Abstract

Current perception models in autonomous driving greatly rely on large-scale labeled 3D data. However, it is expensive and time-consuming to annotate 3D data. In this work, we aim at facilitating research on self-supervised learning from the vast unlabeled 3D data in autonomous driving. We introduce a masked autoencoding framework for pre-training large-scale point clouds, dubbed Voxel-MAE. We take advantage of the geometric characteristics of large-scale point clouds, and propose the range-aware random masking strategy and binary voxel classification task. Specifically, we transform point clouds into volumetric representations, and randomly mask voxels according to their distance to the capture device. Voxel-MAE reconstructs the occupancy values of masked voxels and distinguishes whether the voxels contain point clouds. This simple binary voxel classification objective encourages Voxel-MAE to reason over high-level semantics to recover the masked voxel from only a small amount of visible voxels. Extensive experiments demonstrate the effectiveness of Voxel-MAE across several downstream tasks. For the 3D object detection task, Voxel-MAE reduces half labeled data for car detection on KITTI and boosts small object detection by around 2% mAP on Waymo. For the 3D semantic segmentation task, Voxel-MAE outperforms training from scratch by around 2% mIOU on nuScenes. For the first time, our Voxel-MAE shows that it is feasible to pre-train unlabeled large-scale point clouds with masked autoencoding to enhance the 3D perception ability of autonomous driving. Codes are publicly available at https://github.com/chaytonmin/Voxel-MAE.

1. Introduction

3D perception aims to obtain accurate surrounding information, which is one of the core techniques in autonomous driving. Many autonomous driving datasets [9, 24, 1, 17] have been published and have extensively promoted the capability of environmental perception for unmanned vehicles. However, collecting and annotating 3D data for common tasks, such as 3D object detection and semantic segmentation, are time-consuming and laborious. For example, a skilled worker can only annotate about 100-200 frames per day [17]. Self-supervised learning on large-scale unannotated 3D data is of great significance to the perceptual ability of autonomous driving, which may become the solution of next-generation industry-level powerful and robust autonomous driving perception model [17].

Self-supervised learning enables pre-training rich features without human annotations, which has made tremendous strides in recent years. In particular, the simple masked autoencoding has proved effective in learning representative features, whose task is to reconstruct the masked data from the unmasked input [7, 13, 5, 32]. For example, in natural language processing (NLP), masked autoencoding has enabled the training of large language models, such as BERT [7]. In 2D vision, masked autoencoding has outperformed the supervised pre-training counterparts [13]. However, the existing masked autoencoding methods are mainly designed for language and 2D data. Training from scratch on large-scale labeled data is still the dominant approach in 3D vision [31]. To introduce the idea of masked autoencoding to self-supervised learning on large-scale 3D point clouds, we identify challenges and propose solutions from the following perspectives:
(i) It is trivial to reconstruct the features of large-scale point clouds. There are two typical representations of point clouds, i.e., point-based and voxel-based. We choose the voxel-based representation for its computational efficiency [40]. The voxel features of large-scale point clouds contain the spatial information of voxels [34]. Regression of the voxel features is not suitable, as the positional encoding of voxels would provide a shortcut for decoder [31]. We address this issue by focusing on the occupancy distribution of voxels and designing the binary voxel classification objective as the pretext task. This simple task pushes the network to learn representative features to recover the overall structure of the 3D scene.

(ii) Data distribution of large-scale point clouds in autonomous driving is different from language and 2D images. The density of point clouds will decrease as the distance to the LiDAR sensor rises, thus it is unsuitable to perform the same random masking strategy for all voxels as in NLP and 2D vision. To address this, we propose the range-aware random masking strategy for large-scale point clouds, i.e., the masking ratio decreases as the distance of the voxel to the LiDAR sensor rises.

(iii) The Transformer architecture can not handle massive unmasked voxels for large-scale point clouds. Transformers in NLP and 2D vision perform self-attention on the unmasked patches. However, even after masking 90% voxels, there are still hundreds of thousands of unmasked voxels for large-scale point clouds. We propose to design the encoder with the 3D Spatially Sparse Convolutions [34], which uses the positional encoding module to focus on the visible voxels, similar to positional embeddings in Transformers.

