mathematics

MDPI

Article

Masked Autoencoder for Pre-Training on 3D Point Cloud Object Detection

Guangda Xie [1], Yang Li [2,*], Hongquan Qu [2] and Zaiming Sun [3]

[1] College of Electrical and Control Engineering, North China University of Technology, Beijing 100144, China
[2] College of Information, North China University of Technology, Beijing 100144, China
[3] School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China
* Correspondence: liyang_ncut@ncut.edu.cn

Abstract: In autonomous driving, the 3D LiDAR (Light Detection and Ranging) point cloud data of the target are missing due to long distance and occlusion. It makes object detection more difficult. This paper proposes Point Cloud Masked Autoencoder (PCMAE), which can provide pre-training for most voxel-based point cloud object detection algorithms. PCMAE improves the feature representation ability of the 3D backbone for long-distance and occluded objects through self-supervised learning. First, a point cloud masking strategy for autonomous driving scenes named PC-Mask is proposed. It is used to simulate the problem of missing point cloud data information due to occlusion and distance in autonomous driving scenarios. Then, a symmetrical encoder–decoder architecture is designed for pre-training. The encoder is used to extract the high-level features of the point cloud after PC-Mask, and the decoder is used to reconstruct the complete point cloud. Finally, the pre-training method proposed in this paper is applied to SECOND (Sparsely Embedded Convolutional Detection) and Part-A2-Net (Part-aware and Aggregate Neural Network) object detection algorithms. The experimental results show that our method can speed up the model convergence speed and improve the detection accuracy, especially the detection effect of long-distance and occluded objects.

Keywords: 3D object detection; self-supervised; pre-training; autoencoder

MSC: 68T07; 68T45

Citation: Xie, G.; Li, Y.; Qu, H.; Sun, Z. Masked Autoencoder for Pre-Training on 3D Point Cloud Object Detection. *Mathematics* **2022**, *10*, 3549. https://doi.org/10.3390/math10193549

Academic Editor: Daniel-Ioan Curiac

Received: 22 August 2022
Accepted: 23 September 2022
Published: 28 September 2022

1. Introduction

In the field of autonomous driving, 3D point cloud object detection has attracted much attention. Effective detection of traffic objects in complex environments will significantly improve the safety and efficiency of autonomous vehicles. The main sensors for autonomous driving are cameras and LiDAR. The image captured by the camera is 2D image data. Although rich environmental texture information is included, the depth of information of the image is lost. The LiDAR sensor acquires 3D point cloud data. The point cloud has rich depth value and shape information, which can more accurately perceive the relative distance and significantly improve the understanding ability of 3D autonomous driving scenes. In the field of 3D object detection, state-of-the-art performance has been achieved based on LiDAR-based models [1].

Although LiDAR sensors are widely used in 3D autonomous driving scenarios, in fact, LiDAR frames are technically 2.5D [2]. The LiDAR sensor fires off beams of laser light that return after hitting the first target, causing the information behind the target to be lost. For example, the LiDAR scans the autonomous driving scenarios (Figure 1a to obtain a set of point cloud data (Figure 1b). A tree (red circle in Figure 1) in the scene blocks the laser beam, resulting in the loss of all information behind the tree, which inevitably affects object perception. Through the research of KITTI dataset [3], it is found that there are three main reasons for the missing object shape. Take the target 1, 2 and 3 marked in Figure 1 as an

example. First, for the close object (Car 1, Figure 1c marks the distance of the object, and Figure 1d shows the three views of target 1 in the first row), its parts on the far side are occluded by the parts on the near side. This situation is commonly called self-occlusion. Second, for the occluded object (Car 2), its left side is partially occluded by target 1, resulting in a missing shape. This situation is commonly called external-occlusion. Third, for the long-range object (Car 3), which is not occluded by any object but is far away from the sensor, the number of points is relatively small compared with the number of close-range objects, and it is difficult to obtain the complete object shape. This situation is commonly called signal miss. In autonomous driving scenarios, self-occlusion is unavoidable because on-board LiDAR sensors cannot achieve 360° scanning. Therefore, this paper aims to solve the difficult problem of 3D object detection in the case of external-occlusion and signal miss, wherein signal miss only represents in long-distance situations.

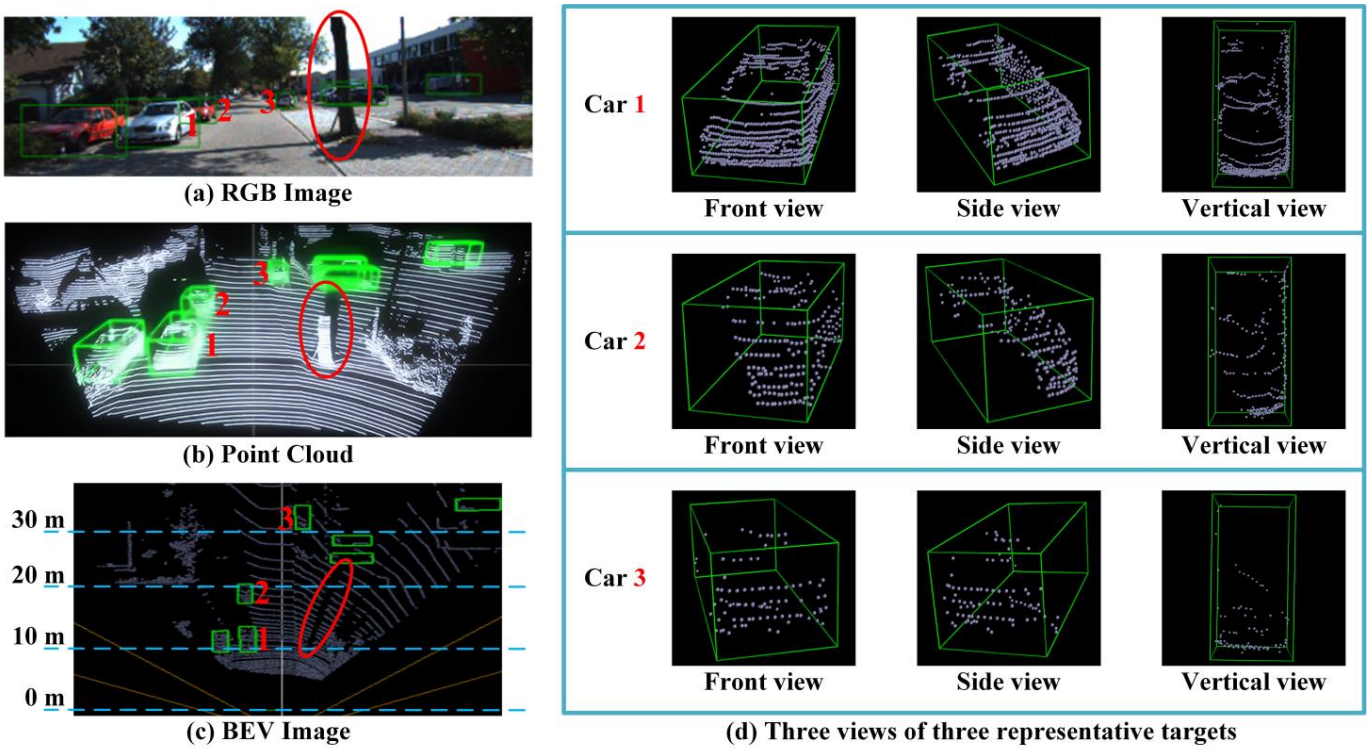

Figure 1. (**a**) is the RGB image of the autonomous driving scene, (**b**) is the point cloud obtained from the LiDAR scanning scene, (**c**) is the BEV (Bird's Eye View) image of the point cloud, marking the distance between the object and the sensor. Mark Car 1, Car 2 and Car 3 represent the close target, occluded target, and long-range target, respectively. (**d**) shows the three views of these three targets.

