PAEN: Efficient Pillar-based 3D Object Detector Based on Attention and Dilated Convolution

Jia Wen[1,2], Guanghao Zhang[1,2(✉)], Qi Zhang[1,2], Kelun Tian[3],and Kejun Ren[1,2]

[1] School of Information Science and Engineering, Yanshan University,
Qinhuangdao 066004, China
[2] The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Provinc
[3] HeBei Construction Material Vocational and Technical College
1349786296@qq.com

Abstract. The Pillar-based 3D object detector can complete the scene-sensing task efficiently and quickly, meeting the basic real-time detection needs of the automatic driving sensing module. In this paper, we propose a Pillar Sequence Attention Encoder and Dilated Expansion Convolution Network. The former addresses issues of coarse encoding methods and limitations in encoding information during the pillar encoding stage, while the latter tackles the problem of insufficient receptive fields in the backbone network. We conducted experiments on the KITTI dataset to validate the performance of our model and the effectiveness of the proposed modules. Experiments show that our method achieved a mean average precision(mAP) of 81.48% for the car category, surpassing the baseline model by 3.12%, while the inference time only increases by about 10ms.

Keywords: 3D object detection, LiDAR, Pillar detector, Attention module, Dilated convolution.

1 Introduction

3D object detection based on LiDAR point clouds finds widespread applications in fields such as autonomous driving perception and robotics, drawing considerable attention. There have been many studies that have achieved great success in utilizing point cloud data for 3D object detection. These methods can be categorized into approaches based on raw points and methods based on regularization processing. Previous research often suggests that approaches utilizing unprocessed point cloud data can harness the full information of the original point clouds, leading to superior performance. Conversely, methods that employ structured data processing with efficient convolutional modules tend to offer faster detection speeds. We decided to explore methods based on regularization processing, specifically focusing on those pillar-based approach.

Pillar-based detection methods usually include the following basic processes: pillar feature encoding, backbone network feature extraction, neck network feature aggregation, and detection head for detection. In the pillar feature encoding phase, recent methods have employed averaging or max-pooling methods on the raw point cloud features within the pillar to obtain pillar features. However, these methods lack exploration and learning of the original point cloud features, resulting in relatively coarse pillar features. In the feature extraction stage of the backbone network, previous methods commonly employed sparse convolution for multi-level feature extraction. However, pillar size limitations and the sparse convolution kernel frequently result in an insufficient receptive field for backbone networks to meet precise detection requirements.

Based on the above analysis, in the pillar feature encoding stage, we propose the Pillar Sequence Attention Feature Encoder. This encoder is primarily composed of two modules: the Pillar-based Sequence Attention (PSA) module and the Pillar Feature Soft Aggregate (PFSA) module. The PSA module capture attention information among points in the local region of the pillar. The PFSA module finely aggregate information from points within the pillar. In the feature extraction section of the backbone network, we propose the Dilated Expansion Convolution Network, which is mainly composed of two convolution modules: Sparse Dilated Expansion Convolution (SDEC) and Dense Dilated Expansion Convolution (DDEC). SDEC and DDEC leverage dilated convolutions to capture feature information with both sparse and dense in wide-ranging receptive fields. Integrating the previously mentioned encoder and backbone network, we construct the Pillar-based Attention Expansion Network(PAEN). Our model achieves competitive performance on the KITTI[1] dataset.

Our work makes the following contributions:

1. We propose a Pillar Sequence Attention Feature Encoder to finely aggregate correlated feature information between points within the pillar. This addresses limitations in encoding information and the issue of coarse encoding methods.
2. We propose a Dilated Expansion Convolution Network that efficiently utilizes dilated convolutions to capture feature information over a large receptive field. This resolves the issue of insufficient receptive fields in traditional convolutional backbone networks.
3. PAEN achieves a notable enhancement in accuracy while maintaining the capability for real-time detection, providing a novel approach for exploring pillar-based 3D object detection models.

2 Related Works

According to the representation form of point cloud data, LiDAR-based 3D object detection methods can be broadly categorized into point-based methods and methods based on regularised processing.

Point-based methods: Point-based methods originated from PointNet[2]. These methods such as PointRCNN[3] and Lidar-RCNN[4] divided the point cloud space into equally sized clusters and obtained local features at the cluster level. They accomplished feature extraction through sampling and aggregation methods.

Regularized processing methods: Regularized processing methods can be divided into voxel-based processing methods and pillar-based processing methods.

