Frozen CLIP Transformer Is an Efficient Point Cloud Encoder

Xiaoshui Huang[1*], Zhou Huang[2*], Sheng Li[3*], Wentao Qu[4]
Tong He[1†], Yuenan Hou[1], Yifan Zuo[2†], Wanli Ouyang[1]
[1]Shanghai AI Laboratory
[2]Jiangxi University of Finance and Economics
[3]University of Electronic Science and Technology of China
[4]Nanjing University of Science and Technology

Abstract

The pretrain-finetune paradigm has achieved great success in NLP and 2D image fields because of the high-quality representation ability and transferability of their pretrained models. However, pretraining such a strong model is difficult in the 3D point cloud field due to the limited amount of point cloud sequences. This paper introduces **E**fficient **P**oint **C**loud **L**earning (EPCL), an effective and efficient point cloud learner for directly training high-quality point cloud models with a frozen CLIP transformer. Our EPCL connects the 2D and 3D modalities by semantically aligning the image features and point cloud features without paired 2D-3D data. Specifically, the input point cloud is divided into a series of local patches, which are converted to token embeddings by the designed point cloud tokenizer. These token embeddings are concatenated with a task token and fed into the frozen CLIP transformer to learn point cloud representation. The intuition is that the proposed point cloud tokenizer projects the input point cloud into a unified token space that is similar to the 2D images. Comprehensive experiments on 3D detection, semantic segmentation, classification and few-shot learning demonstrate that the CLIP transformer can serve as an efficient point cloud encoder and our method achieves promising performance on both indoor and outdoor benchmarks. In particular, performance gains brought by our EPCL are **19.7** AP_{50} on ScanNet V2 detection, **4.4** mIoU on S3DIS segmentation and **1.2** mIoU on SemanticKITTI segmentation compared to contemporary pretrained models. Code is available at https://github.com/XiaoshuiHuang/EPCL.

Introduction

Recently, the pretrain-finetune paradigm has achieved great success in natural language processing (NLP) (Chowdhery et al. 2022; Gu et al. 2021) and 2D image fields (Alayrac et al. 2022; Dosovitskiy et al. 2021; Radford et al. 2021). In the pretrain-finetune paradigm, a backbone is first pretrained on a large-scale dataset to learn general and transferable representations. Then, the pretrained model is finetuned on training samples of the downstream task to learn task-specific knowledge.

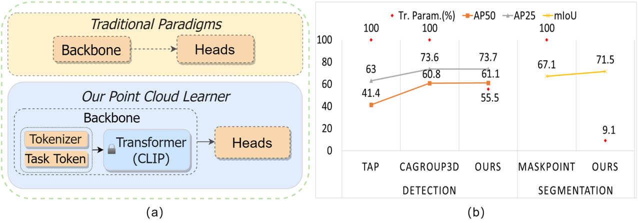

Figure 1: (a) Traditional paradigm fine-tunes the whole model, while our method only fine-tunes the tokenizer (T) and head (H). The CLIP transformer, which is initialized from the original CLIP weight, is kept frozen during training. (b) Our EPCL brings accuracy gains with higher training efficiency compared to SOTA pre-training methods.

The CLIP models (Radford et al. 2021) are strong pretrained models, which are trained in the contrastive manner by leveraging more than 400 million image-text pairs. The impressive performance of CLIP models on few-shot and zero-shot tasks is attributed to the powerful representation learned from the large quantity of pretraining data and the inherent alignment between the image and language domains.

However, directly applying the pretrain-finetune paradigm to the point cloud field will confront great difficulties due to the scarcity of training samples as well as the inherent domain gap between point cloud and image domains. For instance, the majority of pretraining methods (Pang et al. 2022; Qian et al. 2022; Yu et al. 2022; Xie et al. 2020; Huang et al. 2023; Zheng et al. 2023) are trained with limited data, ShapeNet (Chang et al. 2015) or ScanNet datasets (Dai et al. 2017). ShapeNet contains about 50, 000 objects and ScanNet contains 1, 513 room scans (Xie et al. 2020). Compared to the pretraining data of CLIP, the number of training samples in the point cloud field is merely ten thousandth. The prior knowledge learned from the limited training samples is also limited.

Inspired by the great success of CLIP, we ask a question: *can we apply the CLIP transformer to point cloud tasks as a pretrained encoder*? If the answer is yes, the 2D and 3D modalities are bridged and we can leverage the pretrained CLIP transformer for learning effective representations in the point cloud field. In this condition, the heavy reliance on

[*]These authors contributed equally.

[†]Corresponding author.

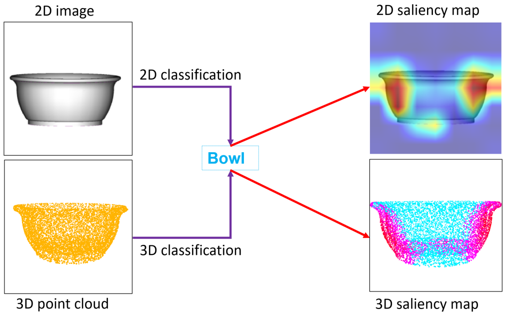

Figure 2: Using the frozen CLIP image transformer as an encoder for 2D and 3D classification, the saliency maps show the frozen CLIP model can attend to similar regions at different modalities.

3D pre-training data can also be relieved.

