Article

SAE3D: Set Abstraction Enhancement Network for 3D Object Detection Based Distance Features

Zheng Zhang , Zhiping Bao , Qing Tian * , Zhuoyang Lyu

Posted Date: 30 October 2023

doi: 10.20944/preprints202310.1823.v1

Keywords: 3D object detection; distance features; SA layer enhancement

Article

SAE3D: Set Abstraction Enhancement Network for 3D Object Detection Based Distance Features

Zheng Zhang [1], Zhiping Bao [1], Qing Tian [1,*] and Zhuoyang Lyu [2]

[1] School of Information, North China University of Technology, Beijing 100144, China; zhangzheng@ncut.edu.cn (Z.Z.); wiskey@mail.ncut.edu.cn (Z.B.).

[2] School of Information, Brown University Cmputer Science and Applied Math, RI, US.; zhuoyang_lyu@brown.edu (Z.L.).

* Correspondence: tianqing@ncut.edu.cn

Abstract: With increasing demand from unmanned driving and robotics, more attention has been paid to point cloud-based 3D object accurate detection technology. However, due to the sparseness and irregularity of the point cloud, the most critical problem is how to utilize the relevant features more efficiently. In this paper, we proposed a point-based object detection enhancement network to improve the detection accuracy in the 3D scenes understanding based on the distance features. Firstly, the distance features are extracted from the raw point sets and fused with the raw features about reflectivity of the point cloud to maximizing the use of information in point cloud. Secondly, we enhanced the distance features and raw features that we collectively refer to them as self-features of the key points in Set Abstraction (SA) layers with the self-attention mechanism, so that the foreground points can be better distinguished from the background points. Finally, we revised the group aggregation module in SA layers to enhance the feature aggregation effect of key points. We conducted experiments on the KITTI dataset and nuScenes dataset and the results show the enhancement method proposed in this paper has excellent performance.

Keywords: 3D object detection; distance features; SA layer enhancement

1. Introduction

With the development of unmanned driving and other technologies, 3D scenes understanding based on point cloud has become one of the popular research directions. Compared to traditional images, point cloud data have the unique advantages. The strong penetration of LiDAR makes the point cloud less susceptible to external factors such as weather and light. However, point cloud are also characterized by sparseness and disorder, and the reflectivity of LiDAR decreases as the measurement distance increases. This led to poor characterization of objects at a distance, causing the drop of detection accuracy. How to deal with these characteristics of the point cloud has become the key to improve the accuracy in 3D detection task.

In recent years, in order to efficiently utilize the information provided in the point cloud, researchers have proposed a number of schemes, which are mainly divided into two types according to the different processing methods:

a) Grid-based methods, which partition the sparse points into regular voxel or pillar grids, and process them through 3D or 2D convolutional networks.

b) Point-based methods, which directly perform feature learning on point sets with SA which are most often utilized to sample the key points and aggregate features.

Compared with the point-based methods, the grid-based methods increase the computational speed of network inference, but also cause the loss of information during the voxelization. Therefore, in order to ensure that the information in the point sets be fully utilized, a point-based methods enhancement network was proposed in this paper.

The core of the point-based 3D object detection methods is the SA layer, which was first proposed by Qi et al. [8]. In the existing research, SA layer has been revised by many methodologies and how to fully utilize the information of each point and reduce inference time became a priority in point-based methods. In 3DSSD [1], to speed up the inference, they first adopted 3D single stage object detector and a feature-based farthest point sample module (F-FPS) is proposed. This module utilizes the feature information of the point sets to sample key points in order to keep adequate interior points of different foreground instances. SASA [2] proposes a semantic segmentation based farthest point sample module (S-FPS), which utilizes the feature of point cloud to distinguish the foreground points from the background points through a small semantic segmentation module to better access to key points. However, the point cloud features used by these algorithms only utilize the raw features of the point cloud, i.e., the reflectivity and 3D coordinates which reveals spatial information of each point in the point cloud, and distance characteristics are not taken into consideration. In the actual measurement, due to the attenuation characteristics of LiDAR and the limitation of the observation angle, the reflectivity of the measured point decreases as the object is further away, and the projection of the object in the point cloud also decreases.

Therefore, based on the distance characteristics related to point cloud, we propose three feature enhancement modules to more efficiently utilize the semantic information contained in the point cloud. Firstly, we propose the initial feature fusion module, in which the distance feature is extracted from the point cloud and incorporated with the raw feature of each point. Secondly, we present a key point feature enhancement module. During the group aggregation in SA, the self-characterization of the key points will be weakened, but it is crucial for distinguishing whether the key point is a foreground or background point. Therefore, after each sampling aggregation, the multi-attention mechanism is used to strengthen the features of key points and fuse them with the aggregated features. Finally, in order to enhance the effect of group aggregation in SA, we revised the original grouping module, in which multiple points nearest to the key points are taken to participate in feature aggregation after sampling over the key points. However, only the spatial location is considered, which may result in features belonging to different categories being mixed together during the aggregation process, so the performance of the semantic segmentation module before S-FPS is decreased, which leads to the performance degradation of the sampling effect of S-FPS. Therefore, we optimize the grouping module by selecting the points with the closest features as the aggregation points from multiple points closest to the key points.