VoxelNet[5] was the first to propose regularizing sparse point cloud space into voxels. Specifically, these methods such as SECOND[6] first divided the discrete point cloud into regular voxel grids, using grid features to replace point cloud features within each grid. Pillar-based methods were represented by PointPillars[7]. These methods such as PillarNet[8] and Pillarnext[9] mapped sparse point cloud data to pillars, forming regular BEV (bird's eye view) pseudo-images.

3 Proposed Methods

In this section, we will provide a detailed explanation of the proposed PAEN model. The overall structure of PAEN is shown in Fig.1. Firstly, we employ the Pillar Sequence Attention Feature Encoder to efficiently obtain attention features between points within pillars. Subsequently, we implement a Dilated Expansion Convolution Network to capture extensive contextual information. Finally, we utilize a traditional Neck Network to perform feature fusion tasks, followed by employing a classic anchor box detection head for object detection and classification.

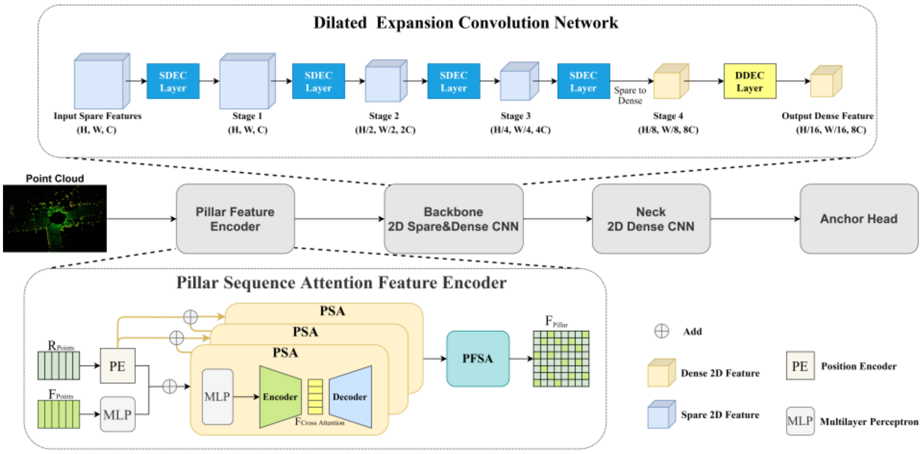

Fig. 1. Illustrates the overall structure of the PAEN model.

3.1 Pillar Sequence Attention Feature Encoder

3.1.1 Pillar Sequence Attention Module

The structure of the PSA is depicted in Fig.2. The PSA module comprises two levels of cross-attention. The original point cloud feature data undergoes the first-level cross-attention in the encoder to obtain cross-attention features, accomplishing the task of unifying the length of the pillar sequence. The cross-attention features indirectly derive point-wise self-attention feature information through the second-level cross-attention in the decoder.

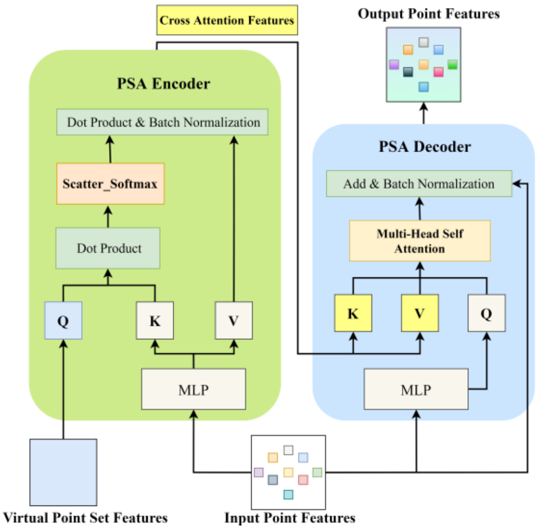

Fig. 2. Illustrates the structure of the Pillar Sequence Attention module.

Encoder. In the encoder, we first obtain the input point cloud feature information, denoted as F^{in}, which consists of both the point cloud feature information F=(x,y,z,r) and the relative grid position information R=(x',y',z').

$$F^{in} = \mathrm{MLP(F)} + \mathrm{PE(R)} \tag{1}$$

$$R = \left(P_i - (C_i \times P_{size})\right)/P_{size} \tag{2}$$

Here, P_i represents the global offset distance, C_i denotes the grid index, and P_{size} represents the grid size of the pillar. PE is the position encoding operation, and MLP refers to the multi-layer perceptron.

