuning. Our model outperforms previous SOTA methods in various zero-shot and data-efficient learning benchmarks. Most importantly, our model successfully inherits CLIP’s rich-structured knowledge, allowing 3D scene understanding models to recognize not only object concepts but also open-world semantics.

1. Introduction

3D scene understanding aims to distinguish objects’ semantics, identify their locations, and infer the geometric attributes from 3D scene data. It has a wide range of applications in virtual reality [53], robot navigation [3, 51] and autonomous driving [38, 75, 76]. However, training traditional 3D scene understanding models requires a large number of human annotations, which are laborious to collect. Besides, the human annotations used in the current 3D scene understanding benchmark only contain close-set semantic information of the objects (e.g., 20 classes in ScanNet [15]). These make it difficult for 3D scene understanding systems to recognize open-vocabulary categories and to infer open-world semantics.

Large-scale vision-language foundation models (e.g., CLIP [57]) capture rich visual and language features. They only require image-text pairs mined from the Internet for unsupervised pre-training, and have demonstrated superior ability in zero-shot and open-vocabulary reasoning for classification and dense prediction tasks [25, 39, 79, 91]. However, due to the difficulty of collecting 3D-text pairs and the complexity of scene data, is extremely challenging to build analogous 3D foundation models.

Some pioneering works explore the potential of extending CLIP’s text feature space to 3D scene representations. LSeg [39]/OpenSeg [25] models (used in OpenScene [54]) and CLIP2Scene [6] align the pixel/point features with class names’ CLIP text embeddings. However, the text embeddings contain much less knowledge than CLIP’s vision encoder, because some attributes, rare and abstract concepts would be difficult to label in the training texts. Besides, OpenScene [54] requires 2D segmentation annotations for the pre-trained models. CLIP2Scene [6] requires close-set pre-defined classes for training. This will result in a loss of CLIP’s powerful open-world properties because training with pre-defined classes restricts the feature space to a lim-
ized set of vocabularies. In this paper, we propose directly transferring CLIP’s
vision feature space to 3D scene understanding models without any supervision (e.g., 2D/3D annotations or vision-language grounding annotations). Once the transfer is complete, our model can complete the open-vocabulary 3D scene understanding task without additional training. This annotation-free paradigm is both effort-saving and able to preserve CLIP’s open-world information to the maximum.

In this direction, MaskCLIP [91] in 2D vision has taken the first step towards extracting pixel-level features from CLIP vision encoder’s final feature map. However, it has difficult adapting to 3D contents since 3D scenes contain much more objects and more complex structures. Besides, MaskCLIP has an inherent defect in precisely locating and segmenting objects’ boundaries due to the limited resolution of CLIP’s final feature map. To address these problems, we propose a new method to extract free pixel-level CLIP features without breaking its feature space. We first crop the input images at multi-scale to accommodate various object sizes. To preserve object-level semantics, we divide each cropped sample into semantically relevant regions and extract a local feature for each region. Specifically, we add several local classification tokens in ViT [21] encoder, which only aggregate information from local patches within a region. By simply forwarding CLIP and combining multiple local features, we extract a pixel-level feature map for each 3D scene’s RGB view.

After extracting pixel-level CLIP features, we adopt the feature projection in 3DMV [16] to project multi-view image features to the point cloud. The resulting point features are aligned with CLIP’s feature space and used as off-line training targets. Then we train the 3D understanding model with feature distillation, minimizing the distance between learned point features and the target features as in [54]. This way, we obtain CLIP-FO3D, which extracts free and open-world 3D scene representations aligned with CLIP.

Our CLIP-FO3D can perform annotation-free open-vocabulary 3D semantic segmentation without additional training processes. Since CLIP’s vision features are well aligned with text features, we can take the text embeddings of each class name’s prompts as classification weights to perform semantic segmentation. CLIP-FO3D performs remarkably on standard close-set ScanNet [15] and S3DIS [4] segmentation benchmark. In addition, to examine CLIP-FO3D’s open-vocabulary capability inherited from CLIP, we extend the label set of the standard ScanNet dataset with NYU labels [63]. We demonstrate that CLIP-FO3D has remarkable segmentation results on NYU-40 classes and other long-tailed categories beyond the NYU label set.

