GeoMAE: Masked Geometric Target Prediction for Self-supervised Point Cloud Pre-Training

Xiaoyu Tian[1] Haoxi Ran[2] Yue Wang[3] Hang Zhao[1]*

[1]IIIS, Tsinghua University [2]CMU [3]NVIDIA

Abstract

This paper tries to address a fundamental question in point cloud self-supervised learning: what is a good signal we should leverage to learn features from point clouds without annotations? To answer that, we introduce a point cloud representation learning framework, based on geometric feature reconstruction. In contrast to recent papers that directly adopt masked autoencoder (MAE) and only predict original coordinates or occupancy from masked point clouds, our method revisits differences between images and point clouds and identifies three self-supervised learning objectives peculiar to point clouds, namely centroid prediction, normal estimation, and curvature prediction. Combined, these three objectives yield an nontrivial self-supervised learning task and mutually facilitate models to better reason fine-grained geometry of point clouds. Our pipeline is conceptually simple and it consists of two major steps: first, it randomly masks out groups of points, followed by a Transformer-based point cloud encoder; second, a lightweight Transformer decoder predicts centroid, normal, and curvature for points in each voxel. We transfer the pre-trained Transformer encoder to a downstream peception model. On the nuScene Datset, our model achieves 3.38 mAP improvment for object detection, 2.1 mIoU gain for segmentation, and 1.7 AMOTA gain for multi-object tracking. We also conduct experiments on the Waymo Open Dataset and achieve significant performance improvements over baselines as well. [1]

1. Introduction

While object detection and segmentation from LiDAR point clouds have achieved significant progress, these models usually demand a large amount of 3D annotations that are hard to acquire. To alleviate this issue, recent works explore learning representations from unlabeled point clouds, such as contrastive learning [20, 46, 53], and mask modeling [32, 51]. Similar to image-based representation learning settings, these representations are transferred to downstream tasks for weight initialization. However, the existing self-supervised pretext tasks do not bring adequate improvements to the downstream tasks as expected.

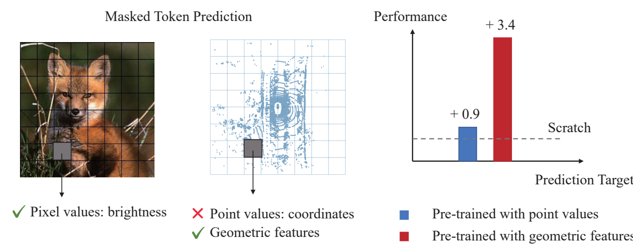

Figure 1. Pixel value regression has been proved effective in masked autoencoder pre-training for images. We find this practice ineffective in point cloud pre-training and propose a set of geometry aware prediction targets.

Contrastive learning based methods typically encode different 'views' (potentially with data augmentation) of point clouds into feature space. They bring features of the same point cloud closer and make features of different point clouds 'repel' each other. Other recent works use masked modeling to learn point cloud features through self reconstruction [32, 51]. That is, randomly sparsified point clouds are encoded by point cloud feature extractors, followed by a reconstruction module to predict original point clouds. These methods, when applied to point clouds, ignore the fundamental difference of point clouds from images – point clouds provide scene geometry while images provide brightness. As shown in Figure 1, this modality disparity hampers direct use of methods developed in the image domain for point cloud domain, and thus calls for novel self-supervised objectives dedicated to point clouds.

Inspired by modeling and computational techniques in geometry processing, we introduce a self-supervised learning framework dedicated to point clouds. Most importantly, we design a series of prediction targets which describe the fine-grained geometric features of the point clouds. These

*Corresponding to: hangzhao@mail.tsinghua.edu.cn
[1]Our code is available at https://github.com/Tsinghua-MARS-Lab/GeoMAE.

geometric feature prediction tasks jointly drive models to recognize different shapes and areas of scenes. Concretely, our method starts with a point cloud voxelizer, followed by a feature encoder to transform each voxel into a feature token. These feature tokens are randomly dropped based on a pre-defined mask ratio. Similar to the original MAE work [18], visible tokens are encoded by a Transformer encoder. Then a Transformer decoder reconstructs the features of the original voxelized point clouds. Finally, our model predicts point statistics and surface properties in parallel branches.

We conduct experiments on a diverse set of large-scale point cloud datasets including nuScenes [4] and Waymo [39]. Our setting consists of a self-supervised pre-training stage and a downstream task stage (3D detection, 3D tracking, segmentation), where they share the same point cloud backbone. Our results show that even without additional unlabeled point clouds, self-supervised pre-training with objectives proposed by this paper can significantly boost the performance of 3D object detection. To summarize, our contributions are:

- We introduce geometry aware self-supervised objectives for point clouds pre-training. Our method leverages fine-grained point statistics and surface properties to enable effective representation learning.
- With our novel learning objectives, we achieve state-of-the-art performance compared to previous 3D self-supervised learning methods on a variety of downstream tasks including 3D object detection, 3D/BEV segmentation, and 3D multi-object tracking.
- We conduct comprehensive ablation studies to understand the effectiveness of each module and learning objective in our approach.