To mitigate the domain gap between point cloud and image domains, we design an extremely efficient module, *i.e.*, the point tokenizer, to map the point cloud and image information into the same embedding space. Since the point cloud and images all describe the surface information, we can consider them as a unified 2D-manifold that every point/pixel has a neighbourhood homeomorphic to a certain region of space $\mathbf{R}^2$ (Pressley 2010). We hypothesise that the frozen CLIP can extract meaningful representation from the 2D-manifold input. The tokenizer embeds the point/pixel neighbourhoods into the unified token space and weakly aligns the features of 2D images and 3D point cloud. Then, the transformer encoder extracts meaningful representations by semantically aligning them further. We validate this hypothesis in the experimental results (Figure 4).

Based on this finding, we propose the **E**fficient **P**oint **C**loud **L**earning (EPCL) framework to directly leverage the frozen CLIP transformer as the encoder for point cloud tasks. The difference between our method and other 3D pre-training methods is illustrated in Figure 1. Our EPCL merely requires the training of the lightweight task token, tokenizer and task head while previous 3D pre-training algorithms need to the train the whole model. Take S3DIS segmentation as an example. The trainable parameters of our EPCL barely account for 9.1% of all trainable model parameters, which strongly demonstrates the superior efficiency of EPCL.

To sum up, our EPCL framework possesses the following merits:

High efficiency. Our EPCL fully leverages the rich and broad knowledge hidden in CLIP models and only requires the training of a small portion of trainable network parameters. Our EPCL is apparently much more efficient compared to contemporary 3D pretraining algorithms that train all network parameters.

Aligning 2D and 3D models without paired data. Aligning multi-modal models has become mainstream to train a strong pretrained model while the existing methods (Radford et al. 2021; Alayrac et al. 2022) usually require paired data, *e.g.*, image-text pairs (Radford et al. 2021) and video-text pairs (Alayrac et al. 2022). Our EPCL does not need 3D-2D paired data to train the model when adapting the CLIP image transformer for 3D tasks. Figure 2 reveals that our method can semantically align similar regions in the 3D point cloud compared to CLIP in the 2D image on the recognition task.

Free from 3D pre-training. The strong 2D pre-trained CLIP transformer is directly used for 3D point clouds in EPCL without requiring 3D pre-training, which helps circumvent the barrier from the scarcity of 3D data.

Facilitating few-shot learning in downstream tasks. Leveraging the rich knowledge learned in CLIP, EPCL is effective when downstream tasks have scarce training samples.

We perform comprehensive experiments on mainstream point cloud tasks including detection, segmentation, recognition, classification and few-shot learning. Experimental results show that EPCL achieves better performance than the state-of-the-art 3D pre-training methods. Notably, our EPCL brings the gains of **19.7** AP_{50} on ScanNet V2 detection, **4.4** mIoU on S3DIS segmentation and **1.2** mIoU on SemanticKITTI segmentation compared to contemporary pretrained encoders.

Difference to prior works. While there have been several existing works proposing the utilization of 2D pretrained models, our method differs from them. For instance, Image2Point (Xu et al. 2022) expands 2D kernels of a CNN into 3D kernels for point cloud feature extraction. Pix4Point (Qian et al. 2022) initializes from 2D pretrained backbones and finetunes the entire neural network, resulting in low training efficiency. PPKT (Liu et al. 2021b) pretrains 3D backbones by distilling from 2D pretrained models, but it exhibits relatively low performance. ACT (Dong et al. 2022) adopts a two-stage strategy, training the teacher from a 2D pretrained model and distilling the teacher to a 3D point cloud Transformer student through masked modeling. However, prior works utilizing 2D pretrained models often suffer from either *low performance* or *low efficiency*.

In contrast to these prior works, our method, EPCL, *directly applies* the frozen CLIP model to extract point cloud features for various tasks, achieving better performance compared to recent pretrained methods. In this GPT era, our approach provides an *efficient* encoder for point cloud feature extraction. Additionally, since CLIP is a general vision pretrained model without task-specific information, we have designed a task token to further embed task-related biases.

Related Work

CLIP-Based Methods

CLIP (Radford et al. 2021), which aims to learn transferable visual representation from natural languages, has attracted increasing attention due to its promising results on various downstream tasks (Lei Ba et al. 2015; Kornblith, Shlens, and Le 2019; Recht et al. 2019). It consists of two encoders for visual and text representations, respectively. The method is jointly trained to align the two modalities with over 400 million image-text pairs. The rich semantic representation shared by both domains inspires many works and has been demonstrated effective in tasks like image caption (Vinyals et al. 2015) and video (Carreira et al. 2019). CLIPCAP (Mokady, Hertz, and Bermano 2021) trained a lightweight mapping network to generate meaningful captions, while the CLIP and language model is frozen. EVL (Lin et al. 2022) addressed the task of zero-shot video understanding via contrastive learning between video and text representations, which is free from the annotation of the downstream tasks. CLIP-ViL (Shen et al. 2021) uses CLIPs as pretrained backbones and finetunes the CLIP model on specific vision-language tasks. LAMM (Yin et al. 2023) uses frozen CLIP to embed multiple modalities into tokens and input these tokens into large language model to conduct multi-modal understanding tasks. Although promising, directly applying CLIP to 3D tasks is non-trivial due to the significant domain gap.

Point Cloud Representation Learning

Learning discriminative point cloud representation plays a critical role in downstream tasks. The existing methods can be divided into two categories, *i.e.*, point-based and voxel-based methods.

