Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6909314, 12 pages
https://doi.org/10.1155/2022/6909314

Research Article

MANet: End-to-End Learning for Point Cloud Based on Robust Pointpillar and Multiattention

Xingli Gan,[1] **Hao Shi**,[1] **Shan Yang**,[2] **Yao Xiao**,[3] **and Lu Sun**[4]

[1] *School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China*
[2] *China Unicom Smart City Research Institute, Beijing 100048, China*
[3] *College of Sericulture, Textile and Biomass Sciences, Southwest University, China*
[4] *Department of Communication Engineering, Institute of Information Science Technology, Dalian Maritime University, China*

Correspondence should be addressed to Hao Shi; 222008855022@zust.edu.cn

Received 11 July 2022; Accepted 22 August 2022; Published 14 September 2022

Academic Editor: Xiaojie Wang

Detecting 3D objects in a crowd remains a challenging problem since the cars and pedestrians often gather together and occlude each other in the real world. The Pointpillar is the leader in 3D object detection, its detection process is simple, and the detection speed is fast. Due to the use of maxpooling in the Voxel Feature Encode (VFE) stage to extract global features, the fine-grained features will disappear, resulting in insufficient feature expression ability in the feature pyramid network (FPN) stage, so the object detection of small targets is not accurate enough. This paper proposes to improve the detection effect of networks in complex environments by integrating attention mechanisms and the Pointpillar. In the VFE stage of the model, the mixed-attention module (HA) was added to retain the spatial structure information of the point cloud to the greatest extent from the three perspectives: local space, global space, and points. The Convolutional Block Attention Module (CBAM) was embedded in FPN to mine the deep information of pseudoimages. The experiments based on the KITTI dataset demonstrated our method had better performance than other state-of-the-art single-stage algorithms. Compared with another model, in crowd scenes, the mean average precision (mAP) under the bird's-eye view (BEV) detection benchmark increased from 59.20% of Pointpillar and 66.19% of TANet to 69.91 of ours, the mAP under the 3D detection benchmark was increased from 62% of TANet to 65.11% of ours, and the detection speed only dropped from 13.1 fps of Pointpillar to 12.8 fps of ours.

1. Introduction

Autonomous driving uses sensors to detect and track moving objects, such as cars, pedestrians, and cyclists in real-time. Lidar is arguably the most important. Point cloud generated by lidar provide geometric structure information of objects and high-precision spatial coordinates, so how to use that information is extremely crucial [1].

With the outstanding achievements of computer vision and deep learning methods in pictures, extensive literature thinks about how to design end-to-end network results for point clouds. Unlike images represented as regular dense grids, 3D point cloud is not only irregular and disordered but also has the characteristics of uneven density and different shape and scaling ratio due to input-output size and order differences. Therefore, previous convolutional neural network (CNN) with regular grids is not suitable for point cloud. The method to solve this problem is to divide the space into regular geometry, such as $3 * 3 * 3$ cm and then manually design the feature extraction method. However, different feature extraction methods need to be designed for different environments and different detection targets, which is lack generality. In order to solve this problem, based on the PointNet designed by Qi et al. [2], an end-to-end detection network VoxelNet is proposed. VoxelNet [3] divides the point cloud into equidistant 3D voxels and encodes each voxel through the stacked VFE layer, and then, 3D convolution further aggregates the local voxel features to convert the point cloud into high dimensional volumetric representation. Finally, RPN produces test results. While the VoxelNet performance is strong, at 4.4 Hz, the inference time is too slow to deploy in real time. SECOND [4, 5] improved the inference speed of VoxelNet but the 3D convolutions remain a bottleneck. Lang et al. [6] use a novel encoder that learns features on pillars (vertical columns) of the point cloud instead of voxel to predict 3D oriented boxes for objects which is highly efficient to compute due to the key operations can be formulated as 2D convolutions and Pointpillar runs at 72 Hz which has the obvious speed advantage. Although Pointpillar enables a trade-off between speed and accuracy, the performance is still unsatisfactory in challenging cases. As shown in Figure 1, the first row shows the corresponding 2D image. The second row demonstrates the 3D detection results produced by Pointpillar. Pedestrians were not detected due to the severe occlusion. We reveal the intrinsic reason that the key parameter in Voxel Feature Encode (VFE) is the size of the voxel. A coarser voxel leads to a smaller feature map and faster inference speed but has inferior performance, especially for small objects.

To solve this problem, TANet [7] introduced attention in the feature extraction stage and also divided FPN-RPN into coarse extraction and fine extraction to effectively solve the occlusion problem. Inspired by the words of TANet, we introduce the mixed-attention network (MA) and the Convolutional Block Attention Module (CBAM) [8]. MA combines channel-wise, point-wise, and voxel-wise to enhance key information and to suppress unstable points. Channel-wise is used to determine which channels in each voxel; point-wise is used to determine which points in a voxel; voxel-wise is used to determine which grids are more important in all voxel grids. The CBAM consists of two complementary attention modules: spatial attention and channel attention. It can assign more weight to the unshaded part and less weight to the shaded part. By inserting CBAM into FPN, more refined features are obtained to improve the accuracy of classification and regression.

The contributions of this paper include the following three aspects. Firstly, we introduce one novel single-stage framework, named MANet, which strikes a balance between accuracy and speed. Secondly, we introduce the MA model which can feedback the original features of the point cloud more effectively and retain better geometric properties of the point cloud. Thirdly, our model runs at 11 frames per second while achieving competitive performance on the KITTI dataset.

The rest of the paper is organized as follows. Section 2 introduces the related achievements on the 3D object detection and analyzes their merit and demerit. Section 3 presents network architecture and the elaboration of mixed attention and CBAM. Section 4 presented the relevant parameter setting of the experiment, evaluation criteria, and comparative results with other models. Section 5 concludes this paper.

2. Relation Work

2.1. 3D Object Detection. Object detection of the point cloud is an integral part of the 3D vision. Like the task of 2D target detection, 3D target detection is to locate all interested targets in a given scene accurately. At present, 3D object detection can generally be divided into region proposal-based and single-shot methods [5]. The methods based on the candidate region firstly predict the region with possible objects (also known as the proposal), then extract the features of each region to determine the object category of each candidate region. Figure 2 shows the development process of the 3D object detection algorithm. According to the different methods for generating proposals, these methods can be further divided into three categories: multi-view-based methods, segmentation-based methods, and point-based methods. Among them, Frustum-PointNet [9], a technique based on the PointNet++ [10] cone proposed by qi, achieves high benchmark performance, but its multistage design makes end-to-end learning impractical. The methods based on single-shot methods predict category probability directly and use a single-level network to regression the 3D bounding box of objects. These methods do not require region proposals and postprocess, making them ideal for real-time applications. According to the type of input data, the methods can be divided into three types: methods based on BEV (projection graph) [3, 11, 12], methods based on discretization, and methods based on the point. VoxelNet is the first method to deploy a point network in a lidar point cloud for target detection. The author transforms the irregular point cloud into regular voxels, then processes them by a set of three-dimensional convolution layers, followed by a two-dimensional backbone and a detection head. This makes end-to-end learning possible, but like earlier work that relied on 3D convolution, requiring 225 ms of reasoning time (4.4 Hz) for a single point cloud. Pointpillar proposes converting voxels into pseudoimages and then using mature 2D target detection methods to carry out detection, which successfully realized real-time reasoning. TANet researches the relationship between point cloud, space, and voxel based on the Pointpillar, suppressed the unimportant and highlighted the important parts through the attention mechanism, and solves the interference of background points to a certain extent. HVNet [13] solves the balance problem of accuracy and speed caused by voxel division by integrating multiscale voxels.