Next, the point cloud feature, denoted as F^{in} is processed through a MLP to derive K vectors and V vectors, and the learnable fixed-length virtual point set features are used as Q vectors for cross-attention modeling. The virtual point set feature can be regarded as several fixed-dimensional virtual points in the point cloud. By using the pillar index of each point as the index, the scatter_softmax operation restricts the attention weights of each virtual point to the original points within each pillar. This results in attention feature information with the same dimensionality, referred to as Cross-Attention Features (CAF). We use the cross-attention features with the same feature dimensions as the pillar sequence features. In this case, the dimensionality of the cross-attention features corresponds to the sequence length. This resolves the issue of inconsistent sequence lengths in attention modeling.

$$Q = F^{v},\ K = MLP(F^{in}),\ V = MLP(F^{in}) \tag{3}$$

$$CAF = CrossAttention(Q, K, V) \tag{4}$$

$$CrossAttention = SoftMax_{scatter}(QK^{T})V \tag{5}$$

Decoder. The PSA decoder utilizes the CAF as the K and V vectors and the original point cloud feature information as the Q vector to perform cross-attention modeling again. This process indirectly obtains the self-attention feature information (SF) between each point within the pillars of the original point cloud.

$$\mathrm{Q} = \mathrm{MLP}(\mathrm{F}^{\mathrm{in}}), \qquad \mathrm{K} = \mathrm{CAF}, \qquad \mathrm{V} = \mathrm{CAF} \tag{6}$$

$$\mathrm{SF} = \mathrm{CrossAttention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) \tag{7}$$

$$\mathrm{CrossAttention} = \frac{\mathrm{SoftMax}(\mathrm{QK}^{\mathrm{T}})\mathrm{V}}{\sqrt{\mathrm{d}}} \tag{8}$$

On one hand, the decoder further enhances the cross-attention features, indirectly obtaining the self-attention features between points within the pillar sequence. On the other hand, it restores the dimensionality of the self-attention features to the dimensionality of the original input point cloud features. This design enables the PSA module to be layered repeatedly, similar to a Transformer, without altering the feature dimensions of the input data. This stacking progressively strengthens the data's representational power, effectively addressing the issue of limited encoded information.

3.2.2 Pillar Feature Soft Aggregation Module

Next, we designed the PFSA module to fully utilize the point feature information with strong representational capabilities. As shown in Fig.3, the PFSA module takes the PSA point cloud features as input and utilizes scatter_softmax function to obtain the weight information of each point within the corresponding pillar. Subsequently, the PSA point cloud features are weighted by the weight information of each point through element-wise multiplication.

$$\mathrm{W}^{\mathrm{P}} \; = SoftMax_{scatter}(F^{PSA}) \tag{9}$$

$$\mathrm{F}^{\mathrm{PFSA}} = \; F^{PSA} \; * \mathrm{W}^{\mathrm{P}} \tag{10}$$

$$\mathrm{PF}^{\mathrm{PFSA}} \; = \; \mathrm{Add}_{\mathrm{scatter}}(\mathrm{F}^{\mathrm{PFSA}}) \tag{11}$$

Here $\mathrm{F}^{\mathrm{PSA}}$ represents the PSA point cloud features, W^{P} represents the weight information of each point within its corresponding pillar, $\mathrm{SoftMax}_{\mathrm{scatter}}$ denotes the scatter softmax operation, $\mathrm{F}^{\mathrm{PFSA}}$ represents the weighted point cloud feature information, $\mathrm{Add}_{\mathrm{scatter}}$ indicates the scatter addition operation, and $\mathrm{PF}^{\mathrm{PFSA}}$ represents the final obtained pillar features.

PSA point cloud features represent the attention level between points within the pillar, where foreground points have higher feature intensities. This weighting approach reduces the contribution of background points in the pillar features, enhancing the contribution of foreground points. Finally, the pillar feature information is obtained by PFSA through a scatter-add operation that aggregates data from all points within the pillar. The PFSA module refines the inter-point attention information provided by the PSA module, further refining and fusing the point cloud feature information through a filtering and selection mechanism. It fully utilizes the powerful expressive point cloud feature information provided by the PSA module, effectively solving the problem of rough aggregation methods.