2. Related Work

2.1. Self-Supervised Learning for Point Clouds

Self-supervised learning for point cloud [14, 20, 23, 32, 37, 38, 42, 46, 51, 53] has drawn considerable attention due to the expensive cost of labeling the 3D point cloud. Some are based on contrastive paradigms [20, 46, 53]. PointContrast [46] learns from correspondences between different point cloud views with a contrastive loss. DepthContrast [53] considers different depth map as an instance and discriminating between them to learn the representation. STRL [20] learns the invariant representation from two augmented temporally-correlated frames from a 3D point cloud sequence. Others [14, 37, 38] utilize a pretext task to promote self-supervised representation learning. [38] phrases the pretext task as a part segmentation task by displacing the part of the parts of the point cloud and then predicting their ordering labels. [14] squeezes learned representations through an implicitly defined parametric discrete generative model bottleneck. [37] introduces a bidirectional reasoning between local and global to capture the underlying semantic knowledge. Motivated by the huge success of 2D masked image modeling, masked point modeling methods [32, 51] have be proposed recently. Point-BERT [51] adopts a BERT-style pre-training strategy by predicting discrete tokens of masked input point parts. Point-MAE [32] simply predicts the original coordinates of the masked point patches tokens.

2.2. Masked Image Modeling

Motivated by the success of BERT [12] for masked language modeling, Masked Image Modeling (MIM) [1, 3, 6, 13, 18, 24, 44, 47, 54] becomes a popular pretext task for self-supervised visual representation learning. BEiT [3] first introduces the pre-training pattern of BERT into the computer vision filed by masking out the random image patches and predicts discrete tokens. MAE [18] and SimMIM [47] both propose to predict the raw pixels of the masked patches. Compared with SimMIM, MAE is more pre-training efficient because it only takes the visible token as the input of the encoder and passes all tokens through a light-weight decoder. Many following works use such asymmetric architecture but explore different prediction targets. MaskFeat [44] uses low-level local features HOG [10] as the prediction target. A^2MIM [24] introduce to learn the frequency component of the masked patch features. PeCo [13] uses an offline visual perceptual codebook to guide the training.

2.3. Geometry Learning in Point Cloud

In computer graphics, previous works [19, 31, 40, 45, 52, 59] propose various methods for calculation of differential properties of 3D discrete geometry. Curvature and normal are two of these most important properties. Taubin algorithm [40] proposes to estimate the curvature of a surface at each point of a polyhedral approximation. CAN [52] introduces Local Fitting for normal curvatures by employing chord, neighbor normal vector and osculating circle. As for surface normal estimation, Hoppe *et al*. [19] first suggests to fit a least square plane to k nearest neighbors of each point to estimate its normal. Mitra *et al*. [31] analyzes the methods of least square with noise added and provides theoretical bound.

In deep learning filed, methods [9, 30, 34, 35, 41, 43, 56] are commonly based on some assumptions of implicit local geometry. Point-based methods [25, 26, 34–36, 41, 43] usually adopt set abstraction to capture local points features in region-wise. Voxel-based methods [9, 22, 55, 56] project the point clouds to 3D voxel grids and encode features of points inside the same voxel by voxelization.

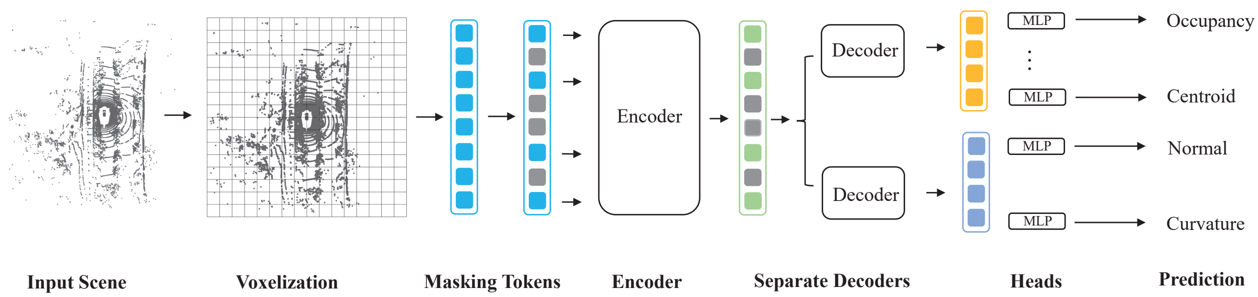

Figure 2. Architecture Overview. The input point cloud scene is first voxelized into voxel grids. After the voxelization, we randomly mask the voxel tokens and fed the visible ones into a sparse encoder-decoder transformer. The encoder is encouraged to capture the geometric information of the point cloud by supervising our proposed geometric prediction targets.

