SAE-PointPillars: Adaptive Spatial Feature Fusion Based PointPillars 3D Target Detection Algorithm

Yuxuan Zhang, Ziwei Zhou

***Abstract*—To tackle the issue of subpar performance in detecting distant and occluded targets within 3D object detection algorithms for autonomous driving road scenarios, the PointPillars algorithm was enhanced and the SAE-PointPillars algorithm was put forward. Despite the fact that PointPillars excels in rapid point cloud processing and real-time target detection, its precision and robustness in identifying distant and occluded targets still present limitations. Firstly, the voxelized feature input is refined based on the SimAM attention mechanism to allow the feature extraction stage of the network to pay greater attention to crucial information and enhance global feature learning. Secondly, the ASFF adaptive spatial feature fusion module is employed to improve the backbone network, boosting the network's feature extraction and feature fusion capabilities, and addressing the information loss problem resulting from feature concatenation. Finally, the EMA attention mechanism is introduced to further strengthen the feature information. The test outcomes on the KITTI dataset reveal that, in samples of simple, medium, and difficult categories, compared with the original PointPillars network, the SAE-PointPillars algorithm improves the average precision of vehicle, pedestrian, and cyclist categories by (3.37%, 3.32%, and 1.56%), (2.85%, 5.35%, and 3.97%), and (3.27%, 4.13%, and 5.36%).**

***Index Terms*—PointPillars, Object Detection, Attention Mechanism, Adaptive Spatial Feature Fusion**

I. Introduction

3D object detection technology furnishes 3D information concerning the position, size, and type of surrounding targets for autonomous driving, constituting the core part of the autonomous driving environment perception system. With the rapid progress of deep learning and LiDAR technology, a multitude of 3D object detection algorithms based on LiDAR point cloud data have emerged. Compared with traditional image data, which is limited to providing two-dimensional spatial information, 3D point clouds, due to their capacity to represent three-dimensional spatial coordinates, can construct a more comprehensive geometric and topological structure of the scene. This three-dimensional representation approach offers substantial advantages for environmental perception in autonomous driving systems. By directly acquiring the spatial coordinate information of target objects, it significantly enhances the accuracy and reliability of environmental perception. It is worth noting that point cloud data, typically acquired through LiDAR scanning, exhibits characteristics such as unstructured distribution, spatial disorder, uneven density, and partial missing data. These features pose challenges for the direct transfer and application of two-dimensional object detection networks to point cloud data processing [1]. Currently, research paradigms in the field of 3D object detection are primarily categorized into three directions based on the type of data source: methods for 3D detection using single LiDAR point clouds, methods for 3D detection based on monocular or multicamera vision [2], and cross-modal 3D detection approaches that integrate multi-sensor information [3]. For diverse point cloud data processing methods, the prevalently employed 3D point cloud detection models encompass Point-based [4], Voxel-based, Point-Voxel-based [5], and Multi-viewbased [6]. These algorithms have enhanced the overall performance and efficiency of target detection, propelling the further advancement of 3D object detection. Nevertheless, accurately detecting distant and occluded targets from sparse and voluminous unstructured point cloud data remains a formidable task.

The detection of point cloud targets represents one of the core tasks in point cloud processing. This thesis focuses on 3D object detection technology using LiDAR. As a core component of intelligent driving perception systems, the algorithmic framework for 3D object detection based on LiDAR can be categorized into four technical branches according to the data processing paradigm: direct point cloud processing, grid-based structuring, point-voxel hybrid approaches, and range-based methods [7]. The performance of point-based 3D object detection methods largely depends on their sampling strategies. Theoretically, increasing the number of points in the environment can improve detection performance; however, this also leads to a significant increase in memory consumption. Notably, the intrinsic non-uniform spatial distribution of 3D point clouds tends to cause sampling bias, where high-density regions produce redundant features while sparse regions suffer from insufficient information representation. This spatial feature imbalance not only constrains the model's generalization capability but also ultimately degrades detection accuracy. Current research in this area includes representative methods such as PointNet++ [8], Pointformer [9], and Point-GNN [10]. The 3D detection method based on voxel representation involves transforming point clouds into structured grid representations, a process known as voxelization. This

Manuscript received December 9, 2024; revised April 6, 2025.

Yuxuan Zhang is a postgraduate student of the University of Science and Technology Liaoning, Anshan, 114051, China (corresponding author phone: 156-1381-5302; e-mail: 15613815302@163.com)

Ziwei Zhou is a Professor at the University of Science and Technology Liaoning, Anshan, 114051, China (e-mail: 381431970@qq.com)

procedure maps 3D point clouds onto 2D feature maps, enabling them to meet the input requirements of 2D convolutional neural networks. Consequently, this approach facilitates the effective transfer of well-established feature extraction techniques from the 2D visual domain to 3D data processing. Current research in this area includes representative methods such as Pointpillars [11], Center-Point [12] and VoTr [13]. The 3D detection method based on point-voxel hybrid representation achieves target recognition by synergistically leveraging the complementary characteristics of the two data modalities: point cloud data preserves high-precision geometric details, while voxelization constructs a regularized grid structure for efficient feature extraction. Through a joint representation strategy, this approach not only retains the capability to represent geometric details but also exploits the structured nature of voxelized data to improve the efficiency of computational resource utilization. Current research in this area includes representative methods such as SASSD [14] and PVGNet [15].During point cloud data processing, the distance-based 3D detection method constructs a distance image using inter-point distance information, as an alternative to directly employing the original 3D coordinates. Current research in this area includes representative methods such as RangeDet [16], to-point [17] and Rsn [18].

PointPillars [11] is a dimensionality reduction technique specifically designed for point cloud object detection networks, which has great potential for practical applications. Firstly, it converts 3D data into 2D, thus reducing the amount of data required. Secondly, this method uses 2D convolution techniques instead of 3D convolution, which is a clearly dominant, computationally intensive, and difficult-to-deploy operator to identify features, thus reducing the complexity of computation and enhancing the convenience of algorithm deployment. However, the detection accuracy of this network is not as good as that of similar detection methods. The main factors causing this problem are as follows: firstly, the detection efficiency of the network is affected by the size of the pillars. As the size of the pillars increases, the resolution of the false image decreases, but its running efficiency is relatively high. Secondly, reducing the size of the pillars will improve the resolution of the false image; although the running speed is slower, the detection results are still excellent; thirdly, the false image is generated by the feature encoding network. The quality of the image generated in this way directly affects the detection results; finally, the features used for detection usually contain a lot of redundant information.