Point-based methods extract discriminative representation from raw points using either multi-layer perception (Qi et al. 2017), graph convolution (Wang et al. 2019) or kernel-based convolution (Thomas et al. 2019) or multimodal fusion (Huang et al. 2022). The objective of these methods is to leverage the global structure information or local property of point neighbours to describe the 3D point cloud. The advantages of these methods are that features can directly extract from analyzing point neighbours, the memory consumption is relatively small and no preprocessing steps are required.

Voxel-based methods require to pre-process the given point clouds into voxels. Then, voxel-based convolution neural networks are applied to extract the representation. Typical examples are VoxelNet (Riegler, Osman Ulusoy, and Geiger 2017) and Minkowski Engine (Choy, Gwak, and Savarese 2019). They design octree-based convolution and sparse convolution to effectively extract the local representation of the point cloud without large GPU memory consumption. The advantage of these methods is that the representation can easily overcome the density variation.

3D Pre-Training

Pre-training aims to learn prior knowledge from the training data. The existing pre-training methods can be divided into three categories, *i.e.*, global contrastive, local contrastive and Masking AutoEncoder (MAE). The global contrastive learning methods (Wang et al. 2021; Mei et al. 2022; Huang et al. 2023) compare the global feature difference of point clouds. In contrast, local contrastive learning methods (Xie et al. 2020; Wang et al. 2023) compare the local point feature differences or local view pixel differences. Recently, several MAE-based pre-training methods (Yu et al. 2022; Pang et al. 2022) are proposed to learn pretrained transformer backbones. These methods leverage the knowledge of pretrained datasets so that the downstream task models initialize from a better starting point.

Recently, several methods are proposed to use the 2D pretrained models on point cloud tasks. PointCLIP (Zhang et al. 2022) projects the point cloud into 2D views and directly uses the frozen 2D pre-trained models for 3D recognition. P2P(Wang et al. 2022c) designs a projector to project the 3D objects into the 2D plane and designs several prompts to use the frozen 2D pre-trained backbones. PPKT (Liu et al. 2021b) transfers the knowledge of 2D pretraiend model to 3D backbones by using point-to-pixel loss. However, these methods require projecting the 3D objects into several 2D views and are sensitive to view projection. Hence, they are used for object-level point clouds but face great difficulty in handling scene-level point cloud perception. Image2Point (Xu et al. 2022) expands 2D kernels of a 2D CNN into 3D kernels and applied them to voxel-based point cloud tasks, which suffers from relatively low accuracy as the parameter domain gap. Pix4Point (Qian et al. 2022) initializes from 2D pretrained backbones and finetunes the whole neural network, which is not efficient. ACT (Dong et al. 2022) requires two stage training to transfer the knowledge of 2D pretrained model to 3D point cloud transformer. However, the training process is not efficient and the performance has a large gap to task-specific model.

Different to previous approaches, our EPCL directly utilizes the pre-trained 2D CLIP transformer as an efficient encoder to extract point cloud features. And our method is applicable to both real-world and synthetic point cloud tasks. In this GPT era, our work provides the insights that frozen CLIP can achieve comparable or better performance to recent SOTA pretrained methods with higher efficiency.

Method

This section first introduces the Vision Transformer (ViT) (Dosovitskiy et al. 2021) in the 2D image field and the transformer in the point cloud field. Then, we present our EPCL and the rationale behind the workability of EPCL.

Preliminary

2D Vision Transformer. Given an image $I \in \mathbb{R}^{H \times W \times C}$, the ViT (Dosovitskiy et al. 2021) divides the image into a sequence of flattened local image patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$ and uses a tokenizer to convert these patches into a 1D sequence of visual token embeddings $E_I(I) \in \mathbb{R}^{N \times D}$, where N is the number of tokens, $P \times P$ is the image patch size, D is the dimension of each image token. H and W are the height and width of the given image, respectively. The total number of patches is $N = HW/(P^2)$. The position embedding is concatenated to the visual token embeddings. Visual tokens and class tokens are fed into the transformer for feature extraction. Afterwards, the feature is fed into the classification head to yield the classification results. Mathematically, the 2D ViT can be formulated as follows:

$$z_0 = [x_{\text{cls}}, E_I(I_{1,1}), ..., E_I(I_{\frac{H}{P},\frac{W}{P}})] + E_{\text{pos}}, \quad (1)$$

$$\widetilde{z}_l = \text{MSA}(\text{LN}(z_{l-1}) + z_{l-1}), \quad (2)$$

$$z_l = \text{MLP}(\text{LN}(\widetilde{z}_l)) + \widetilde{z}_l, \quad (3)$$

$$y = H^{\text{cls}}(\text{LN}(z_L^0)), \quad (4)$$

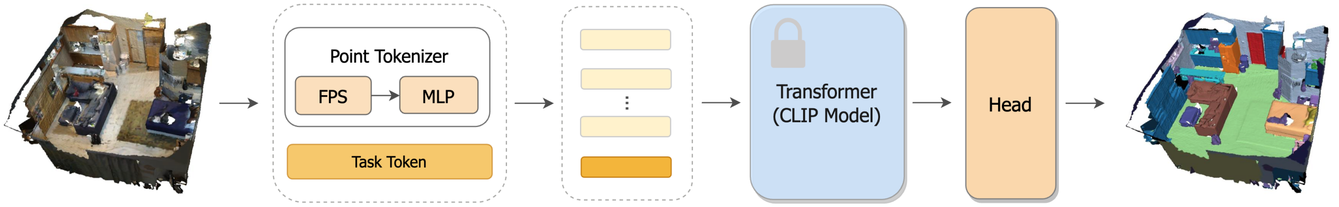

Figure 3: Schematic overview of EPCL. The Point Tokenizer contains two successive steps, that are Farthest Point Sampling (FPS) for downsampling the input point cloud and Multi-Layer Perceptron (MLP) for extracting features from the downsampled point cloud. The Task Token is task-specific and learnable. Tokens from the point tokenizer and task token are fed into the frozen CLIP Transformer. The Head uses the tokens from the Transformer to yield the predictions for each specific downstream task. The CLIP transformer, which is initialized from the original CLIP weight, is kept frozen during the training stage, while the point cloud tokenizer, task token and head are trainable.