In summary, the main contribution of the article is summarized as follows:

- We proposed a key points self-features enhancement module to enhance the self-features of the key points. In this module, we introduce the multi-attention mechanisms to enhance the raw features and distance features to retain the semantic information of the key points as much as possible during each SA layer.
- We proposed an initial feature fusion module to extract the distance features of the point cloud and fuse the distance features into the raw features of the point sets. This module makes the features of the distant points more significant and thus improve detection accuracy of the distant instances.
- We revised the group aggregation module in the set abstraction. We made a second selection after the first selection of points within a fixed distance around the key point. In second selection we take the features into account to enhance the sampling effect of S-FPS.

2. Related

1.1. Grid-based methods

Grid-based methods are mainly divided into two categories: voxel-based methods and pillar-based methods. In voxel-based methods, irregular point cloud is first con-verted into regular voxels, which are then fed into the network. VoxelNet [3] is a pioneering network that converts point cloud into voxels, and then utilizes 3D convolutional network to predict 3D bounding box. Yan et al. [14] proposed 3D sparse convolution, which reduces the computation of traditional 3D convolution, and makes the detection efficiency of voxel-based detection net-works greatly improved. Voxel Transformer [17] and Voxel Set Transformer [13] introduce modules such as Transformer [15] and Set Transformer [16] respectively on the basis of voxels to improve the detection accuracy. Pillar-based methods such as PointPillars [5] divide the space into regular pillars, which are compressed and then fed into a 2D convolutional network, which will dramatically increase the network inference speed. PillarNet [6] uses a sparse convolutional-based encoder network for spatial feature learning, and uses the Neck module for high-level and low-level feature fusion to improve the accuracy of the accuracy of pillar-based detection methods.

Grid-based methods lose more semantic information in the process of converting irregular point cloud into regular voxels or pillars. This may lead to poor performer in the final detection accuracy.

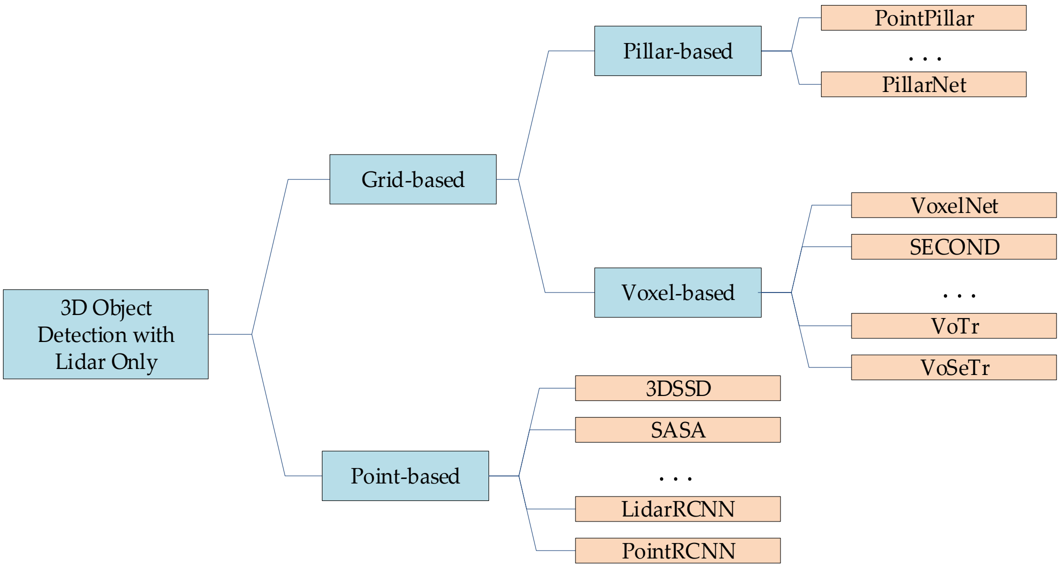

Figure 1. Overview of related work

1.2. Point-based methods

Point-based methods generally perform feature extraction directly on the o point sets. This approach obtains key points and aggregates points around them by means of sampling and group aggregation for feature extraction. Point-based methods were first proposed by Qi et al. [7] and later improved and refined by Qi et al. [8]. Shi et al. [4] first proposed to extract the foreground points by segmentation and utilize the features of these points for the bounding box regression to improve the detection accuracy. Yang et al. [1] utilized one-stage detection to improve the inference speed and proposed the F-FPS, to make the sampled key points closer to the foreground instances. SASA [2] predicts scores of each point by a small semantic segmentation module to make abstracted point sets focus on object areas.

Since point-based 3D object detection is directly processing the point sets, relative to the voxel-based methods, point-based methods can maximize the retention of the semantic information of the point cloud and achieve higher detection accuracy. Therefore, this paper adopts the point-based object detection network and tends to utilize the original information of the point cloud more efficiently.

3. Proposed Methods

In this section, we will introduce in detail the network architecture of the SAE3D proposed in this paper. This enhancement network consists of three main parts: an initial feature fusion module, a key points self-features enhancement (KSFE) module and a revised group aggregation (RGA) module.

As shown in Figure 2, the overall architecture is a one-stage point-based 3D object detection network. Firstly, we define the raw points fed into the network as P, the initial feature fusion module extracts the distance features and integrate with initial features of each point in P. After integration, we feed P into the backbone, which contains three SA layers, and we call the input of each SA layer as P_1. In SA layers we first sample the key points K from P_1, and then we feed K into the key points feature enhancement module to enhance the self-features of K, and then inputs K into the key points feature enhancement module to enhance the features of K. After enhancement we got K_1. Finally, the revised group aggregation module is used to aggregate the points around K_1 to obtain the aggregated key points K_2. K_2 is the finally output of each SA layer.