This paper tackles the issue of subpar performance in detecting distant and occluded targets using PointPillar and proposes a 3D target detection algorithm for laser point cloud based on an enhanced PointPillars. The specific work is presented as follows:

Relying on the SimAM attention mechanism [19], this paper refines the voxelized feature input in PointPillars to heighten the network's focus on key information during the feature extraction stage. This refinement augments the global feature learning, thereby effectively enhancing the precision and robustness of target detection.

To address the information loss problem caused by feature fusion, an adaptive spatial feature fusion module (ASFF) [20] is introduced to improve the 2D backbone network. This module aims to dynamically and adaptively adjust the weight of each feature element to achieve effective feature fusion. This process not only promotes the deep integration of local fine-grained information and global context information, but also improves the model's performance in detecting distant and occluded targets. Finally, the Exponential Moving Average (EMA) attention mechanism [21] is added to further enhance the feature information of the pseudo image in all dimensions and aggregate the target feature.

The model is trained and the method is verified on the KITTI dataset [22], and the performance of the proposed SAE-PointPillars algorithm and the original PointPillars algorithm under the same conditions is compared for three different difficulty scenarios. Through the experimental comparison and visualization results, the effectiveness of the proposed algorithm is evaluated and verified from two perspectives.

II. PointPillars Algorithm

The PointPillars algorithm operates rapidly, and its core concept lies in converting point clouds into two-dimensional pseudo-images, followed by the utilization of two-dimensional convolutional neural network technology to carry out object recognition and bounding box regression on the image. The main procedures of the PointPillars algorithm can be divided into three parts: the Pillar Feature Network (PFN), the Backbone, and the SSD detection head.

The working principle of the Pillar Feature Network (PFN) is presented as follows: It transforms point clouds into pillars and partitions them into multiple Pillar units in a square pattern. Each Pillar unit is a small three-dimensional entity that is segmented from the point cloud in the Cartesian coordinate system (X-Y plane) in accordance with a specific stride, containing multiple points. Subsequently, the Pillar units are stacked for feature learning, and the features are mapped to a pseudo-image, facilitating the utilization of two-dimensional convolutional neural networks for feature extraction.

The working principle of the Backbone network is as follows: It employs a 2D convolutional neural network to carry out multiple downsampling operations on the pseudo-image, generating features with a gradually diminishing spatial resolution.Subsequently, it upsamples the downsampled features and concatenates them successively to generate the final feature map.

The working principle of the SSD [23] detection head lies in the fact that it takes the feature map processed by 2D convolution as input, conducts bounding box regression and object classification operations, thereby completing the object detection task in a single forward propagation. This detection head is capable of generating predictions regarding the position and type of objects by forecasting the offset and probability of multiple prior boxes at each location and combining the confidence score in the end.

The PointPillars algorithm is highly efficient and straightforward, being suitable for real-time systems and readily deployable in autonomous driving. Nevertheless, it demonstrates poor recognition performance for distant and occluded objects. The main reason is that the learning of point cloud features is confined to the local space of the

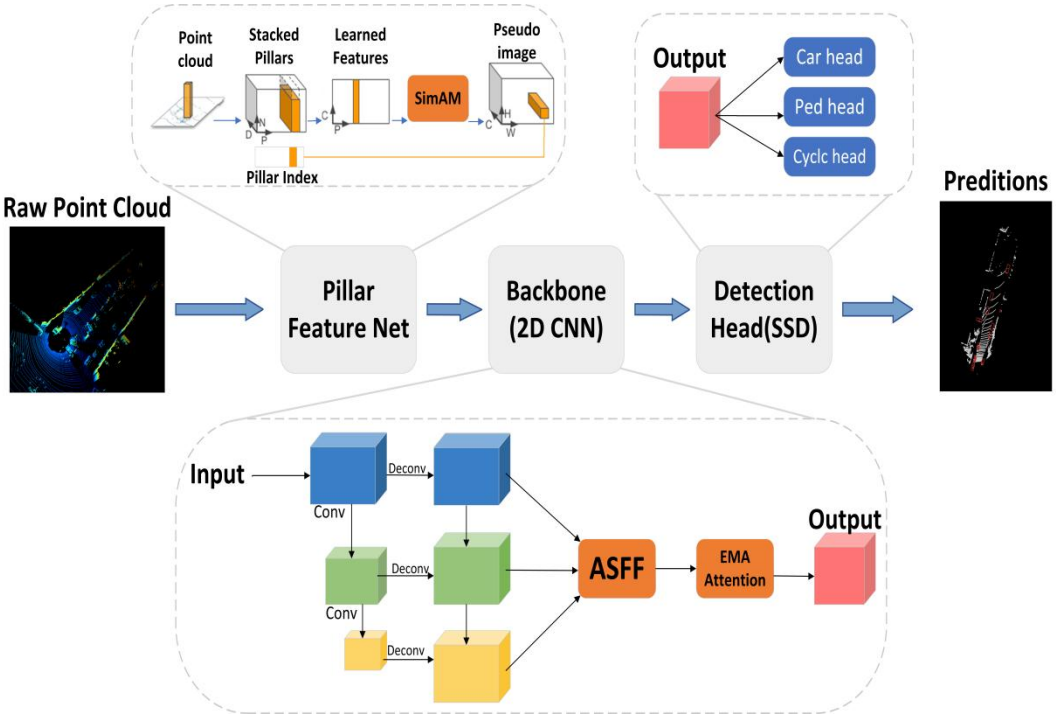

Fig. 1. SAE-PointPillars network architecture diagram

pillars, and the global context information of adjacent pillars cannot be exploited, thereby influencing the detection accuracy of the target. Meanwhile, the main trunk network employs traditional 2D CNN for feature extraction, which might overlook some crucial features and context information, leading to a weak feature extraction capability.