3. Method

3.1. Architecture Overview

We propose a simple yet effective method for self-supervised point cloud representation learning, named **GeoMAE** . GeoMAE predicts both point statistics and surface geometric properties from point clouds. The overall pipeline is illustrated in Figure 2. First, we voxelize the original point cloud and transform it into voxel patch tokens. We randomly masked out voxel tokens based on predefined ratios for a challenging pre-text task. We define a set of learnable tokens for masked tokens. These visible tokens (corresponding to masked tokens) are fed into a sparse transformer encoder. Conditioned on features of visible tokens, learnable masked tokens are processed by separated decoders to predict both point statistics (centroid and occupancy) and surface properties (normal and curvature). Next, we will elaborate on each step.

Voxel Token Embedding and Masking. We follow recent 3D perception architectures [48] and transform sparse input point clouds into regular voxel grids. Then, these voxels are processed by 3D convolutional neural networks or transformer-based networks. We adopt the widely-used dynamic voxelization [55] to perform voxelization: First, the input scene is divided into equally spaced voxels as shown in Figure 2. Each point p_i will be assigned to a voxel v_j where the point resides. Then, we pass non-empty voxels through VFE [56] layers to obtain per-voxel features/tokens T_v. Based on evidences from 2D masked modeling methods [18], we choose a high mask ratio (70%) when removing tokens. Our method predicts target properties per learnable masked token.

Sparse Encoder. After random masking, only visible voxel tokens are fed into an encoder. Due to the sparse and long-range nature of the input scene, we choose a sparse transformer proposed in SST [15] as our encoder. Similar to Swin-Transformer [27], self-attention is only calculated among non-empty voxels within the same region in SST. The output token of the encoder is T_e, together with the learnable masked token T_m to form the input T_d of the decoder.

Decoders. We use two separate decoders to decode point statistics and surface properties, respectively. Each decoder consists of two sparse transformer blocks. These two decoders take the same input T_d and generate two output features T_point and T_surface. Empirically, we found such separate design better facilitated models to learn point statistics and surface properties than a single shared decoder did. Finally, we use separate prediction heads with lightweight MLPs to predict each target $P \in \mathbb{R}^{N \times K}$ based on features produced by previous decoders.

Prediction Targets. The prediction targets include the point statistics and surface properties of a point cloud region. The point statistics contain two objectives: pyramid centroid T_cent and occupancy T_occ. There are also two objectives for surface properties: surface normal T_norm and surface curvature T_curv. The details of each prediction target will be discussed in Section 3.2 and Section 3.3.

We train our network to learn the point statistics and surface properties of uneven point clouds by supervising those prediction targets:

$$\begin{aligned}\mathcal{L}_\text{point} &= \mathcal{L}_\text{cent}(P_\text{cent}, T_\text{cent}) + \mathcal{L}_\text{occ}(P_\text{occ}, T_\text{occ}),\\ \mathcal{L}_\text{surface} &= \mathcal{L}_\text{curv}(P_\text{curv}, T_\text{curv}) + \mathcal{L}_\text{nor}(P_\text{nor}, T_\text{nor}),\end{aligned} \tag{1}$$

For centroid, curvature and normal prediction, we use MSE loss, and for occupancy prediction we use Cross-Entropy loss. The overall loss function of our framework is defined as:

$$\mathcal{L} = \mathcal{L}_\text{point} + \mathcal{L}_\text{surface} \tag{2}$$

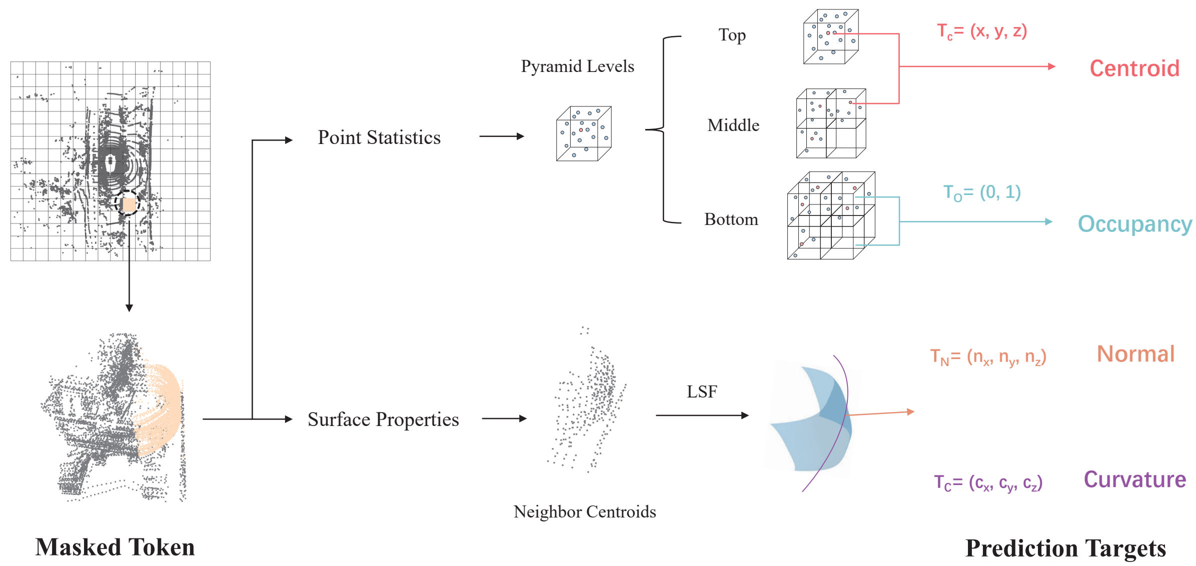

Figure 3. Prediction Targets. We introduce point statistics and surface properties prediction targets to guide the model learning the geometric features of the point cloud. The point statistics targets contain the occupancy of each voxels and centroid of non-empty voxels in a pyrmaid level. The surface properties prediction targets include surface normal and surface curvature which are obtained by an estimation calculation.

3.2. Point Statistics Prediction

Different from 2D images and 3D indoor point clouds, outdoor point clouds are sparse and occluded. Point density varies much in a point cloud, which prevents models directly predicting original point coordinates. The pilot study in Figure 1 also shows that such a prediction target is not available. To deal with non-uniform points, we opt to predict centroid of points in each voxel. In addition, to incorporate multi-scale information, we aim to predict these statistics in different scales by building a voxel pyramid. As shown in Figure 3, we break each masked voxel into three sub-voxel levels (top, middle, and bottom) and compute the voxel occupancy and centroid at each level.

Centroid and Occupancy. Let $\mathcal{G}^l = \{G^l_i = \{I_{G^l_i}\} | i = 1, ..., N_l; l \in \{top, middle, bottom\}\}$ be the set of non-empty grids in the l-th pyramid level where $I_{G^l_i}$ is the grid index, and N_l is the number of non-empty grids. Points that are within the same grid G^l_i are grouped together into a set $\mathcal{N}(G^l_i)$ by calculating their belonging grid index $I_{G^l_i}$ from their spatial coordinates. The point centroid of each grid G^l_i is then calculated as:

$$c_{G^l_i} = \frac{1}{|\mathcal{N}(G^l_i)|} \sum_{x_{p_j} \in \mathcal{N}(G^l_i)} x_{p_j} \tag{3}$$

We also introduce an occupancy prediction target to judge whether a grid is empty or not:

$$o_{G^l_i} = \begin{cases} 1, & \text{at least one point in the grid} \\ 0, & otherwise \end{cases} \tag{4}$$

The point statistics prediction targets for each masked token v_j can be formalized as:

$$P^j_{cent} = \{c_{G^l_i}\}, \quad P^j_{occ} = \{o_{G^l_i}\} \tag{5}$$

3.3. Surface Properties Prediction

LiDAR point clouds naturally preserve geometric information. Although point statistics provide a rough estimation of shapes, they cannot describe the fine-grained geometric information that are usually critical to recognition tasks. Therefore, in addition to point statistics prediction, we further leverage 3D shape geometry of point clouds for self-supervised learning. Our desiderata include: these geometric features should be easy to compute and accurately approximate local shape geometry; we can estimate these features from local point groups. Therefore, our choices are surface curvature and surface normals which can be computed in closed-form from local points. To obtain a more stable geometric representation, we incorporate points from 8 neighboring voxels in addition to inside points in each voxel.

Surface Normal and Curvature. Inspired by surface feature estimation (i.e., curvature estimation [52] and noraml estiamtion [31]) in geometry processing, we adopt local least square fitting to handle noisy LiDAR point clouds. Given a set of K gathered points $\mathbf{p}_i$ $(1 \leq i \leq K)$, we compute a covariance matrix

$$M = \frac{1}{K} \sum_{i=1}^{K} \mathbf{p}_i \mathbf{p}_i^T - \bar{\mathbf{p}} \bar{\mathbf{p}}^T, \tag{6}$$

Method	Waymo		nuScenes	
	L1 AP/APH	L2 AP/APH	mAP	NDS
Scratch	70.68/66.39	64.28/60.42	50.39	55.04
BYOL [16]	70.15/65.72	63.71/59.94	50.01	54.67
PointContrast [46]	71.73/67.28	65.34/61.45	50.96	55.39
SwAV [5]	71.85/67.43	65.41/61.63	51.57	55.72
STRL [20]	71.91/67.64	65.52/61.77	51.72	55.84
GeoMAE (Ours)	**73.71/70.24**	**67.30/63.97**	**53.77**	**57.23**

Table 1. Performances of 3D object detection on the Waymo Open Dataset and nuScenes Dataset validation split.