After the backbone is finished, in order to improve the prediction accuracy, this paper adopts the bounding box prediction mechanism in VoteNet [12] to predict the bounding box as same as SASA [2].

We will explain each module in detail below.

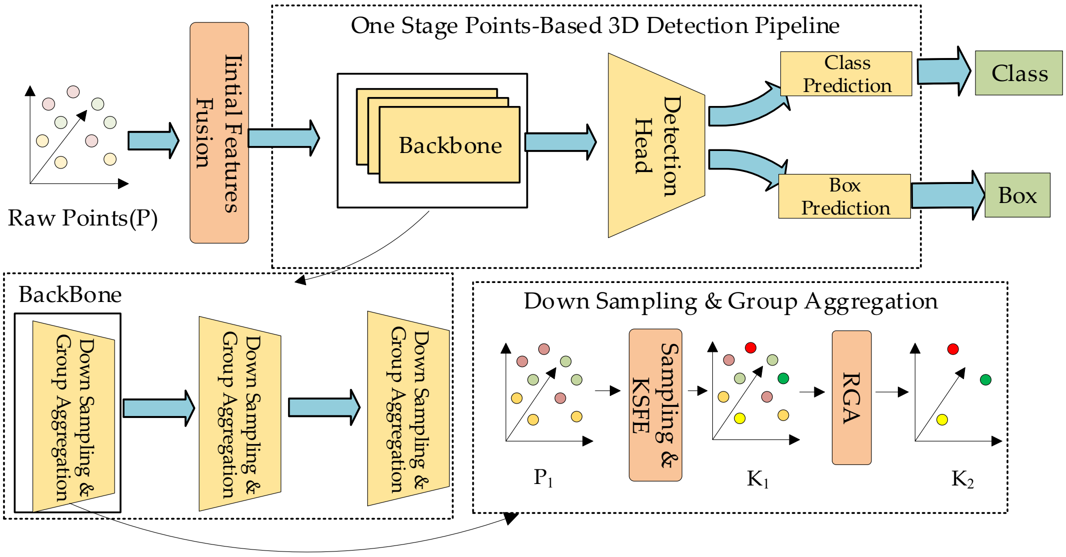
One Stage Points-Based 3D Detection Pipeline
Raw Points(P)
Iintial Features Fusion
Backbone
Detection Head
Class Prediction
Box Prediction
Class
Box
BackBone
Down Sampling & Group Aggregation
Down Sampling & Group Aggregation
Down Sampling & Group Aggregation
Down Sampling & Group Aggregation
Sampling & KSFE
RGA
P1
K1
K2

Figure 2. Overall flowchart (the raw point cloud P through the initial feature fusion module to get P_1, P_1 input to the backbone, backbone consists of three SA (Set Abstraction) layers. P_1 first through the down sampling and then through the KSFE (Key points Self-Feature Enhancement module) to the K_1, and finally through the RGA (Revised Group Aggregation module) to get K_2.

3.1. Initial features fusion module

Before the SA layers we utilize the Initial features fusion module to extract the distance features and integrate with features of the raw point sets. The relevant features of the raw point cloud are very sensitive to the measurement distance. In the actual measurement, as the distance increases, the reflectivity of LiDAR decreases, which leads to the problem that the features of the long-distance points are not obvious and thus reduce the detection accuracy, so we believe that the distance features are very important for the improvement of the accuracy of the target detection.

3.1.1. Distance features

Due to the high computational consumption of traditional calculation method, our distance features are defined as follows:

$$DF_p = \frac{|x_p| + |y_p| + |z_p|}{Scale} \qquad p \in P \qquad (1)$$

where P is the raw point sets, DF_p and x_p, y_p, z_p is the distance feature and the coordinate of the p and $Scale$ is the scaling factor.

3.1.2. Feature fusion

We process the initial feature fusion as shown in Figure 3. Since the reflectivity of each point decreases with the increase of the measurement distance, we adopt the method of adding the distance features with the initial features of the point cloud to strengthen the features of the long-distance points. Finally, we perform fusion operations on the coordinates and related features of the point cloud through the Multilayer Perceptron (MLP).