III. MPROVED POINTPILLARS MODEL

To tackle the issue of poor recognition of distant and occlusion targets with PointPillars, improvements were made to both the data input and backbone network module. The input of each Pillar feature in the model was processed through a SimAM attention mechanism module, allowing the network to pay more attention to valuable input features. Meanwhile, based on the ASFF space-adaptive network, the original backbone network RPN is refined to enhance the ability of feature fusion. Subsequently, the EMA attention

mechanism is introduced to further fortify the feature information. The improved model, namely SAE-PointPillars, has the following network structure as show in Fig. 1.

A. SimAM Attention Module

The SimAM attention module is a parameter-free 3D attention module based on a framework of optimizing energy functions. In the network layer, it derives 3D attention weights self-adaptively through the energy function, without adding extra parameters to the original network. This makes SimAM significantly faster in computational inference than traditional low-dimensional attention modules. Additionally, SimAM achieves function computation by applying 3D attention weights to each feature point and attaching a scalar to each feature, Thus, the final result possesses advantages such as global scope and flexibility.

Based on the definition of the optimized energy function, we derive the minimum energy function as in (1).

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t-\hat{u})^2 + 2\hat{\sigma}^2 + 2\lambda} \tag{1}$$

In this formula, t stands for the target neuron of the input characteristic; λ is a function parameter; and the parameter $\hat{u}$ is given by the following two formulas,as in (2) and (3).

$$\hat{u} = \frac{1}{M} \sum_{i=1}^{M} x_i \tag{2}$$

$$\hat{\sigma}^2 = \frac{1}{M} \sum_{i=1}^{M} (x_i - \hat{u})^2 \tag{3}$$

In this formula, x_i represents the adjacent neuron of the input characteristic target neuron.

According to Formulas (1) to (3), the smaller the energy of the feature neuron, the greater the difference between neuron t and its neighboring neurons, and the higher its importance. Finally, the feature is enhanced through Formula,as in (4).

$$\tilde{X} = \text{sigmoid}\left(\frac{1}{E}\right) \odot X \tag{4}$$

B. Improved 2D CNN Backbone Network

a. Feature Fusion Enhancement Module

In the 2D CNN backbone network of the PointPillars algorithm, the input features are downsampled multiple times to obtain features of different scales: (W/2) × (H/2) × C, (W/4) × (H/4) × 2C, (W/8) × (H/8) × 4C. This helps to retain the local features of the point cloud and expand the receptive field to capture contextual information. Then, upsampling is used to produce features of the same dimension (W/2) × (H/2) × 2C, which are used for more precise predictions or segmentation. Finally, the features of the same scale are concatenated together to form a feature information of (W/2) × (H/2) × 6C.

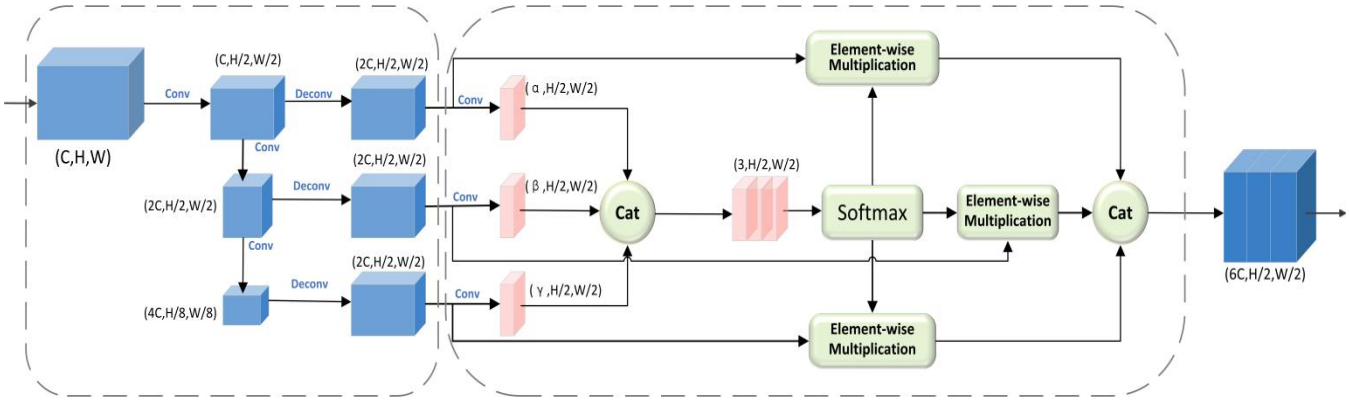

Fig. 2. Feature Fusion Enhancement Module

However, this direct feature fusion approach might result in information loss and subsequently give rise to missed detection issues. Therefore, this study incorporates the Adaptive Spatial Feature Fusion (ASFF) module into the CNN of PointPillars. The ASFF module is capable of adaptively adjusting the weights of features, thereby enhancing the accuracy and robustness of multi-scale object detection. The integration of the ASFF module is intended to improve the feature fusion process in the PointPillars algorithm and strengthen the system's ability to detect objects of different scales.

The schematic diagram of the feature fusion enhancement module based on the ASFF module is shown in Fig. 2. By inputting the three feature maps of the same dimension (W/2) × (H/2) × 2C, which are output from the up-sampling operation of the 2D CNN backbone network of PointPillars, into the two-dimensional convolution, we can obtain three feature weight vectors of size (W/2) × (H/2) × 1, which are then concatenated along the channel dimension to form a weight fusion map of size (W/2) × (H/2) × 3. By using the Softmax normalization along the channel dimension, we obtain three weight maps of size (W/2) × (H/2) × 1, which further establishes the mutual dependence between the three feature weight vectors to achieve adaptive feature fusion. By multiplying the weight maps with the corresponding features and element-wise operations, we can obtain new features. Finally, by concatenating and fusing these new features, we obtain a result of (W/2) × (H/2) × 6C. The normalizationfunction Softmax is represented by formula, as in (5).

$$\mathrm{Softmax}(\alpha,\beta,\gamma)=\left(\frac{e^{\alpha}}{e^{\alpha}+e^{\beta}+e^{\gamma}},\frac{e^{\beta}}{e^{\alpha}+e^{\beta}+e^{\gamma}},\frac{e^{\gamma}}{e^{\alpha}+e^{\beta}+e^{\gamma}}\right) \quad (5)$$

In this formula, α, β, and γ are different spatial weight vectors at the same location.

b. EMA Muti -scale Attention Module

By enhancing the fusion module with feature fusion, the feature map utilizes the EMA attention mechanism to enhance the feature connections among the channels. EMA uses a parallel subnetwork to effectively integrate features of different scales, which can consider both local details and global context simultaneously, thereby improving the ability to capture distant targets. By cross-channel interaction, the expression of distant target features is enhanced, while avoiding channel dimensionality reduction operations, which can retain the necessary detail information of distant targets, thereby improving the recognition accuracy for distant targets. Its network structure is shown in Fig. 3.