Driven by these analyses, we present a novel self-supervised learning framework for pre-training large-scale point clouds based on masked voxel autoencoding, termed as **Voxel-MAE**. As illustrated in Figure 2, our Voxel-MAE first randomly masks voxels with the range-aware masking strategy and then puts the unmasked voxels into the 3D sparse encoder, and the output of the 3D decoder is the probability that each voxel contains points. At last, we calculate the binary voxel classification loss for pre-training the network. The pretext task of masked voxel classification pushes the encoder network to be voxel-aware of the whole shape of objects, thus learning representative features for 3D perception.

Our Voxel-MAE is simple and generalizes well on various downstream tasks. For the 3D object detection task, Voxel-MAE outperforms the state-of-the-art self-supervised learning methods by 0.5% ~ 6% mAP on ONCE [17] dataset, which is a recently published dataset for self-supervised learning on large-scale point clouds. And with Voxel-MAE pre-training, we can improve the performance of the popular 3D detectors [34, 21, 37, 22] trained from scratch on KITTI [9], Waymo [24], and nuScenes [1], especially for small objects. As shown in Figure 1, Voxel-MAE is a data-efficient learner that allows to effectively train large-scale point clouds only from limited annotated 3D data. For the 3D semantic segmentation task, Voxel-MAE with a two-layer decoder can improve training from scratch by about 2% mIOU. We also validate the effectiveness of Voxel-MAE on the unsupervised domain adap-
tation task, which proves the transfer learning ability of our Voxel-MAE. As point clouds in autonomous driving are information redundant, Voxel-MAE with a 90% masking ratio can still learn representative features and improves the performance of 3D perception. These observations are consistent with those in self-supervised pre-training in NLP [7] and 2D vision [13], and we hope they will promote the development of self-supervised learning for autonomous driving.

The main contributions of this work are listed below:

• To our best knowledge, we present the first masked autoencoding framework for self-supervised learning on large-scale point clouds. Our method is a data-efficient learner that can reduce the demand for expensively annotated 3D data.

• We propose the range-aware random masking strategy to improve the pre-training performance of far objects. The masking ratio of voxels is related to the distance from the LiDAR sensor.

• We design the binary voxel classification task to enforce the network to reason over high-level semantics to recover the masked occupancy distribution of the 3D scene from only a small amount of visible voxels.

• Voxel-MAE significantly outperforms training from scratch over a variety of downstream tasks, including 3D object detection, semantic segmentation, and unsupervised domain adaptation. Experiments on ONCE show that our method achieves state-of-the-art performance among competing self-supervised learning methods.

2. Related Work

2.1. LiDAR-based 3D Perception

LiDAR-based 3D perception methods with accurate 3D spatial information have been widely used in the field of autonomous driving. In this section, we give an overview of the recent advances in both LiDAR-based 3D object detection and 3D semantic segmentation. LiDAR-based 3D object detectors can be categorized into point-based [23, 36], voxel-based [40, 34, 15], and point-voxel-based [21, 6]. Point-based object detectors extract discriminative features from raw point clouds with PointNet [20] and generate proposals centered at each point with high computation cost. Voxel-based detectors transform the irregular point clouds into volumetric representations, which will degrade the fine-grained localization accuracy. Point-voxel-based methods take advantage of the localization accuracy of point-based detectors and the computational efficiency of voxel-based detectors. LiDAR-based 3D segmentation methods are divided into grid-based [29, 18] and voxel-based [41, 25]. Grid-based methods focus on converting the 3D point clouds into a 2D frontal-view or range image, which fails to model the 3D geometric information. Voxel-based methods convert the point clouds into volumetric representations. The existing 3D perception methods are trained with large-scale labeled 3D data. How to design the self-supervised learning network to minimize dependence on 3D annotation has rarely been studied.

2.2. Self-supervised Learning

Self-supervised Learning (SSL) has been popular in recent years without the expensive data annotation. [8] first proposes the pretext of predicting the relative location of image patches. Methods in [10, 33, 30] design the pretext task of rotation prediction that has shown promising results in representative feature learning. In [2], a jigsaw puzzle prediction task is proposed and generalizes well in domain adaptation object recognition. DeepCluster [3] and SwAV [4] obtain the pseudo label with k-means clustering and use the label to train the network. Moco [14], PointContrast [31], BYOL [12], and DepthConstrast [39] construct contrastive views for self-supervised learning. Recently, MAE [13] first masks random patches of the input image and reconstructs the missing pixels with the simple autoencoder framework, showing promising results in self-supervised learning. VideoMAE [27] extends the MAE into spatiotemporal representation learning from videos that are more information redundant. Point-BERT [38] first introduces MAE to pre-train small-scale point clouds. Point-MAE [19] reconstructs the point patches with the Chamfer distance. MaskPoint [16] designs the decoder for discriminating the masked point patches. However, the existing self-supervised learning methods are focused on small-scale datasets, while it has rarely been studied on automotive millions of point clouds.