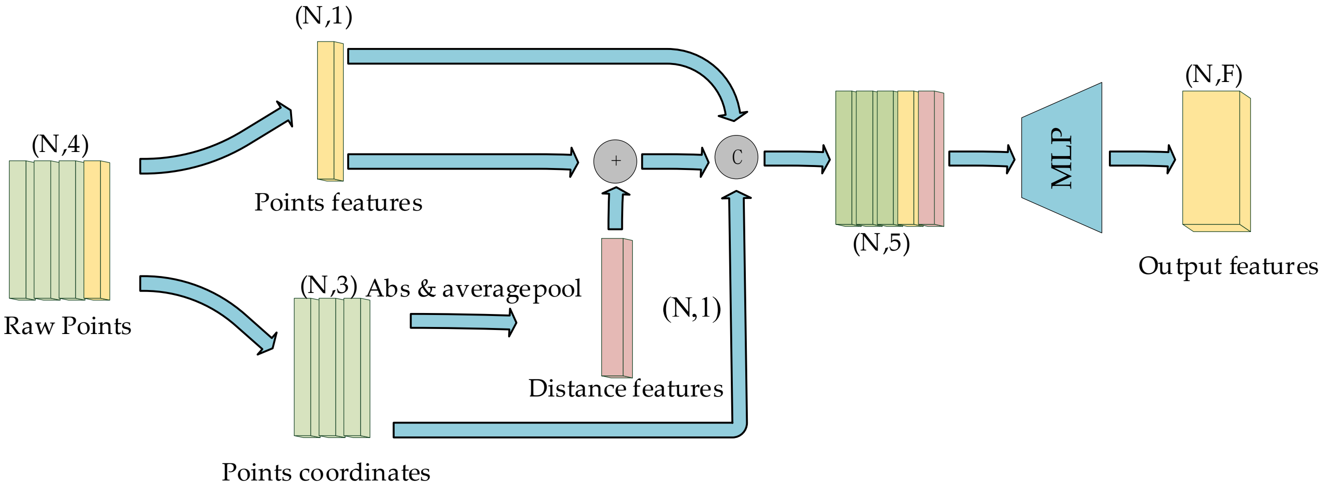

Figure 3. Initial features fusion module (where *N* stands for the number of input point clouds and *F* stands for the number of feature layers for each point in the output. "C" represents the stitching operation and "+" represents the numerical summing operation.)

3.2. Key points features enhancement module

In SA layers, the key points get from the sampling will process feature aggregation with its surrounding points, and the self-features of the key points will be diminished after aggregation with max or average pooling. However, each key point has its own unique features in the point cloud, and these features contain important information included where the key point is located and what kind of instance the key point stand for. However, the feature aggregation will cause the loss of such information. Therefore, we propose a key points self-feature enhancement module as shown in Figure 4, which enhances the distance features and the raw features of the key point, and integrate them with the aggregated features.

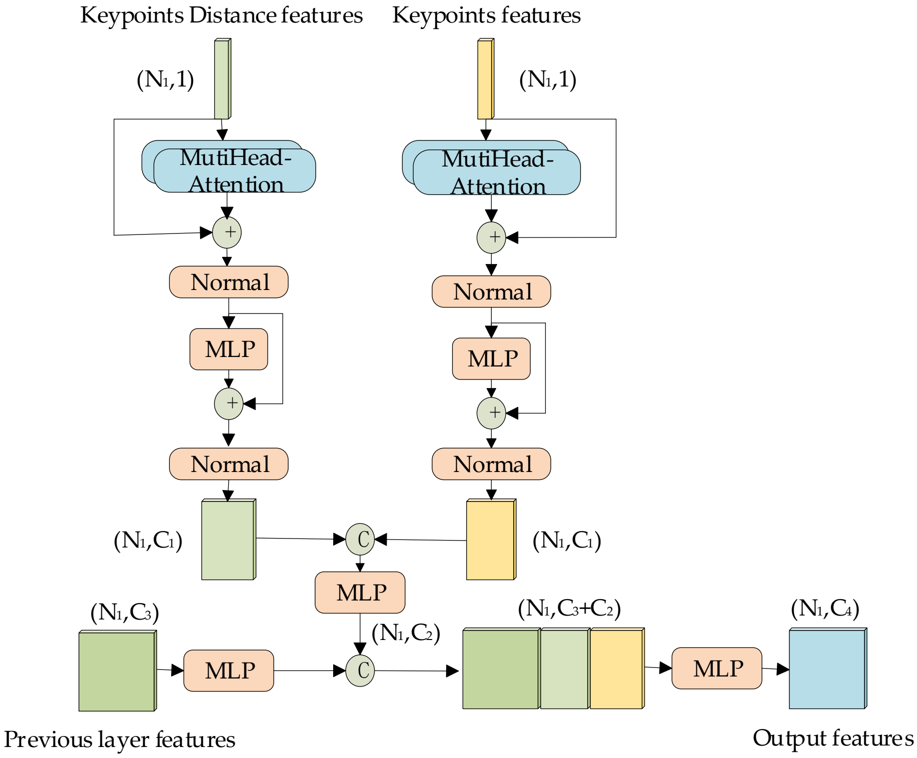

Figure 4. Key points self-feature enhancement module (where N_1 is the number of key points after sampling, and C_i is the number of feature channel in each stage. "C" stands for the stitching operation and "+" stands for the numerical summing operation.)

3.2.1. Feature Enhancement Module

In order to make the self-features of the key points more distinctive, we adopt the multi-attention mechanism to enhance the distance features and raw features of the key points. The features are strengthened by the multi-head self-attention mechanism, the self-attention algorithm essentially uses a matrix multiplication to calculate the relationship between each patch and the other patches in the query. The specific formulas are as follows:

$$Attention(Q,K,V)=Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

$$Q=F\times W_q \tag{3}$$

$$K=F\times W_k \tag{4}$$

$$V=F\times W_v \tag{5}$$

where F is the self-features of the key points, W_q, W_k and W_v are the learnable weight matrices. After that we utilize the splicing method to combine them with each other. Finally, we perform the integration of the aggregated features of the key points with their self-features through the MLP to accomplish the enhancement of self-features of key points.

3.3. Revised group aggregation module

In the process of sampling key points, we follow the S-FPS and D-FPS combining sample strategy as same as SASA [2]. A small semantic segmentation module is adopted in the network to compute the classification score of each point to distinguish between foreground and background points in the point cloud. The input features to the segmentation network are those obtained from group aggregation of the point sets. In the general grouping operation, the selection of points used for aggregation around the key points only considers the spatial location from the key points, and not taken features distance from the key points into account. In this paper, it is argued that this aggregation operation diminishes the border lines of the different instances and reduces the effectiveness of the segmentation module in the network in predicting the classification scores of each point, thus affecting the sampling performance of S-FPS.To avoid these problems, we make a second selection after selecting points within a certain distance from each key point. In the second selection, we introduce the feature distance to ensure the features of the selected points are close to the features of the key point. By virtue of this we can enhance the performance of the segmentation in this network.