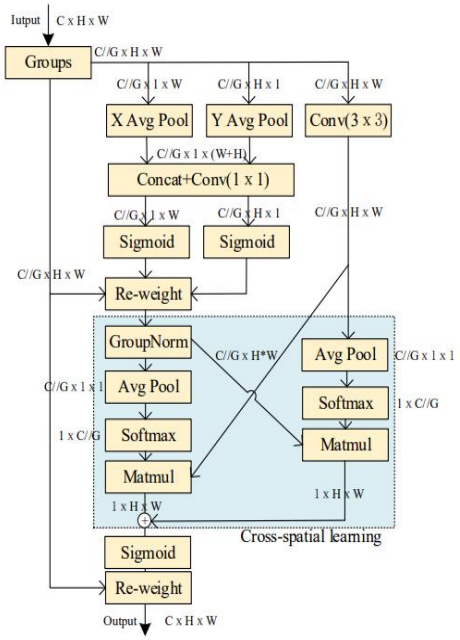

Fig. 3. EMA Attention Network Architecture Diagram

The input feature map will be divided into G sub-features along the channel dimension, each feature group is used to learn different semantic information as in (6).

$$X=[X_0,X_1,\ldots,X_{G-1}],X_i\in R^{C\times H\times W} \quad (6)$$

In this formula, $X\in R^{C\times H\times W}$ is the input feature vector, and G is the number of groups along the channel dimension.

The input is divided into three parts to capture the relationships between channels, similar to the CA attention mechanism [24]. The first two parts are combined from two spatial directions after one-dimensional average pooling, and then learned features are obtained through one-dimensional convolution. A Sigmoid function is used to generate a binomial distribution. In the third branch, only a 3 × 3 convolution is used to capture multi-scale features, so that the EMA can adjust the importance of different channels while accurately preserving spatial structure information in the channels. The calculation process is as follows:

$$Z_C^H(H) = \frac{1}{W} \sum_{0 \le i \le w} x_C(H, i) \quad (7)$$

$$Z_C^H(W) = \frac{1}{H} \sum_{0 \le i \le H} x_C(j, W) \quad (8)$$

$$x' = Conv_{3\times3}(x) \quad (9)$$

$$x'' = x \times Sigmoid(Conv_{1\times1}(Conact(Z_C^H, Z_C^W))) \quad (10)$$

In this formula, $Z_C^H(H)$ represents the pooling output of channel C at height H, W represents the width, and i denotes the position in the width dimension. $Z_C^W(W)$ represents the pooling output of channel C at height W, H represents the width, and j denotes the position in the width dimension. x' represents the output of the feature map after a 3 × 3 convolution, and x'' represents the output of the feature map after multiplying it by the normalization of the previous two channels.

In the end, the outputs of the first two branches and the output of the third branch will jointly participate in the cross-space feature learning process. The output features of one branch will first be processed by 2D average pooling, followed by activation through the Softmax function, and the spatial distribution weights obtained will be multiplied element-wise with the output features of the other branch x'. This step aims to fully utilize the global information and enhance the correlation between features. Next, the result after multiplication will be further processed through the Sigmoid function to generate spatial attention weights, which will be multiplied with the original features to obtain the output. The entire calculation process is as follows:

$$Z_C' = \frac{1}{H\times W} \sum_j^H \sum_i^W x_C'(i, j) \quad (11)$$

$$Z_C'' = \frac{1}{H\times W} \sum_j^H \sum_i^W x_C''(i, j) \quad (12)$$

$$x_{out} = Sigmoid((x'' \times Soft\max(Z'_c)) + (x' \times Soft\max(Z''_c))) \times x \quad (13)$$

In this formula, Z_C' and Z_C'' are respectively the previous outputs x' and x'', which are the pooling outputs of channel C at height H and width W.

IV. Experimental Results and Analysis

A. Experimental Data Set

In this paper, we selected the KITTI dataset as the experimental dataset. This dataset is a large-scale public one that encompasses not only abundant point cloud data but also image data, precisely reflecting diverse driving scenarios such as those in urban, rural, and highway areas. In each image, up to 15 vehicles and 30 pedestrians may emerge, accompanied by varying degrees of occlusion. During the model training phase, we exclusively utilized the point cloud data within the KITTI dataset and divided it into 7481 training samples and 7518 testing samples. Additionally, the 7481 training samples were further subdivided into 3712 training set samples and 3769 validation set samples. The detection task within this large-scale public dataset was retrieved from the evaluation server on the KITTI official website, featuring corresponding difficulty levels of Easy, Moderate, and Hard. The performance of the model was comprehensively evaluated by the following four evaluation indicators: 2D detection box precision (bbox), bird's eye view detection box precision (bev), 3D detection box precision (3d), and detection target rotation angle precision (aos). These indicators jointly constitute a comprehensive and detailed evaluation system for the model's performance.