where $E_I(.)$ is the image tokenizer that extracts the token embedding for each image patch, and x_{cls} is the class token. The transformer consists of L layers of layer normalization $LN(.)$, multi-head self-attention $\mathrm{MSA}(.)$ and multilayer perceptron $\mathrm{MLP}(.)$. The residual connection is applied after every block. H^{cls} represents the classification head and takes the feature of the class token at the last layer as input. Take the 1000-class image classification task as an example. H^{cls} refers to a single MLP that maps the input feature into a 1000-dimension classification output.

Transformer in point cloud. Before the standard transformer is applied to the point cloud field, there are some transformer layers (Zhao et al. 2021; Guo et al. 2021) specifically designed for point cloud processing. Pioneered by PointBERT (Yu et al. 2022), the standard transformer has been applied to point cloud tasks. Similar to the ViT, the point cloud $P \in \mathbb{R}^{A\times 3}$ is divided into a sequence of point cloud patches $P_p \in \mathbb{R}^{M\times(3K)}$, where A and M denote the number of points and patches, respectively, and K denotes the number of points in each patch. These patches are sent to a tokenizer to extract point token embeddings. Then, these point token embeddings, position embedding and class token are fed into the standard transformer for feature extraction. Afterwards, these features are fed into a task head for downstream tasks. The Transformer for point cloud can be formulated as follows:

$$f_0 = [x_{\mathrm{cls}}, E_p(P_1), ..., E_p(P_M)] + E_{\mathrm{pos}}, \tag{5}$$

$$\widetilde{f_l} = \mathrm{MSA}(\mathrm{LN}(f_{l-1}) + f_{l-1}), \tag{6}$$

$$f_l = \mathrm{MLP}(\mathrm{LN}(\widetilde{f_l})) + \widetilde{f_l}, \tag{7}$$

$$y = H_p^{\mathrm{cls}}(\mathrm{LN}(f_L^0)), \tag{8}$$

where the E_p is the point cloud tokenizer, the three-layer MLP is usually applied for obtaining point cloud token embeddings. H_p^{cls} refers to three-layer MLPs that map the input feature into the C-dimension classification predictions for C-class point cloud classification tasks.

Comparison between 2D and 3D transformers. By comparing the equation (1)-(3) and (5)-(8), the standard transformer module is the same, which consists of a series of the LN, MSA and MLP. The only difference lies in the tokenizer during the feature extraction. Then, the deep features are fed into different 2D/3D task heads for 2D/3D downstream tasks. Here, we want to investigate whether the same standard transformer module pretrained on 2D could be directly applied to 3D point cloud tasks.

The Proposed Algorithm: EPCL

The motivation of our method is to leverage the frozen 2D CLIP model for downstream point cloud understanding tasks. To this end, we propose the Efficient Point Cloud Learning (EPCL) framework to use the 2D frozen CLIP transformer and only finetune the tokenizer, task token and task head. The overall framework is shown in Figure 3. This section introduce the details of tokenizer, task token and Frozen CLIP transformer. The details of task head are attached in the supplement.

Point cloud tokenizer. Given a point cloud $P \in \mathbb{R}^{A\times 3}$, similar to the objective of the 2D tokenizer, the point cloud tokenizer aims to convert the input point cloud into a sequence of token embeddings. Specifically, we first sample M points as centers of point patches by the Farthest Point Sampling (FPS) algorithm, and then group K points from each center by the K-Nearest Neighbourhood (KNN) algorithm and thus obtain M patches. These patches are further fed into several MLPs to obtain the token embeddings $E_p(P) \in \mathbb{R}^{M\times D_p}, i \in [1...M]$, where D_p is the dimension of each point token, *i.e.*, 768.

Task token. Since the CLIP is trained by a large-scale text-image pair dataset, it is lacked of task information. To further embed the given point clouds into a shared token space that benefit for the task, we design a task token to learn a global task-related bias. Our task token module is implemented by a fully connected layer with learnable parameters. Following (Liu et al. 2021a), we initialize the task token as enumerated numbers.

Frozen CLIP transformer. After the input point cloud is converted into a sequence of visual tokens, we feed visual tokens and task tokens into the CLIP image transformer, which is initialized from the original CLIP weight and kept frozen during training. The frozen CLIP transformer serves as the feature extractor for downstream point cloud tasks.