3.3.1. Group aggregation method

The particular operation is shown in Figure 5. Firstly, we select N points as a point set P_N within the sphere with radius R around the key point, and calculate the feature distance D_f between the points and the key points, which we define as follows:

$$D_f = \left| f_{keypoints} - f_n \right| \qquad n \in N \qquad (6)$$

where $f_{keypoints}$ and f_n is separately stand for the features of the key points and the features of the points around the key points. Before calculation, these features will through a simple MLP to make sure the features channel is one. After obtaining D_f, we select the N_k points with the smallest D_{fk} (k = 1, 2, …, N_k) in P_N as a point set P_{Nk}. The P_{Nk} will be used for subsequent features aggregation. In this way, we further strengthen the semantic information of the key points. This can help S-FPS to better distinguish the foreground points from the background points before sampling.

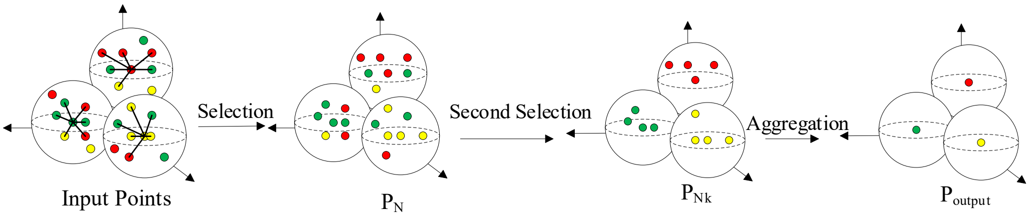

Figure 5. Revised group aggregation module (points with similar colors in the figure represent similar features, P_N is the point obtained by the first selection around the key point, P_{Nk} is the point obtained by the second selection, and P_{output} is the final point output after group aggregation).

3.4. Prediction Head

The overall architecture in this paper consists of three SA layers with a bounding box prediction network. Same as our baseline network, our bounding box prediction network follows the bounding box prediction mechanism in VoteNet [12]. A voting point indicating the center of mass of the corresponding object is first computed from the candidate point features, and then the points in the vicinity of each voting point are aggregated to estimate the bounding box of the detected target.

3.5. Loss

The loss function in the SAE3D inherited from SASA [2]. The overall loss function expressed as follows:

$$L = L_v + L_c + L_r + L_{seg} \qquad (7)$$

where L_c and L_r is the loss of the classification and regression, L_v is the loss generated when calculating the vote in the point voting head proposed in VoteNet [12]. L_{seg} is total segmentation loss proposed in SASA [2].

4. Experiment

4.1. Datasets

The network we proposed are validated on the KITTI dataset and nuScenes dataset:

4.1.1. KITTI Dataset [9]

The KITTI dataset is a widely used public dataset in the field of computer vision, which is mainly used to study and evaluate tasks such as autonomous driving, scene understanding and target detection. The dataset is based on the streets of Karlsruhe, Germany, and contains rich urban driving scenarios. The KITTI dataset has been the mainstream standard for 3D object detection in traffic scenes because it provides data from real-world scenarios with a high degree of realism and representativeness.

In the original KITTI dataset, each sample contains multiple consecutive frames of point cloud data. In the experiment, a total of 7481 point clouds are included along with 3D bounding boxes for training and 7581 samples for testing. We use a general setup where the training samples are subdivided into 3712 training samples as well as 3769 testing samples, and our experimental network is trained on the training samples and validated on the testing samples.

4.1.2. NuScenes Dataset [10]

The nuScenes dataset is one of the more challenging autopilot datasets,380k LiDAR scans from 1,000 scenes. It is labeled with up to 10 object categories, including 3D bounding boxes, object velocities, and attributes. The detection range is 360 degrees. nuScenes dataset are evaluated using metrics such as the commonly used mean Average Precision (mAP) and the novel nuScenes Detection Score (NDS), which reflects the overall quality of measurements across multiple domains.

When transferring the nuScenes dataset, we combine LiDAR points from the current key frame and previous frames within 0.5 seconds, which involves up to 400k LiDAR points in a single training sample. We then reduce the number of input LiDAR points. Specifically, we voxelize the point cloud from the key frame as well as the stacked previous frames with pixel sizes of (0.1m, 0.1m, 0.1m), then randomly select 16,384 and 49,152 voxels from the key frame and the previous frames, and randomly select one internal LiDAR point from each selected voxel. A total of 65,536 points were fed into the network with 3D coordinates, reflectivity, and timestamps.

4.1.3. Evaluation indicators

In the experiment on the KITTI dataset, two precision metrics are used, one is the 11-point Interpolated average precision (AP) proposed by Gerard et al. [19], and the other is the average precision AP$|_{R40}$ for 40 recalled positions proposed by Simonelli et al. [18], and all the precision IoU (Intersection over Union) thresholds are 0.70. The specific formulas of AP$|_{R}$ are as follows:

$$AP|_R = \frac{1}{|R|}\sum_{r\in R}\rho_{interp}(r) \tag{8}$$

$$\rho_{interp}(r) = \max_{r':r'\geq r} p(r') \tag{9}$$

where *p(r)* gives the precision at recall *r*. AP applies exactly 11 equally spaced recall levels: R_{11} = {0, 0.1, 0.2, …,1} and AP$|_{R40}$ applies recall levels: R_{40} = {1/40, 2/40, 3/40, ..., 1}. We mainly use AP as an accuracy indicator and AP$|_{R40}$ will be applied in ablation experiment in section 4.5.