where M is a 3×3 symmetric matrix, $\bar{\mathbf{p}}$ is the centroid of this point cluster. After the eigen-decomposition of M, we obtain eigenvalues λ_1, λ_2, λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) and their corresponding eigenvectors and $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$. In fact, we use singular value decomposition. Following the aforementioned work [31], the normal vector for each voxel is $\mathbf{n}_3$ (the corresponding eigenvector of the least eigenvalue). Moreover, we compute three pseudo curvature vectors c_m for each point:

$$c_m = \frac{\lambda_m}{\sum_{i=1}^{3} \lambda_i}, m \in \{1, 2, 3\}. \tag{7}$$

Therefore, surface properties prediction targets for each masked token v_j can be formalized as:

$$P^j_{nor} = n^j_3, \quad P^j_{curv} = \{c^j_0, c^j_1, c^j_2\} \tag{8}$$

4. Experiments

In this section, we evaluate our proposed GeoMAE on two widely used benchmarks: Waymo Open Dataset [39] and nuScene Dataset [4]. We first elaborate the experiment setting in Section 4.1. In Section 4.2 we compare our method with previous self-supervised point cloud representation learning methods. In Section 4.3 we show the generalization of our method on different downstream tasks. In Section 4.4, we conduct various ablation studies to evaluate the effectiveness of our approach.

4.1. Experimental Setup

Waymo Open Dataset. Waymo Open Dataset [39] consists of 798 training sequences and 202 validation sequences. The point cloud scene is collected by a 64-beam LiDAR with around 158k point cloud samples in the training split and 40k point cloud samples in the validation split. For 3D detection, the official evaluation metric includes standard 3D mean Average Precision (mAP) and mAP weighted by heading accuracy (mAPH). These metrics are based on an IoU threshold of 0.7 for vehicles and 0.5 for other categories. These metrics are further broken down into two difficulty levels: L1 for boxes with more than five LiDAR points and L2 for boxes with at least one LiDAR point.

nuScenes Dataset. The nuScenes Dataset [4] is a large-scale autonomous driving dataset that contains 700, 150, 150 sequences for training, validation, and testing, respectively. For 3D detection, the major official metrics are mean Average Precision (mAP) and nuScenes detection score (NDS). The mAP uses a bird-eye-view center distance threshold (0.5m, 1m, 2m, 4m) instead of bounding box IoU. NDS is a weighted average of mAP and other attribute metrics, including translation, scale, orientation, velocity, and other box attributes. For 3D tracking, nuScenes mainly uses AMOTA, which penalizes ID switches, false positive, and false negatives and is averaged among various recall thresholds.

Model. Our proposed GeoMAE use a standard SST [15] as the encoder, which contains two consecutive blocks. Each attention module in the encoder has two heads, 128 input channels, and 256 hidden channels. We use two parallel decoders, and each decoder has two transformer blocks. Given the masked voxels V_m with the voxel grid size g_x, g_y, g_z along the x, y, and z axes of the 3D space, respectively, we sub-divide each voxel's spatial space into three pyramid levels (as shown in Figure 3): top, middle, and bottom. The grid in the top level has the same grid size as the original voxel grid. The sub-grids in the middle level have a grid size of $g_x/2$, $g_y/2$, and $g_z/4$, while the grid size in the bottom level is $g_x/4$, $g_y/4$, and $g_z/8$. The number of grids in each level, from top to bottom, is 1, 16, and 128, respectively.

Training Details. On the Waymo Open Dataset, we use all the samples for pre-training and uniformly sample 20% of frames for finetuning following common practice. On the nuScenes Dataset, we use all the frames (including the unlabeled sweeps) for pre-training and the labeled samples for finetuning. For pre-training, all the self-supervised methods are trained for 72 epochs (denoted as 6x) with the AdamW optimizer [21]. The initial learning rate is 1e-5. For finetuning on downstream tasks, we follow the original training

Method	Detection Head	Waymo		nuScenes	
		L1 AP/APH	L2 AP/APH	mAP	NDS
SpConv*	Anchor-base	66.74/63.05	60.27/57.12	48.46	53.92
SST		70.68/66.39	64.28/60.42	50.39	55.04
GeoMAE		**73.71/70.24**	**67.30/63.97**	**53.77**	**57.23**
SpConv*	Center-base	69.45/65.27	63.19/59.44	55.67	64.03
SST		70.23/66.29	64.00/60.39	55.71	64.07
GeoMAE		**73.14/69.68**	**66.95/63.75**	**58.73**	**66.68**

Table 2. Performances of 3D object detection on the Waymo Open Dataset and nuScenes Dataset validation split. *: re-implemented by MMDetection3D.