Analysis on 2D-3D Semantic Alignment of CLIP Transformer To analyse the workability of frozen 2D CLIP

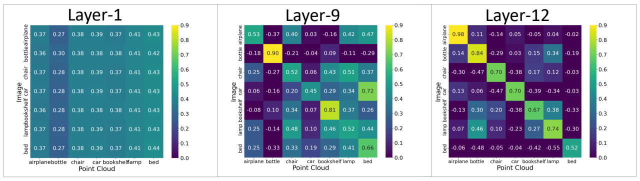

Figure 4: The cross-correlation between CLIP image features and point cloud features at layers 1, 9, and 12 for different object categories.

transformer for point cloud representation learning, we calculate the semantic similarity between image features and point cloud features. Specifically, we first calculate their feature cross-correlation at different layers of the same CLIP transformer. Then, we use the transformer explanation tool in (Chefer, Gur, and Wolf 2021) to obtain the significance map. For the 2D image, we crop a view from the ShapeNet model to keep the texture and apply the CLIP model to classify the image view.

Statistical results. Figure 4 shows that the tokenizer can weakly align the 2D and 3D features. At shallow layers, the features from the point cloud and image for the same category have lower cross-correlation in the left sub-figure. As the layers go deeper, the 2D-3D features are matched with high cross-correlation in the same category (see the right sub-figure).

Visual results. Figure 5 shows the roughly similar significance maps at a 2D image and 3D point cloud. This figure shows that the frozen CLIP model can capture similar semantic regions from 2D and 3D modalities.

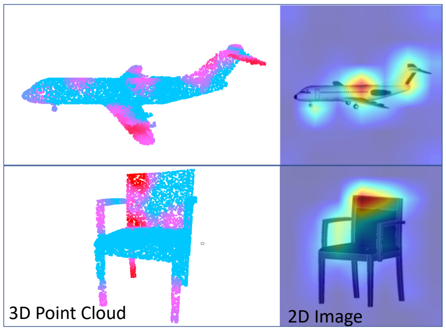

Figure 5: The semantic similarity between 2D image and 3D point cloud from significance maps.

The Rationale Behind EPCL To better understand why the frozen CLIP transformer is workable for the point cloud, we provide an intuitive explanation from the manifold aspect. We define the input token space as Ω_I and output token space as Ω_O. The CLIP image transformer learns a function f to map the input tokens $X \in \Omega_I$ into semantically meaningful tokens $Y \in \Omega_O$: $Y = f(X)$. Since the CLIP has been trained in a large-scale dataset that contains diverse web image-text pairs, the input token space Ω_I is large and diverse.

The image tokenizer uses convolution to aggregate local information into the image token space, denoted by Ω_I^I. Similarly, our point cloud tokenizer uses the FPS + KNN + MLP to aggregate local neighbourhood information token space, denoted by Ω_I^P. Since a point cloud frame only records points on the surface, according to the manifold definition (Pressley 2010), a small local point cloud patch is approximately a plane and the 3D point cloud lies in the 2D manifold. Since the given point cloud consists of many local planes, our tokenizer and the task token learn to project the 2D-manifold point cloud into the token space Ω_I^P that is similar to the CLIP image token space Ω_I^I projected from 2D image plane. Since the images and local point cloud patches are both 2D-manifold planes, the above tokenizer learning is achievable. Previous research shows that the transformer inherently extracts shape-biased features (Park and Kim 2022; Naseer et al. 2021) for 2D images. Therefore, the CLIP image transformer can extract shape-based features from the token space Ω_I^P for the given point cloud.

Experiments

We introduce the details of used datasets and baselines in section . Then, experiments of the downstream tasks are described in section , section and section , respectively. Afterwards, the ablation studies are presented in section .

Datasets and Baselines

Datasets. We conduct real-world detection on ScanNet (Dai et al. 2017), indoor semantic segmentation on S3DIS (Armeni et al. 2016) and outdoor semantic segmentation on SemanticKITTI Behley et al. (2019). Also, we evaluate the accuracy of few-shot learning and classification on synthetic ModelNet40 (Wu et al. 2015).

Baselines. Our EPCL focuses on leveraging pre-training for downstream tasks. For a fair comparison, the state-of-the-art (SOTA) point cloud pre-training methods with the transformer-based architectures are selected as the baselines.

- **Detection:** Following MaskPoint (Yu et al. 2022), we compare with MaskPoint (Yu et al. 2022), Point-BERT(Pang et al. 2022), TAP (Wang et al. 2023), Simple3D-Former (Wang et al. 2022b), SoftGroup (Vu et al. 2022) and CAGroup3D (Wang et al. 2022a).
- **Segmentation:** Following Simple3D-former (Wang et al. 2022b), we compare with MaskPoint (Pang et al. 2022) and other transformer-based method (Zhao et al. 2021).
- **Classification:** Following MaskPoint (Yu et al. 2022), we compare with MaskPoint (Yu et al. 2022), Point-BERT(Pang et al. 2022), Simple3D-Former (Wang et al. 2022b) and P2P (Wang et al. 2022c).

Detection

For many contemporary 3D pre-training works, they report the performance mainly on object-level classification and part segmentation tasks, which is insufficient in real-world point cloud tasks. Moreover, the most recent P2P (Wang et al. 2022c) needs to project the point cloud into 2D views, which is confronted with great challenges in solving real-world point cloud tasks. Our EPCL does not require any projections and thus can be widely applied to real-world point cloud scenarios. In this section, the detection on ScanNet V2 is evaluated and compared with the state-of-the-art 3D pre-training approaches (Pang et al. 2022; Yu et al. 2022) and object detection methods(Vu et al. 2022; Wang et al. 2022a).