3. Methodology

Given the instances from large-scale point clouds, self-supervised pre-training is to train the network with the unlabeled data to generate representative features. Inspired by the excellent performance of masked autoencoding in NLP [7] and 2D vision [13], we design the masked voxel autoencoding network for 3D perception. The proposed Voxel-MAE randomly masks the voxels and then reconstructs the occupancy values of voxels with an autoencoder network. The pretext task of binary voxel classification is trained with the cross-entropy loss. The detailed architecture of Voxel-MAE is demonstrated in Table 1.

3.1. Masked Voxel Modeling

With the $n_s$ unlabeled point cloud data $\{X^i\}_{i=1}^{n_s}$, we aim to first pre-train the masked autoencoding network $\phi_{pre}$ to
learn the high-level semantics. Then we use the pre-trained model to warm up the network $\phi_s$ of downstream tasks. We also extend the pre-training method to domain adaptive task on the target point clouds \{${X}_j$\}$_{j=1}^n$.

### 3.1.1 Range-aware Random Masking

Following the typical setting of 3D perception models [34, 21, 37, 41], the point clouds are divided into spaced voxels. For the point clouds with range $W \times H \times D$ along the $X \times Y \times Z$ respectively, the size of a voxel is $v_W \times v_H \times v_D$ accordingly. The total number of voxels is $n_l$, and the number of occupied voxels that contain points is $n_v$.

Similar to language and 2D images, large-scale point clouds are information-redundant, thus the masking strategy is also applicable to point clouds [31]. However, the data distribution of large-scale point clouds is different from that of language and 2D images. The sparsity levels of the 3D point clouds are correlated to their distances to the LiDAR sensor. The points near the LiDAR sensor are pretty dense, while the points in the far distance are very sparse. Thus we can not perform the same masking strategy for both the near-range and far-range points. We should mask a small percentage of data for the far-range points.

To this end, we use distance information to design the Range-aware random masking strategy, namely, the masking ratio will decrease as the distance from the LiDAR sensor rises. According to the distance of points to the LiDAR sensor, the occupied voxels are divided into three groups: $0$-$30$ meters, $30$-$50$ meters, and $>50$ meters, and the corresponding numbers of voxels are $n_{v1}$, $n_{v2}$, and $n_{v3}$. We take the random masking strategy for each group with the descending masking ratio $r_1$, $r_2$, and $r_3$ (i.e., $r_1 > r_2 > r_3$). Thus the number of unmasked occupied voxels is $n_{un} = n_{v1}(1 - r_1) + n_{v2}(1 - r_2) + n_{v3}(1 - r_3)$, and the set of voxels $V_{input} \in R^{n_{un} \times 4}$ is used as training data. The occupied voxels containing point clouds (the value of each voxel is 1) and the empty voxels (the value of each voxel is 0) are used as ground truth $T \in R^{n_l \times 1}$ for voxel classification loss.

### 3.1.2 3D Sparse Convolutional Encoder

Masked autoencoding in NLP [7] and 2D vision [13] adopts the Transformer network to perform self-attention on the unmasked portions of the training data, which are not influenced by the masked portions. For a 3D scene that contains millions of points, even after masking 90% of voxels, there are still hundreds of thousands of unmasked voxels. The Transformer network can not aggregate information from huge unmasked input data [7]. In the field of processing large-scale point clouds, the 3D Sparse Convolution [11, 34] is proposed, in which the positional encoding module, and thus our voxel masking strategy can reduce the memory complexity of train-

| Input | Layer Destination | Output | Output Size |
|-------|------------------|--------|-------------|
| Input point clouds | Mean operator | Voxel representation | $1600 \times 1408 \times 41 \times 4$ |