Besides, given that collecting 3D point clouds and annotating are laborious, CLIP-FO3D can also be regarded as an unsupervised cross-modal pre-training framework to bene-
fit data efficiency. CLIP-FO3D achieves promising performance in traditional benchmarks where limited annotations are provided, such as zero-shot and data-efficient learning.

Most importantly, CLIP-FO3D encodes rich open-world knowledge inherited from CLIP. It understands not only object concepts, but also text queries with open-world semantics (e.g., affordances, colors, activities), broadening the application of 3D scene understanding.

Our contributions can be summarized as follows:

- We propose directly transferring CLIP’s vision feature space to 3D scene understanding models without any annotation, which preserves CLIP’s open-world properties to the maximum.
- We present a novel method to extract pixel-level CLIP features, and a feature distillation method to align 3D point representations with CLIP’s feature space.
- Our model achieves promising annotation-free 3D semantic segmentation performance on large vocabularies, and shows remarkable open-world properties.
- As an unsupervised pre-training method, our model outperforms previous state-of-the-art methods in zero-shot and data-efficient learning.

2. Related Work

2.1. Open-vocabulary Dense Prediction in 2D Vision

Open-vocabulary dense prediction aims to recognize and localize objects with open-set semantics. Pioneering works [43, 71, 81, 92] in 2D vision mainly utilize large-scale image-caption annotations as weak supervision source to enlarge the vocabulary set. Other works [22, 23, 25, 26, 39, 41, 47, 49, 58, 70, 80, 90] distill vision-language foundation models’ (e.g., CLIP) knowledge to transfer their open-vocabulary capability. However, they all require some form of human annotations, such as box/mask proposals and pixel semantic annotations. Open-world knowledge encoded in CLIP is forgotten when fine-tuning with human annotations. In contrast, MaskCLIP [91] proposes directly utilizing CLIP for dense prediction tasks without training. However, it struggles to handle 3D scene’s RGB views with more complex contents, as shown in Section 4. We propose a new method to extract free pixel-level CLIP features from 3D scene’s RGB views by only modifying CLIP’s inputs and forwarding process without fine-tuning.

2.2. Zero-shot 3D Visual Recognition

Zero-shot learning is relatively under-explored in 3D vision. Initial works focus on classification tasks on object-level data [10–13, 29, 55, 72]. Recent works [33, 42, 83, 86] adopt CLIP to perform zero-shot 3D classification with open-vocabularies. Multi-modal foundation models are also
used in other 3D tasks, such as NeRF [35, 37, 65] and 3D visual question answering [17].

For 3D scene understanding, Michele et al. [48] and Chen et al. [7] study zero-shot semantic segmentation with generative model and word embedding prototypes. Lu et al. [46] propose a zero-shot 3D object detection method with pseudo-labels generated with 2D classifier. Zhang et al. [89], Chen et al. [6] and Rozenberszki et al. [61] propose fully unsupervised 3D semantic/instance segmentation methods. But these methods still result in a model with close-set vocabularies. Recent works achieve promising results in applying CLIP to open-vocabulary 3D scene understanding. However, they all require human annotations for training, such as 2D pixel labels [54], 3D point labels [27] and pre-trained masks [34, 64]. PLA [18] and RegionPLC [73] utilize an image-captioning model to generate captions for 3D content, enabling open-vocabulary segmentation by aligning 3D representations with text embeddings. Unlike existing works, we propose directly transferring CLIP’s vision feature space to 3D scene representations with no annotations. Without additional training, CLIP-FO3D shows superior open-world understanding properties inherited from CLIP.

2.3. 3D Representation Learning

Inspired by the self-supervised representation learning in 2D vision, 3D pre-training methods [30, 82] achieve better fine-tuning performance and efficiency in various downstream tasks by leveraging contrastive learning [8, 9, 44, 68, 88], masked auto-encoder [20, 52, 78, 85], or both [55].

Beyond self-supervised pre-training, cross-modal learning methods propose to distill knowledge in images/text and pre-trained models to 3D representations [2, 20, 32, 55, 59]. For scene understanding tasks, recent works enrich 3D point representations by utilizing 2D/text annotations [50, 77, 84], pixel-point alignments [40, 45, 62, 66], neural rendering [31] and pre-trained models [56, 60, 69, 74, 87]. Our method can be regarded as an unsupervised cross-modal 3D representation learning method. We demonstrate that distilling CLIP’s richly-structured vision knowledge to 3D models can benefit 3D scene understanding when limited annotations are available, outperforming previous self-supervised and cross-modal pre-training methods.