In the evaluation framework of the KITTI dataset, Average Precision (AP) is the core metric for evaluating target detection performance. This metric measures the model performance by calculating the area under the precision-recall curve, which follows a strict spatial matching criterion: differentiated intersection and concurrency (IoU) thresholds are set for different target categories, and targets are classified into three difficulty levels, Easy,Moderate,Hard, based on their degree of occlusion, truncation ratio and height. Hard three difficulty levels. For the specific calculation, the detection results are first arranged in descending order of confidence, and the original PR curve is generated and then smoothed by 40-point interpolation, i.e., 40 sampling points are uniformly selected on the recall axis (1/40, 2/40, ... , 1), and take the maximum precision value on the right side of each point for averaging. The formula is as follows:

$$AP = \frac{1}{40} \sum_{K=0}^{40} \max_{r' \ge r_k} P(r'), r_k = \frac{k}{40} \quad (14)$$

B. Experimental Analysis

a. Experimental platforms

The experimental environment is based on the Ubuntu22.04 operating system, with a processor of Intel(R) Core(TM) i9-10940X, the GPU employs NVIDIA GeForce RTX 3090, featuring 24G of video memory. SAE-PointPillars constitutes an improvement of the PointPillars algorithm model based on the MMdet3d framework and is coded in Python 3.8.

During the training process, the Adam optimizer was selected, with a batch size of 2, a learning rate of 0.0002, a momentum term of 0.8, and a maximum iteration limit of 160.

b. Analysis of Losses

The PointPillars algorithm's loss function must be calculated separately for category classification and prior box direction classification. Each ground truth (GT) box in this algorithm contains seven parameters: (x, y, z, w, l, h, θ), representing length, width, height, the detection box's center coordinates in three-dimensional space, and the box's orientation.

The regression residuals, which are used in the loss calculation, are defined as follows:

$$\Delta_x = \frac{x^{gt} - x^a}{d^a}, \Delta_y = \frac{y^{gt} - y^a}{d^a}, \Delta_x = \frac{z^{gt} - z^a}{h^a} \quad (15)$$

$$\Delta_w = log\frac{w^{gt}}{w^a}, \Delta_l = log\frac{l^{gt}}{l^a}, \Delta_w = log\frac{h^{gt}}{h^a} \quad (16)$$

$$\Delta_\theta = sin(\theta^{gt} - \theta^{a}) \quad (17)$$

In the formula, x^{gt}, y^{gt}, z^{gt} represent the lengths of the bounding box along the x, y, z axes, respectively; x^{a}, y^{a}, z^{a} denote the lengths of the prior box along the x, y, and z axes; and d^{a} indicates the length of the diagonal distance of the prior box.

Given the above regression residuals, the total regression loss is calculated using the following formula:

$$L_{loc} = \sum_{b\in(x,y,z,w,l,h,\theta)} SmoothLI(\Delta b) \quad (18)$$

For the classification task of prior box categories, this algorithm employs Focal Loss to balance positive and negative samples. The calculation formula is as follows:

$$L_{cls} = -\alpha_\alpha (1 - p^{\alpha})^{\gamma} \log p^{\alpha} \quad (19)$$

In the equation, the parameters α and γ are two dynamic parameters, set here to 0.25, and 2, respectively.

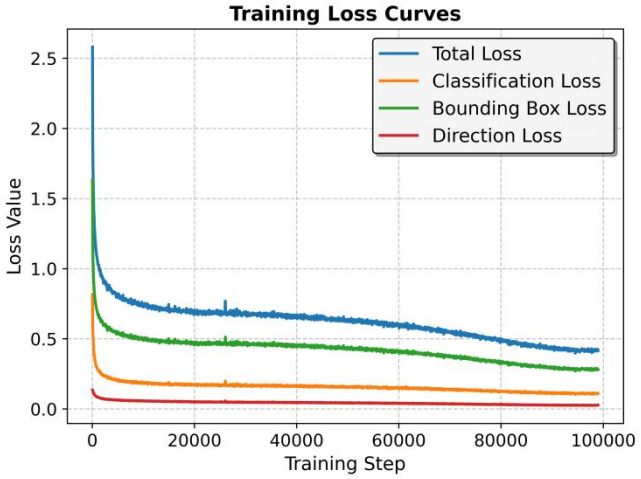

Fig. 4. Changes in PointPillars Loss

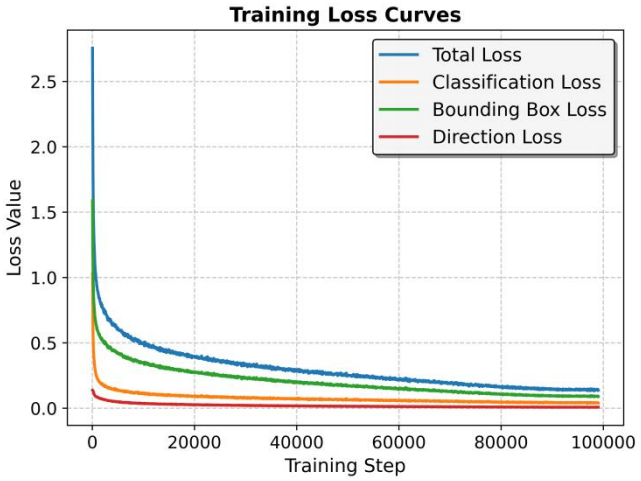

Fig. 5. Changes in SAE-PointPillars Loss

For the a priori frame direction classification task, the algorithm employs the Softmax function to predict direction category. The formula is as follows:

$$L = \frac{1}{N_{pos}}(\beta_{loc}L_{loc} + \beta_{cls}L_{cls} + \beta_{dir}L_{dir}) \quad (20)$$

In the equation, β_{loc} , β_{cls} , and β_{dir} are the coefficients of the loss function, which are 2, 1 and 0.2, respectively.

The changes in the classification loss, bounding box loss, direction loss, and total loss of the SAE-PointPillars algorithm and the PointPillars algorithm during training are shown in detail in Fig. 4 and Fig. 5. Comparing these two figs, shows that the SAE-PointPillars algorithm in this paper has stronger feature learning ability.

C. Controlled Experiment

To assess the detection performance of the proposed algorithm, the official KITTI test set was employed for evaluation. The detection outcomes of representative mainstream object detection algorithms were chosen for comparison, encompassing the algorithm F-PointNets that integrates point clouds and images, the voxel-based algorithm VoxeNet, the point cloud-based algorithm PointRCNN, as well as SECOND and TANet. The detection results are presented in TABLE I, which contains the comparison of the detection accuracy of the proposed algorithm with other algorithms in the Car, Pedestrian, and Cyclist classes of the KITTI test set.