Method	mAP	NDS
PointPillars [22]	30.5	45.3
3DSSD [49]	56.4	42.6
CBGS [57]	63.3	52.8
CenterPoint [50]	58.0	65.5
VISTA [11]	63.0	69.8
Focals Conv [7]	63.8	70.0
TransFusion-L [2]	65.5	70.2
LargeKernel3D [8]	65.4	70.6
GeoMAE†	**67.8**	**72.5**

Table 3. Performances of 3D object detection on the nuScenes test split. † means that we use a multi-stride structure.

settings in each downstream approach to finetune the detector, segmenter, and tracker.

4.2. Comparison on 3D Object Detection

Unlike 2D MIM methods which adopt image classification task as the benchmark to evaluate the effectiveness of their pre-training methods, we do not have a classification task for scene-level 3D point clouds. So we choose the 3D object detection task to compare our method with previous 3D self-supervised methods.

Settings. We compare our GeoMAE with several typical 3D self-supervised learning methods including PointContrast [46], STRL [20], BYOL [16], and SwAV [5]. We follow the strategy in [29] to apply these methods for pre-training the SST backbone. Details are presented in the Appendix. After pre-training, we evaluate the pre-trained backbones based on the 3D object detector benchmark proposed in SECOND [48]. Detectors with different pre-trained SST backbones are all finetuned for 24 epochs on the Waymo Open Dataset and 20 epochs on the nuScenes Dataset.

Results. The finetuning results on 3D object detection are shown in Table 1. Our proposed GeoMAE significantly improves SST, which is 3.03/3.85 L1 AP/APH better than training from scratch on the Waymo Open Dataset and 3.38 mAP on the nuScenes Dataset. Compared with other self-supervised methods, GeoMAE outperforms the second best method STRL [20] by a significant margin, 1.78/2.20 L1 AP/APH on the Waymo Open Dataset and 2.05 mAP on the nuScenes Dataset, demonstrating the effectiveness of our model and prediction target designs.

4.3. Comparison on other Downstream Tasks

We further evaluate the effectiveness and generalization of GeoMAE in different 3D downstream tasks (including detection, segmentation and tracking) with different model architectures (backbones and heads). For each model in a downstream task, we evaluate three variants with different backbones: 1. the original sparse convolutional networks without pre-training; 2. SST without pre-training; 3. SST pre-trained by our GeoMAE .

4.3.1 3D Object Detection

Settings. We comprehensively evaluate GeoMAE on both the anchor-based detector SECOND and a center-based detector CenterPoint [50]. For the Waymo Open Dataset, the detection point cloud range is set to [-74.88m, 74.88m] for X- and Y-axes, [-2m, 4m] for Z-axes, and the voxel size is set to (0.32m, 0.32m, 6m). For nuScenes Dataset, the detection range is set to [-51.2m, 51.2m] for X- and Y-axes, [-5m, 3m] for Z-axes, and the voxel size is set to (0.256m, 0.256m, 8m).

Results. As shown in Table 2, both anchor-based and center-based detectors with pre-trained SST by our GeoMAE achieve better performance than the baselines. For anchor-based detector, our GeoMAE outperforms the baseline by 3.03 L1 AP on the Waymo Open Dataset and 3.38 mAP on the nuScenes Dataset. While for the center-based detector, our approach improves the results of training from scratch by 2.91 L1 AP on the Waymo Open Dataset and

Methods	mIoU	barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other	sidewalk	terrain	manmade	vegetation
Cylinder3D [58]	76.1	76.4	40.3	91.2	93.8	51.3	78.0	78.9	64.9	62.1	84.4	96.8	71.6	76.4	75.4	90.5	87.4
Cylinder3D-SST	76.5	76.2	40.0	91.8	94.2	51.6	78.1	80.1	64.7	62.5	84.7	97.1	71.7	76.7	75.8	90.8	87.7
Cylinder3D-GeoMAE	**78.6**	**78.2**	**42.6**	**93.5**	**95.8**	**55.4**	**79.8**	**83.5**	**66.8**	**65.6**	**87.3**	**97.7**	**73.3**	**78.2**	**77.4**	**92.6**	**89.5**

Table 4. Performances of 3D semantic segmentation on the nuScenes Dataset validation split.

Methods	Modality	mIoU	Drivable	Ped. Cross.	Walkway	Stop Line	Carpark	Divider
CenterPoint [50]		48.6	75.6	48.4	57.5	36.5	31.7	41.9
CenterPoint-SST	L	49.7	77.2	49.5	58.7	37.2	32.5	43.1
CenterPoint-GeoMAE		**52.4**	**79.5**	**53.1**	**61.6**	**39.7**	**34.9**	**45.6**
BEVFusion [28]		62.7	85.5	60.5	67.6	52.0	57.0	53.7
BEVFusion-SST	C + L	63.1	86.1	60.8	68.7	53.5	53.8	55.5
BEVFusion-GeoMAE		**65.2**	**86.5**	**62.3**	**70.2**	**55.7**	**59.4**	**56.8**

Table 5. Performances of BEV map segmentation on the nuScenes Dataset validation split.