3. Method

This section describes the process of transferring CLIP’s feature space to 3D scene understanding models. We first extract pixel-level CLIP features from the 3D scene’s RGB views by modifying CLIP’s inputs and forwarding process, introduced in Section 3.1. We then get target point features from pixel features and train the 3D model by feature distillation, introduced in Section 3.2.

3.1. Extract Free Pixel-level CLIP Features

Although CLIP only aligns image-level global features with text embeddings, it should inherently encode local and dense semantics at the front layers. As demonstrated in MaskCLIP [91], CLIP must divide image-level semantics
into local segments, and properly align each segment’s semantics with independent concepts in the text. MaskCLIP proposes to discard the global pooling layer and extract dense features from the final feature map with reformulated 1 × 1 convolutional layers.

However, adapting MaskCLIP to 3D scene contents brings poor dense prediction results, as shown in Section 4. On the one hand, CLIP’s feature map has a much lower resolution than the input images (downsampled by 16² in ViT/16 [21] and 32² in ResNet-50 [28]). Although MaskCLIP is reasonably capable of recognizing salient semantics in the images, it struggles to segment the numerous objects in 3D scenes at different scales. On the other hand, the pixel features in CLIP’s final feature map contain much global semantics regarding the entire image. They may contain multiple objects’ semantics since each pixel’s feature aggregates information from all other pixels in forwarding. However, ideally, we hope to extract object-level features of different objects in a 3D scene. We propose increasing the resolutions and extracting local features from CLIP to address the aforementioned problems, described as follows.

**Multi-scale region extraction.** Firstly, the input view is cropped at multi-scales to adapt to the recognition of objects of various sizes in 3D scenes, as shown in Figure 2 (a). For each cropped sample, we divide into many local regions. This effectively improves the feature resolution. Specifically, we divide the image sample into super-pixels with SLIC [1] as in [62]. The super-pixel roughly covers an object or object’s part, resulting in locally visually similar regions. After processing the input image, we extract an embedding vector for each super-pixel with modified CLIP’s ViT [21] encoder, which is introduced below.

**Extracting local features from CLIP.** We hope the super-pixel feature aggregates information from local patches rather than from the entire image like the global classification token in ViT’s encoder. This process is shown in Figure 2 (b). Given a cropped image sample Ic, we segment it into N super-pixels: Ic = S1 ∪ S2 ∪ ... ∪ SN, where Si ∩ Sj = ∅, ∀ i ≠ j. We then upsample the exact cropped image to CLIP’s input size and obtain Ic. In ViT, Ic is first reshaped into M flattened patches: Ic = P1 ∪ P2 ∪ ... ∪ PM, where M = 14² in ViT-B/16. We then assign each patch in Ic to a specific super-pixel in Ic based on the their spatial locations by interpolation, since the number of super-pixels N in our method is always less than the number of patches M. We denote the patch Pi being assigned to the super-pixel Sj as Pi ∼ Sj.

To represent each super-pixel’s feature, we add N local classification tokens beyond the original global classification token in CLIP’s forwarding process. The local tokens have similar functions with the group tokens in [71], but are not learnable and have different updating mechanism. The local tokens are initialized the same as the global one and are updated by the same self-attention mechanism and pre-trained weights in ViT. The only difference in updating the local tokens during inference is how attention scores are computed. Namely, each local token is only attended to its assigned patch tokens, which is implemented as masked-attention mechanism as in [19, 71]. Recall that in ViT’s forwarding process, the attention score of the classification token is calculated as:

\[
A_{\text{global}} = \sum_i \text{softmax} \left( \frac{q^g \cdot k_i^T}{C} \right) v_i, \tag{1}
\]

where \(q^g = \text{Emb}_g(x^g)\), \(k_i = \text{Emb}_k(x_i)\), \(v_i = \text{Emb}_v(x_i)\), and \(C\) is a constant scaling factor and \(\text{Emb}(\cdot)\) denotes the linear layers encoding the query, key, and value embeddings. \(x^g\) is the global classification token and \(x_i\) represents the input feature of patch \(P_i\).