Based on the data in TABLE I, the algorithm proposed in this paper exhibits a higher average precision (AP) on the KITTI dataset compared with other mainstream 3D object detection algorithms. Under diverse sample difficulties, the AP of Car was enhanced by 3.37%, 3.32%, and 1.56%; the AP of Pedestrian was elevated by 2.85%, 5.35%, and 3.97%; and the AP of Cyclist was increased by 3.27%, 4.13%, and 5.36%. These results amply illustrate that on the foundation of the PointPillars algorithm, the introduction of the SimAM attention mechanism for enhancing voxel-based feature input, the improvement of the backbone network structure based on the adaptive spatial feature fusion module (ASFF), and the utilization of the EMA method to boost the effectiveness of feature information have enhanced the performance of the algorithm in 3D object detection tasks.

D. Ablation Experiment

Ablation experiments are designed to demonstrate the effectiveness of the proposed modules. Each group of models is trained on the KITTI dataset for 160 rounds, and the average detection accuracy (AP) is chosen as the evaluation metric. The evaluated modules are the improved voxelised feature input module (SimAM), the improved backbone network module by applying spatial adaptive feature fusion (ASFF), and the feature information enhancement module (EMA), respectively. All experiments utilise pre-trained weights and ensure that the optimiser and loss function are of the same class. TABLE II illustrates the AP comparison of the detection results of each module in the KITTI dataset.

The results of the detection of the SimAM module alone and of the original model demonstrate that the SimAM module generates the attention weights by calculating the local self-similarity of the feature maps. This is a parameter-free attention mechanism that automatically

TABLE I
COMPARISON OF AP BETWEEN DIFFERENT ALGORITHMS(%)

Model	Car			Pedestrian			Cyclist		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
F-PointNets[25]	83.76	70.92	63.65	70.00	61.32	53.59	77.15	56.49	53.37
VoxeNet[26]	85.10	72.54	70.38	63.65	59.36	54.71	79.36	60.39	53.3
SECOND[27]	85.78	75.79	74.39	51.84	45.86	39.48	82.64	65.1	58.34
TANet[28]	85.34	74.92	72.48	66.64	59.29	54.06	86.88	66.96	63.24
PointRCNN[29]	86.18	75.94	75.27	58.53	51.20	47.44	86.15	67.04	61.50
PointPillars	85.1	75.30	73.78	63.56	56.52	51.17	85.96	66.19	61.53
SAE-PointPillars	**88.47**	**78.62**	**75.34**	**66.41**	**61.87**	**55.14**	**89.23**	**70.32**	**66.89**

TABLE II
COMPARISON OF AP WITH DIFFERENT MODULES ADDED(%)

	Car			Pedestrian			Cyclist		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
PointPillars	85.1	75.30	73.78	63.56	56.52	51.17	85.96	66.19	61.53
PointPillars+SimAm	86.99	76.25	74.27	64.32	58.49	52.18	86.59	68.52	62.98
PointPillars+ASFF	87.11	77.58	74.55	65.73	57.63	53.45	87.48	67.75	64.78
PointPillars+EMA	86.19	77.82	74.19	64.56	59.41	52.49	87.59	68.72	62.86
SAE-PointPillars	88.47	78.62	75.34	66.41	61.87	55.14	89.23	70.32	66.89

derives the attention weights by optimising the energy function without adding additional parameters to the network. This reduces the complexity of the model and the risk of overfitting. The AP enhancements for Car, Pedestrian, and Cyclist are (1.89%, 0.95%, 0.49%),(0.76%, 1.97%, 1.01%) and (0.63%, 2.33%, and 1.45%)for the easy , moderate, and hard samples, respectively.

The detection results of the improved backbone network module by adding spatial adaptive feature splicing fusion (ASFF) and the detection results of the original model show that the ASFF module is able to adaptively adjust the fusion weights of features from different scales according to the spatial information in the input feature map. The accuracy and robustness of target detection are improved by constructing the feature weight matrix and performing Softmax normalisation. The AP enhancements for Car, Pedestrian, and Cyclist are (2.01%, 2.28%, 0.77%), (2.17%, 1.11%, 2.28%), and (1.52%, 1.56%, 3.25%) for the easy , moderate, and hard samples.

The detection results of the EMA module and the original model detection results shows that the EMA attention module enhances the association between feature dimensions through cross-channel interaction, effectively amplifies the response strength of key features, and at the same time tracks the dynamic changes of model parameters through the exponential moving average mechanism to improve the stability of the model. The AP enhancements for Car, Pedestrian, and Cyclist are (1.09%, 2.52%, 0.41%), (1.00%, 2.89%, 1.32%), and (1.63%, 2.53%, 4.33%) for the easy, moderate, and hard samples.

The three modules are then added together to enable a comparison with the original model. The AP improvement for the Car, Pedestrian, and Cyclist samples are (3.37%, 3.32%, 1.56%), (2.85%, 5.35%, 3.97%), and (3.27%, 4.13%, 5.36%) for the easy, moderate, and hard samples. The experimental results demonstrate that the addition of individual modules enhances the detection accuracy. Furthermore, the incorporation of all modules surpasses the accuracy achieved through the standalone addition of each module, substantiating the efficacy of each module and indicating minimal intermodule interference.

E. Visual Analysis

In order to visually assess the performance of the target detection algorithm, this study visualizes and compares the detection results of the SAE-PointPillars algorithm with the baseline PointPillars algorithm on the KITTI dataset (shown in Fig. 6-Fig. 11). Three typical scenarios were selected for validation, and the following labeling methods were used by intercepting the point cloud data from a bird's-eye view: red circles mark the missed targets in the original image, red arrows indicate the occluded detected targets, and yellow circles mark the corresponding targets in the bird's-eye view. In scenario (a), the original PointPillars algorithm is unable to effectively detect the vehicle at the greatest distance, resulting in a missed detection. In contrast, the SAE-PointPillars algorithm markedly enhances the situation by successfully and accurately detecting the vehicle at the greatest distance. This improvement demonstrates that the SAE-PointPillars algorithm has enhanced both the feature extraction and the judgement processes in challenging scenarios.