|   |   |   |   |
|---|---|---|---|
| Masked voxels $V_{input}$ | 3DSparseConv, filter=(3,3,3), stride=(1,1,1) | Spconv $\_1$ | $1600 \times 1408 \times 41 \times 16$ |
| Spconv $\_1$ | 3DSparseConv, filter=(3,3,3), stride=(2,2,2) | Spconv $\_2$ | $800 \times 704 \times 21 \times 32$ |
| Spconv $\_2$ | 3DSparseConv, filter=(3,3,3), stride=(2,2,2) | Spconv $\_3$ | $400 \times 352 \times 11 \times 64$ |
| Spconv $\_3$ | 3DSparseConv, filter=(3,3,3), stride=(2,2,2) | Spconv $\_4$ | $200 \times 176 \times 5 \times 64$ |
| Latent feature tensor | 3DTransConv, filter=(1,1,3), stride=(1,1,2) | Dense $\_1$ | $200 \times 176 \times 2 \times 128$ |

Table 1. The details of our Voxel-MAE architecture which consists of a 3D Encoder and a 3D Decoder. 3DSparseConv and 3DTransConv denote 3D Sparse Convolution proposed in SECOND [34] and common 3D Deconvolution, respectively. Here we display the output size for pre-training SECOND on the KITTI dataset.
## 4. Experiments

### 4.1. Experimental Setup

We perform three downstream tasks on four autonomous driving datasets [17, 9, 24, 25] to verify the effectiveness of our proposed model. For 3D object detection and unsupervised domain adaptation tasks, we adopt the training settings of the popular point clouds detection codebase OpenPCDet [26] (version 0.5.2). For the 3D semantic segmentation task, we take the open-sourced Cylinder3D [41] as the pre-trained backbone, which adopts the cylindrical voxel partition. For the ONCE dataset, we first pre-train the backbone with Voxel-MAE on the unlabeled raw set and then fine-tune the perception model on the training set. For KITTI, Waymo, and nuScenes datasets, pre-training and fine-tuning are both on the training set.

For voxels within 0-30 meters, 30-50 meters, and >50 meters, we set the masking ratio to 90%, 70%, and 50%, respectively. The number of self-supervised pre-training epochs is 3 in our experiments. In training Voxel-MAE, we use common augmentations, including random world flipping and random world scaling. We choose the ADAM optimizer and cosine annealing learning rate strategy during model training. More detailed parameter setups can be found in OpenPCDet [26, 17] and Cylinder3D [41].

### 4.2. Results on Downstream Tasks

#### 4.2.1 3D Object Detection

We first compare the proposed method with some state-of-the-art self-supervised learning methods on ONCE dataset [17] val set, including three contrastive learning methods (PointContrast [31], BYOL [12], and DepthContrast [39]), as well as two other methods (SECOND [34], DeepCluster [3]).

| Method          | Car   | Pedestrian | Cyclist | Method           | Car   | Pedestrian | Cyclist |
|-----------------|-------|------------|---------|------------------|-------|------------|---------|
| SECOND [34]     | 78.62 | 52.98      | 67.15   | PV-RCNN [21]     | 83.61 | 57.90      | 70.47   |
| Voxel-MAE + SECOND | 78.90 | 53.14      | 68.08   | Voxel-MAE + PV-RCNN | 83.82 | 59.37      | 71.99   |

Table 3. Performance comparison on the KITTI val split evaluated by the mean Average Precision with 11 recall positions at moderate difficulty level.

---

**Table 2. Quantitative detection performance achieved by different self-supervised learning methods on the ONCE val set. The pre-training process is on the unlabeled small set that contains 100k scenes.**