In our method, the attention score of each local classification token is computed from local patches as:

\[
A_{\text{local}, j} = \sum_{i: P_i \sim S_j} \text{softmax} \left( \frac{q^l_j \cdot k_i^T}{C} \right) v_i, \tag{2}
\]

where \(P_i \sim S_j\) means that patch \(P_i\) is assigned to the super-pixel \(S_j\). \(x^l_j\) is the local classification token of super-pixel \(S_j\) and \(x_i\) represents the input feature of patch \(P_i\).

Note that the original tokens in ViT are not affected during inference. By forwarding CLIP with additional tokens, we obtain a local feature for each super-pixel that is aligned with CLIP’s feature space. We then apply the same local token feature to all pixels within a super-pixel to preserve object-level semantics. In this way, we obtain a feature map for each cropped image of the same size as CLIP’s input.

**Multi-scale feature fusion.** After extracting local feature maps for all the cropped samples, we resize them back to the cropping sizes and stitch them back to the original input image, as shown in Figure 2 (c). Since one pixel may belong to different super-pixels from different cropped samples, we average all features as the final pixel feature. We extract pixel features for each 3D scene’s RGB view. While the whole process is slow and difficult to apply to real-time inference, we only do the above process once for each view and use the pixel-level features as offline training targets.

### 3.2. Feature Distillation with 2D Teacher

To connect the pixel-level 2D features with 3D point features, we project each point in a 3D scene back to the RGB views following 3DMV [16]. Point-to-pixel projection is computed based on the camera pose, intrinsics, and the world-to-grid transformation matrix. Since we can obtain each projected pixel’s depth value from the RGB-D images, we only keep the points that are in the correct depth ranges.
<table>
<thead>
<tr>
<th>Inference by feature projection</th>
<th>Backbone</th>
<th>ScanNet mIoU</th>
<th>mAcc</th>
<th>S3DIS mIoU</th>
<th>mAcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaskCLIP-3D</td>
<td>CLIP</td>
<td>9.7</td>
<td>21.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Target Feature (Ours)</td>
<td>CLIP</td>
<td>27.6</td>
<td>47.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CLIP-FO3D (Ours)</td>
<td>Res-Unet</td>
<td>30.2</td>
<td>49.1</td>
<td>22.3</td>
<td>32.8</td>
</tr>
</tbody>
</table>

Table 1. Annotation-free semantic segmentation on standard ScanNet and S3DIS datasets. MaskCLIP-3D and our Target Feature use pixel features to infer points’ semantics, which is slow and unsuitable for practical use. Our pixel-level target features capture significantly better semantics than MaskCLIP. CLIP-FO3D trained with feature distillation even outperforms its target.

for further training. As some points will be associated with multiple pixels from different views, we compute the average of all 2D features as the final point feature \( f_{3D} \in \mathbb{R}^C \), where the channel number \( C \) is the same as CLIP’s vision encoder. As a result, we obtain point-level features for each 3D scene denoted as \( F_{\text{target}} = \{ f_{3D,i} \}_{i=1}^{N_p} \), where \( N_p \) is the number of points in the scene.

After extracting offline target point features for each scene, we train the 3D understanding model to learn from these targets by feature distillation like OpenScene [54]. Denoting the learned point features for a scene as \( F_{\text{learn}} = \{ f_{3D,i} \}_{i=1}^{N_p} \), the loss function of feature distillation is:

\[
\mathcal{L} = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathcal{D} \left( f_{3D,i}, \hat{f}_{3D,i} \right). \tag{3}
\]

The distance \( \mathcal{D}(\cdot, \cdot) \) is the negative cosine similarity:

\[
\mathcal{D}(f_1, f_2) = -\frac{f_1}{\|f_1\|_2} \cdot \frac{f_2}{\|f_2\|_2}, \tag{4}
\]

where \( \| \cdot \|_2 \) is \( L_2 \)-norm. We use cosine distance because CLIP-driven classification relies on the cosine distance between vision and text embeddings.