3.02 mAP on the nuScenes Dataset. All the results verify the efficacy of our proposed method. In Table 3, we also test our method on the nuScenes test split and achieve a new state-of-the-art result.

4.3.2 3D Object Tracking

Settings. We also conduct experiments in a 3D multi-object tracking (MOT) task on the nuScenes Dataset by performing tracking-by-detection algorithms proposed by CenterPoint [50] and SimpleTrack [33]. The point cloud range and voxel size are the same as the 3D object detection settings.

Results. From Table 6, we can see that our GeoMAE outperforms the baseline (SST) by 1.7 AMOTA for Centerpoint and 1.1 AMOTA for SimpleTrack. These observations are consistent with those in 3D object detection.

Method	AMOTA↑	AMOTP↓	MOTA↑	IDS↓
Centerpoint* [50]	57.3	0.681	0.522	594
Centerpoint-SST	59.9	0.660	0.514	586
Centerpoint-GeoMAE	**61.6**	**0.635**	**0.635**	**582**
SimpleTrack* [33]	63.2	0.678	0.548	520
SimpleTrack-SST	63.8	0.653	0.541	514
SimpleTrack-GeoMAE	**64.9**	**0.624**	**0.561**	**473**

Table 6. Performances of 3D multi-object tracking on the nuScenes Dataset validation split. *: re-implemented by MMDetection3D.

4.3.3 LiDAR Semantic Segmentation

Settings. To demonstrate the generalization capability, we evaluate our method on the nuScenes Dataset for the LiDAR segmentation task. We follow the official guidance to leverage mean intersection-over-union (mIoU) as the evaluation metric. We adopt the Cylinder3D [58] as our baseline architecture and replace the last stage of the backbone from sparse convolutions into SST. Other training settings are the same as in Cylinder3D.

Results. As reported in Table 4, the Cylinder3D obtains 0.4 performance gain by replacing the backbone from sparse convolutions with our modified SST. When applying our GeoMAE to pre-train the backbone, it achieves 2.1 mIoU gain than training from scratch.

4.3.4 BEV Map Segmentation

Settings. We further experiment our method in the BEV Map Segmentation task on the nuScenes Dataset. We perform the evaluation in the [-50m, 50m]×[-50m, 50m] region following the common practice in BEVFusion [28]. We develop the CenterPoint-SST and BEVFusion-SST by replacing the last two stages of the LiDAR backbone with SST.

Results. We report the BEV map segmentation results in Table 5. For the LiDAR-only model, our method surpass the SST baseline by 2.7 mIoU. In the multi-modality setting, GeoMAE further boosts the performance of BEVFusion-SST about 2.1 mIoU, which demonstrates the strong generalization capability of our method.

4.4. Ablation Study

We adopt standard SST [15] the default backbone in our ablation study. To get efficient validation and reduce experimental overhead, all the experiments are pre-trained for 72 (6x) epochs if not specified.

Prediction Targets. We present ablation studies in Table 7 to justify our design choices. Scratch means train-

Methods	Type	Waymo L1 AP	nuScenes mAP
Scratch		70.68	50.39
+ Centroid	Point Statistics	71.60	51.25
+ Occupancy		72.65	52.12
+ Surface Normal	Surface Properties	73.37	52.94
+ Surface Curvature		**73.71**	**53.31**

Table 7. Ablation study on prediction targets. Detection results on the Waymo and nuScenes Dataset.

ing on the detection from scratch without pre-training the backbone. From the table we can see that by adopting centroid and occupancy as the prediction targets, the model performs better than training form scratch with about 1.87 L1 AP gain on the Waymo Open Dataset and 1.7 mAP gain on the nuScene Dataset. As shown in the last two rows, we investigate the effect of predicting the surface normal and curvature. The additional prediction targets further improve the performance 1.06 AP on the Waymo Open Dataset and 1.19 mAP on the nuScenes Dataset.

Decoder Design. The separate decoder is one key component in our GeoMAE . We ablate the design in Table 8. Shared means we use a single decoder to decode both Point Statistics and Surface Properties information. Same target means we use two decoders but the separate decoders decode and predict the same Point Statistics targets or the same Surface Properties targets. It can be observed that our final design achieves the best performance, which indicates that such separate decoder truly disentangles the different representations and enable the model to learn different geometry information.

Decoder	Waymo L1 AP	nuScenes mAP
Shared Decoder	72.45	52.04
Separate (Only Point Statistics)	72.65	52.28
Separate (Only Surface Properties)	72.03	51.87
Separate (Different Targets)	**73.71**	**53.31**

Table 8. Ablation study on decoder design. Detection results on the Waymo and nuScenes Dataset.

Masking Ratio. We study the influence of the masking ratio in Table 10. Similar to the observation on images, the optimal masking ratio is high (70%). The performance degrades largely with too low or too high masking ratios.

Training Schedule. Table 9 shows the effect of the training schedule length. We ablate the pre-training schedule of GeoMAE from 24 (2x) to 96 (8x) epochs and fix the finetuning epoch as 24 epochs. The accuracy improves steadily with longer training schedule until 72 epochs.