| Method          | overall 0-30m | 30-50m | 50m-inf | overall 0-30m | 30-50m | 50m-inf | overall 0-30m | 30-50m | 50m-inf | mAP   |
|-----------------|---------------|--------|---------|---------------|--------|---------|---------------|--------|---------|-------|
| SECOND [34]     | 71.19         | 84.04  | 63.02   | 47.25         | 26.44  | 29.33   | 24.05         | 18.05  | 58.04   | 69.96 | 52.43 | 34.61 | 51.89 |
| BYOL [12]       | 68.02         | 81.01  | 60.21   | 44.17         | 22.16  | 16.68   | 12.06         | 50.61  | 62.46   | 44.29 | 28.18 | 40.04 | 46.04 |
| PointContrast [31] | 71.07       | 83.31  | 64.90   | 49.34         | 22.52  | 23.73   | 21.81         | 16.06  | 56.36   | 68.11 | 50.35 | 34.06 | 49.98 |
| DepthContrast [39] | 71.88       | 84.26  | 65.58   | 49.97         | 23.57  | 26.36   | 21.15         | 14.39  | 56.63   | 68.26 | 50.82 | 34.67 | 50.69 |
| SwAV [4]        | 72.71         | 83.68  | 66.95   | 50.10         | 25.13  | 27.77   | 22.77         | 16.36  | 58.05   | 69.99 | 52.23 | 34.86 | 51.96 |
| DeepCluster [3] | 73.19         | 84.25  | 66.86   | 50.47         | 24.00  | 26.36   | 21.73         | 16.79  | 58.99   | 70.80 | 53.66 | 36.17 | 52.06 |
| Voxel-MAE       | 72.78         | 83.77  | 66.01   | 50.26         | 27.49  | 30.54   | 25.28         | 16.11  | 57.26   | 69.71 | 52.31 | 33.51 | 52.51 |

---

3.1.3 3D Deconvolutional Decoder

Our decoder consists of light-weight 3D Deconvolution layers, whose last layer outputs the probability that each voxel contains points. The output of decoder is \( P \in R^{n_l \times 1} \). The decoder is only used to perform the occupied voxel reconstruction during pre-training. Shifting the masked tokens to the decoder can promote the encoder to learn better latent features for the downstream tasks.

#### 3.1.4 Reconstruction Target

In NLP [7] and 2D vision [13], the goal of masked auto-encoding is to reconstruct the masked patches as a regression task. However, regression of the voxel features is trivial as the network can find a shortcut for the decoder with the positional encoding of voxels. For 3D perception, the occupancy structure of the 3D scene is of great significance to perception models [40, 34, 41]. Inspired by this, we design the binary voxel classification task for large-scale point clouds pre-training, whose goal is to enforce the network to reason over high-level semantics to recover the masked occupancy distribution of the 3D scene from a small number of visible voxels. Given the predicted occupied voxels \( P \) and the ground truth occupied voxels \( T \), we calculate the binary voxel classification loss of cross-entropy:

\[
loss = -\frac{1}{batch} \sum_{i=1}^{batch} \sum_{j=1}^{n_i} T^i_j \log P^i_j ,
\]

where \( P^i_j \) is the predicted probability of voxel \( j \) of the \( i \)-th training sample, and \( T^i_j \) is the corresponding ground truth whether the voxel contains point clouds.

---

Table 2. Quantitative detection performance achieved by different self-supervised learning methods on the ONCE val set. The pre-training process is on the unlabeled small set that contains 100k scenes.
Table 4. Quantitative detection performance achieved by different methods on the Waymo val set. We subsample a single frame of 20% data (about 32k frames) of all the training samples as the training set.

| Method               | Vec_L1 mAP | Vec_L2 mAP | Ped_L1 mAP | Ped_L2 mAP | Cyc_L1 mAP | Cyc_L2 mAP |
|----------------------|------------|------------|------------|------------|------------|------------|
| SECOND [34]          | 70.96      | 73.34      | 65.23      | 57.22      | 57.13      | 55.62      |
| Voxel-MAE + SECOND   | 71.12      | 70.58      | 66.27      | 59.03      | 59.73      | 56.18      |
| CenterPoint [37]     | 71.33      | 70.76      | 63.16      | 64.27      | 68.68      | 67.39      |
| Voxel-MAE + CenterPoint | 71.89    | 71.33      | 64.08      | 65.78      | 70.29      | 69.03      |
| PV-RCNN [21]         | 75.41      | 74.74      | 67.44      | 63.70      | 65.88      | 64.25      |
| Voxel-MAE + PV-RCNN  | 75.94      | 75.28      | 67.94      | 64.91      | 67.21      | 65.49      |
| PV-RCNN++ [22]       | 77.82      | 77.32      | 69.07      | 69.92      | 63.39      | 61.82      |
| Voxel-MAE + PV-RCNN++| 78.23      | 77.72      | 69.54      | 71.07      | 64.62      | 63.02      |

Table 5. Quantitative detection performance achieved by different methods on the nuScenes val set.