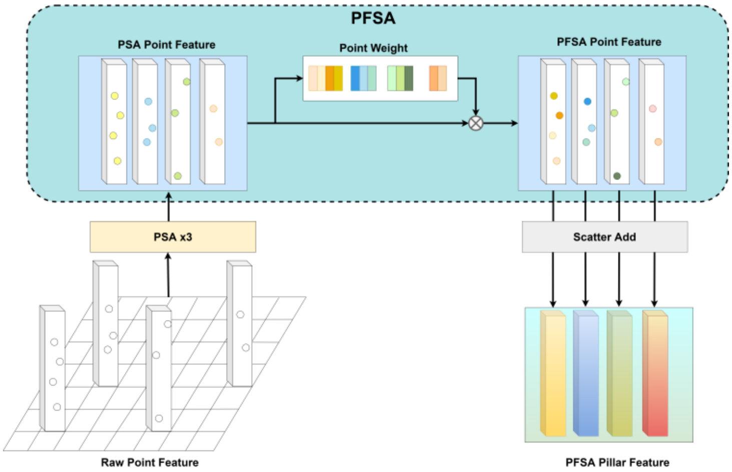

Fig. 3. Illustrates the structure of the Pillar Feature Soft Aggregation module.

3.2 Dilated Expansion Convolution Network

In previous studies in the fields of semantic segmentation and object detection, such as Deeplabv3[10] and YOLOF[11] models, dilated convolutions have been widely used to effectively expand the model's receptive field . As illustrated in Fig.4, convolution kernel with a dilation rate of 1 can handle central local information, while rates greater than 1 enlarge the receptive field. We employ an Inception-like structure to combine the convolutional results of kernels with different dilation rates. This approach preserves the original model's receptive field to cover smaller object sizes while leveraging convolutions with larger dilation rates to expand the receptive field. We refer to this structure as dilated expansion convolution.

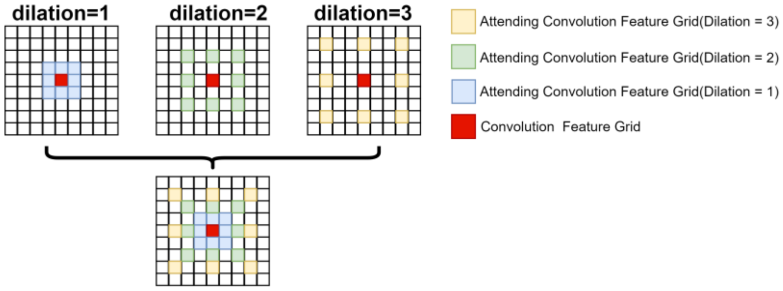
dilation=1
dilation=2
dilation=3
Attending Convolution Feature Grid(Dilation = 3)
Attending Convolution Feature Grid(Dilation = 2)
Attending Convolution Feature Grid(Dilation = 1)
Convolution Feature Grid

Fig. 4. Illustrates the receptive field of dilated expansion convolution.

SDEC processes sparse 2D feature data and is composed of multiple layers of sparse dilated expansion convolution. Each layer consists of a sparse convolution block and two sparse dilated expansion convolution blocks. The sparse convolution block primarily used for downsampling sparse 2D feature maps. As shown in Fig.5, the sparse dilated expansion convolution block consists of two layers of multiple parallel submanifold sparse convolutions. The sparse feature map is initially subjected to sparse convolution operations at different receptive field ranges. Subsequently, the convolutional feature maps from multiple receptive field sizes are combined with the original input feature map to obtain the dilated expansion feature map. We refer to this operation as dilated expansion convolution processing. Next, the obtained dilated expansion features undergo further dilated expansion convolution processing to further extract sparse features.

The DDEC structure is similar to the SDEC, but as it deals with dense 2D feature data, the DDEC block consists of traditional 2D convolutions. As shown in Fig.5, the dense dilated expansion convolution block adopts the same structure as the sparse dilated expansion convolution block, replacing the submanifold sparse convolution with ordinary convolution, performing 2D convolution operations at different receptive field ranges.