In 2D vision tasks, large-scale datasets (such as ImageNet [4]) are commonly used to pre-train models. Then, the pre-trained parameters are used as the initial values of the model, and the model is trained by transfer learning on small-scale datasets of specific downstream tasks, such as object detection and instance segmentation. Compared with random initialization, transfer learning is helpful to improve the performance of the model [5–7]. For point cloud object detection in autonomous driving scenarios, the pre-training method is considered to solve the false and missed detections in the case of external occlusion and signal miss. However, in the field of 3D object detection, especially in autonomous driving scenarios, supervised learning is difficult because there is no large-scale dataset containing category annotation like ImageNet.

Therefore, a 3D point cloud pre-training dataset for autonomous driving scenarios is collected and produced. A masked method for autonomous driving point cloud data, PC-Mask, is proposed to simulate signal miss caused by external occlusion and distance. A Point Cloud Mask Autoencoder (PCMAE) is designed, which consists of an encoder and

a decoder. The encoder extracts the point cloud features after PC-Mask, and the decoder is responsible for reconstructing the point cloud. Grid operation is introduced to solve the problem of irregular point cloud. Through this self-supervised learning method, the backbone of the target detection model is pre-trained to enhance the feature extraction ability of the point cloud target and improve the detection accuracy. On KITTI dataset, our pre-trained model improves the accuracy of the detection algorithm SECOND [8] and Part-A2-Net [9].

2. Related Work

2.1. 3D Point Cloud Object Detection

At present, 3D point cloud object detection methods are mainly divided into two categories: point-based and voxel-based. The point-based method takes the original point cloud as the input of the detector, extracts the point cloud feature set through iterative sampling and grouping, and then performs object detection [10]. The representative point-base methods [11–13] mainly use the nearest neighbor search operation after the Farthest Point Sampling (FPS), which greatly limits its efficiency. Voxel-based methods divide the point cloud into a uniform grid for voxelization. Then the 3D convolutional neural network is used for feature extraction to achieve object detection [14]. The advanced voxel-based method [8,15,16] replaces CNN with sparse 3D convolution, which solves the problem of memory consumption and large computational load caused by empty voxels, but voxelization still brings inevitable information loss. This paper introduces Gridding [17] instead of voxelization to normalize the original point cloud. The above methods usually randomly initialize the model parameters at the beginning of training. In the field of object detection, initializing model parameters after pre-training can greatly improve the model's ability to understand data [18]. Therefore, a PCMAE pre-training method is proposed to improve the detection ability of 3D point cloud objects through transfer learning.

2.2. Application of Transfer Learning in 3D Point Cloud

Transfer learning is widely used in the field of deep learning. The backbone is pre-trained with data-rich upstream tasks, and then transfer learning is used to initialize the downstream model weights. Compared with the features fine-tuned on the detection dataset or trained from scratch, Vasconcelos et al. [18] verified that the features learned in the upstream classification task are more suitable for object detection. However, there is little available research on the application of transfer learning in 3D object detection. The reason is that there are differences in point cloud data captured by different sensors or different natural scenes. Besides, there is a lack of large-scale datasets containing category annotations for autonomous driving scenarios, which cannot meet the pre-training method of supervised learning. Compared with supervised learning, He et al. [19] proved that reasonable self-supervised learning pre-training on small-scale data can achieve better effects.

Therefore, some work adopts self-supervised learning methods for pre-training. For example, Proxy tasks are constructed by deformation [20], rotation [21] and partial rearrangement [22]. Different from these methods, we propose a Proxy task for autonomous driving scenarios, PC-Mask, which performs self-supervised pre-training on the backbone of 3D object detection in a masked manner.

2.3. Masked Self-Supervised Pre-Training

Self-supervised learning designs Proxy Tasks to mine the representation features of data and use them as supervision information to improve the feature extraction ability of the model. Compared with the supervised learning method, self-supervised learning can achieve good results without label data. The self-supervised pre-training method based on masked methods is widely used in various fields due to its simple concept and easy expansion. There are Bert [23], Bart [24], GPT [25–27] and so on in NLP. In the field of 2D image processing, there are DAE [28], Igpt [29], MAE [19], etc. In the field of 3D point cloud processing, Point-bert [30] uses dAVE mapping for mask point modeling.

The disadvantage is that it relies on complex data augmentation and costly two-stage pre-training. Zhang et al. [31] proposed a one-stage pure masked auto-encoding method using transfer learning for 3D object detection. However, due to differences in application methods and scenarios, this method is not suitable for autonomous driving scenarios. Of course, mask pre-training also has some shortcomings. For example, masked data usually exist only in pre-training, not in downstream tasks. Source domain bias is generated, which leads to data mismatch between pre-training stage and fine-tuning stage, and the poor effect of transfer learning [32]. Therefore, the proposed PC-Mask strategy is used for pre-training on the source domain of the dataset, which alleviates the source domain bias problem caused by data mismatch in the traditional mask strategy.

3. Approach

This paper proposes PCMAE, which can provide pre-training for the Backbone of the 3D Object Detection algorithm by means of transfer learning, as shown in Figure 2. In this section, the overall architecture of the PCMAE model and the details of each module are described.

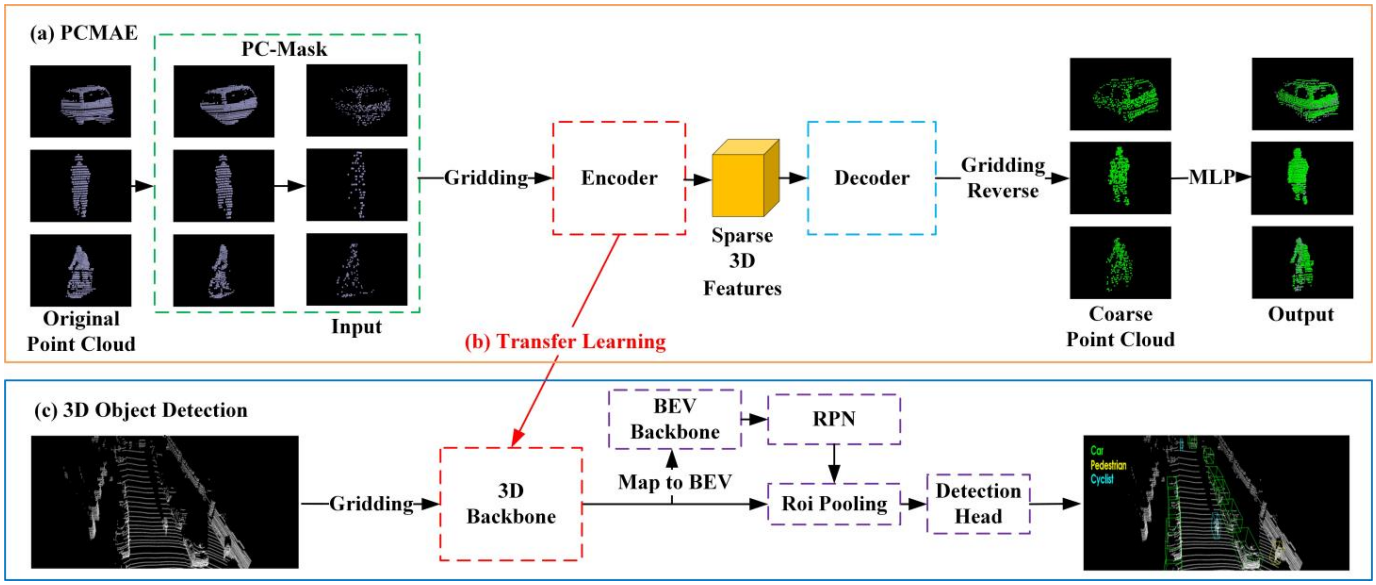

Figure 2. (**a**) PCMAE structure. (**b**) Shared pre-trained model weights through transfer learning. (**c**) 3D object detection model.