Method	3D Pretrain	AP_{50}	AP_{25}
PointBERT	✓	38.3	61.0
MaskPoint	✓	42.1	64.2
TAP	✓	41.4	63.0
Simple3D-Former	✗	40.7	59.4
CLIP frozen + 3DETR	✗	**43.0**	**62.6**
SoftGroup	✗	59.4	71.6
CAGroup3D	✗	60.8	73.6
CLIP froz.+CAGoup3D	✗	**61.1**	**73.7**

Table 1: Detection on ScanNet V2.

Table 1 shows that the frozen CLIP model achieves better accuracy than baseline methods. This observation shows that the CLIP transformer can effectively learn 3D representation to solve real-world 3D detection and achieve better performance than state-of-the-art 3D pre-training method, TAP (Wang et al. 2023). Note that the CLIP model is frozen and has not seen any 3D point cloud in their learned parameters. Our method only fine-tunes the same training dataset with other baselines. These results demonstrate that the CLIP transformer achieves better accuracy. Notably, EPCL achieves better performance than the state-of-the-art object detection method CAGroup3D when using the head of CAGroup3D. The impressive performance is attributed to the strong ability of the CLIP model to align the features in different modalities.

Semantic Segmentation

Indoor semantic segmentation. This section introduces the experiments on indoor segmentation dataset S3DIS(Armeni et al. 2016). To ensure fair comparison, we put all these encoders on the same code base, which shares the same hierarchical tokenizer and semantic task head. The MaskPoint initializes the encoder with the pre-trained model of MaskPoint, and the Simple3D-Former initializes the encoder with 2D ViT. Then, the MaskPoint and Simple3D-Former methods finetune the *whole model* on the S3DIS training samples. In contrast, our method keeps the CLIP model frozen and *only* finetunes the tokenizer and task head.

Method	OA	mAcc.	mIoU.
Point Transformer	90.8	76.5	70.4
MaskPoint	89.0	73.8	67.1
Simple3D-Former	-	72.5	67.0
Ours	**90.8**	**77.8**	**71.5**

Table 2: Indoor semantic segmentation on S3DIS (Area5).

Table 2 shows that the frozen CLIP model obtains obviously better accuracy (*i.e.*, mAcc. and mIoU) than other state-of-the-art 3D pre-training methods as well as Point Transformer on S3DIS (Area5) dataset. This observation illustrates that the frozen CLIP model is an efficient point cloud learner in the real-world semantic segmentation task.

Outdoor semantic segmentation. We also evaluate our EPCL on outdoor segmentation task and compare with recent task-specific methods (Cylinder3D (Zhou et al. 2020), PVKD (Hou et al. 2022), 2DPASS (Yan et al. 2022), RPVNet (Xu et al. 2021)) and pretrained method (RangeFormer (Kong et al. 2023)). Quantitative comparison on SemanticKITTI validation set is shown in Table 3, which demonstrates better accuracy than these compared methods. It is worth noting that the frozen CLIP achieves the highly competitive performance compared to the task-specific model, even without any engineering tricks such as test time augmentation and model ensemble.

Method	mIoU. (val.)
Cylinder3D	65.2
PVKD	66.4
2DPASS	69.3
RPVNet	69.6
RangeFormer	69.6
Ours	**72.4**

Table 3: Outdoor semantic segmentation on SemanticKITTI.

Method	3D Pretrain	OA
PointBERT	✓	93.2
MaskPoint	✓	**93.8**
P2P	✗	92.7
Simple3D-Former	✗	92.0
Ours + w/o CLIP frozen	✗	92.3
Ours	✗	**92.9**

Table 4: Classification on ModelNet40.

Classification

Supervised classification. Table 4 summarizes the classification results on the synthetic ModelNet40 dataset. Our method achieves comparable performance to the state-of-the-art 3D pre-training methods although our method does not pre-train the model on the object dataset, *i.e.*, ShapeNet. Compared to the P2P, which is the recent state-of-the-art method that directly used a 2D pre-trained model, our method achieves better accuracy. P2P designs a projection module to render images from 3D objects which is tailored for 3D classification. Our method does not need to project 3D to 2D images and directly processes the 3D tokens, which shows potential in other applications except classification. Compared to Simple3D-Former, which finetunes the whole model from a 2D pre-trained model, our method still achieves better accuracy. These positive results verify our argument that *the frozen CLIP transformer is an effective encoder to learn 3D representation for point cloud understanding*.

Tuning Method	**5-w,10-s**	**5-w,20-s**	**10-w,10-s**	**10-w,20-s**	**30-w,10-s**
PointBERT	94.6 ± 3.1	96.3 ± 2.7	91.0 ± 5.4	92.7 ± 5.1	81.4 ± 2.4
MaskPoint	95.0 ± 3.7	97.2 ± 1.7	**91.4** ± **4.0**	93.4 ± **3.5**	80.7 ± 4.9
Ours	**95.1** ± **2.7**	**97.3** ± **1.6**	91.1 ± 4.2	**93.5** ± 3.8	**81.7** ± **0.7**

Table 5: Few-shot learning accuracy of 3D pre-training methods and EPCL on ModelNet40.

Few-shot learning. One important advantage of pre-trained models is that fewer training samples are required in downstream tasks. This is usually evaluated by the few-shot learning task. To evaluate the few-shot learning ability, we follow MaskPoint (Pang et al. 2022) to conduct experiments with the setting of "K-way N-shot", *i.e.*, *5way-10shot, 5way-20shot,10way-10shot and 10way-20shot*.

Table 5 summarizes the comparison experiments on "K-way N-shot" few-shot learning. Our EPCL obtains more accurate classification results than the state-of-the-art 3D pre-training methods. This observation shows that the frozen CLIP transformer is an effective representation learning encoder in the challenging few-shot learning setting.