| Method               | mAP  | NDS  | mATE | mASE  | mAOE  | mAVE  | mAAE  |
|----------------------|------|------|------|-------|-------|-------|-------|
| SECOND [34]          | 50.59| 62.29| 31.15| 25.51 | 26.64 | 26.26 | 20.46 |
| Voxel-MAE + SECOND   | 50.82| 62.45| 31.02| 25.23 | 26.12 | 26.11 | 20.04 |
| CenterPoint [37]     | 56.03| 64.54| 30.11| 25.55 | 38.28 | 21.94 | 18.87 |
| Voxel-MAE + CenterPoint | 56.45| 65.02| 29.73| 25.17 | 38.38 | 21.47 | 18.65 |

4.2.2 3D Semantic Segmentation

For self-supervised learning on the 3D semantic segmentation task, We design the lightweight decoder of Voxel-MAE with only two 3D Deconvolutional layers. Table 6 shows that Voxel-MAE outperforms training from the scratch by around 2% in terms of mIOU on nuScenes val set. This implies that Voxel-MAE is well suited for the 3D semantic segmentation task since our masked voxel classification objective promotes the model to capture the occupancy distribution of the 3D space, which is also very important for the dense prediction task, i.e., 3D semantic segmentation. Results in Table 6 were obtained by retraining Cylinder3D [41].

4.2.3 Unsupervised Domain Adaptation

The domain shifts in LiDAR-based 3D perception models are apparent as the LiDARs have different patterns. Evaluation of datasets captured in different locations, conditions, or sensors results in a drop in model performance, due to the gap in distribution between the training (source) data and the test (target) data. To further investigate the generalization ability of Voxel-MAE in representation learning, we conduct experiments in two scenarios of unsupervised domain adaptive (UDA) 3D object detection: different collection locations and time (i.e., Waymo → KITTI) and Waymo (64-beam LiDAR). The high masking ratio of 90% is not applicable to very sparse point clouds. Results for SECOND [34], CenterPoint [37], and PV-RCNN [21] are from OpenPCDet [26] and ONCE [17].

Table 6. Quantitative segmentation performance achieved by different methods on the nuScenes val set.

| Method                  | Epoch | miIoU |
|-------------------------|-------|-------|
| Cylinder3D [41]         | 15    | 70.22 |
|                         | 25    | 70.83 |
| Voxel-MAE + Cylinder3D [41] | 15    | 71.61 |
|                         | 25    | 72.85 |

Table 7. Quantitative results of unsupervised domain adaptation methods on the KITTI val set.

| Task       | Method                  | PV-RCNN BEV 3D |
|------------|-------------------------|----------------|
| Waymo → KITTI | Oracle                  | 88.98 82.50 |
|            | Source-only             | 61.18 22.01 |
|            | SN [28]                 | 79.78 63.60 |
|            | ST3D [35]               | 84.10 64.78 |
|            | ST3D(w/SN) [35]         | 86.65 76.86 |
|            | Voxel-MAE + ST3D        | 85.52 65.24 |
|            | Voxel-MAE + ST3D(w/SN)  | 87.15 77.13 |

nuScenes → KITTI

| Method                  | Epoch | mIoU |
|-------------------------|-------|------|
| Oracle                  |       |      |
| Source-only             |       |      |
| SN [28]                 |       |      |
| ST3D [35]               |       |      |
| ST3D(w/SN) [35]         |       |      |
| Voxel-MAE + ST3D        |       |      |
| Voxel-MAE + ST3D(w/SN)  |       |      |

Table 8. Comparison of different pretext tasks.

| Target   | Car | Pedestrian | Cyclist |
|----------|-----|------------|---------|
| PV-RCNN  [21] | 83.61 | 57.90 | 70.47 |
| Regression | 83.60 | 57.92 | 70.52 |

Table 9. Impacts of pre-training times.

| Epoch | Easy | Moderate | Hard |
|-------|------|----------|------|
| 2     | 81.26 | 67.23 | 61.68 |
| 3     | 82.09 | 68.05 | 62.08 |
| 4     | 82.12 | 68.00 | 61.99 |

4.3. Further Analysis

4.3.1 Data-efficient Learner

Pre-training allows fine-tuning of the model using a small amount of labeled data. We study the data-efficiency of Voxel-MAE by varying the amount of labeled data used for fine-tuning. We use the 3D object detector SECOND [34] as the backbone and report the detection performance on the KITTI val set. As shown in Figure 1 and Figure 4 (a), our Voxel-MAE with 50%, 75%, and 75% of labeled data gets the same performance as training from scratch with full dataset on the car, pedestrian, and cyclist class, respectively. When using 25% samples for fine-tuning, our Voxel-MAE improves training from scratch by 1%~2% mAP. This indicates that Voxel-MAE is a data-efficient learner, which can reduce the demand for expensively human-annotated 3D data.