In the nuScenes dataset, as mentioned above, we apply the NDS and mAP as the evaluation indicator. The specific formulas of NDS expressed as follows:

$$NDS = \frac{1}{10}\left[5mAP + \sum_{mTP\in TP}(1-\min(1, mTP))\right] \tag{10}$$

where mTP is the mean Ture Positive metrics consists of 5 metrics: average translation error, average scale error, average orientation error, average velocity error and average attribute error.

4.2. Experimental Setting

SAE3D is implemented based on the OpenPCdet [11] and is trained on a single GPU. All experiments were performed on Ubuntu 16.04 and NVIDIA RTX-2080Ti.

3.2.1. Setting in KITTI dataset

During the training process, batch size takes the value of 2, and 16384 points are randomly selected from the remaining points in each batch to input into the detector. In terms of network parameters, the number of key points in the three SA layers is set to 4096, 1024, and 512, respectively, and the scaling factor Scale for the distance feature takes the value of 120.

Adam optimizer [20] and a periodically varying learning rate were adopted in the training for a total 80 epochs, with the initial value of the learning rate set to 0.001. Additionally, we used three commonly used data augmentation methods during training: randomly flipping the X-axis with respect to the Y-axis, random scaling, and randomly rotating the Z-axis.

4.2.2. Setting in nuScenes dataset

During the training process, batch size takes the value of 1. Adam optimizer and a periodically varying learning rate were adopted in the training for a total 10 epochs, with the initial value of the learning rate set to 0.001.

To handle the huge number of points in the nuScenes dataset, four SA layers are adopted. The number of key points in four SA layers is set to 16384, 4096, 3072 and 2048.

4.3. Results

The detection performance of the SAE3D model is evaluated on the KITTI dataset and nuScenes dataset against some existing methods proposed in the literature.

In KITTI dataset, based on the difficulty of detecting the test set is categorized into three levels of difficulty: "Easy", "Moderate" and "Hard". We take the 3D bounding box average precision (3D AP) of the "Car" category as the main evaluation which is usually adopted as main indicator in KITTI datasets. As shown in Table 1, compared with the baseline network SASA, 3D AP is improved by 0.544% and 0.648% in the difficulty levels of "Moderate" and "Hard", respectively. (The rest of the precision improvement will be shown in detail in Section 4.5).

In nuScenes dataset, as shown in Table 2. Compared with the baseline network, SAE3D got 3.3% and 1.7% improvement in the indicators of NDS and mAP, respectively.

Table 1. The detection results of 3D AP for "Car" in KITTI.

Methods	**Car 3D AP (%)**		
	Easy	Moderate	Hard
SECOND	84.656	75.966	68.712
VoxelNet	77.478	65.119	57.736
PointPillars	82.588	74.317	68.995
PointRCNN	89.023	78.246	77.554
Vox Set Tran	88.869	78.766	77.576
SASA	**89.108**	78.847	77.588
SAE3D	89.059	**79.391**	**78.236**

Table 2. Results on the nuScenes validation set. Evaluation metrics include NDS, mAP and 10 classes. Abbreviations: pedestrian (PED.), traffic cone (T.C.), construction vehicle (C.V.).

Methods	**NDS**	**mAP**	**Car**	**Truck**	**Bus**	**Trailer**	**C.V.**	**Ped.**	**Motor**	**Bicycle**	**T.C.**	**Barrier**
PointPillars	45.2	25.8	70.3	32.9	44.9	18.5	4.2	46.8	14.8	0.6	7.5	21.3

3DSSD	51.7	34.5	75.9	34.7	60.7	21.4	10.6	59.2	25.5	7.4	14.8	25.5
SASA	55.3	36.1	71.7	42.2	63.5	29.6	12.5	62.6	27.5	9.1	12.2	30.4
SAE3D	**58.6**	**37.8**	72.4	44.1	62.7	31.2	15.9	60.4	30.1	12.8	10.1	31.6

4.4. Enhancement Validation

In order to verify the enhancement effect of the proposed network in this paper, we utilize SASA [2] and PointRCNN [4] two baseline networks, for testing respectively in KITTI dataset. Both baseline networks are point-based 3D object detection networks, where SASA [2] is a one-stage object detection network and PointRCNN [4] is a two-stage object detection network. The experiments introduce the enhanced network proposed in this paper into the above two networks respectively and effectively improve the detection performance of the original benchmark network.

Table 3. shows the improvement in the accuracy of the 3D detection frames of the "Car" category in the enhanced networks of SASA [2] and PointRCNN [4], respectively.

After the introduction of the enhanced network in SASA [2], the 3D AP of the “Car” decreases slightly in "Easy" difficulty, and increases by 0.544% and 0.648% in "Moderate" and "Hard" difficulties, respectively.

After introducing the augmented network in PointRCNN [4], the accuracy of the 3D AP is improved by 0.137%, 0.593% and 0.885% in "Easy", "Moderate" and "Hard" difficulties, respectively.

Table 3. Enhancement effectiveness. Abbreviations: Distance features-based enhancement network proposed in this paper (SAE3D).