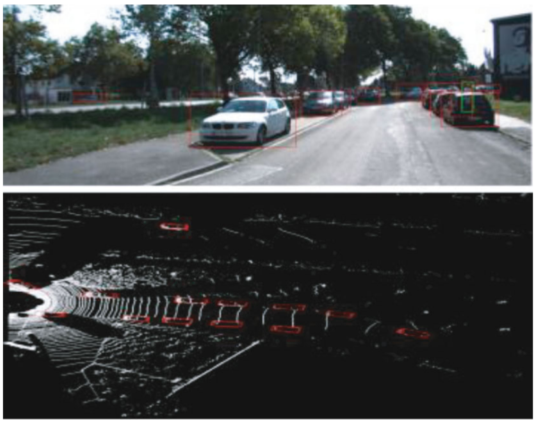


Figure 1: Detection results for Pointpillar.

2.2. Attention. In recent years, attention has gained popularity as a plug-and-play module for the existing basic convolutional neural network (CNN) architecture [13–17].

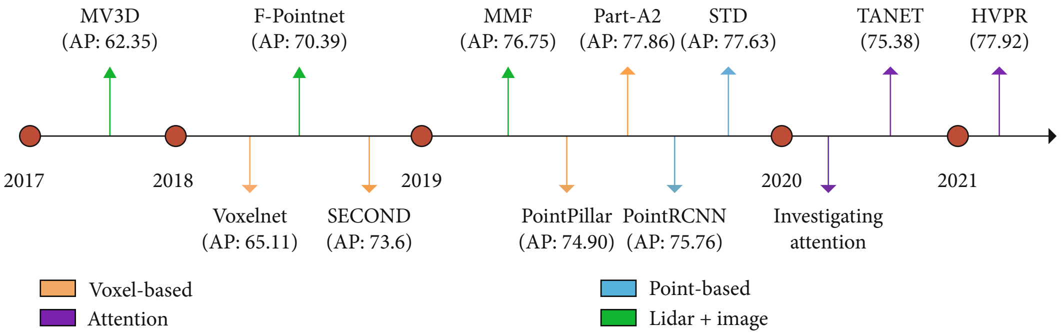


Figure 2: The development process of a 3D object detection algorithm based on point cloud representation.

Attention mechanisms are aimed at mimicking the human visual system by focusing on more relevant features to the target rather than an entire scene containing some unrelated background. Many methods have been introduced to estimate attention (weight) maps to reweight the original feature maps learned from CNN. SENet uses the global average set feature to calculate channel-level attention in their squeeze and excitation module for image-related tasks. They ignore spatial attention, which plays an essential role in deciding "where" as shown in [18]. After analyzing the defects of SENet and SKNet [19], a lightweight module CBAM is proposed to realize the mixing of space and channel by serial instead of parallel, which dramatically reduces the running time. Some scholars have tried to introduce attention to point cloud network architecture in recent years. The experiment of [20] verified that both 2D and 3D attention modules could be inserted into the existing modules to improve the feature extraction capability of the point cloud network. TANet combines channel attention, point attention, and voxel attention to enhance the critical information of the target and suppress unstable points, thus improving the robustness of the network. In addition, [21–23] introduce transformer into point cloud classification to enhance it by four points over PointNet with half the number of parameters.

2.3. Datasets. Semantic 3D is a large-scale point cloud classification benchmark, which provides a 3D point cloud dataset of natural scenes with large labels, totaling over 4 billion points and 8 category labels. And it also covers a wide variety of urban scenarios. KITTI [24] can not only have lidar, image, GPS, and INS data but also have manually labeled segmentation tracking results, which can be used to objectively evaluate the effect and performance of a large-range of 3D modeling and fine classification. The 3D object detection benchmark consists of 7,481 training images and 7,518 test images along with the corresponding point clouds, including a total of 80,256 labeled objects. The recent H3D dataset [25] records the crowded and highly interactive urban scenes, including a total of 1 million labeled instances in 27721 frames. The KAIST multispectral dataset [26] is a multispectral pedestrian detection dataset, which provides black-and-white thermal imaging image pairs during the day and night. Through the complementary advantages of color image and thermal imaging, the dataset improves the accuracy of pedestrian detection and overcomes the problems of previous pedestrian detection data, such as blocked pedestrians, messy background, and unclear imaging at night. Other noteworthy multimodal datasets include [27] providing driving behavior tags, [28] providing location classification tags, and raw data without semantic tags. The nuScenes dataset [29] contains 3D bounding box of 23 classes and 8 properties, with his annotation number being more than one times KITTI 7, resolving errors due to data enhancement.

3. Approach

This part introduces our object detection network based on attention object networks shown in Figure 3, called mixed-attention-Pp. The model structure diagram can be divided into VFE, multiscale pseudograph feature learning module, and detection head. Firstly, the original point cloud is transformed into a grid composed of voxels and obtains a more discriminative representation through mixed attention. Then, the features are aggregated by maximum pooling and finally dispersed back into a pseudoimage of $H * W * C$. In the feature extraction stage, we added CBAM to the original FPN [30–33]. The ability to capture detail is improved by inserting a CBAM module in the upsampling stage. Finally, a single shot multibox detector (SSD) is used to detect the position and classify categories.

We defined some common variables of the point cloud in advance. Point cloud set $P = \{p_i = [x_i, y_i, z_i, r_i]^t \in \mathrm{R}\}_{i=1,2,\cdots,M}$ where x_i, y_i, and z_i represent the coordinates of the midpoint in lidar space. r_i represents other spatial features, such as reflectivity and normal. M represents the number of point clouds. We use $(c_x, c_y, c_z, h, w, \ell, \theta)$ to define a 3D bounding box where c_x, c_y, and c_z represent the center point of the box; h , w, and ℓ represent the size of the box; and θ represents the direction of motion of the object.