Scenario (a)

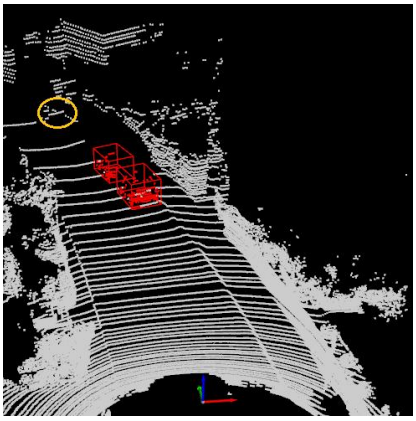

Fig. 6 Results of PointPillars

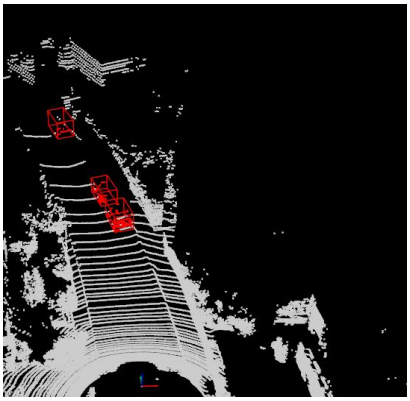

Fig. 7 Results of SAE-PointPillars

In scenario (b), SAE-PointPillars also exhibits its superior detection capability, effectively identifying targets that are obscured by other objects. In this scene, the original algorithm may fail to detect some crucial targets due to the presence of occluded objects. However, the enhanced SAE-PointPillars model demonstrates enhanced flexibility in feature learning and information fusion, thereby improving its ability to detect targets in complex environments.

Scenario (b)

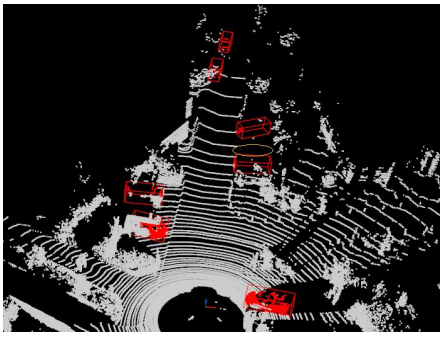

Fig. 8 Results of PointPillars

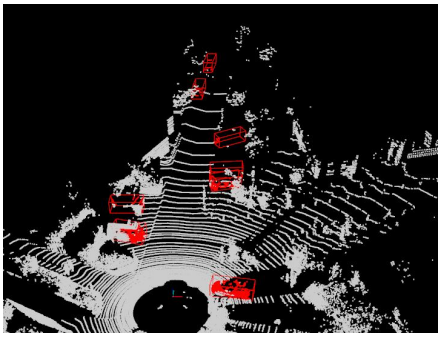

Fig. 9 Results of SAE-PointPillars

In scenario(c), the SAE-PointPillars algorithm demonstrates even greater capabilities in accurately identifying targets that are distant and partially obscured by the vehicle in front. While the original PointPillars algorithm may be unable to detect vehicles due to occlusion in this complex situation, the introduction of the attention mechanism to SAE-PointPillars enhances the algorithm's sensitivity to key features, thus ensuring improved detection accuracy.

Scenario (c)

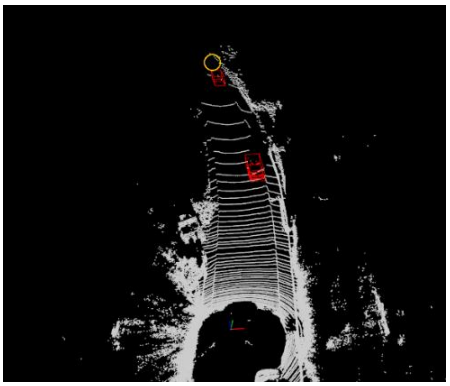

Fig. 10 Results of PointPillars

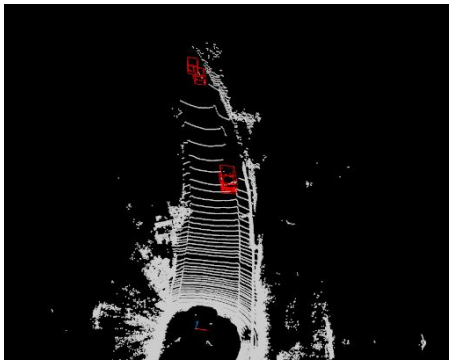

Fig. 11 Results of SAE-PointPillars

V. Conclusion

In order to address the issue of poor detection of distant and occluded targets under the 3D target detection algorithm for self-driving road scenes, the PointPillars algorithm has been enhanced and the SAE-PointPillars algorithm has been proposed. Firstly, in the point cloud voxelization processing stage, the SimAM attention mechanism is introduced to generate learnable weight scalars for each feature channel by constructing an energy function, thereby reducing model parameters while enhancing global feature representation capabilities. Secondly, the backbone network is improved based on the adaptive spatial feature fusion module ASFF, which improves the feature extraction and fusion ability of the network and solves the information loss problem of feature splicing. Finally, the EMA attention mechanism is embedded in the feature decoding stage to amplify the response intensity of critical features through cross-channel interactions.The experimental results show that the proposed SAE- PointPillars have improved 3D accuracy compared to the original algorithmic model under Easy, Moderate, and Hard difficulties, with average accuracy improvements in Car, Pedestrian, and Cyclist categories of (3.37%, 3.32%, and 1.56%), (2.85%, 5.35% and 3.97%) and (3.27%, 4.13% and 5.36%), respectively. The visualisation results show that the SAE-PointPillars algorithm can effectively detect distant as well as occluded targets.

The SAE-PointPillars algorithm proposed in this study has made significant advancements in enhancing the detection performance of distant and occluded targets in autonomous driving environments by optimising the PointPillars algorithm. The introduction of the SimAM and EMA attention mechanisms and the ASFF adaptive spatial feature fusion module not only optimises the feature input and information fusion process, but also improves the model's ability to learn key features. These improvements enable SAE-PointPillars to demonstrate better performance than the traditional PointPillars algorithm at multiple difficulty levels. Therefore, this study provides new ideas and methods for target detection in complex scenes for future autonomous driving systems, which has important theoretical value and application prospects.