3.1. Framework

The pre-training of PCMAE adopts the method of Auto-encoding. The network structure mainly includes three parts: PC-Mask module, encoder and decoder modules, as shown in Figure 2a. Firstly, the PC-Mask module masks the original point cloud to simulate the missing point cloud caused by the two cases of external occlusion and signal miss. Then, the obtained subset of point clouds is used as the input of the model and sent to the Gridding layer for normalization. The 3D backbone network, which needs to be pre-trained, is used as the encoder to learn the 3D features of a subset of point clouds. The encoder consists of 3D Sparse Convolution [8], and the decoder consists of 3D Sparse Inverse Convolution. The decoder and encoder are symmetrical. The decoder is used to reconstruct the sparse 3D features output by the encoder. After that, the coarse point cloud is obtained through the Gridding Reverse layer. Finally, a three-layer MLP is used to restore the details to complete the point cloud reconstruction.

3.2. Point Cloud Masking

In autonomous driving scenarios, the original point cloud obtained through LiDAR usually has two types of problems: external-occlusion and signal miss. PC-Mask is pro-

posed to simulate the above two missing point clouds, as shown in Figure 3. For external-occlusion, three points are randomly selected within the original point cloud to form a plane, which is cut into two parts, as shown in Figure 3b. Then the parts with fewer points are deleted and the ones with more points are kept. This operation can not only simulate the real situation more realistically, but also retain most of the structure of the target, which is beneficial to the reconstruction of the target. For signal miss caused by long distance, random sampling operation of high masking ratio is performed on the point cloud after simulated external-occlusion, as shown in Figure 3c. The high masking ratio follows a uniform distribution, which can eliminate redundancy to some extent [19], and the shape information of the point cloud is preserved to the greatest extent. It makes pre-training more efficient.

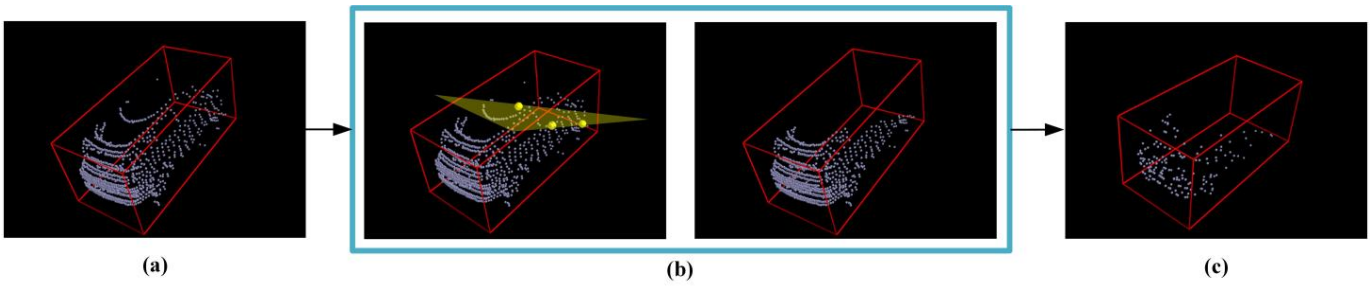

Figure 3. The procedure of point cloud mask. (**a**) is the original point cloud of a car, (**b**) is an example of simulated "External-occlusion", (**c**) is an example of simulated "Signal miss".

3.3. Gridding and Gridding Reverse

Since point clouds are unordered and irregular, it is necessary to normalize them. Converting point clouds into a 3D voxel and then performing a convolution operation is a commonly used method. However, two problems arise: first, the voxelization process introduces a quantization effect, which will lead to the irreversible loss of geometric information; the second is that voxelization is not differentiable and is not suitable for point cloud reconstruction.

To solve the above problems, Gridding [17] is introduced to normalize the original point cloud. The input of the Gridding layer is the subset of point clouds after PC-Mask, $P = \{p_i\}_{i=1}^{n}$, where $p_i \in \mathbb{R}^3$, n is the number of point clouds in p, and the output is a regular 3D grid, each grid cell is a cube with 8 vertices. The difference between Gridding and voxelization lies in the calculation method of the value w_i at each vertex v_i. For voxelization, as shown in Figure 4a, the calculation method is:

$$w_i = \begin{cases} 0 & \forall p \notin \mathcal{N}(v_i) \\ 1 & \exists p \in \mathcal{N}(v_i) \end{cases} \tag{1}$$

where $\mathcal{N}(v_i)$ represents the neighboring points of vertex v_i. If a vertex has no neighboring points, the value of w_i is 0. If there are one or more neighboring points, the value of w_i is 1. Therefore, voxelization is non-differentiable. For gridding, as shown in Figure 4b, the value w_i at each vertex v_i is calculated as:

$$w_i = \begin{cases} \sum\limits_{p \in \mathcal{N}(v_i)} \frac{(1-|x_i^v - x|)(1-|y_i^v - y|)(1-|z_i^v - z|)}{|\mathcal{N}(v_i)|} & if\ |\mathcal{N}(v_i)| > 0, \\ 0 & if\ |\mathcal{N}(v_i)| = 0, \end{cases} \tag{2}$$

where x_i^v, y_i^v, z_i^v are the coordinates of the vertex, and x, y, z are the coordinates of its neighboring original point cloud. $|\mathcal{N}(v_i)|$ is the number of neighboring points of v_i, and the value of each vertex is related to the number and distance of its neighboring points. Compared with voxelization, gridding can better preserve some details of objects, and gridding is differentiable.

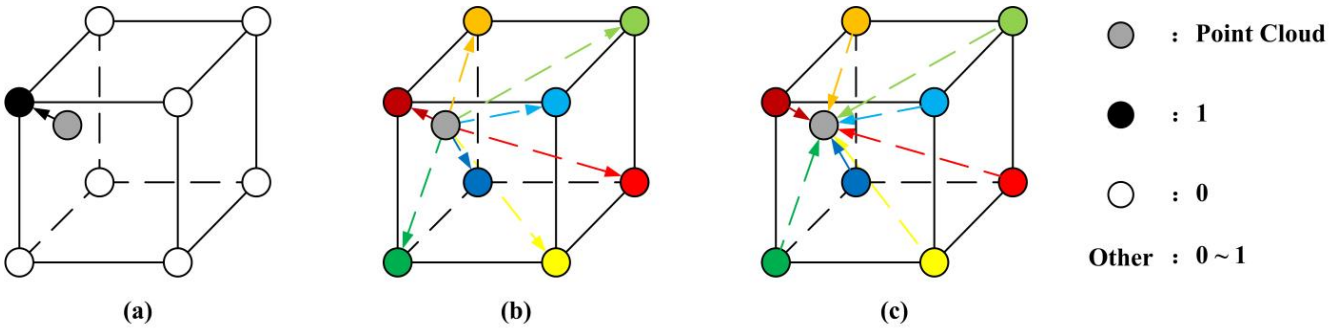

Figure 4. Point cloud normalization. (**a**) is voxelization, (**b**) is gridding, (**c**) is gridding reverse.