4.3.2 Range-aware Random Masking

In this section, we study the effectiveness of the proposed range-aware random masking strategy. For range-aware random masking, the masking ratio for voxels within 0-30 meters, 30-50 meters, and > 50 meters is set to 90%, 70%, and 50%, respectively. We compare it with random masking (the masking ratio is set to 90% for all voxels). We can see from Figure 4 (b) that range-aware random masking surpasses random masking by 2.5%, 0.7%, and 2.1% mAP with ONCE raw set as the pre-training datasets. This shows that the masking ratio for large-scale point clouds should be inversely proportional to the distance to the LiDAR sensor.
4.3.3 Binary Voxel Classification

Developing the proper pretext task is the primary concern in self-supervised learning. This section studies the effectiveness of the proposed binary voxel classification task. We replace the binary voxel classification loss with voxel feature regression loss, denoted by voxel regression task. As shown in Table 8, the performance of the voxel regression task is similar to that of training from scratch. The classification task surpasses the regression task by 0.2%, 1.6%, and 1.4% of mAP on the car, pedestrian, and cyclist class, respectively. We argue that the binary voxel classification task is harder for the network to learn and whether the voxel contains points is more critical for 3D perception models. With the simple binary voxel classification task, the pre-trained network can be voxel-aware of object shape, thus improving the performance of downstream tasks.

4.3.4 Visualization

We provide a visualization of reconstructed occupancy voxels with masking ratios of 90% and 70% in Figure 3. Voxel-MAE produces satisfying reconstructed results. Even with a high 90% masking ratio, Voxel-MAE can still recover the masked voxels. This implies that our Voxel-MAE is able to learn useful representations that capture the overall structure of the 3D scene.

4.4. Ablation Studies

In this section, we perform thorough ablation experiments to investigate the individual components of our Voxel-MAE. All experiments are conducted with the 3D object detector SECOND [34].

4.4.1 Pre-training Times

In Table 9, we first study the influence of pre-training times. With the increment of pre-training times, the performance increases at first and afterward will remain stable. We pretrain Voxel-MAE for three epochs in experiments.

4.4.2 Pre-training Data

We then investigate the effects of the amount of pre-training data. Table 10 shows that the performance of Voxel-MAE benefits from the increasing amount of unlabeled data on the ONCE medium raw set. In practice, autonomous driv-

![Figure 3. Masked voxel reconstruction results of Voxel-MAE with masking ratio 90% and 70% on KITTI dataset.](image)

![Figure 4. (a) Data efficiency of Voxel-MAE. (b) Comparison of different masking strategies.](image)

![Table 10. Impacts of the amount of pre-training data.](image)

| Data size | 120k | 240k | 360k | 480k | 600k |
|-----------|------|------|------|------|------|
| mAP       | 52.51| 52.71| 53.23| 53.57| 53.61|

| Masking ratio | 0-30m | 30-50m | 50m-inf | overall 0-30m | 30-50m | 50m-inf | mAP |
|---------------|-------|--------|---------|---------------|--------|---------|-----|
| 0.3           | 23.73 | 26.00  | 21.49   | 17.25         |        |         |     |
| 0.5           | 24.07 | 26.42  | 22.33   | 17.57         |        |         |     |
| 0.7           | 26.29 | 29.10  | 23.38   | 17.80         |        |         |     |
| 0.7           | 26.57 | 29.31  | 24.93   | 16.01         |        |         |     |
| 0.9           | 29.15 | 33.06  | 26.00   | 18.27         |        |         |     |
| 0.9           | 27.94 | 31.25  | 24.49   | 18.06         |        |         |     |
| 0.9           | 27.49 | 30.54  | 25.28   | 16.11         |        |         |     |
| 0.95          | 23.68 | 26.10  | 21.65   | 15.98         |        |         |     |
| 0.98          | 21.61 | 23.88  | 19.16   | 15.74         |        |         |     |

| Masking ratio | 0-30m | 30-50m | 50m-inf | overall 0-30m | 30-50m | 50m-inf | mAP |
|---------------|-------|--------|---------|---------------|--------|---------|-----|
| 0.7           | 26.57 | 29.31  | 24.93   | 16.01         |        |         |     |
| 0.9           | 27.49 | 30.54  | 25.28   | 16.11         |        |         |     |
| 0.95          | 23.68 | 26.10  | 21.65   | 15.98         |        |         |     |
| 0.98          | 21.61 | 23.88  | 19.16   | 15.74         |        |         |     |

![Table 11. Impacts of masking ratio.](image)
ing companies accumulate a large amount of unlabeled data. Voxel-MAE can utilize these easily available data to improve the performance of 3D perception.