3.1. Stacked Mixed-Attention. The entire range of space S is discretized into the point cloud P, where the range of P is (W^*, H^*, D^*). P is equally divided into a specific voxel grid $V = \{v^1, \ldots, v^k\}$ where $v^k \in R^{N*C}$, k represents the index of voxels, N represents the maximum number of point clouds per voxel, and c represents the characteristic number of point clouds. Each grid size is $w = W^*/v_w^*$, $h = H^*/v_h^*$, and

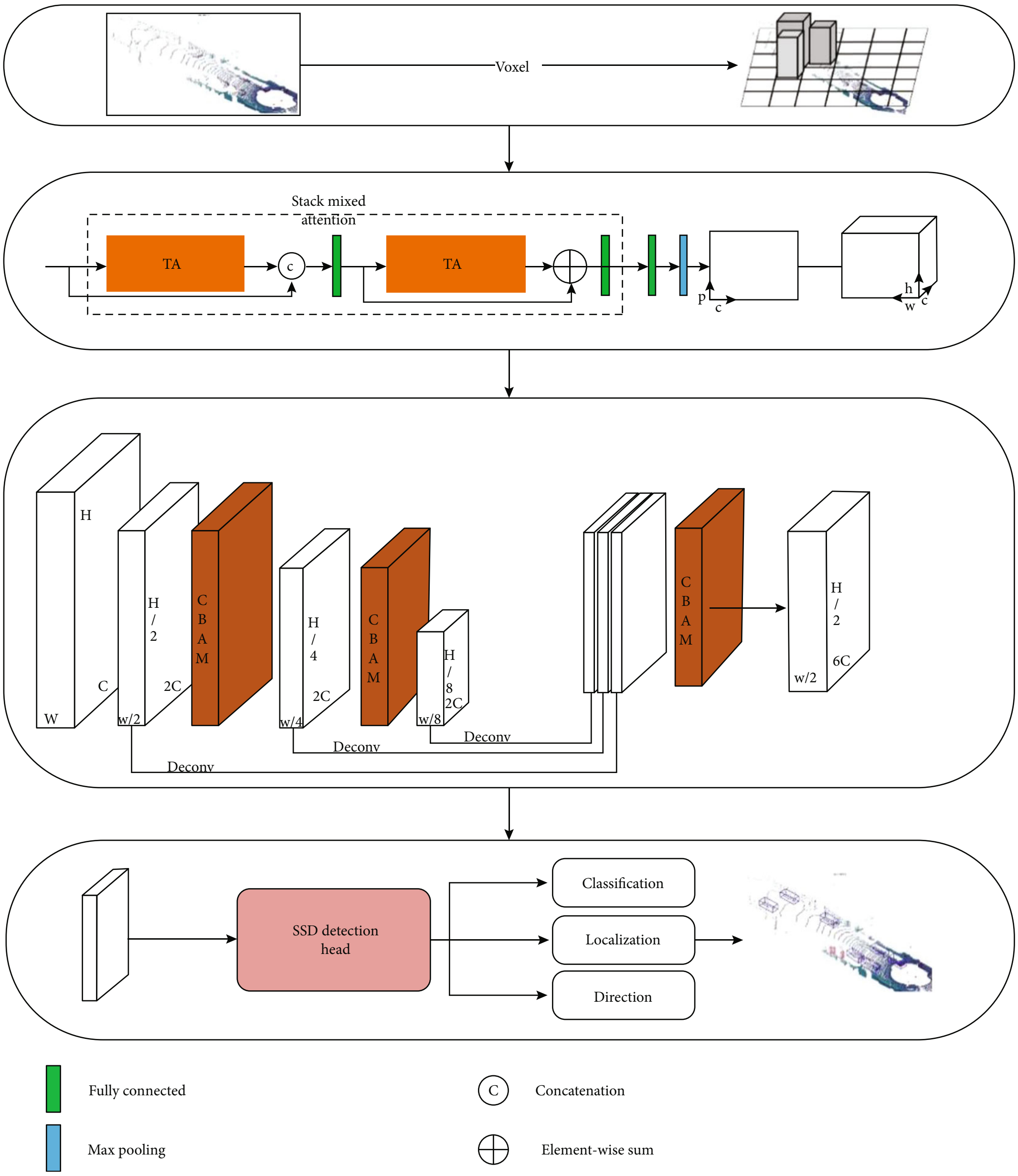


Figure 3: The full pipeline of network.

$D = D^* / v_d^*$. In the z-axis, we regard it as a whole, so the D is 1. They are shown in Figure 4.

3.1.1. Point-Wise Attention. Given the input v^k where its shape is (K, N, C). K represents the number of voxels, N represents the number of voxels, and C represents the number of channels. Firstly, we use maxpooling to make the feature transfer from the previous layer a vector then we use two MLP to obtain global coding features S^k.

$$S^k = W_2 \delta\left(W_1 E^K\right), \tag{1}$$

where E^K is point-wise, W_1 and W_2 are the weight parameters of two MLP, respectively, and δ is the ReLU activation function.

3.1.2. Channel-Wise Attention. Channel-wise attention is very similar to channel-wise attention. The only difference is maxpooling in the first dimension of input v^k. Specifically, we do maxpooling to convert the feature map of the previous layer into vector $U^k \in R^{1*C}$, which aggregates the features of all points on each channel. Then, through the two MLP to estimate the attention characteristic map $T^k = W_2' \delta (W_1' (U^k))$ where $W_2' \in R^{R*C}$, $W_1' \in R^{R*C}$. Then, S^k and T^k are combined through multiply, and the attention weight M^k is obtained through sigmoid activation function. Finally, F_1^K is obtained by dot product of v^k and M^k.

$$M^k = \delta\left(S^k * T^k\right). \tag{2}$$

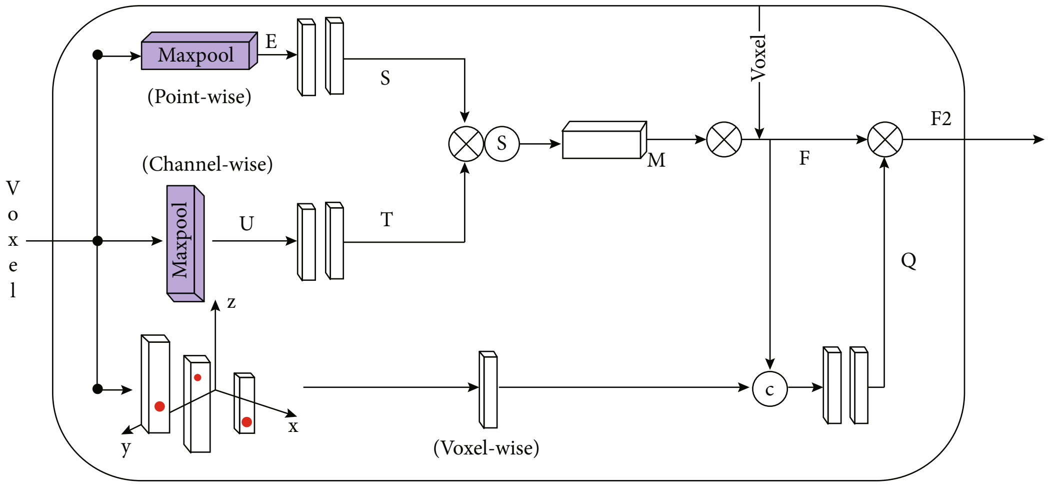


Figure 4: The architecture of the mixed attention module.