Gridding Reverse is to convert the regular feature grid output by the network into an irregular point cloud $p^c = \{p_i^c\}_{i=1}^m$. Contrary to the operation of Gridding, as shown in Figure 4c, the 8 vertices $\Theta^i = \left\{\theta_j^i\right\}_{j=1}^8$ of each grid cell are transformed into a point p_i^c inside this grid cell. Its coordinates are calculated as follows:

$$p_i^c = \begin{cases} \frac{\sum_{\theta \in \Theta^i} w'_\theta v_\theta}{\sum_{\theta \in \Theta^i} w'_\theta} & if \sum_{\theta \in \Theta^i} w'_\theta \neq 0, \\ \text{ignore} & if \sum_{\theta \in \Theta^i} w'_\theta = 0, \end{cases} \tag{3}$$

where v_θ represents the coordinates $\{v_\theta \mid \theta \in \Theta^i\}$ of the 8 vertices of the cell, and w'_θ represents the corresponding values $\{w'_\theta \mid \theta \in \Theta^i\}$ of each vertex. When the weighted sum of the corresponding values of the 8 vertices of a cell is 0, no point cloud is generated in the cell.

3.4. Loss-Function in Pre-Training

The unordered and irregular point clouds make it difficult to calculate the loss directly. The original and reconstructed point clouds are gridding processed to obtain 3D grids $\mathcal{G}_{gt} =< V^{gt}, W^{gt} >$ and $\mathcal{G}_{\text{pred}} =< V^{\text{pred}}, W^{\text{pred}} >$, respectively. To reduce the influence of outliers in the original point cloud on training, the L1 distance is calculated as the pre-training loss:

$$\mathcal{L}\left(W^{\text{pred}}, W^{gt}\right) = \frac{1}{N_G^3} \sum \| W^{\text{pred}} - W^{gt} \| \tag{4}$$

where $W^{\text{pred}} \in \mathbb{R}^{N_G^3}$, $W^{gt} \in \mathbb{R}^{N_G^3}$, and N_G is the resolution of the two 3D grids.

4. Pre-Training Experiments

4.1. Pre-Training Datasets

The pre-training data are obtained from the KITTI dataset. Including single, close and no external-occlusion ground-truth. There are four targets: car, cyclist, pedestrian and background. For car and background, the extraction ranges are z: [−3 m, 1 m], y: [−20 m, 20 m], x: [5 m, 20 m]. For cyclist and pedestrian, the extraction ranges are z: [−3 m, 1 m], y: [−10 m, 10 m], x: [2 m, 10 m]. Among them, z represents the vertical direction, y represents the left and right direction, and x represents the front and rear direction coordinates. Figure 5 shows the visualization of the target. In particular, the KITTI dataset is obtained by on-board LiDAR sensors. For the same object, its parts on the far side are occluded by the parts on the near side, which is called self-occlusion. This is an inevitable situation in autonomous driving scenarios, so we treat the target with self-occlusion as a complete target. As a result, 18,866 instances were extracted. Next, individual targets are processed using PC-mask. In order to simulate the actual possible external-occlusion situation, the occluded mask operation is performed on each target once and twice, and two sets of training data are obtained to simulate the occlusion situation. Then random sampling of high masking ratio is performed. To simulate the signal miss

situation at different distances, a random mask ratio of 25%, 45%, 65%, and 85% is set. Based on the above operations, the total number of point cloud samples in the pre-training dataset is 18,866 × 2 × 4 = 150,928. The subset of point clouds after PC-Mask is used as the input of PCMAE, and the original point cloud is used as the label. The data are divided into training, validation and test sets according to 7:1:2.

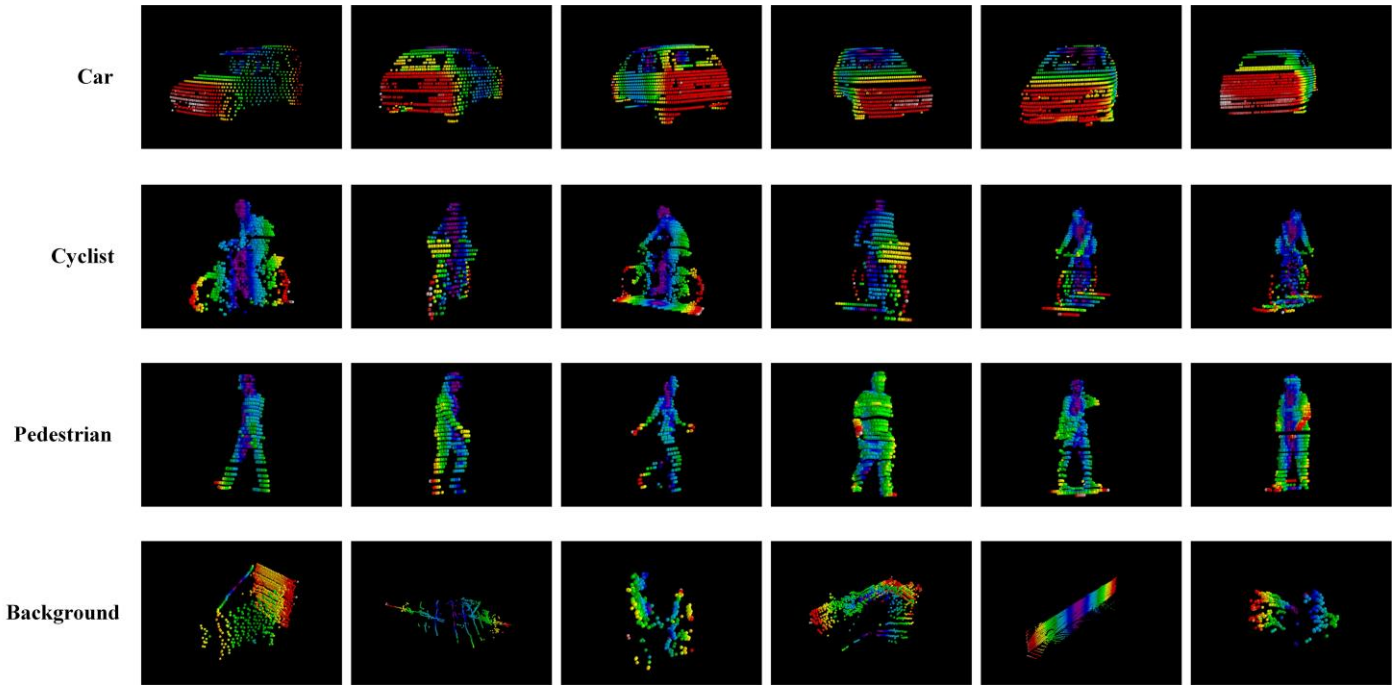

Figure 5. Visualization of different classes of targets in the KITTI dataset.

4.2. Pre-Training Implementation Details

Two commonly used 3D backbone networks, SpMiddleFHD [8] and SparseConvUNet [9] are used as encoders for PCMAE. The pre-training process includes a training phase and a validation phase, as shown in Figure 6.