Methods	**Car 3D AP (%)**		
	Easy	Moderate	Hard
SASA	89.108	78.847	77.588
SASA+ SAE3D	89.059	79.391	78.236
Improvement	-0.049	+0.544	+0.648
PointRCNN	89.023	78.246	77.554
PointRCNN+ SAE3D	89.160	78.839	78.439
Improvement	+0.137	+0.593	+0.885

4.5. Ablation Experiment

In this paper, ablation experiments are designed to verify the actual effect of each module. All modules are trained on the training set of the KITTI dataset and evaluated on the validation set for the "Car" category of the KITTI dataset.

4.5.1. Initial feature fusion module

As shown in Table 4, the initial feature fusion module proposed in this paper is of great help to improve the precision of 3D bounding box. The improvement of this module is most obvious in the difficulty levels of "Moderate" and "Hard". Compared with the baseline network used in this paper, in the difficulty levels of "Moderate" and "Hard", 3D bounding box accuracy improvement of this module is 0.551% and 0.811% respectively, and the improvement of the accuracy of the 2D bounding box accuracy is 0.186% and 0.811%, and bounding box accuracy improvement in BEV view is 0.257% and 1.048%.

As shown in Table 5, when using the $AP|_{R40}$, the improvement in the accuracy of the 3D bounding box is 2.549% and 2.582% for the difficulties of "Moderate" and "Hard", respectively. The improvement in the accuracy of 2D bounding box is 1.976% and 0.533% respectively, the improvement in the accuracy of bounding box in BEV view is 0.295% and 2.257% respectively.

4.5.2. Key points self-features enhancement module

As shown in Table 4, this module improves the detection accuracy of 3D bounding box and the accuracy of bounding box detection in BEV view. The detection accuracy of the 3D bounding box is improved by 0.339% and 0.746% under the difficulty levels of "Moderate" and "Hard" respectively, and the detection accuracy of the bounding box in BEV view is improved by 0.118%, 0.565%, and 1.349% in “Easy", "Moderate", and "Hard" levels of difficulty, respectively.

As shown in Table 5, the accuracy of the 3D bounding box is improved by 2.362% and 2.467% for the "Moderate" and "Hard" levels of difficulty, respectively, when using AP|*R40*. The accuracy of the bounding box in BEV view was improved by 1.778%, 1.906% and 2.367% for "Easy", "Moderate", and "Hard" levels of difficulty, respectively.

4.5.3. Revised group aggregation module

As shown in Table 4, the detection accuracy of this module on BBOX is improved by 0.316% and 0.376% under the difficulty levels of "Moderate" and "Hard", respectively. And compared with the baseline network, the module improves other accuracies such as 3D bounding box and steering angle.

As shown in Table 5, when AP|*R40* are used, the detection accuracy improvement on BBOX is 2.051% and 0.555% at "Moderate" and "Hard" levels, respectively.

Table 4. Comparison table of the general accuracy enhancement effect of different modules. Abbreviations: initial feature fusion module(I), KSFE module(K), and RGA module(F).

+I	+K	+F	Car 3D AP (%)			Car BBOX AP (%)			Car BEV AP (%)			Car AOS AP (%)		
			Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
			89.108	78.847	77.588	96.742	89.855	89.036	90.199	87.855	85.993	96.71	89.75	88.88
	√		88.971	79.246	78.334	96.473	89.847	89.163	**90.317**	**88.420**	**87.342**	96.44	89.81	89.07
		√	89.213	79.324	78.114	**96.813**	**90.171**	**89.412**	89.876	89.397	86.976	96.54	90.08	89.11
√			**89.167**	**79.398**	**78.399**	96.668	90.041	89.287	90.149	88.112	87.041	96.64	89.98	89.12
√	√	√	89.059	79.391	78.236	96.758	90.169	89.382	89.978	88.382	86.824	**96.71**	**90.10**	**89.22**

Table 5. Comparison table of the AP|*R40* enhancement effect of different modules. Abbreviations: initial feature fusion module(I), KSFE module(K), and RGA module(F)

+I	+K	+F	Car 3D AP R40 (%)			Car BBOX AP R40(%)			Car BEV AP R40(%)			Car AOS AP R40(%)		
			Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
			91.592	80.705	77.902	**98.289**	92.972	92.104	93.277	89.128	86.465	**98.26**	92.85	91.92
	√		91.457	83.067	80.369	98.111	94.583	92.469	**95.055**	**91.034**	**88.832**	98.09	94.52	92.35
		√	91.432	82.913	78.956	98.023	95.023	92.659	93.124	89.223	88.624	98.21	94.89	92.39
√			91.555	**83.254**	**80.484**	98.097	94.948	**92.637**	93.197	89.423	88.722	98.08	94.85	**92.44**
√	√	√	91.426	83.236	80.191	98.266	**95.036**	92.616	93.014	90.902	88.525	98.23	**94.93**	92.42

4.6. Detection effect

Figure 6 shows the actual detection effect, although there is still a small part of the missed detection problem exists, but most of the vehicles are detected and the accuracy of the 3D bounding box is high.