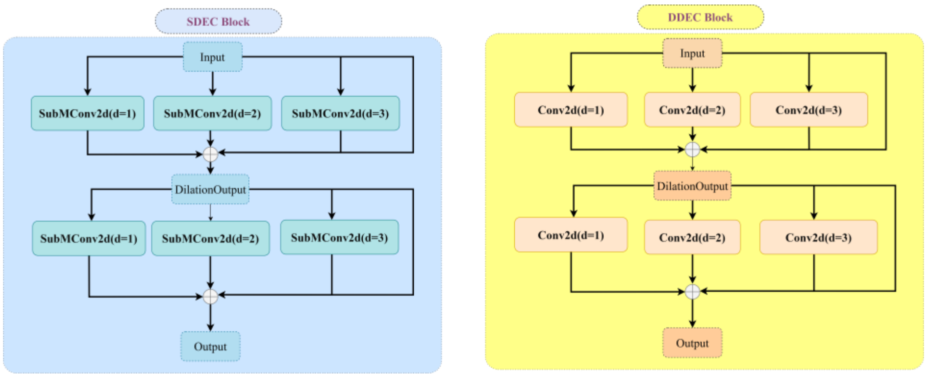

Fig. 5. Illustrates the Dilated Expansion Convolution Blocks.

3.3 Neck Network Feature Aggregation

In PAEN, the neck network adopts the approach used in previous methods like VoxelNet[5] and SECOND[6]. It obtains dense feature maps with downsampling strides of 8 and 16 from the backbone network. By employing convolution and transposed convolution techniques, feature maps with varying resolutions are processed to yield feature maps of a uniform size. Finally, the two processed feature maps are stacked along the channel dimension. The neck network aggregates deep holistic abstract feature information and shallow local fine information generated by the backbone network, providing more powerful feature information for the detection head.

3.4 Detection Head and Loss Function

For the detection head and loss function, we follow the classical anchor-based detection head. The loss function employs the same methodology as in previous models such as VoxelNet[5], SECOND[6], and PointPillars[7]. It includes the following components: 1)Classification Loss (L_{cls}): Formed by the Focal loss[13], this component addresses the classification loss. 2)Regression Loss (L_{reg}): Constructed using the Smooth L1 loss, this component addresses the regression loss. 3)Orientation Loss (L_{dir}): Comprising the Cross-Entropy loss, this component handles the orientation loss.

The regression loss (L_{reg}) is computed for the deviations between the predicted target center point, predicted box dimensions, and predicted orientation angle, compared to the corresponding ground truth labels, i.e.,the regression value between [x_a, y_a, z_a, l_a, w_a, h_a, θ_a] and [x_g, y_g, z_g, l_g, w_g, h_g, θ_g].

The regression loss formula is defined as follows:

$$L_{reg} = \sum_{r \in [x,y,z,l,w,h,\theta]} \mathrm{SmoothL1}(\Delta r) \tag{12}$$

The categorical loss formula is defined as follows:

$$L_{cls} = -\alpha_t (1 - P_t)^{\gamma} \log(P_t) \tag{13}$$

Where P_t is the confidence information obtained by the model for this prediction frame, and are hyperparameters, which we set as in the previous method,α=0.25,γ=2, respectively.

The total loss function for training is as follows:

$$L = \beta_{cls} L_{cls} + \beta_{reg} L_{reg} + \beta_{dir} L_{dir} \tag{14}$$

We employ a similar strategy to previous methods, assigning proportional weights to the three losses in the overall loss function: β_{cls}=1、 β_{reg} = 2 and β_{dir}= 0.2.

4 Experiments

4.1 Dataset Introduction

As a dataset for 3D object detection in autonomous driving, KITTI[1]primarily focuses on detecting object categories include cars, pedestrians, and cyclists. Detection difficulty is categorized into easy, moderate, and hard based on factors such as the distance of the object in the scene and whether it is occluded. The calculation of Average Precision (AP) for the final detection results on the KITTI dataset is divided into two methods: R11 and R40, based on the number of recall points set.

4.2 Implementation Details

PAEN employs the Adam[14] optimizer for end-to-end training. During training, the initial learning rate is set to 0.003, and a one-cycle policy is used for updates, with a weight decay of 0.01. On the KITTI dataset, we train for 80 epochs using one NVIDIA RTX 4090 with a batch size of 4. Our model is implemented based on the OpenPCDet project, and for data augmentation, we employ strategies such as ground truth sampling and random horizontal flipping

4.3 Experimental Results on KITTI Dataset

We evaluated PAEN on the KITTI dataset, focusing on the detection performance of the car category objects at an IOU threshold of 0.7. We used the R40 method to compute the Average Precision (AP). Evaluation was performed using BEV metrics from KITTI, comparing with existing single-modal and multi-modal models. PillarNet[8] is employed as the baseline model, and as it has not been experimented with on the KITTI dataset, we use the released PillarNet from the OpenPCDet project as our baseline for evaluation.