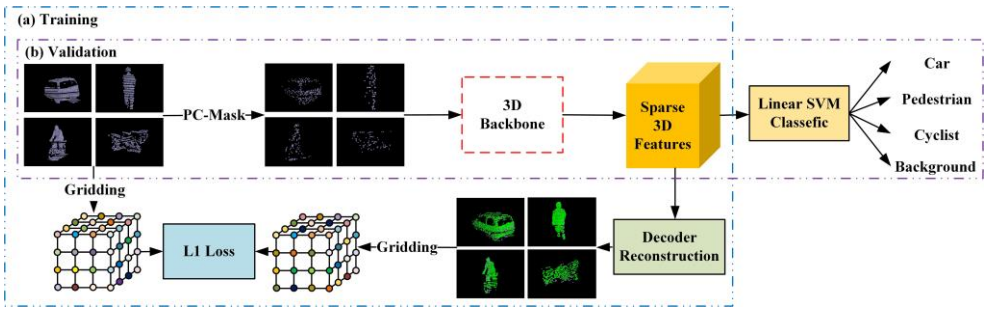

Figure 6. Pre-training flow. (**a**) Training phase. (**b**) Validation phase.

In the training phase, PCMAE is based on PyTorch [33], and both 3D backbones are trained with the same configuration and parameters. The models are optimized with an AdamW [34] optimizer with $\beta 1 = 0.9$ and $\beta 2 = 0.999$. The network trained with a batch size of 16 on two NVIDIA Quadro RTX 6000 GPUs. Initial learning rate as 5×10^{-4} and weight decay as 1×10^{-3}. The optimization is set to stop after 160 epochs.

In the validation phase, a linear SVM (Support Vector Machine) is trained to verify the effect of pre-training. Since the effect of pre-training depends on the quality of the sparse 3D features extracted by the encoder, the pre-trained encoder is frozen. The linear SVM classifier is trained by extracting sparse 3D features of the point cloud subset after PC-Mask.

The classification accuracy of SpMiddleFHD is 83.4%, and the classification accuracy of SparseConvUNet is 87.1%. Using a confusion matrix to show the classification results, as shown in Figure 7, The horizontal axis shows the Predicted label, and the vertical axis shows the True label. Each cell represents the probability that a 'True Label' is predicted to be a 'Predicted Label'. For the model SpMiddleFHD (Figure 7a), take the first row as an example. For all samples with the true label "car", 87.1% were predicted correctly, 0.9% were predicted as "Cyclist", 0.2% were predicted as "Pedestrian", and 11.8% were predicted as "Background". The probability of "car" being mispredicted as "Background" is high because the background includes many objects similar to "car", such as trucks, buses, etc. In addition, "Cyclist" and "Pedestrian" also have high similarities, resulting in a high probability of being wrongly detected by each other, as shown in the second and third rows. This is hard to avoid, especially after simulating external occlusion and loss of signal. The classification results of the two models are credible, of which the SparseConvUNet model is relatively good. After pre-training, the different target features extracted by the 3D backbone can be distinguished by the linear SVM classifier, which proves that the PCMAE pre-training effect is better.

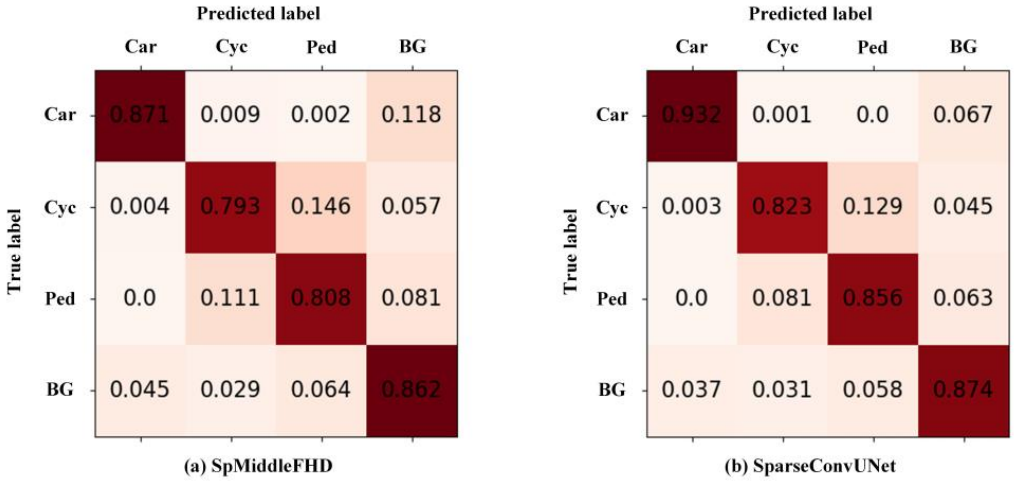

Figure 7. Confusion matrix. (**a**) shows the classification results of SpMiddleFHD. (**b**) shows the classification results of SparseConvUNet. The horizontal axis shows Predicted label, and the vertical axis shows True label. Each cell represents the prediction probability of the output. For darker colors, the prediction probability is larger. The main diagonal cell represents the probability of correct classification.

4.3. Visualization of Point Clouds in PCMAE

The reconstruction results on the test set are visualized as shown in Figure 8. For the four types of targets of car, cyclist, pedestrian and background, after simulating the incomplete point cloud in the case of external-occlusion and signal miss through PC-Mask, the PCMAE model can complete the reconstruction well.

The reconstruction results of the real scene are visualized, as shown in Figure 9. The object of the real scene is extracted from the KITTI dataset and reconstructed with PCMAE. The model can better complement the real object's shape.

PCMAE can not only reconstruct the point cloud in the case of simulating external-occlusion and signal miss, but it can also better supplement the object shape in the real scene. It is proved that PC-Mask can better simulate the point cloud of the real scene and the pre-training model has certain robustness.

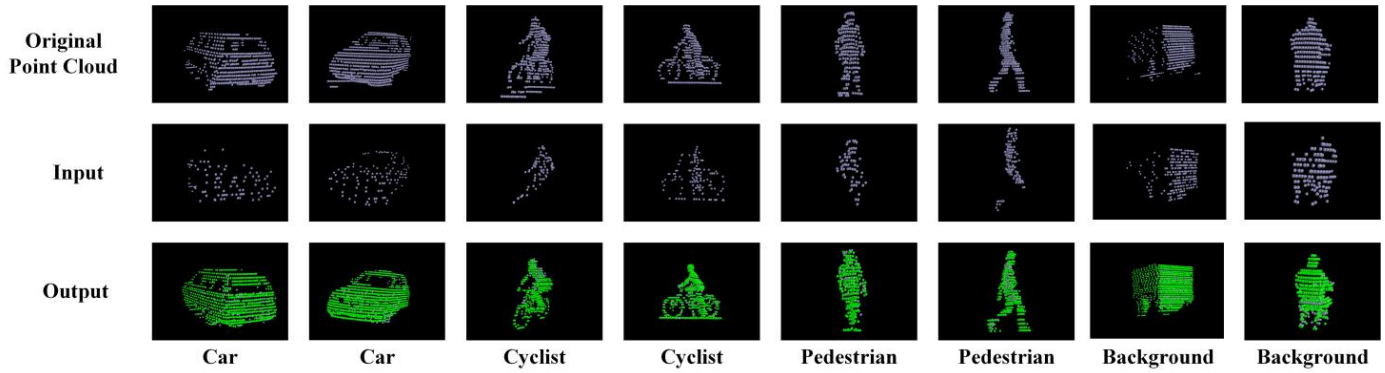

Figure 8. Test set reconstruction visualization. The first row is the original point cloud. The second row is the subset of point clouds after PC-Mask, which is used as the input of PCMAE. The third row is the reconstruction result of PCMAE.