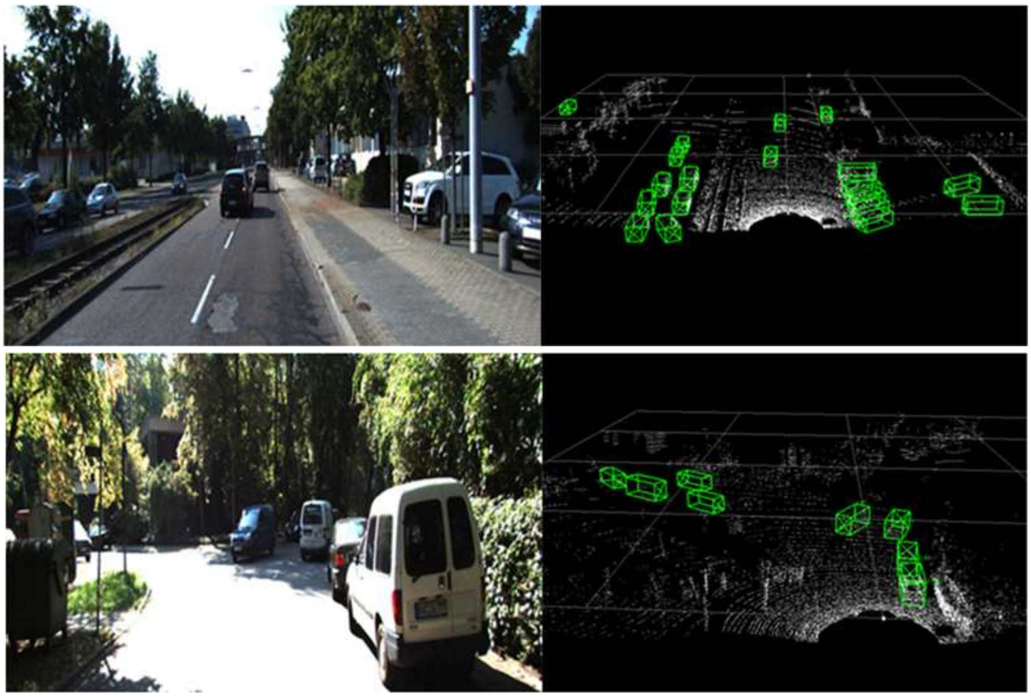

Figure 6. Actual detection effect diagram in KITTI dataset (Left is the pictures of the real scenes, Right is the detection 3D bounding box predicted in the point cloud)

5. Discussion

In this paper, we continue to explore the possibilities of the point-based 3D object detection. Point cloud has a huge amount of data and there is a lot of information contained, useful and useless. We believe there is still information in point cloud that is underutilized. Therefore, we proposed the SAE3D. The results show that find more useful information and enhance the useful information in point cloud can improve the final detection accuracy.

6. Conclusions

In this paper, we proposed SAE3D with three enhancement modules: an initial feature fusion module, key point self- feature enhancement module, and a revised group aggregation module. We describe the design ideas and implementation of the three modules in detail in the paper. In this paper, we use KITTI and nuScenes datasets for testing, and design ablation experiments on the KITTI dataset to analyze the enhancement of each module in detail. The results show that all three enhancement modules we proposed serve to increase the detection accuracy.

Author Contributions: Conceptualization, Z.Z. and Z.B.; methodology, Z.Z. and Z.B.; software, Z.B.; validation, Z.Z.; formal analysis, Z.B.; investigation, Q.T. and Z.B.; resources, Q.T.; data curation, Z.B.; writing, Z.Z. and Z.B.; original draft preparation, Z.B.; visualization, Z.Z.; supervision, Z.B., Z.L. and Q.T.; project administration, Q.T. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program under Ministry of Science and Technology of the People's Republic of China.

Data Availability Statement: Not applicable.

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

References

1. Yang Z, Sun Y, Liu S, et al. 3dssd: Point-based 3d single stage object detector[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11040-11048.

2. Chen C, Chen Z, Zhang J, et al. Sasa: Semantics-augmented set abstraction for point-based 3d object detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2022, 36(1): 221-229.
3. Zhou Y, Tuzel O. Voxelnet: End-to-end learning for point cloud based 3d object detection[C]//Proceedings of the IEEE con-ference on computer vision and pattern recognition. 2018: 4490-4499.
4. 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.
5. 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.
6. Shi G, Li R, Ma C. Pillarnet: High-performance pillar-based 3d object detection[J]. arXiv preprint arXiv:2205.07403, 2022.
7. 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.
8. Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. Advances in neural information processing systems, 2017, 30.
9. Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: The kitti dataset[J]. The International Journal of Robotics Research, 2013, 32(11): 1231-1237.
10. Caesar H, Bankiti V, Lang A H, et al. nuscenes: A multimodal dataset for autonomous driving[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11621-11631.
11. OD Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds[J]. OD Team, 2020.
12. Ding Z, Han X, Niethammer M. Votenet: A deep learning label fusion method for multi-atlas segmentation[C]//Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22. Springer International Publishing, 2019: 202-210.
13. He C, Li R, Li S, et al. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 8417-8427.
14. Yan Y, Mao Y, Li B. Second: Sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): 3337.
15. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
16. Lee J, Lee Y, Kim J, et al. Set transformer: A framework for attention-based permutation-invariant neural net-works[C]//International conference on machine learning. PMLR, 2019: 3744-3753.
17. Mao J, Xue Y, Niu M, et al. Voxel transformer for 3d object detection[C]//Proceedings of the IEEE/CVF International Con-ference on Computer Vision. 2021: 3164-3173.
18. Simonelli A, Bulo S R, Porzi L, et al. Disentangling monocular 3d object detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1991-1999.
19. Gerard Salton and Michael J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA, 1986.
20. Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.