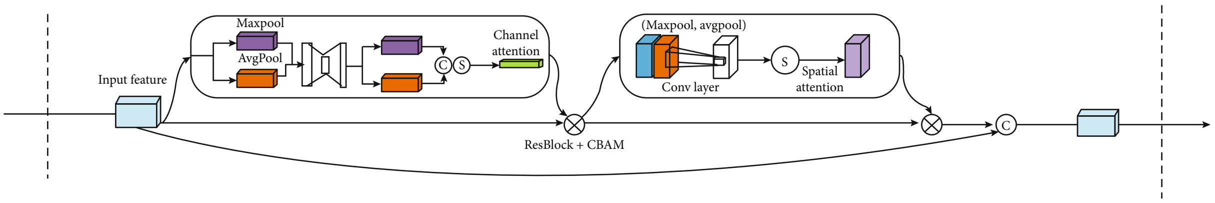


Figure 5: The architecture of CBAM.

3.1.3. Voxel-Wise Attention. Further, the importance of voxels is judged by focusing on voxels. The input for voxel-wise is V_C and F_1^K where V_C represents the center of gravity point of each 3D voxel grid. Its size is $(K, 1, 3)$. We first expand V_C to $(K, N, 3)$ in the first dimension and then concatenate V_C and F_1^K in the first dimension and then obtain voxel level attention weight Q which size is $(K, 1, 1)$ through two full connection layer and sigmoid. Finally, $F_2^K = q_k * F_1^K$.

3.1.4. Stack-TA. Considering that TA module directly acts on point cloud, it does not contain high-dimensional semantic features. The generalization ability of this network is insufficient. We chose to stack TA in order. We choose to stack TA in a sequential manner. Specifically, VX and VC are input into the first TA, and the output is M. M is used as the input of the second TA attention. The difference from the former is that we directly add the input and output instead of splicing. Finally, maxpooling is used to aggregate the features of all voxels.

3.2. Convolutional Block Attention Module (CBAM). As shown in Figure 5, CBAM is the mixed-attention composed of two continuous attention blocks. To be specific, with the intermediate feature graph $F \in R^{C*H*W}$ as input. CBAM successively deduced the one-dimensional channel attention graph $M_C \in R^{C*1*1}$ and the 2D space attention graph $M_s \in R^{1*w*h}$. The whole attention process can be summarized as follows: graph $M_s \in R^{1*w*h}$. The whole attention process can be summarized as follows:

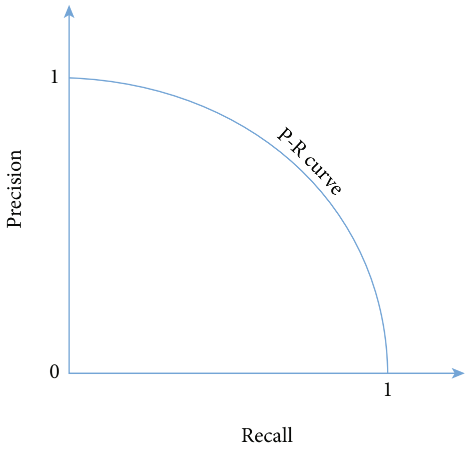
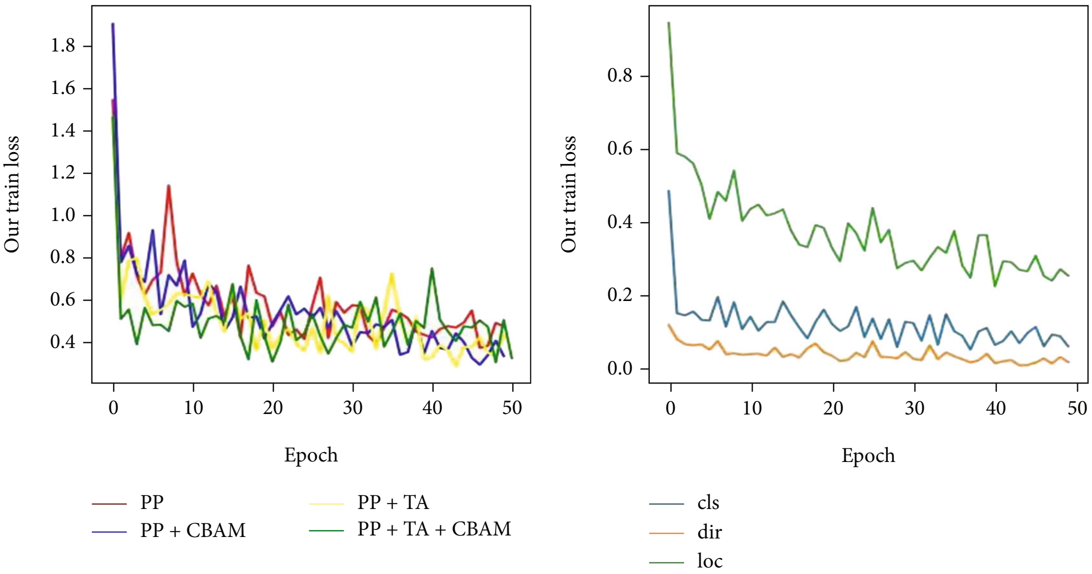


Figure 6: The P-R curve.

$$\begin{aligned} F' &= M_C(F) \otimes F, \\ F' F' &= M_S\left(F'\right) \otimes F'. \end{aligned} \tag{3}$$

3.3. Backbone. We used a trunk similar to FPN [31], whose structure is shown in Figure 3. The backbone network can be divided into three parts. The one on the left is the



(a) (b)

Figure 7: The loss of train. (a) The ablation experiments of different attention. (b) The component of the total loss.

downsampling network, which produces features with smaller and smaller spatial resolutions. The CBAM module in the middle inputs the feature map into CBAM after convolution, and CBAM extracts useful information while inhibiting irrelevant information. On the right is the upsampling module, which combines the feature layer $(6c, w/2, h/2)$ through deconvolution. The final output features are a concatenation of all features that originated from different strides.