The BEV evaluation results of PAEN on the KITTI test set are presented in Table 1. PAEN achieves detection accuracies of (93.83%, 88.52%, 85.41%) for the easy, moderate, and hard difficulty levels, respectively. Our model exhibits the best performance in BEV metrics for the car category across all difficulty levels. The improved performance validates the effectiveness of the components proposed in our model. Observations indicate that models like PointPillar[7], PillarNet[8], and PAEN, which utilize a pillar representation, often outperform on BEV evaluation metrics. We believe that these models effectively preserve the spatial information of point clouds in a two-dimensional format and leverage sophisticated 2D detection techniques to process BEV features efficiently. This approach allows these models to achieve higher detection accuracy while maintaining real-time performance.

Table 1. BEV Metrics Detection Results of Car Category Objects in KITTI Test Set .

Method	Modality	AP(BEV) @Car		
		Easy	Mod	Hard
VoxelNet	**L**	89.35	79.26	77.39
PointPillars	**L**	90.07	86.56	82.81
SECOND	**L**	88.01	79.37	77.95
AFDet	**L**	89.42	85.45	80.56
CenterNet3D	**L**	91.08	88.46	83.62
MV3D	**L+R**	86.62	78.93	69.80
AVOD	**L+R**	89.75	84.95	78.32
F-PointNet	**L+R**	91.17	84.67	74.77
PointPainting	**L+R**	92.45	88.11	83.36
PL++	**L+R**	84.61	73.80	65.59
PI-RCNN	**L+R**	91.44	85.81	81.00
PillarNet(baseline)	**L**	91.29	86.98	84.55
PAEN(ours)	**L**	**93.83**	**88.52**	**85.41**

4.4 Ablation Experiment

We performed ablation studies on the KITTI validation set using 3D evaluation criteria to assess the influence of each proposed components on the overall results.

The effects of PSA and PFSA in the Pillar Sequence Attention Feature Encoder:

As shown in Table 2, after adding the PSA module for enriching point cloud features, the mAP increased to 72.81%. Particularly, there was a significant improvement in detecting small targets. The combination of PSA and PFSA modules resulted in an increase in mAP to 73.52%, which is a significant improvement of 3.69% compared to the baseline model. The PFSA module further filtered and aggregated the rich point cloud feature information obtained by the PSA module, leading to improved detection performance across all object categories.

Table 2. Contribution of PSA and PFSA modules to PAEN.

Component			3D Object Detection(%)			
Baseline	**PSA**	**PFSA**	**Cars**	**Cyclists**	**Pedestrians**	**mAP**
√			85.80	45.37	77.51	69.56
	√		87.10	**50.24**	81.09	72.81
		√	86.30	48.51	73.99	69.60
	√	√	**87.53**	49.96	**83.07**	**73.52**

The effects of SDEC and DDEC in the dilated expansion convolution network:

Table 3 demonstrates that when the original basic sparse convolution block is replaced individually with the SDEC block or the DDEC block, the detection performance for all three categories improves notably as the receptive field of the sparse convolution is enlarged. When both SDEC and DDEC blocks are employed together, the mAP reaches 73.85%, representing a 4.29% increase over the baseline model. This suggests that our dilated expansion convolution network effectively expands the model's receptive field through dilated convolutions, which in turn enhances the model's detection capabilities.

Table 3. Contribution of SDEC and DDEC modules to PAEN**.**

Component			3D Object Detection(%)			
Baseline	**SDEC**	**DDEC**	**Cars**	**Cyclists**	**Pedestrians**	**mAP**
√			85.80	45.37	77.51	69.56
	√		88.28	48.10	79.06	71.81
		√	87.16	50.07	80.12	72.45
	√	√	**88.35**	**51.11**	**82.09**	**73.85**

Inference Time Comparison:

We conducted experiments on a NVIDIA RTX4090 and obtained comparative results of inference time between PAEN and previous classic real-time detection models. Table 4 illustrates that the PAEN strikes a balance between speed and precision, maintaining the capability for real-time detection while significantly improving the detection accuracy.

Table 4. Inference Time Comparison.