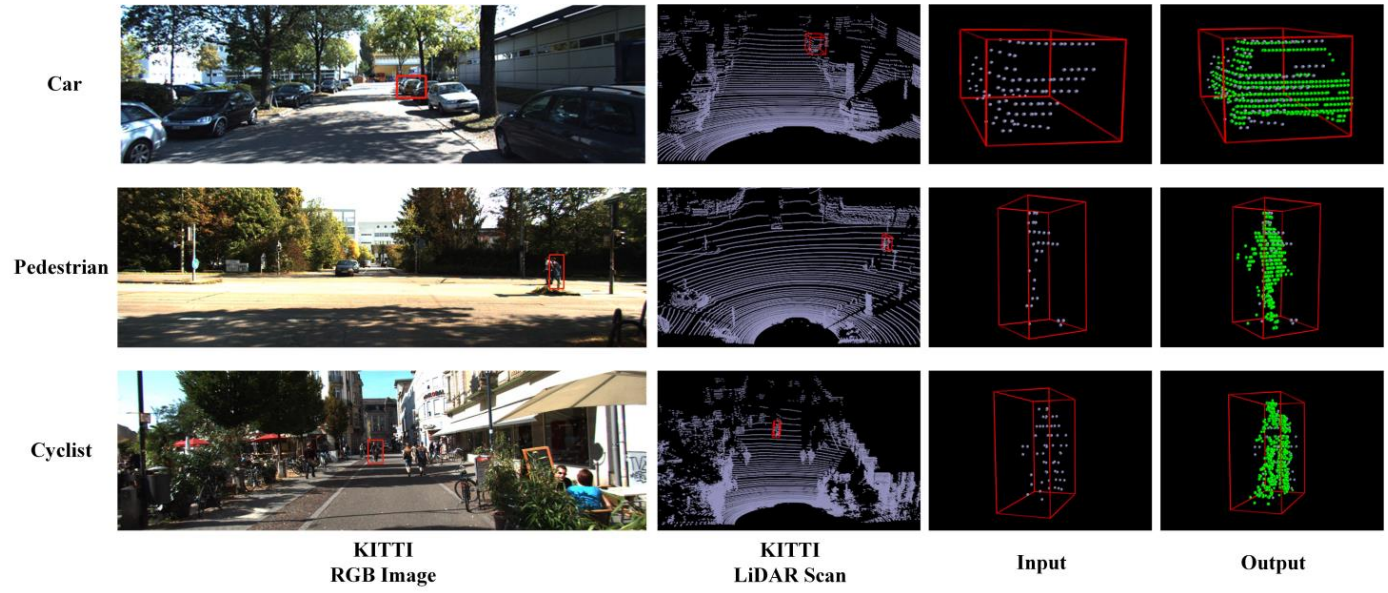

Figure 9. Visualization of the reconstructed target. The first column is the RGB image of the real scene. The second column is the LiDAR scan. The third column is the incomplete point cloud extracted from the real scene. The fourth column is the reconstruction result.

5. 3D Object Detection Experiments

This section introduces 3D object detection experiments. PCMAE pre-trains the backbone of the 3D object detection model, and then initializes the 3D backbone with pre-trained weights through transfer learning. Then compare it with the way of random initialization. Experimental design is presented in Section 5.1, experimental details are presented in Section 5.2, quantitative evaluation and qualitative results are presented in Sections 5.3 and 5.4, respectively.

5.1. Experimental Design

To demonstrate the wide applicability of the PCMAE pre-training method to 3D object detection algorithms, representative SECOND [8] and Part-A2-Net [9] algorithms are selected for experiments. The SECOND algorithm designed a 3D backbone called SpMiddleFHD, and first proposed to replace the traditional 3D convolution with sparse 3D convolution. This backbone reduces the massive memory consumption and computational load of empty voxels, an approach that continues to this day. The Part-A2-Net algorithm designs a 3D backbone similar to UNet, namely SparseConvUNet, including part awareness and part aggregation. It makes full use of the internal location information in the label

and is a very popular 3D backbone network at present. This paper pre-trains the above two 3D backbones in Section 4 and verifies that pre-training can perform effective feature extraction for self-occlusion and long-distance targets. Therefore, in this section, two sets of comparative experiments are conducted using PCMAE pre-training (pre-training parameters to initialize model weights) and no pre-training (randomly initializing model weights). The first group is the comparison between SECOND and PCMAE + SECOND. The second group is the comparison of Part-A2-Net and PCMAE + Part-A2-Net. Use the same hyperparameters to verify whether PCMAE can improve the detection performance of SECOND and Part-A2-Net algorithms.

5.2. Implementation Details

The experiment is carried out on KITTI dataset, which has 7481 training samples. These training samples are divided into the train set (3712 samples) and validation set (3769 samples) as the frequently used partition of KITTI dataset. In order to verify the universality of PCMAE, experiments are performed based on SECOND and Part-A2-Net algorithms, respectively. In order to verify the effectiveness of PCMAE pre-training, a comparison experiment is performed with the way of randomly initializing weights, and the same hyperparameters are used.

Experiment 1 trains SECOND by using SpMiddleFHD 3D backbone, the trained model weights are initialized with two strategies: random initialization and initialization from ImageNet checkpoints. There is a total of 3 point clouds per minibatch using the Adam optimizer running on an NVIDIA Quadro RTX 6000 GPU. All models were trained for 42,000 iterations. The initial learning rate was 0.0002, with an exponential decay factor of 0.8 and a decay every 3712 iterations. A decay weight of 0.0001, a β1 value of 0.9 and a β2 value of 0.999 were used. The experimental process is shown in Figure 10a. After 42,000 iterations, the random initialization method (gray) converges around 35,000 times (The way to judge the convergence is that the change in loss is small, and the accuracy of the validation set no longer increases). The mAP (mean Average Precision) of the validation set is 65.89. The weights are initialized by pre-training (purple) to converge at around 15,000 times, and the mAP of the validation set is 67.85.

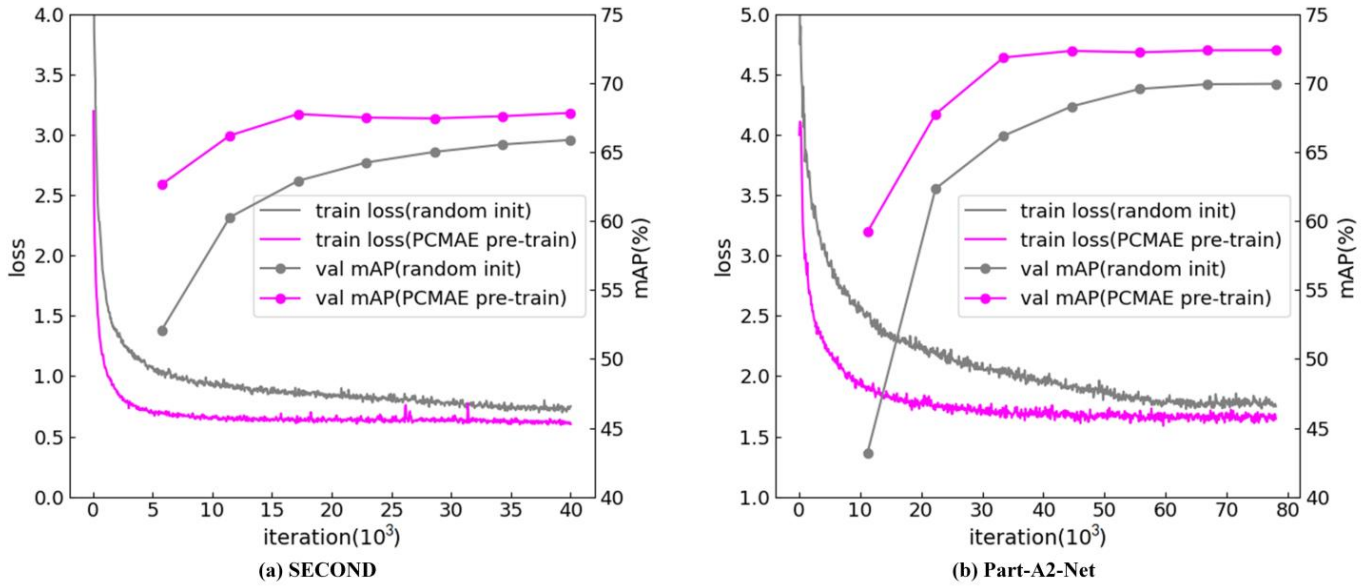

Figure 10. (**a**) SECOND training process. (**b**) Part-A2-Net training process. The Grey curve represents parameters that are randomly initialized. The purple curve represents the parameters initialized by PCMAE pre-training.