3.4. Detection Head and Loss. We use the same detection header as SSD [5, 34] to locate 3D objects. Ground truth (gt) and anchors are defined by $(x, y, z, w, l, h, \theta)$ where (x, y, z), (w, l, h), and θ are the center point, size, and angle of the box, respectively. Local residuals between the anchors and the ground truth are defined:

$$\begin{aligned}
\Delta x &= \frac{x^{\mathrm{gt}} - x^{\alpha}}{d}, \\
\Delta y &= \frac{x^{\mathrm{gt}} - x^{\alpha}}{d}, \\
\Delta z &= \frac{z^{\mathrm{gt}} - z^{\alpha}}{h^{\alpha}}, \\
\Delta w &= \log \frac{w^{\mathrm{gt}}}{w^{\alpha}}, \\
\log \frac{l^{\mathrm{gt}}}{l^{\alpha}} \Delta h &= \log \frac{h^{\mathrm{gt}}}{h^{\alpha}}, \\
\Delta \theta &= \sin\left(\theta^{\mathrm{gt}} - \theta^{\alpha}\right),
\end{aligned} \tag{4}$$

where $d = \sqrt{(w^a)^2 + (l^a)^2}$ and gt and a are, respectively, the ground truth and anchor box. In order to train our model, we carried out regression, classification and memory update, the total loss is

Table 1: Ablation experiments on the effect of CBAM, SE, and TA as well as different combination settings.

Methods	Car	Pedestrian	Cyclist	mAP
Baseline [6]	74.9	43.5	64.5	59.0
SE [18]	68.3	44.94	51.72	54.9
CBAM [8]	76.1	50.2	64.3	63.5
TA [7]	78.8	46.47	59.60	62
TA + CBAM	78.8	51.74	64.74	65.1

$$\mathscr{L} = \frac{1}{N_{\mathrm{pos}}} \begin{pmatrix} \lambda_{\mathrm{reg}} \mathscr{L}_{\mathrm{reg}} + \lambda_{\mathrm{dir}} \mathscr{L}_{\mathrm{dir}} \\ + \lambda_{\mathrm{cls}} \mathscr{L}_{\mathrm{cls}} + \lambda_{\mathrm{mem}} \mathscr{L}_{\mathrm{mem}} \end{pmatrix}, \tag{5}$$

where N_{pos} represents positive anchors; the equilibrium parameter of the corresponding loss and the total localization loss is defined as follows:

$$\mathscr{L}_{\mathrm{reg}} = \sum_{r \in (\mathscr{X}, \mathscr{Y}, \mathscr{Z}, \mathscr{W}, \ell, \hbar, \theta)} \mathrm{Smooth}L1(\Delta \mathbf{r}). \tag{6}$$

4. Experiment and Discussion

4.1. Parameter Settings. We set the threshold on the XY of the scene of the point cloud to (0, 70.4) and (-40, 40). The resolution of the pillar is 0.16 m, the maximum number of columns is 12000, and the maximum number of points per column is 100. If the points are less than 32, then we randomly sample the points in the column. If the points' number is more than 100, then we use farthest point sampling for downsampling. We share the matching strategy for the box and prediction box as Pointpillar. Specifically, the anchors consist of the

Table 2: Results on the KITTI test BEV detection benchmark.

Methods	3D mAP	Car			Pedestrian			Cyclist		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
VoxelNet [3]	58.25	89.25	79.26	77.39	46.13	40.74	38.11	66.70	54.76	50.55
Pp [6]	66.19	88.35	86.10	79.83	58.6	50.23	47.19	79.14	62.25	56.00
F-P [9]	65.39	88.70	84.00	75.33	58	50.22	47.20	75.38	61.96	54.98
PIXOR [12]	N/A	89.38	87.30	77.97						
MV3D [38]	N/A	86.02	76.90	68.49						
SECOND [39]	60.56	88.07	79.37	77.95	55.1	46.27	44.76	73.67	56.04	48.78
MANet	69.91	89.21	86.36	83.10	61.4	55.01	51.23	82.81	68.36	63.76

Table 3: Results on the KITTI test 3D detection benchmark.

Methods	Mod	Bev mAP	Car			Pedestrian			Cyclist		
			Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
MV3D [38]	Lidar&img	N/A	71.09	62.35	55.12						
F-P [9]		57.35	81.20	70.39	62.19	51.21	44.89	40.23	71.96	56.77	50.39
VoxelNet [3]	Lidar	49.05	77.47	65.11	57.73	39.48	33.69	31.5	61.22	48.36	44.37
SECOND [39]		56.69	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90
Pp [6]		59.20	79.05	74.99	68.30	52.08	43.53	41.49	75.78	59.07	52.92
TANet [7]		62	83.81	75.38	67.66	54.92	46.67	38.63	73.93	59.60	53.59
MANet		65.11	83.47	78.85	71.89	56.02	51.74	45.58	80.08	64.74	60.79

following 7 parameters (c_x, c_y, c_z, h, w, l, θ), and the anchor direction, θ, is applied to two orientations (0 and $\pi/2$). We regard anchors with intersection over union (IOU) greater than 0.6 as positive samples and those less than 0.2 as negative samples. We ignore those anchors with IOU between (0.2, 0.6) when calculating the loss. We select the nonmax suppression (NMS) score of 0.5 in the postprogress step. Due to the large body size gap between cars and people, we set the corresponding parameters for different objects.

Car. Thresholds are set to (0, 70.4) and (-40, 40), the size of the prior box is set to (1.6, 30.9, 1.5), and the size of the confidence interval was set to (0.45, 6).

Pedestrian and Cyclist. Since pedestrians are blocked and the number of point clouds is sparse, we set the scene range to (0, 48) and (-20, 20), the size of the prior box is set to (0.6, 0.8, 1.73), and the confidence interval to (0.2, 0.6). The prior size of all boxes is set to (1.6, 3.9, 1.5) and (0.6, 0.8, 1.73) for pedestrians and individuals. The confidence interval for the vehicle is (0.45, 0.6), and the pedestrian is (0.5, 0.35).

4.2. Sample Ground Truths from the Database.

This paper takes the KITTI dataset as the benchmark. Since the dataset samples are only 7361 frames and the number is small, we can manually add some objects to the point cloud to improve the effect of model training. This paper adopts the same data enhancement method as Pointpillar. Firstly, points in the prior box are removed and recorded from the training set. Secondly, N samples are randomly selected, and the prior box and point cloud of the selected samples were randomly rotated (- $\pi/20$, $\pi/20$) and translated (0, 0.25); then, we added the samples to the training set.

Table 4: Results on the test nuScenes 3D detection benchmark.