References

[1] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li,"Deep learning for LiDAR point clouds in autonomous driving: A review," IEEE Trans. Neural Netw. Learn. Syst., vol. 32, pp. 3412–3432, 2020.

[2] X. Ma, W. Ouyang, A. Simonelli, and E. Ricci, "3D object detection from images for autonomous driving: A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 46, pp. 3537–3556, 2023.

[3] A. Singh, "Transformer-based sensor fusion for autonomous driving: A survey, " in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV) Workshops, Paris, France, Oct. 2–6, 2023, pp. 3312–3317.

[4] Z. T. Yang, Y. N. Sun, S. Liu et al., "3DSSD: Point-based 3D single stage object detector," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 11037–11045.

[5] S. S. Shi, C. X. Guo, L. Jang et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 10526–10535.

[6] B. Yang, W. J. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7652–7660.

[7] J. Mao, S. Shi, X. Wang, H. Li, "3D object detection for autonomous driving: A comprehensive survey," Int. J. Comput. Vis., vol. 131, pp. 1909–1963, 2023.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," Adv. Neural Inf. Process. Syst., vol. 30, pp. 5105–5114, 2017.

[9] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3D object detection with Pointformer," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Nashville, TN, USA, Jun. 20–25, 2021, pp. 7463–7472.

[10] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Seattle, WA, USA, Jun. 13–19, 2020, pp. 1711 –1719.

[11] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA, Jun. 15–20, 2019, pp. 12697–12705.

[12] T. Yin, X. Zhou, and P. Krahenbuhl, " Center-based 3D object detection and tracking, " in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Nashville, TN, USA, Jun. 20–25, 2021, pp. 11784–11793.

[13] J. Mao, Y. Xue, M. Niu, H. Bai, J. Feng, X. Liang, H. Xu, and C. Xu, "Voxel transformer for 3D object detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis., Montreal, QC, Canada, Oct. 10 – 17, 2021, pp. 3164–3173.

[14] C. He, H. Zeng, J. Huang, X. S. Hua, and L. Zhang, "Structure aware single-stage 3D object detection from point cloud," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Seattle, WA, USA, Jun. 13–19, 2020, pp. 11873–11882.

[15] Z. Miao, J. Chen, H. Pan, R. Zhang, K. Liu, P. Hao, J. Zhu, Y. Wang, and X. Zhan, "PVGNet: A bottom-up one-stage 3D object detector with integrated multi-level features, " in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Nashville, TN, USA, Jun. 20–25, 2021, pp. 3279–3288.

[16] Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang, "RangeDet: In defense of range view for LiDAR-based 3D object detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis., Montreal, QC, Canada, Oct. 10–17, 2021, pp. 2918–2927.

[17] Y. Chai, P. Sun, J. Ngiam, W. Wang, B. Caine, V. Vasudevan, X. Zhang, and D. Anguelov, "To the point: Efficient 3D object detection in the range image with graph convolution kernels, " in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Nashville, TN, USA, Jun. 20–25, 2021, pp. 16000–16009.

[18] P. Sun, W. Wang, Y. Chai, G. Elsayed, A. Bewley, X. Zhang, C. Sminchisescu, and D. Anguelov, "RSN: Range sparse net for efficient, accurate LiDAR 3D object detection, " in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Nashville, TN, USA, Jun. 20–25, 2021, pp. 5725–5734.

[19] L. Yang, R. Y. Zhang, L. Li et al., "Simam: A module for simple, parameter-free attention convolutional neural networks," in Proc. Int. Conf. Mach. Learn., PMLR, 2021, pp. 11863–11874.

[20] S. Liu, D. Huang, and Y. Wang, " Learning spatial fusion for single-shot object detection, " arXiv preprint arXiv:1911.09516v2, 2019.

[21] D. Ouyang, S. He, G. Zhang et al., "Efficient multi-scale attention module with cross-spatial learning," in Proc. ICASSP 2023-2023 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), IEEE, 2023, pp. 1–5.

[22] A. Geiger, P. Lenz, C. Stiller et al., "Vision meets robotics: The KITTI dataset," Int. J. Robot. Res., vol. 32, no. 11, pp. 1231–1237, 2013.

[23] W. Liu, D. Anguelov, D. Erhan et al., "SSD: Single shot MultiBox detector," in Computer Vision-ECCV 2016, B. Leibe, J. Matas, N. Sebe et al., Eds., Lecture Notes in Computer Science, Springer, 2016, vol. 9905, pp. 21–37.

[24] Q. B. Hou, D. Q. Zhou, and J. S. Feng, "Coordinate attention for efficient mobile network design," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Los Alamitos: IEEE Computer Soc, 2021, pp. 13708–13717.

[25] C. R. Qi, W. Liu, C. Wu et al., "Frustum pointnets for 3D object detection from RGB-D data," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Piscataway: IEEE, 2018, pp. 918–927.

[26] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Piscataway: IEEE, 2018, pp. 4490–4499.

[27] Y. Yan, Y. Mao, and B. Li, " SECOND: Sparsely embedded convolutional detection," Sensors, vol. 18, no. 10, pp. 3337 – 3353, 2018.

[28] Z. Liu, X. Zhao, T. Huang et al., "Tanet: Robust 3D object detection from point clouds with triple attention," in Proc. AAAI Conf. Artif. Intell., Palo Alto: AAAI, 2020, pp. 11677–11684.

[29] S. Shi, X. Wang, and H. Li, " PointRCNN: 3D object proposal generation and detection from point cloud," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Piscataway: IEEE, 2019, pp. 770–779.

YUXUAN ZHANG was born in Hebei Province, P. R. China, received the B.S. degree in Communication Engineering from University of Science and Technology Liaoning, Anshan, P. R. China, in 2026.

He is currently pursuing the M.S. degree in Electronic Information with University of Science and Technology Liaoning, Anshan, P. R. China. He research interest is artificial intelligence.

ZIWEI ZHOU (1974), male, from Anshan, Liaoning, associate professor, master's supervisor, received bachelor's and master's degrees from Liaoning University of Science and Technology in 1997 and 2007, respectively; Ph.D. from Harbin Institute of Technology in 2013, with main research directions in artificial intelligence, 3D vision, deep learning and robotic system research. Email: 381431970@qq.com.