Experiment 2 trains Part-A2-Net by using SparseConvUNet backbone, the model weights were initialized for training using two strategies: random initialization and initialization from

ImageNet checkpoints. The entire network was trained end-to-end using the Adam optimizer and a batch size of 6 for 78,000 iterations. The cosine annealing learning rate strategy is used with an initial learning rate of 0.001. The experimental process is shown in Figure 10b. After 78,000 iterations, the random initialization experiment converges around 66,000 times, and the validation set mAP is 70.44. The weights are initialized by pre-training to converge at around 40,000 times, and the mAP of the validation set is 71.92.

The above two experimental results show that the pre-training method based on PCMAE has faster convergence, lower loss and higher accuracy. The importance of the pre-training scheme is demonstrated.

5.3. Quantitative Evaluation

The PCMAE pre-training method proposed in this paper is applied to the SECOND [8] and Part-A2-Net [9] 3D object detectors, respectively, and compared with the original algorithm. The evaluation indicators refer to the AP under 40 recall thresholds (Apr40) of SECOND and Part-A2-Net. The KITTI dataset divides the detection difficulty into three levels: easy, moderate, and hard according to the size of the object, the occlusion, and the degree of truncation. Table 1 shows the quantitative evaluation of the two groups of experiments. Reported the 3D APs of three types of objects at different difficulty levels under Apr40. SECOND and Part-A2 in the table use the training method of randomly initializing weights. After adding PCMAE, the training method of pre-training initialization weight is adopted. The blue part shows the gain of PCMAE to the two algorithms. It can be seen that after PCMAE pre-training, the detection accuracy of the two 3D object detectors has been improved, indicating that the PCMAE pre-training method has wide applicability. It is particularly noted that the performance gain mainly comes from the two detection levels of moderate and hard. Among them, for moderate, the SECOND algorithm improves each category by 0.55%, 3.49%, and 2.71%, respectively. The Part-A2-Net algorithm improves each category by 1.40%, 2.95%, and 0.31%, respectively. For the hard difficulty level, the SECOND algorithm improves each category by 1.67%, 3.11%, and 2.93%, respectively. The Part-A2-Net algorithm improves each category by 1.53%, 2.67%, and 1.57%, respectively. It is shown that the PC-Mask proposed in this paper simulates real scenes and performs self-supervised pre-training, which can improve the detection effect of occluded and long-distance situations.

Table 1. Comparisons on the KITTI validation set, evaluated by the 3D Average Precision (AP) under 40 recall thresholds (R40). The rotated IoU (Intersection-over-Union) threshold is 0.7 for the car and 0.5 for the pedestrian/cyclist. The two comparison experiments are shown separately as test results (The difference between each comparative experiment is whether the PCMAE pre-training parameters are used for initialization, and the model structure and hyperparameters are the same), the blue part shows the gain of PCMAE to the two algorithms.

Method	Car 3D Apr40			Ped 3D Apr40			Cyc 3D Apr40		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
SECOND	90.97	79.94	77.09	58.01	51.88	47.05	78.50	56.74	52.83
PCMAE + SECOND	91.09	80.49	78.76	59.85	55.37	50.16	79.71	59.45	55.76
Improvement	+0.12	+0.55	+1.67	+1.84	+3.49	+3.11	+1.21	+2.71	+2.93
Part-A2	89.42	79.13	78.51	58.21	54.07	49.74	86.76	71.38	66.71
PCMAE+ Part-A2	89.63	80.53	80.04	59.56	57.02	52.41	88.09	71.69	68.28
Improvement	+0.21	+1.40	+1.53	+1.35	+2.95	+2.67	+1.33	+0.31	+1.57

5.4. Qualitative Results

Figure 11 shows the 3D point cloud detection results and the corresponding 2D images on the KITTI test set.