Method	mAP(3D) @Car	Interfence Time
SECOND	80.92	64ms
PointPillar	75.29	56ms
PillarNet(Baseline)	78.36	**51ms**
PAEN(ours)	**81.48**	62ms

Visualization of Detection Results:

We visually compared the detection results between the baseline model and our proposed PAEN. Fig.6 illustrates the detection results for three different scenes, with each column representing one detection scenario. The second row presents the detection results of the baseline model, and the third row displays the detection results of PAEN. The PAEN exhibits enhanced abilities in both localization and categorization compared to the baseline model.

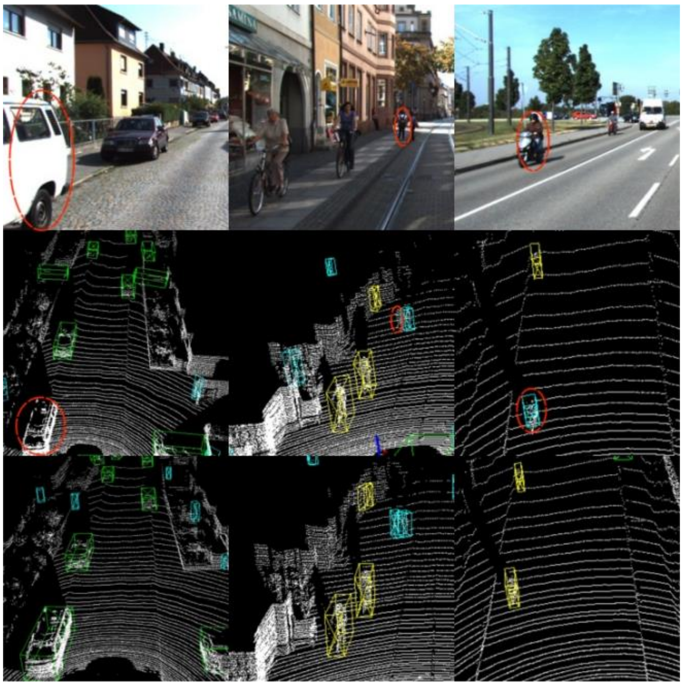

Fig. 6. Visual Comparison of Detection Results between PAEN and Baseline Model.

5 Conclusion

This paper presents a novel 3D object detector based on attention mechanisms and dilated convolution, named PAEN. In the pillar feature encoding stage, we propose a Pillar Sequence Attention Feature Encoder. On one hand, the PSA module enriches the features of individual raw points by utilizing inter-point attention feature information, addressing the limitations of traditional pillar encoders. On the other hand, the PFSA module encodes the point features inside the pillars in a more refined manner to obtain more powerful pillar feature information, solving the problem of rough encoding in traditional pillar encoders. In the backbone network feature extraction stage, we propose a Dilated Expansion Convolution Network. By aggregating features from sparse and dense feature maps with larger receptive field ranges through SDEC and DDEC, we compensate for the insufficient receptive field of traditional convolutional backbone networks. The experiments demonstrate that the PAEN is capable of achieving higher precision in detection while maintaining real-time performance.

Acknowledgements. This work was supported by the National Natural Science Foundation of China No.62273293, Shandong Provincial Natural Science Foundation, China under Grant ZR2022LZH002. and Innovation Capability Improvement Plan Project of Hebei Province (No. 22567626H).

References

1. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite[C]//2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012: 3354-3361.
2. Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.
3. Shi S, Wang X, Li H. Pointrcnn: 3d object proposal generation and detection from point cloud[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 770-779.
4. Li Z, Wang F, Wang N. Lidar r-cnn: An efficient and universal 3d object detector[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 7546-7555.
5. Zhou Y, Tuzel O. Voxelnet: End-to-end learning for point cloud based 3d object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4490-4499.
6. Yan Y, Mao Y, Li B. Second: Sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): 3337.
7. Lang A H, Vora S, Caesar H, et al. Pointpillars: Fast encoders for object detection from point clouds[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 12697-12705.
8. Shi G, Li R, Ma C. Pillarnet: High-performance pillar-based 3d object detection[J]. arXiv preprint arXiv:2205.07403, 2022.
9. Li J, Luo C, Yang X. PillarNeXt: Rethinking network designs for 3D object detection in LiDAR point clouds[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 17567-17576.
10. Chen L C, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation[J]. arXiv preprint arXiv:1706.05587, 2017.
11. Chen Q, Wang Y, Yang T, et al. You only look one-level feature[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13039-13048.
12. Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.
13. Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.
14. Loshchilov I, Hutter F. Fixing weight decay regularization in adam[J]. 2018.