The whole training process only requires the 3D dataset and the pre-trained CLIP vision encoder, without any form of supervision (e.g., 2D/3D annotations or vision-language grounding annotations). Since the learned point features are consistent with CLIP’s feature space, the 3D model can perform open-vocabulary semantic segmentation and open-world reasoning once the feature distillation is finished.

4. Experiments

4.1. Experimental Setup

Dataset. We train CLIP-FO3D on ScanNet’s training set [15]. We use the RGB-D images and the 3D scene meshes for training, and no labels are used. We sub-sample RGB-D images from the raw ScanNet videos every ten frames for each scene. We evaluate our method on ScanNet [15] and S3DIS [4] datasets. ScanNet’s validation set and S3DIS’s “Area 5” are used for all the evaluation experiments. Notice that raw categories’ names and the mapping to NYU40 label set are provided in the ScanNet dataset, which we use to enlarge the vocabulary. We remove the “other furniture” category in ScanNet and the “clutter” category in S3DIS because they do not have specific semantics that can be classified with any text embeddings.

Implementation Details. We use CLIP’s ViT-B/16 encoder as the 2D backbone and use the corresponding text encoder for generating text embeddings. For all the experiments, we adopt MinkowskiNet16 [14] as 3D scene understanding backbone. We remove the final classifier and change the output feature dimension to 512 to match CLIP’s feature dimension. We train CLIP-FO3D using SGD optimizer with a learning rate of 0.8 and a batch size of 4. The model is trained for 80K steps, and the learning rate is decreased by 0.99 for every 1,000 steps. The fine-tuning experiments on CLIP-FO3D are trained with a batch size of 4 for 40K steps. We set the initial learning rate of the pre-trained CLIP-FO3D networks to 0.001, with polynomial decay with power 0.9, because we find that a small learning rate leads to better results when fine-tuning CLIP’s feature space. The initial learning rate of the classifier in data-efficient learning is set to 0.1 following the original setting in [30]. For zero-shot learning and data-efficient learning experiments, we follow the original benchmarks in [7] and [30]. More implementation details can be found in the supplementary material.

4.2. Annotation-free Semantic Segmentation

Semantic segmentation on standard benchmarks. We first present the results of annotation-free semantic segmentation on standard ScanNet and S3DIS benchmarks, where no supervision is provided during training. This is a rather challenging setting. After training CLIP-FO3D with feature distillation, we can take the text embeddings of each class name’s prompts as classification weights to perform semantic segmentation.

The results are shown in Table 1. The first two rows show the results of using multi-view pixel features to infer each point’s semantics by feature projection (introduced in Section 3.2). MaskCLIP-3D is a baseline method that we use MaskCLIP to compute pixel features for RGB views as in [91]. Target Feature is our proposed method to extract free dense CLIP features from RGB views introduced.

<table>
<thead>
<tr>
<th>Head mIoU</th>
<th>mAcc</th>
<th>Common mIoU</th>
<th>mAcc</th>
<th>Tail mIoU</th>
<th>mAcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaskCLIP-3D</td>
<td>19.8</td>
<td>33.1</td>
<td>13.3</td>
<td>26.8</td>
<td>7.5</td>
</tr>
<tr>
<td>Target Feature (Ours)</td>
<td>37.1</td>
<td>63.4</td>
<td>39.0</td>
<td>55.4</td>
<td>40.6</td>
</tr>
<tr>
<td>CLIP-FO3D (Ours)</td>
<td>44.3</td>
<td>65.9</td>
<td>37.6</td>
<td>50.6</td>
<td>26.5</td>
</tr>
</tbody>
</table>

Table 2. Open-vocabulary semantic segmentation on ScanNet with extended labels from NYU label set. We divide all categories into Head, Common and Tail based on the point numbers.
We show the visualization of TGP [7] and MaskCLIP-3D in Figure 3 (left), which demonstrate that our model is capable of recognizing long-tailed categories that are not annotated in traditional benchmarks, indicating that some misleading target features can be corrected during feature distillation.