Pre-training Dataset Scale. We also investigate the effect of different scales of pre-training dataset. As shown in Table 11, performance grows as the scale of pre-training

Training Epochs	Waymo L1 AP	nuScenes mAP
24	72.67	52.64
48	73.32	52.97
72	**73.71**	**53.31**
96	73.71	53.30

Table 9. Ablation on training schedule. Detection results on the Waymo and nuScenes Dataset.

Masking Ratio	Waymo L1 AP	nuScenes mAP
40%	72.27	52.47
60%	73.28	53.00
70%	**73.71**	**53.31**
80%	71.94	51.85

Table 10. Ablation on masking ratio. Detection results on the Waymo and nuScenes Dataset.

Dataset Scale	Waymo L1 AP	nuScenes mAP
0%	70.68	50.39
20%	72.27	51.64
50%	72.88	52.74
80%	73.48	53.03
100%	**73.71**	**53.31**

Table 11. Impacts of different scale of the pre-training dataset. Detection results on the Waymo and nuScenes Dataset.

data increases. And our method still achieves about 1 points performance improvements on both datasets, which indicates the effectiveness of our GeoMAE.

5. Conclusion

We present GeoMAE, a geometry-aware self-supervised pre-training approach for point clouds. GeoMAE achieves strong performance on a variety of downstream tasks including 3D detection, segmentation, and tracking. GeoMAE leverages recent development in masked modeling. In addition to the commonly used occupancy prediction target, our method proposes three additional learning objectives, which jointly become a challenging and informative pretext task. Our key observation is that geometric features provide strong information for models to reason objects and scenes, therefore improve downstream recognition performance. Our results also suggest several venues for future inquiry. First, pre-training using GeoMAE on larger unlabeled dataset will further boost the performance (*e.g.*, pre-training on DDAD dataset [17]). Besides, exploring other types of geometric features remains an open and intriguing question.

Acknowledgement This work was supported by National Key R&D Program of China (2022ZD0161700).

References

[1] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022.

[2] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022.

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[6] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.

[7] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5428–5437, 2022.

[8] Yukang Chen, Jianhui Liu, Xiaojuan Qi, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Scaling up kernels in 3d cnns. *arXiv preprint arXiv:2206.10555*, 2022.

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[11] Shengheng Deng, Zhihao Liang, Lin Sun, and Kui Jia. Vista: Boosting 3d object detection via dual cross-view spatial attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8448–8457, 2022.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021.

[14] Benjamin Eckart, Wentao Yuan, Chao Liu, and Jan Kautz. Self-supervised learning on 3d point clouds by learning discrete generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8257, 2021.

[15] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing Single Stride 3D Object Detector with Sparse Transformer. In *CVPR*, 2022.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[17] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[19] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, pages 71–78, 1992.

[20] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.

[23] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406, 2018.

[24] Siyuan Li, Di Wu, Fang Wu, Zelin Zang, Baigui Sun, Hao Li, Xuansong Xie, Stan Li, et al. Architecture-agnostic masked image modeling–from vit back to cnn. *arXiv preprint arXiv:2205.13943*, 2022.

[25] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed

points. *Advances in neural information processing systems*, 31, 2018.

[26] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8778–8785, 2019.

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[28] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022.

[29] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, et al. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*, 2021.

[30] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015.

[31] Niloy J Mitra and An Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 322–328, 2003.

[32] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. *arXiv preprint arXiv:2203.06604*, 2022.

[33] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. *arXiv preprint arXiv:2111.09621*, 2021.

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[36] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18942–18952, 2022.

[37] Yongming Rao, Jiwen Lu, and Jie Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5376–5385, 2020.

[38] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32, 2019.

[39] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[40] Gabriel Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proceedings of IEEE International Conference on Computer Vision*, pages 902–907. IEEE, 1995.

[41] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019.

[42] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9782–9792, 2021.

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[44] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022.

[45] William Welch and Andrew Witkin. Free-form shape design using triangulated surfaces. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 247–256, 1994.

[46] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, pages 574–591. Springer, 2020.

[47] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022.

[48] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[49] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020.

[50] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[51] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. 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*, pages 19313–19322, 2022.

[52] Xiaopeng Zhang, Hongjun Li, Zhanglin Cheng, et al. Curvature estimation of 3d point cloud surfaces through the fitting of normal section curvatures. *Proceedings of ASIAGRAPH*, 2008:23–26, 2008.

[53] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021.

[54] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

[55] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, pages 923–932. PMLR, 2020.

[56] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[57] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019.

[58] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9939–9948, 2021.

[59] Matthias Zwicker, Mark Pauly, Oliver Knoll, and Markus Gross. Pointshop 3d: An interactive system for point-based surface editing. *ACM Transactions on Graphics (TOG)*, 21(3):322–329, 2002.