4.4.3 Masking Ratio

In this section, we investigate the influence of the voxel masking ratio on the cyclist class of ONCE small raw set. As shown in Table 11, when increasing the masking ratio to 90%, the overall performance boosts from 23.27% to 29.15%. For the points within 30-50 meters and > 50 meters, the masking ratio of 70% and 50% are suitable for downstream tasks, respectively. The proper masking ratio enforces networks to capture the gestalt of the 3D scene to learn representative features.

5. Conclusion

In this paper, we propose Voxel-MAE, the first masked autoencoding framework for self-supervised learning on large-scale point clouds, to utilize the vast unlabeled 3D data in autonomous driving. The core insight of Voxel-MAE is to classify whether the reconstructed voxels contain point clouds, thus enforcing the network to learn representative features to deduce the masked voxels. Moreover, we propose the range-aware random masking strategy to improve the pre-training performance of far objects. Experimental results demonstrate its effectiveness across various downstream tasks, including 3D object detection, semantic segmentation, and unsupervised domain adaptation.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Gi-ancarlo 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. 1, 2

[2] Fabio M Carlucci, Antonio D’Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2229–2238, 2019. 3

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Proceedings of the European conference on computer vision (ECCV), pages 132–149, 2018. 3, 5, 6

[4] 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. 3, 5, 6

[5] 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. 1

[6] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In Proceedings of the IEEE/CVF international conference on computer vision, pages 9775–9784, 2019. 3

[7] 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. 1, 3, 4, 5

[8] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE international conference on computer vision, pages 1422–1430, 2015. 3

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In 2012 IEEE conference on computer vision and pattern recognition, pages 3354–3361. IEEE, 2012. 1, 2, 5

[10] Spyros Gidaris, Praveen Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728, 2018. 3

[11] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold fold sparse convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9224–9232, 2018. 4

[12] Jean-Bastien Grill, Florian Strub, Florent Altch´e, 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. 3, 5

[13] 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. 1, 3, 4, 5

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9729–9738, 2020. 3

[15] 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. 3

[16] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrim-ination for self-supervised learning on point clouds. arXiv preprint arXiv:2203.11183, 2022. 3

[17] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Lian, Yamin Li, Wei Zhang, Zhenguo Li, et al. One million scenes for autonomous driving: Once dataset. arXiv preprint arXiv:2106.11037, 2021. 1, 2, 5, 6

[18] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic
segmentation. In 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS), pages 4213–4220. IEEE, 2019.

[19] 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.

[20] 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.

[21] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Py-rcnn: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10529–10538, 2020.

[22] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Py-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. arXiv preprint arXiv:2102.00463, 2021.

[23] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointr-cnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 770–779, 2019.

[24] 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.

[25] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In European conference on computer vision, pages 685–702. Springer, 2020.

[26] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020.

[27] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videda: Masked autoencoders are data-efficient learners for self-supervised video pre-training. arXiv preprint arXiv:2203.12602, 2022.

[28] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11713–11723, 2020.

[29] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1887–1893. IEEE, 2018.

[30] Liang Xiao, Jiaolong Xu, Dawei Zhao, Zhiyu Wang, Li Wang, Yiming Nie, and Bin Dai. Self-supervised domain adaptation with consistency training. In 2020 25th International Conference on Pattern Recognition (ICPR), pages 6874–6880. IEEE, 2021.

[31] 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.

[32] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhiliang 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.

[33] Jiaolong Xu, Liang Xiao, and Antonio M López. Self-supervised domain adaptation for computer vision tasks. IEEE Access, 7:156694–156706, 2019.

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

[35] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10368–10378, 2021.

[36] 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.

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

[38] 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.

[39] 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.

[40] 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.

[41] 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.