Semantic segmentation with open vocabularies. The standard ScanNet benchmark only contains a small vocabulary of 20 classes. To examine the open-vocabulary capability of CLIP-FO3D inherited from CLIP, we first extend the original vocabulary size with the NYU-40 label set. We remove the NYU-40 labels that do not have specific semantics (e.g., “other structure”, “other furniture”, “other prop”) and evenly divide all the rest categories into Head, Common and Tail based on the point numbers of each category. We show the semantic segmentation results on the three category sets in Table 2. Our Target Features outperform the baseline MaskCLIP-3D by a large margin. Since categories in Common and Tail set usually contain objects with smaller sizes (e.g., bag, box, pillow, book), MaskCLIP-3D struggles to extract their semantics due to the limited feature resolution.

### Table 3. Zero-shot semantic segmentation on ScanNet with different settings. “Unseen-i” indicates that there are i classes that have no labels during training. $S$ and $U$ represent the performance of seen and unseen classes, respectively.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Unseen-2</th>
<th>Unseen-4</th>
<th>Unseen-6</th>
<th>Unseen-8</th>
<th>Unseen-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>mIoU $S$</td>
<td>hIoU $U$</td>
<td>mIoU $S$</td>
<td>hIoU $U$</td>
<td>mIoU $S$</td>
</tr>
<tr>
<td>3DGenZ [48]</td>
<td>33.4</td>
<td>12.8</td>
<td>18.5</td>
<td>32.8</td>
<td>7.7</td>
</tr>
<tr>
<td>TGP [7]</td>
<td>58.6</td>
<td>51.6</td>
<td>54.9</td>
<td>57.9</td>
<td>34.1</td>
</tr>
<tr>
<td>MaskCLIP-3D</td>
<td>67.3</td>
<td>38.7</td>
<td>49.1</td>
<td>64.4</td>
<td>31.0</td>
</tr>
<tr>
<td>CLIP-FO3D (Ours)</td>
<td>70.6</td>
<td>60.3</td>
<td>65.0</td>
<td>69.5</td>
<td>54.8</td>
</tr>
</tbody>
</table>

Table 4. Zero-shot semantic segmentation on S3DIS with different settings. “Unseen-i” indicates that there are i classes that have no labels during training. $S$ and $U$ represent the performance of seen and unseen classes, respectively.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Unseen-4</th>
<th>Unseen-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>mIoU $S$</td>
<td>hIoU $U$</td>
</tr>
<tr>
<td>3DGenZ [48]</td>
<td>53.1</td>
<td>7.3</td>
</tr>
<tr>
<td>TGP [7]</td>
<td>60.4</td>
<td>20.6</td>
</tr>
<tr>
<td>CLIP-FO3D (Ours)</td>
<td>64.8</td>
<td>26.1</td>
</tr>
</tbody>
</table>

While our method even extracts higher quality features for Tail categories than Head, thanks to the multi-scale inputs and local super-pixel features. CLIP-FO3D also achieves promising results in all categories, while it does not perform as well as our Target Features for Common and Tail categories, due to the limited number of examples in the training set. However, CLIP-FO3D can be easily used for indoor open-vocabulary scene understanding applications.

Qualitative Results. We show the visualization of annotation-free semantic segmentation results on standard ScanNet benchmark and recognizing long-tailed categories. As shown in Figure 3 (left), decent segmentation masks are obtained compared to the ground truth. Our method differs from ground-truth results for some objects with ambiguous semantics, such as table and desk, cabinet and its door, chair and sofa. In Figure 3 (right), our method successfully recognizes long-tailed categories that are not annotated in traditional benchmarks, demonstrating our model’s promising open vocabulary capability.

4.3. Zero-shot Semantic Segmentation

Zero-shot semantic segmentation methods train the 3D scene understanding model only with the labels on a subset of classes (seen classes), and evaluate both seen classes and unseen classes. All the existing methods in 3D scene understanding use transductive settings, in which the unlabeled points are accessible during training. CLIP-FO3D can be applied to zero-shot semantic segmentation with minor effort. Specifically, CLIP-FO3D can be used to generate pseudo-labels for the unlabeled points.

We compare our method with state-of-the-art zero-shot semantic segmentation methods on ScanNet and S3DIS following the benchmarks in [7]. We also use a new benchmark where six classes are chosen as unseen classes in S3DIS. We use the metric of mean intersection over union (mIoU) for both seen classes ($S$) and unseen classes ($U$), and use the harmonic mean IoU (hIoU) to demonstrate the overall performance of zero-shot learning as in [5, 67].