Methods	Car	Pedestrian	Cyclist	mAP
Baseline [6]	62.5	50.2	64.5	57.6
PVRCNN	64.8	46.7	—	—
Centerpoint	66.1	62.4	67.6	65.3
Point augmenting	62.2	64.6	73.3	66.7
Ours	65.1	63.7	65.9	64.9

4.3. Algorithm Performance Evaluation.

In the field of target detection [35–37], recall and precision are mainly used as the performance measure of the algorithm. Precision (P) and recall (R) are, respectively, defined as follows:

$$\begin{aligned} R &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \\ P &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \end{aligned} \tag{7}$$

where TP, FP, TN, and FN are represented as true and false positive and true and false negative examples, respectively. It can be seen that the denominator of P is the total number of boxes detected by the detec-tor, and the denominator of R is the total number of boxes given by GT. Since R and P are a pair of paradoxical quantities. To balance them, the "P-R" curve obtained with P as the vertical axis and R as the horizontal axis is used to reflect the relationship in the Figure 6.

The average accuracy comes from the PR curve. In practice, we do not directly calculate the PR curve, but instead smooth the PR curve. That is, for each point on the PR

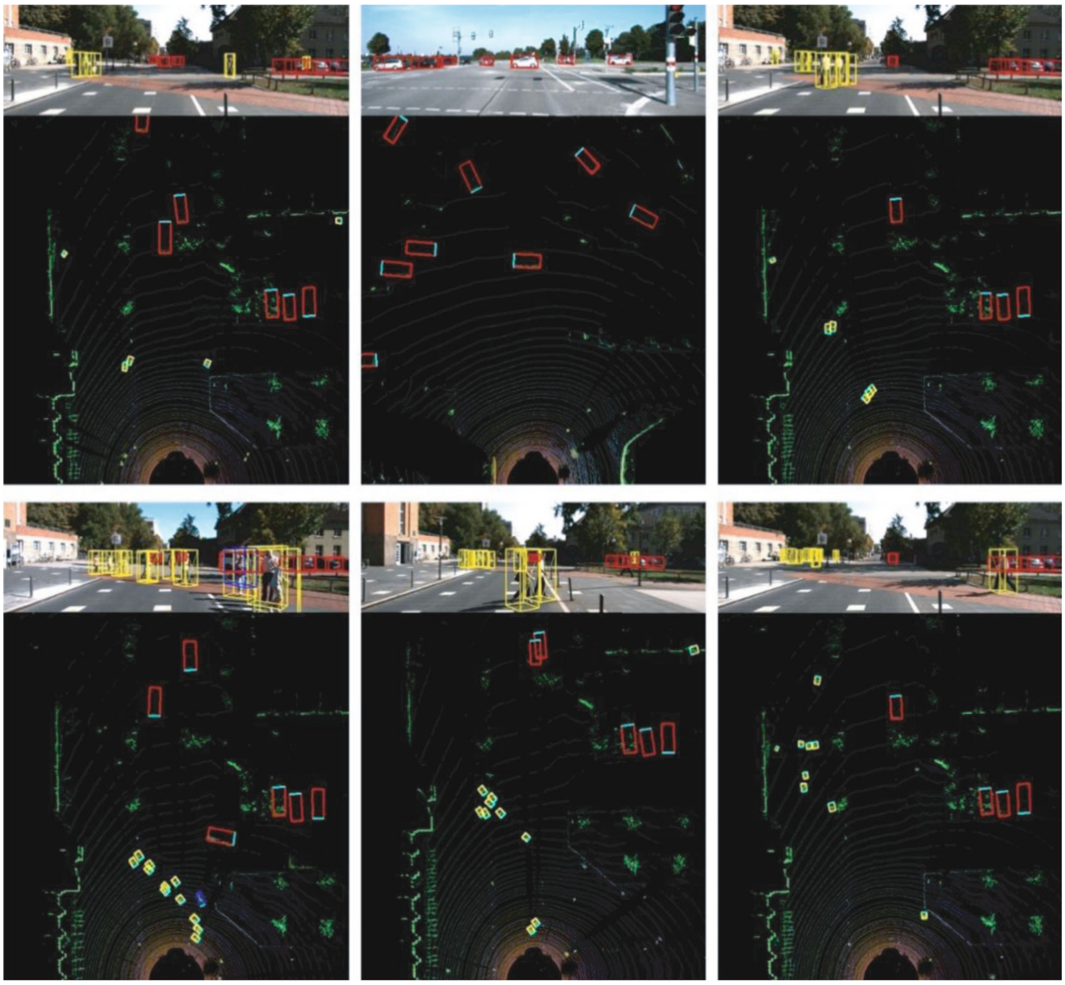


Figure 8: Results of 3D detection on the KITTI test set. For better visualization, the 3D boxes detected using lidar are projected onto images from the left camera.

curve, the value of precision takes the value of the largest precision on the right side of that point. In addition, for a better performance of the reaction model, we also introduced mAP (mean average precision) to represent the mean of AP values for all types.

$$\mathrm{mAP} = \frac{\left(\sum_{i=1}^{C} \mathrm{AP}_i\right)}{C}. \quad (8)$$

In this paper, according to the complexity of the environment, we divided the objects into easy, moderate, and hard and counted the models for 3D bounding box AP, Bev AP, Bev mAP, and Bev bounding box, respectively.

4.4. Performance Analysis

4.4.1. Loss. Train runtime is measured on a GTX 1050 Ti GPU. As mentioned in 3.4, loss consists of three parts: classification, localization, and direction. Figure 7(b) shows the changes of loss in the training process. As can be seen in the figure, the model directly generates the category probability and position coordinate value of the object without generating the candidate region first and then classifying the candidate region, so it has a faster detection speed. In Figure 7(a), we made statistics on the decrease of loss in 4 situations: Pointpillar, Pointpillar +CBAM, Pointpillar+TA, and Pointpillar +TA + CBAM during the training process. The results in Figure 7(a) show that whether inserting CBAM or TA, the loss value of final convergence is lower than the original Pointpillar.

4.4.2. Analysis of the Attention Mechanisms. Table 1 presents ablation studies of the proposed attention mechanisms. The recall for cars, pedestrians, and cyclists is set to (0.7, 0.5, 0.5), and the score threshold is set to 0.4. We removed both the TA and the CBAM from our model as the baseline and achieve a 3D mAP of 59.0%; with only CBAM and SENET, we can see that if only the channel attention mechanism is added, the accuracy is reduced by KITTI 4% while CBAM outperforms the baseline model by 4.5% from Table 1. This suggests that the spatial information is beneficial for the regression of the 3D bounding box. We add the TA module to the baseline base, and the performance was promoted to 62. Finally, we reinserted TANet and CBAM into the model,