Ablation Studies and Discussion

In this section, we introduce several key ablation studies to examine the effect of each component of our EPCL. More ablation studies and discussions can be found in the supplement.

Task token. The task token module aims to learn task embedding for a specific task. As the task embedding module is only trainable in the training stage and the parameters and initialization are frozen during the inference stage. To demonstrate its effectiveness, we conduct an ablation study by removing the task token module on ScanNet V2 at the detection task. Table 6 shows that the detection accuracy decreases (1.9%) when the task embedding is discarded. This experiment illustrates that learning additional task-related feature bias is beneficial to the CLIP model in point cloud tasks.

task token	CLIP Frozen	AP_{50}	Train Para. (%)
✗	✓	59.2	55.23
✓	✗	60.1	100
✓	✓	**61.1**	55.51

Table 6: Ablation studies of the task embedding and the frozen strategy on ScanNet V2 detection task.

Task token transferability. To demonstrate the transferability of our task token, we use the detection task token to replace the classification task token. The classification result improves by 5.7% by using the detection task token compared to the random one. The improved performance of using the detection task token over the random task token demonstrates the transferability. Moreover, the accuracy will drop $>$10% when directly using other task tokens. This result shows that the task token needs to be fine-tuned together with the tokenizer. Simply replacing the task token will lead to inferior performance.

CLIP Frozen or not? Recall that we freeze the CLIP model during the entire training process. It is natural to wonder what the performance will be if turning the parameters of the CLIP model during training. To answer this question, we turn the whole neural network on during the training stage and conduct an ablation study on real-world detection task. Table 6 shows that the accuracy drops 1.0% when the CLIP model is turned on. *The reason* is that the relatively small-scale 3D training dataset fine-tunes the CLIP model to a worse parameter space compared to the one trained on the large-scale dataset. Also, our method shows that the frozen CLIP transformer achieves better training efficiency than the version without freezing.

Method	AP_{50}	AP_{25}
SAM	59.5	73.7
DINO	60.0	72.7
Ours	**61.1**	**73.7**

Table 7: Detection results of other 2D pre-trained models.

Is CLIP better than other 2D pretrained models? The CLIP model is trained on a large-scale dataset that pairs internet images with text, containing a wide range of real-world multimodal knowledge. To demonstrate its effectiveness in 3D representation learning, we replace the frozen CLIP transformer with other 2D pretrained models (SAM (Kirillov et al. 2023), DINO (Caron et al. 2021)) that are solely trained on images (single modality). We then freeze the model during the training stage. Table 7 illustrates that CLIP achieves higher detection accuracy on ScanNet-V2 compared to other 2D pretrained models. This result demonstrates that the CLIP transformer, with its multimodal knowledge, outperforms 2D pretrained models that are only trained on images.

Conclusion

This paper proposes an efficient yet effective method to construct point cloud understanding models by using the frozen CLIP transformer. Our method converts the input point cloud into sequential tokens with a point tokenizer. These tokens and the learnable task token input into the frozen CLIP transformer can generate robust 3D representation. We conduct thorough analyses of the inner mechanism and find the tokenizer can weakly align the 3D and 2D features at different modalities. Then, the CLIP transformer can align them further. Our method achieves appealing performance on a wide range of downstream tasks, including both real-world detection and segmentation tasks as well as synthetic object-level classification tasks.

Acknowledgements

This work is partially supported by the National Key R&D Program of China(NO.2022ZD0160101), National Natural Science Foundation of China General Program (NO. 62271237) and Jiangxi Provincial Natural Science Foundation Outstanding Young Scientist Program (NO. 20224ACB212005).

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*.

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1534–1543.

Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; and Gall, J. 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9297–9307.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Carreira, J.; Noland, E.; Hillier, C.; and Zisserman, A. 2019. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Chefer, H.; Gur, S.; and Wolf, L. 2021. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 397–406.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3075–3084.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5828–5839.

Dong, R.; Qi, Z.; Zhang, L.; Zhang, J.; Sun, J.; Ge, Z.; Yi, L.; and Ma, K. 2022. Autoencoders as Cross-Modal Teachers: Can Pretrained 2D Image Transformers Help 3D Representation Learning? In *The Eleventh International Conference on Learning Representations*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*.

Gu, Y.; Han, X.; Liu, Z.; and Huang, M. 2021. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7(2): 187–199.

Hou, Y.; Zhu, X.; Ma, Y.; Loy, C. C.; and Li, Y. 2022. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8479–8488.

Huang, T.; Dong, B.; Yang, Y.; Huang, X.; Lau, R. W.; Ouyang, W.; and Zuo, W. 2023. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22157–22167.

Huang, X.; Qu, W.; Zuo, Y.; Fang, Y.; and Zhao, X. 2022. IMFNet: Interpretable multimodal fusion for point cloud registration. *IEEE Robotics and Automation Letters*, 7(4): 12323–12330.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. *arXiv:2304.02643*.

Kong, L.; Liu, Y.; Chen, R.; Ma, Y.; Zhu, X.; Li, Y.; Hou, Y.; Qiao, Y.; and Liu, Z. 2023. Rethinking range view representation for lidar segmentation. *arXiv preprint arXiv:2303.05367*.

Kornblith, S.; Shlens, J.; and Le, Q. V. 2019. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2661–2671.