As shown in Table 3, our method outperforms the previous state-of-the-art method [7] by 10.1%, 18.3%, 33.8%, 35.0% and 34.7% on hIoU metric when there are 2, 4, 6, 8 and 10 unseen classes during training. Our CLIP-FO3D also outperforms the method of pseudo-labeling with

<table>
<thead>
<tr>
<th>Setting</th>
<th>Unseen-2</th>
<th>Unseen-4</th>
<th>Unseen-6</th>
<th>Unseen-8</th>
<th>Unseen-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>mIoU $S$</td>
<td>hIoU $U$</td>
<td>mIoU $S$</td>
<td>hIoU $U$</td>
<td>mIoU $S$</td>
</tr>
<tr>
<td>3DGenZ [48]</td>
<td>33.4</td>
<td>12.8</td>
<td>18.5</td>
<td>32.8</td>
<td>7.7</td>
</tr>
<tr>
<td>TGP [7]</td>
<td>58.6</td>
<td>51.6</td>
<td>54.9</td>
<td>57.9</td>
<td>34.1</td>
</tr>
<tr>
<td>MaskCLIP-3D</td>
<td>67.3</td>
<td>38.7</td>
<td>49.1</td>
<td>64.4</td>
<td>31.0</td>
</tr>
<tr>
<td>CLIP-FO3D (Ours)</td>
<td>70.6</td>
<td>60.3</td>
<td>65.0</td>
<td>69.5</td>
<td>54.8</td>
</tr>
</tbody>
</table>

Table 3. Zero-shot semantic segmentation on ScanNet with different settings. “Unseen-i” indicates that there are i classes that have no labels during training. $S$ and $U$ represent the performance of seen and unseen classes, respectively.

Table 4. Zero-shot semantic segmentation on S3DIS with different settings. “Unseen-i” indicates that there are i classes that have no labels during training. $S$ and $U$ represent the performance of seen and unseen classes, respectively.
Table 5. Data-efficient semantic segmentation on ScanNet with limited scene reconstructions. “Sup.” indicates whether points’ semantic labels are used during pre-training.

Table 6. Data-efficient semantic segmentation on ScanNet with limited point annotations. “Sup.” indicates whether points’ semantic labels are used during pre-training.

MaskCLIP-3D by large margins. It is observed that our method is more effective than the baselines when there are more unseen classes, demonstrating superior zero-shot capability. The results on S3DIS with two settings in Table 4 reflect a similar phenomenon, although S3DIS’s data is not accessible during the training of CLIP-FO3D.

4.4. Data-efficient 3D Scene Understanding

As the collection and annotation of 3D point cloud data are very laborious, data-efficient learning methods have been proposed to train a better 3D model when training data or labels are scarce. Existing works have explored self-supervised [30] or cross-modal [60] pre-training methods for data-efficient fine-tuning. CLIP-FO3D can also be considered an unsupervised cross-modal pre-training method to benefit data efficiency. Unlike existing works, our method leverages the richly-structured feature space inherited from CLIP to improve data-efficient fine-tuning results.

We follow the official data-efficient learning benchmarks [30]: Limited Reconstructions (only a few labeled scenes are used for training) and Limited Annotations (only a few points are labeled in each scene). “Scratch” denotes the training from scratch baseline. We also remove the “other furniture” category in ScanNet as in the annotation-free segmentation experiments. The results of limited scene reconstructions are shown in Table 5, when only 1%, 5%, 10%, and 20% scenes are used during training. Our method achieves mIoU improvements of 8.0%, 4.3%, 4.2% and 3.0% over training from scratch, and achieves superior results compared to the previous state-of-the-art supervised pre-training method. The results of limited point annotations are shown in Table 6, when 20, 50, 100, and 200 annotated points are randomly sampled for each scene. Our method achieves mIoU improvements of 11.3%, 6.0%, 5.4%, and 4.1% over training from scratch, and outperforms previous state-of-the-art supervised pre-training methods.

5. Discussion

Ablation study. Table 7 shows the ablation study on ScanNet for two benchmarks. “Multi-scale” represents cropping input images at multi-scales when extracting pixel-level features with CLIP. “Local Feature” represents using local classification tokens to extract super-pixel features.