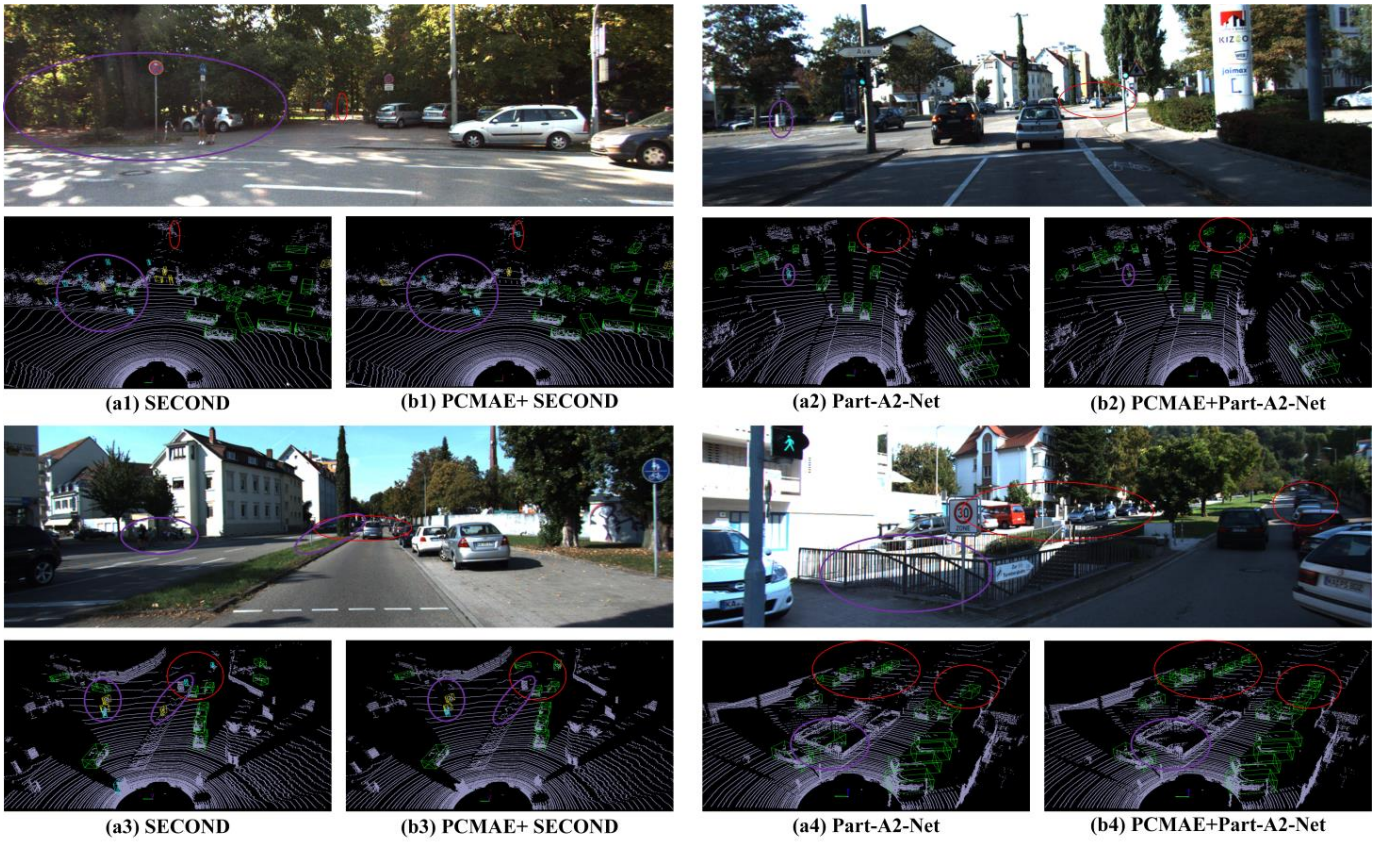

Figure 11. Qualitative results on the KITTI test set. (**a1**–**a4**) Detection results of the original algorithm. (**b1**–**b4**) The optimized algorithm detection results. Green boxes represent cars. Blue boxes represent pedestrians. Yellow boxes represent cyclists. Red ovals indicate missed detections. Purple ovals indicate false detections.

6. Discussion and Conclusions

In the field of deep learning, transfer learning is one of the most successful cases. The resulting model of after pre-training on a large-scale dataset is transferred to subtasks and fine-tuned, and its performance is much higher than the training method with random initialization; yet very little is known about its usefulness in 3D point cloud understanding. This paper aims to promote research on pre-training methods for object detection in 3D point clouds. Pre-training usually requires large-scale labeled datasets for supervised learning. However, the lack of large-scale 3D datasets containing category annotations in autonomous driving scenarios cannot satisfy pre-training methods for supervised learning. Studies have found that in natural language processing and computer vision, simple self-supervised learning methods such as [23,28] pre-train with a smaller amount of unlabeled datasets and even outperform supervised learning. Among them, the mask auto-encoder method is simple and scales well. On the other hand, 3D point clouds are signals of different nature relative to language and images, and usually only contain certain shape information. Simply removing objects or randomly removing patches is likely to be detrimental to the shape representation of the point cloud. Therefore, it is necessary to propose a masked strategy suitable for autonomous driving scenarios to simulate point cloud data with missing shapes in real scenes. Moreover, this masking strategy is combined with an auto-encoder as a self-supervised pre-training method for 3D object detection in the field of autonomous driving.

This paper explores providing pre-training for 3D point cloud object detection algorithms through self-supervised learning. First, is the analysis of LiDAR point cloud dataset in autonomous driving scenarios. It is found that there are three reasons for the missing object shape: namely self-occlusion, external-occlusion and signal miss caused by distance.

Therefore PC-Mask, a masking strategy for LiDAR point cloud data in autonomous driving scenes, is proposed for self-supervised pre-training of point clouds. Then combined with the idea of auto-encoder, PCMAE, a symmetrical encoder–decoder network structure, is designed. Take the 3D backbone that requires to be pre-trained as the encoder to extract the features of the point cloud after PC-Mask. The decoder is responsible for reconstructing the original point cloud. The linear SVM experiment verifies that the pre-trained 3D backbone can effectively extract the features of self-occlusion objects and long-distance objects. Finally, the proposed PCMAE pre-training method is applied to two representative 3D backbone point cloud object detectors SECOND and Part-A2-Net. The effectiveness and wide applicability of the proposed pre-training method are verified from two aspects of quantitative evaluation and qualitative results.

PCMAE is simple and extensible. This paper improves the feature representation ability of the 3D backbone for long-distance and occluded targets through self-supervised pre-training, and uses transfer learning to improve the detection performance of target detectors for complex scenes. PCMAE can reconstruct the point cloud data after PC-Mask. It is shown that it helps the 3D backbone to learn the characteristics of point cloud data in autonomous driving scenes. Therefore, PCMAE can be extended to other point cloud intelligent perception tasks, such as point cloud segmentation, which will serve as future work.

Author Contributions: Conceptualization, Y.L.; Data curation, G.X.; Formal analysis, Y.L.; Funding acquisition, Y.L.; Investigation, Z.S.; Methodology, G.X.; Project administration, Y.L. and H.Q.; Resources, H.Q.; Software, G.X.; Supervision, H.Q. and Z.S.; Validation, H.Q. and Z.S.; Visualization, G.X.; Writing – original draft, G.X.; Writing – review & editing, G.X., H.Q. and Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the National Natural Science Foundation of China: 61971456. General project of science and technology plan of Beijing Municipal Commission of Education: KM202010009011. Yuyou talent training program: 218051360020XN115/014.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10529–10538.
2. Xu, Q.; Zhong, Y.; Neumann, U. Behind the curtain: Learning occluded shapes for 3D object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 22 February–1 March 2022; pp. 2893–2901.
3. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
4. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
5. Kornblith, S.; Shlens, J.; Le, Q.V. Do better imagenet models transfer better? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2661–2671.
6. Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; Van Der Maaten, L. Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 181–196.
7. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the IEEE international Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 843–852.
8. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
9. Shi, S.; Wang, Z.; Shi, J.; Wang, X.; Li, H. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 2647–2664. [CrossRef] [PubMed]
10. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
11. Shi, S.; Wang, X.; Li, H. Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 770–779.
12. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. Std: Sparse-to-dense 3d object detector for point cloud. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1951–1960.

13. Yang, Z.; Sun, Y.; Liu, S.; Jia, J. 3dssd: Point-based 3d single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11040–11048.
14. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
15. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
16. He, C.; Zeng, H.; Huang, J.; Hua, X.-S.; Zhang, L. Structure aware single-stage 3d object detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11873–11882.
17. Li, Y.; Ma, L.; Tan, W.; Sun, C.; Cao, D.; Li, J. GRNet: Geometric relation network for 3D object detection from point clouds. *ISPRS J. Photogramm. Remote Sens.* **2020**, *165*, 43–53. [CrossRef]
18. Vasconcelos, C.; Birodkar, V.; Dumoulin, V. Proper Reuse of Image Classification Features Improves Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 13628–13637.
19. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 16000–16009.
20. Achituve, I.; Maron, H.; Chechik, G. Self-supervised learning for domain adaptation on point clouds. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 123–133.
21. Poursaeed, O.; Jiang, T.; Qiao, H.; Xu, N.; Kim, V.G. Self-supervised learning of point clouds via orientation estimation. In Proceedings of the 2020 International Conference on 3D Vision (3DV), Fukuoka, Japan, 25–28 November 2020; pp. 1018–1028.
22. Sauder, J.; Sievers, B. Self-supervised deep learning on point clouds by reconstructing space. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
23. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
24. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1910.13461.
25. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training. *Comput. Lang.* **2017**, *4*, 212–220.
26. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
27. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
28. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.
29. Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Luan, D.; Sutskever, I. Generative pretraining from pixels. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 1691–1703.
30. Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; Lu, J. 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, New Orleans, LA, USA, 19–24 June 2022; pp. 19313–19322.
31. Zhang, R.; Guo, Z.; Gao, P.; Fang, R.; Zhao, B.; Wang, D.; Qiao, Y.; Li, H. Point-M2AE: Multi-scale Masked Autoencoders for Hierarchical Point Cloud Pre-training. *arXiv* **2022**, arXiv:2205.14401.
32. Yang, J.; Shi, S.; Wang, Z.; Li, H.; Qi, X. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10368–10378.
33. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
34. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.