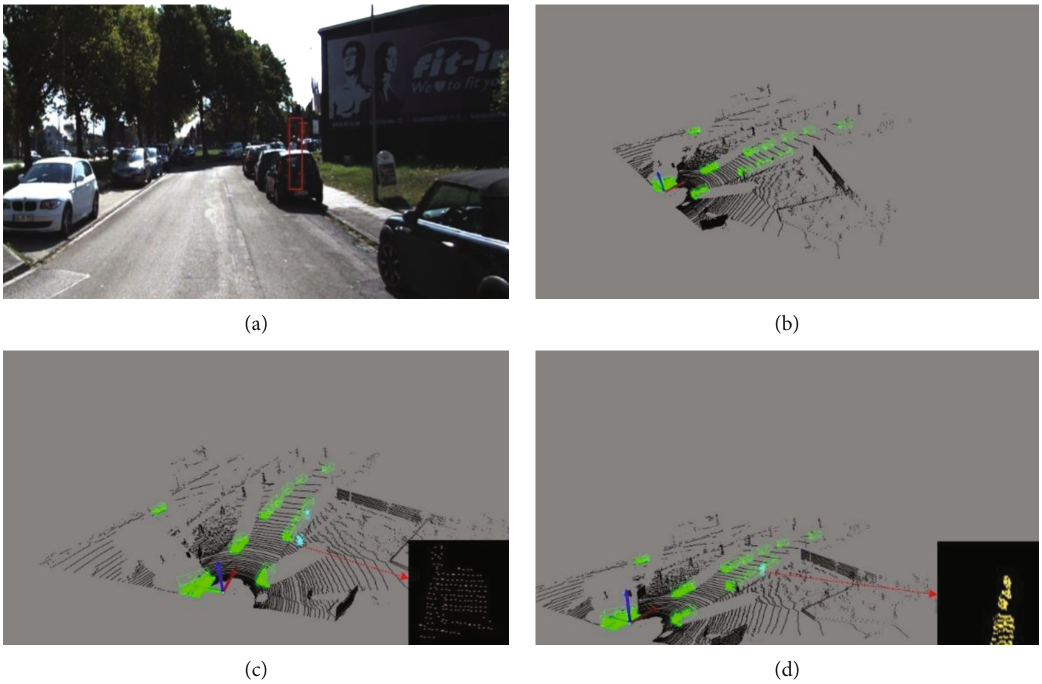


Figure 9: Results of 3D detection on the KITTI test set: (a) the corresponding 2D image; (b) the detection results of the Pointpillar; (c) the detection results of the Pointpillar+SE; (d) the detection results of ours.

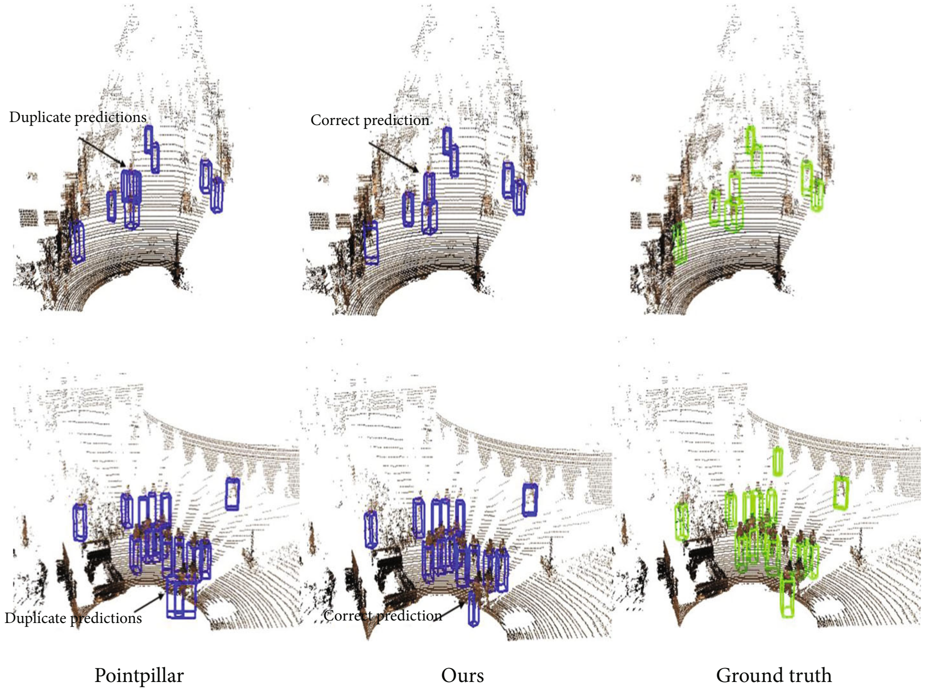


Figure 10: Results of 3D detection on the nuScenes test set. The first column is Pointpillar, the second column is our algorithm, and the third is the ground truth.

with the model accuracy further improved to 65. The experiments show that the performance of the network can be effectively improved by assigning weights to different points for some objects with occlusion. Figure 7 shows the loss of train. Figure 7(a) shows the ablation experiments of different attention. Figure 7(b) shows the component of the total loss.

4.4.3. Quantitative Analysis. In Tables 2 and 3, we compared our model with the others. For comparison convenience, the car, pedestrian, and cyclists' recall are set to (0, 7, 0.5, 0.5), and the score threshold is set to 0.1. All detection results are measured using the official KITTI evaluation detection metrics which are bird's-eye view (BEV) and 3D. We classified the 3D detection model as lidar and lidar and image-based. As can be seen from the table, our mAP rose from Pointpillar at 66.19% and 59.20%, respectively, to 69.91 and 65.1; the 3D bounding box in difficult environments has increased by 4%, which is an exciting result. This suggests that adding attention modules makes the network still work in the face of complex environments. In Table 4, we also add the latest detection model-TANet; we all adopt TA attention in the VFE stage; in the pseudoimage feature extraction stage, our model adopts FPN + CBAM mode, and TANet adopts CFR (coarse regression module and a fine regression); the results show that our model outperforms TANet. We also tested on the large dataset nuScenes, and the experimental results are shown in Table 4. Our method achieves some improvement in single-stage centerpoint compared to pillar and PVRCNN, but point augment achieves high accuracy in cycling and pedestrian detection results. The reason is that point augment uses a multimodal approach, which integrates the semantic information of images, and pictures have a natural advantage for capturing small objects.

4.5. Test Result. Figure 8 shows some test results on the KITTI test set to visually show the detection effect of the model. Each image consists of 2 parts: the first row is the predicted 3D bounding box projected into the image, and the second row is the predicted results of Bev where the red represents the car, yellow is the cyclist, and blue is the pedestrian. The images in the first row select the road with few pedestrians and no occlusion. It can be seen that our algorithm detects all objects without occlusion, and the second row selects the scene when there are many pedestrians. It can be seen that pedestrians will block each other and vehicles, and our algorithm can still have a high recall rate.