We follow the official data-efficient learning benchmarks [30]: Limited Reconstructions (only a few labeled scenes are used for training) and Limited Annotations (only a few points are labeled in each scene). “Scratch” denotes the training from scratch baseline. We also remove the “other furniture” category in ScanNet as in the annotation-free segmentation experiments. The results of limited scene reconstructions are shown in Table 5, when only 1%, 5%, 10%, and 20% scenes are used during training. Our method achieves mIoU improvements of 8.0%, 4.3%, 4.2% and 3.0% over training from scratch, and achieves superior results compared to the previous state-of-the-art supervised pre-training method. The results of limited point annotations are shown in Table 6, when 20, 50, 100, and 200 annotated points are randomly sampled for each scene. Our method achieves mIoU improvements of 11.3%, 6.0%, 5.4%, and 4.1% over training from scratch, and outperforms previous state-of-the-art supervised pre-training methods.
method outperforms the MaskCLIP-3D baseline, increasing the feature resolution with multi-scale inputs brings more significant improvements than extracting local features.

Open-world 3D scene understanding. Since CLIP is trained with massive image-text pairs mined from the Internet, it inherently encodes rich real-world knowledge which can guide open-world applications such as visual navigation and embodied AI [24, 36]. Since we directly transfer CLIP’s feature space to 3D models, CLIP-FO3D has the potential to accomplish open-world 3D scene understanding.

Inspired by the qualitative results in [47] and [54], we use text embeddings to query open-world semantics for CLIP-FO3D’s scene representations. Given a text description, we extract its feature with CLIP’s text encoder, calculate its similarity with point features, and then threshold to produce a 3D mask. The visualization results are shown in Figure 4. Our model successfully finds the location that is most relevant to the text descriptions. For example, given “write”, CLIP-FO3D finds the whiteboard and desk on which one can write. Our model is also able to understand other color, activity and affordance descriptions. These results demonstrate that our 3D scene representations encode rich and well-structured knowledge about the real world.

Advantages of Annotation-free Training. We demonstrate that annotation-free training preserves CLIP’s rich information to the maximum. In contrast, fine-tuning CLIP with human annotations restricts the feature space, as demonstrated in [34].

We compare our pixel-level feature extraction with a supervised 2D segmentation method, LSeg [39]. LSeg aligns pixel embeddings to the CLIP text embedding of the corresponding semantic class from 7 segmentation datasets. Visualization results of capturing long-tailed concepts are shown in Figure 5. The example picture is used in [34]. The similarity of the pixels and text queries are used for visualization. Our method has outstanding advantages of recognizing long-tailed concepts directly inherited from CLIP. LSeg produces non-discriminative results because its feature space is restricted to pre-trained concepts and fails to generalize to open-world understanding.

6. Conclusion
This paper proposes transferring CLIP’s feature space to 3D scene representations. We first extract pixel-level features from CLIP for RGB views. Then we adopt feature projection to get the target point features and train a 3D model. CLIP-FO3D achieves promising annotation-free semantic segmentation results on open-vocabulary concepts. It also outperforms previous state-of-the-art methods in zero-shot and data-efficient learning tasks. Moreover, our model inherits CLIP’s open-world properties, encoding open-world knowledge beyond object concepts.

Acknowledgement
This research was partially supported by National Key R&D Program of China (2022YFB2804103), Key Research and Development Program of Shaanxi (2021ZDLGY01-05), Tsinghua University Dushi Program, National Natural Science Foundation of China (20211710187), and Tsinghua University Talent Program.
References


(ICLR), 2021.


[56] Gordon Guocheng Qian, Xingdi Zhang, Abdullah Hamdi,
and Bernard Ghanem. Pix4point: Image pretrained transformers for 3d point cloud understanding. *ArXiv*, abs/2208.12259, 2022. 3


[73] Jihan Yang, Runyu Ding, Zhe Wang, and Xiaojuan Qi. Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. *ArXiv*, abs/2304.00962, 2023. 3

[74] Yuan Yao, Yuanhan Zhang, Zhen fei Yin, Jiebo Luo, Wanli Ouyang, and Xiaoshui Huang. 3d point cloud pre-training with knowledge distillation from 2d images. *ArXiv*, abs/2212.08974, 2022. 3


[85] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bingyan