Lei Ba, J.; Swersky, K.; Fidler, S.; et al. 2015. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE international conference on computer vision*, 4247–4255.

Lin, Z.; Geng, S.; Zhang, R.; Gao, P.; de Melo, G.; Wang, X.; Dai, J.; Qiao, Y.; and Li, H. 2022. Frozen CLIP Models are Efficient Video Learners. In *European Conference on Computer Vision*, 388–404. Springer.

Liu, X.; Ji, K.; Fu, Y.; Du, Z.; Yang, Z.; and Tang, J. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Liu, Y.-C.; Huang, Y.-K.; Chiang, H.-Y.; Su, H.-T.; Liu, Z.-Y.; Chen, C.-T.; Tseng, C.-Y.; and Hsu, W. H. 2021b. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.04687*.

Mei, G.; Huang, X.; Liu, J.; Zhang, J.; and Wu, Q. 2022. Unsupervised Point Cloud Pre-Training Via Contrasting and Clustering. In *2022 IEEE International Conference on Image Processing (ICIP)*, 66–70. IEEE.

Mokady, R.; Hertz, A.; and Bermano, A. H. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

Naseer, M. M.; Ranasinghe, K.; Khan, S. H.; Hayat, M.; Shahbaz Khan, F.; and Yang, M.-H. 2021. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34: 23296–23308.

Pang, Y.; Wang, W.; Tay, F. E.; Liu, W.; Tian, Y.; and Yuan, L. 2022. Masked autoencoders for point cloud self-supervised learning. *arXiv preprint arXiv:2203.06604*.

Park, N.; and Kim, S. 2022. How do vision transformers work? *arXiv preprint arXiv:2202.06709*.

Pressley, A. N. 2010. *Elementary differential geometry*. Springer Science & Business Media.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qian, G.; Zhang, X.; Hamdi, A.; and Ghanem, B. 2022. Pix4Point: Image Pretrained Transformers for 3D Point Cloud Understanding. *arXiv preprint arXiv:2208.12259*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763. PMLR.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2019. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, 5389–5400. PMLR.

Riegler, G.; Osman Ulusoy, A.; and Geiger, A. 2017. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3577–3586.

Shen, S.; Li, L. H.; Tan, H.; Bansal, M.; Rohrbach, A.; Chang, K.-W.; Yao, Z.; and Keutzer, K. 2021. How Much Can CLIP Benefit Vision-and-Language Tasks? In *International Conference on Learning Representations*.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164.

Vu, T.; Kim, K.; Luu, T. M.; Nguyen, T.; and Yoo, C. D. 2022. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2708–2717.

Wang, H.; Ding, L.; Dong, S.; Shi, S.; Li, A.; Li, J.; Li, Z.; and Wang, L. 2022a. CAGroup3D: Class-Aware Grouping for 3D Object Detection on Point Clouds. *arXiv preprint arXiv:2210.04264*.

Wang, H.; Liu, Q.; Yue, X.; Lasenby, J.; and Kusner, M. J. 2021. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9782–9792.

Wang, Y.; Fan, Z.; Chen, T.; Fan, H.; and Wang, Z. 2022b. Can We Solve 3D Vision Tasks Starting from A 2D Vision Transformer? *arXiv preprint arXiv:2209.07026*.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5): 1–12.

Wang, Z.; Yu, X.; Rao, Y.; Zhou, J.; and Lu, J. 2022c. P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. *arXiv preprint arXiv:2208.02812*.

Wang, Z.; Yu, X.; Rao, Y.; Zhou, J.; and Lu, J. 2023. Take-A-Photo: 3D-to-2D Generative Pre-training of Point Cloud Models.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

Xie, S.; Gu, J.; Guo, D.; Qi, C. R.; Guibas, L.; and Litany, O. 2020. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, 574–591. Springer.

Xu, C.; Yang, S.; Galanti, T.; Wu, B.; Yue, X.; Zhai, B.; Zhan, W.; Vajda, P.; Keutzer, K.; and Tomizuka, M. 2022. Image2point: 3d point-cloud understanding with 2d image pretrained models. In *European Conference on Computer Vision*, 638–656. Springer.

Xu, J.; Zhang, R.; Dou, J.; Zhu, Y.; Sun, J.; and Pu, S. 2021. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16024–16033.

Yan, X.; Gao, J.; Zheng, C.; Zheng, C.; Zhang, R.; Cui, S.; and Li, Z. 2022. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *European Conference on Computer Vision*, 677–695. Springer.

Yin, Z.; Wang, J.; Cao, J.; Shi, Z.; Liu, D.; Li, M.; Sheng, L.; Bai, L.; Huang, X.; Wang, Z.; et al. 2023. LAMM: Language-Assisted Multi-Modal Instruction-Tuning Dataset, Framework, and Benchmark. *arXiv preprint arXiv:2306.06687*.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19313–19322.

Zhang, R.; Guo, Z.; Zhang, W.; Li, K.; Miao, X.; Cui, B.; Qiao, Y.; Gao, P.; and Li, H. 2022. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8552–8562.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16259–16268.

Zheng, X.; Huang, X.; Mei, G.; Hou, Y.; Lyu, Z.; Dai, B.; Ouyang, W.; and Gong, Y. 2023. Point Cloud Pre-training with Diffusion Models. *arXiv preprint arXiv:2311.14960*.

Zhou, H.; Zhu, X.; Song, X.; Ma, Y.; Wang, Z.; Li, H.; and Lin, D. 2020. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv preprint arXiv:2008.01550*.