To further verify the robustness of our algorithm, we compared our method with other current state-of-the-art algorithms. In Figure 9(a), most of the pedestrian's body is covered by the vehicle, and the radar has collected less than 30 points. The original Pointpillar experiment results are shown in Figure 9(b), and the pedestrians were not found. The Pointpillar+SE algorithm (Figure 9(c)) mistakenly detects the road signs as pedestrians. We analyze that the fake image weakens the spatial feature and strengthens the channel feature. SE attention further strengthens the channel feature, so there will be a problem of false detection. While we propose a mixed-attention model that the global spatial information, local spatial information and point features are integrated so the pseudoimage contains more spatial features. CBAM combines space with channels, so it can show excellent performance. In addition to comparing the effect on KITTI, we also verified it in nuScenes. As shown in Figure 10, our algorithm detected the sparse pedestrians missed by second and avoid the repeated detection problem caused by crowding.

5. Conclusion

In this paper, we have designed the MA-CBAM-Pp for single-stage 3D object detection, which improves detection performance, particularly in crowd scenes. Two attention models are inserted into Pointpillar, where the mixed attention is added in VFE which feedback the original features of the point cloud more effectively and retain better geometric properties of the point cloud, and the CBAM attention module was embedded in FPN to mine the deep information of pseudoimages from spatial dimension and channel dimension. Significantly, our result on KITTI with MA-CBAM-Pointpillar outperforms the previously best result that uses TANet by about 3.11%. Detailed experimental comparisons have demonstrated the value of our method, which improves detection accuracy by a large margin in occlusion scenarios. Meanwhile, the visualization results also demonstrate that our inserted modules help improve the accuracy of single-stage 3D object detection. Future work will do fine regression based on this model to further improve the accuracy of the model.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

All authors participated in the discussion of the initial design, methodology, and derivation of the method presented in this paper. X.G. and H.S. designed, conceived the simulation experiments, and analyzed the data. H.S. and D.F.D. performed the training, analyzed the results, and wrote the paper. Z.H.H. and Y.S. reviewed the paper, and all authors participated in amending the manuscript. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (project: Underground space disaster information perception and emergency communication equipment) (No. 2020YFC1511705), the National Natural Science Foundation of China (62101088), and the Radar Signal Processing National Defense Science and Technology Key Laboratory Fund (6142401200101).

Supplementary Materials

The pictures in the paper are provided. *(Supplementary Materials)*

References

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The Kitti vision benchmark suite," in *the 2012*

IEEE conference on computer vision and pattern recognition, pp. 3354–3361, Providence, RI, USA, 2012.

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: deep learning on point sets for 3D classification and segmentation,” in *the Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, Hawaii,USA, 2017.

[3] Y. Zhou and O. Tuzel, “VoxelNet: end-to-end learning for point cloud based 3D object detection,” in *the Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490–4499, Salt Lake City, USA, 2018.

[4] S. Ji, X. Wei, M. Yang, and Y. Kai, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[5] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Single-shot refinement neural network for object detection,” in *the Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4203–4212, Salt Lake City, USA, 2018.

[6] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: fast encoders for object detection from point clouds,” in *at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, Long Beach,USA, 2019.

[7] Z. Liu, X. Zhao, T. Huang, R. Hu, Z. Yu, and X. Bai, “Tanet: robust 3D object detection from point clouds with triple attention,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 11677–11684, 2020.

[8] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: convolutional block attention module,” in *the Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[9] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum pointnets for 3D object detection from RGB-D data,” in *the Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 918–927, Salt Lake City, USA, 2018.

[10] C. R. Qi, L. Yi, S. Hao, and L. J. Guibas, “PointNet++: deep hierarchical feature learning on point sets in a metric space,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, “Joint 3D proposal generation and object detection from view aggregation,” in *the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, Madrid, Spain, 2018.

[12] B. Yang, W. Luo, and R. Urtasun, “Pixor: real-time 3D object detection from point clouds,” in *the Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, Salt Lake City, USA, 2018.

[13] M. Ye, S. Xu, and T. Cao, “HVNet: hybrid voxel network for lidar based 3D object detection,” in *the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1631–1640, Seattle,USA, 2020.

[14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model,” 2016, https://arxiv.org/abs/1602.07360.

[15] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, https://arxiv.org/abs/1409.1556.

[17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-V4, inception-resnet and the impact of residual connections on learning,” in *the Thirty-first AAAI conference on artificial intelligence*, San Francisco, USA, 2017.

[18] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, Salt Lake City, USA, 2018.

[19] W. Wu, Y. Zhang, D. Wang, and Y. Lei, “SK-Net: deep learning on point cloud via end-to-end discovery of spatial keypoints,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 6422–6429, 2020.

[20] S. Qiu, W. Yunfan, S. Anwar, and C. Li, “Investigating attention mechanism in 3D point cloud object detection,” in *2021 International Conference on 3D Vision (3DV)*, pp. 403–412, London, United Kingdom, 2021.

[21] M.-H. Guo, J.-X. Cai, Z.-N. Liu, M. Tai-Jiang, R. R. Martin, and S.-M. Hu, “Pct: point cloud transformer,” *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021.

[22] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, Salt Lake City, USA, 2018.

[23] H. Zhao, L. Jiang, J. Jia, P. H. S. Torr, and V. Koltun, “Point transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: the Kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[25] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, “The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9552–9557, Montreal, QC, Canada, 2019.

[26] Y. Choi, N. Kim, S. Hwang et al., “Kaist multi-spectral day/night data set for autonomous and assisted driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018.

[27] Y. Chen, J. Wang, J. Li et al., “Lidar-video driving dataset: learning driving policies effectively,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5870–5878, Salt Lake City, USA, 2018.

[28] M. Brossard, A. Barrau, and S. Bonnabel, “AI-IMU dead-reckoning,” *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 585–595, 2020.

[29] H. Caesar, V. Bankiti, A. H. Lang et al., “nuScenes: a multimodal dataset for autonomous driving,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, Seattle,USA, 2020.

[30] Z. Ding, H. Xu, and M. Niethammer, “VoteNet: a deep learning label fusion method for multi-atlas segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 202–210, Cham, 2019.

[31] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, Hawaii,USA, 2017.

[32] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 770–779, Long Beach,USA, 2019.

[33] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5589–5598, Seattle,USA, 2020.

[34] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: point-based 3D single stage object detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11040–11048, Seattle,USA, 2020.

[35] F. Nobis, E. Shafiei, P. Karle, J. Betz, and M. Lienkamp, "Radar voxel fusion for 3D object detection," *Applied Sciences*, vol. 11, no. 12, p. 5598, 2021.

[36] J. Zhang, J. Wang, X. Da, and Y. Li, "HcNet: a point cloud object detection network based on height and channel attention," *Remote Sensing*, vol. 13, no. 24, p. 5071, 2021.

[37] W. Zheng, H. Xie, Y. Chen, J. Roh, and H. Shin, "PIFNet: 3D object detection using joint image and point cloud features for autonomous driving," *Applied Sciences*, vol. 12, no. 7, p. 3686, 2022.

[38] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

[39] Y. Yan, Y. Mao, and B. Li, "Second: sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.