



RPV-CASNet: range-point-voxel integration with channel self-attention network for lidar point cloud segmentation

Jiajiong Li¹ · Chuanxu Wang¹ · Chenyang Wang¹ · Min Zhao¹ · Zitai Jiang¹

Accepted: 18 May 2024 / Published online: 14 June 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Maximizing the advantages of different views and mitigating their respective disadvantages in fine-grained segmentation tasks are an important challenge in the field of point cloud multi-view fusion. Traditional multi-view fusion methods ignore two fatal problems: 1. the loss of depth and quantization information due to mapping and voxelization operations, resulting in “anomalies” in the extracted features; 2. how to pay attention to the large differences in object sizes among different views during point cloud learning, and fine-tune the fusion efficiency in order to improve the performance of network. In this paper, we propose a new algorithm that uses channel self-attention to fuse range-point-voxel, abbreviated as **RPV-CASNet**. **RPV-CASNet** integrates the three different views: range, point and voxel in a more subtle way through an interactive structure (range-point-voxel cross-adaptive layer known as **RPVLayer** for short), to take full advantage of the differences among them. The **RPVLayer** contains two key designs: the **Feature Refinement Module (FRM)** and the **Multi-Fine-Grained Feature Self-Attention Module (MFGFSAM)**. Specifically, the **FRM** allows for a re-inference representation of points with entrained anomalous features, correcting the features. The **MFGFSAM** addresses two challenges: efficiently aggregating tokens from distant regions and preserving multiscale features within a single attention layer. In addition, we design a **Dynamic Feature Pyramid Extractor (DFPE)** for network deployment, which is used to extract rich features from spherical range images. Our method achieves impressive mIoU scores of 69.8% and 77.1% on the SemanticKITTI and nuScenes datasets, respectively.

Keywords Multi view fusion · Feature refinement · Multiple fine-grained self-attention mechanism · 3D semantic segmentation

1 Introduction

Semantic segmentation of 3D point clouds has attracted a lot of attention in areas such as autonomous driving, augmented reality (AR) and virtual reality (VR). For autonomous

systems such as self-driving cars, it becomes especially important to understand the scene accurately. Because errors in the perception process can lead to serious accidents and scheduling problems. In this paper, we aim to improve the performance of semantic segmentation in self-driving scenarios, which allows us to perceive the entire 3D scene in a high-quality and point-by-point manner. 3D data is collected by optical or radar sensors, usually in the form of point clouds. And there are also other forms of views to represent as shown in Fig. 1.

More specifically, the semantic segmentation of point clouds can be broadly divided into three directions. Researchers would divide the original point cloud into voxel units, defined in detail in Cartesian coordinates, depicted in Fig. 1(b), and use 3D convolution [1] to construct a dedicated network. A voxel-based view preserves physical size and has friendly memory localization. However, it is relatively sparse and requires very high resolution to eliminate the problem of lost quantisation information, which can increase the

✉ Chuanxu Wang
wangchuanxu_qd@qust.edu.cn

Jiajiong Li
19937731387@163.com

Chenyang Wang
wangchenyang981101@126.com

Min Zhao
zhaomin__99@163.com

Zitai Jiang
jzt1540381517@gmail.com

¹ School of Information Science and Technology, Qingdao University of Science and Technology, Zhongshan Street, Qingdao, Shandong Province 266100, China

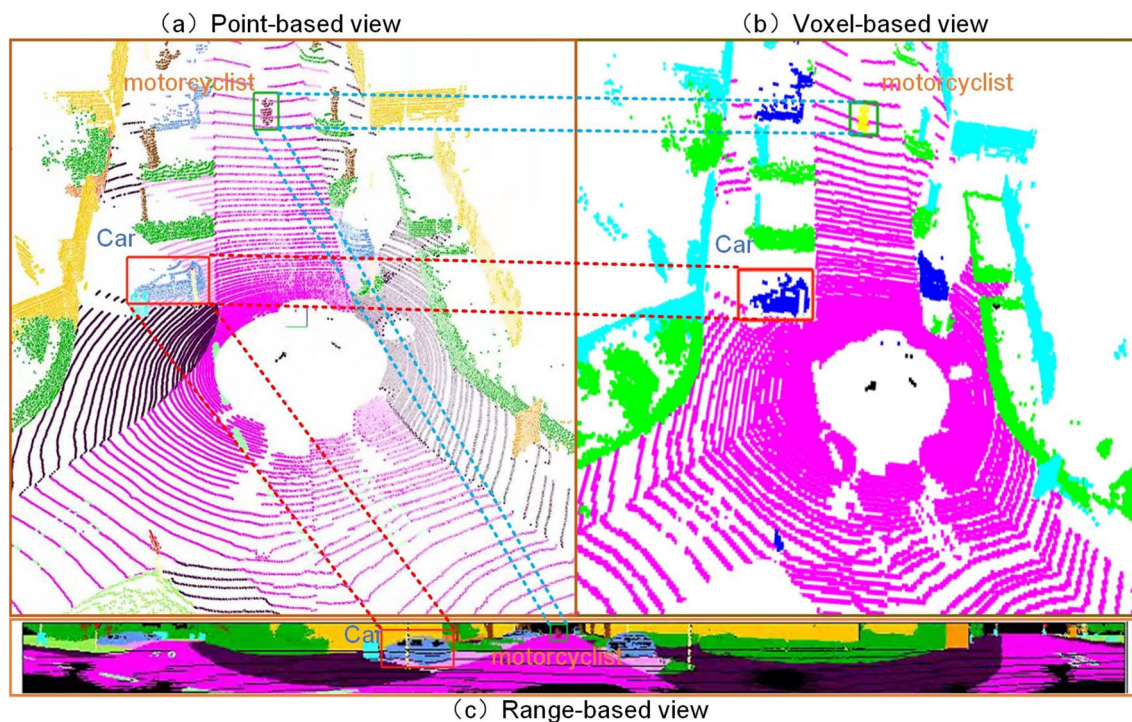


Fig. 1 Three different views of a point cloud. (a) Point-based view (top left): points are irregular and chaotic, which makes finding neighbors of points inefficient. (b) Voxel-based view (top right): the voxelization operation discretizes continuous point cloud data into discrete voxels,

introducing quantization loss and a sharp increase in computation with increasing resolution. (c) Range-based view (bottom): The range image distorts the physical dimensions due to spherical projection, while the depth information of the object is lost

computational and memory footprint by a cubic order of magnitude. Moreover, the different voxel sizes can greatly affect the performance of the network.

Recently, point-based views, depicted in Fig. 1(a), have attracted more and more attention so that much of the work uses raw point cloud data directly in the network. PointNet [2] served as a pioneering work for extracting point features using a multilayer perceptron (MLP) for each point, but it lacked local context modeling capability. Based on PointNet, later studies [3, 4] focused extensively on each point to extract local features by aggregating its neighborhood features. However, points in a point cloud are unstructured and searching the neighbourhood of a point is inefficient due to the random nature of memory access.

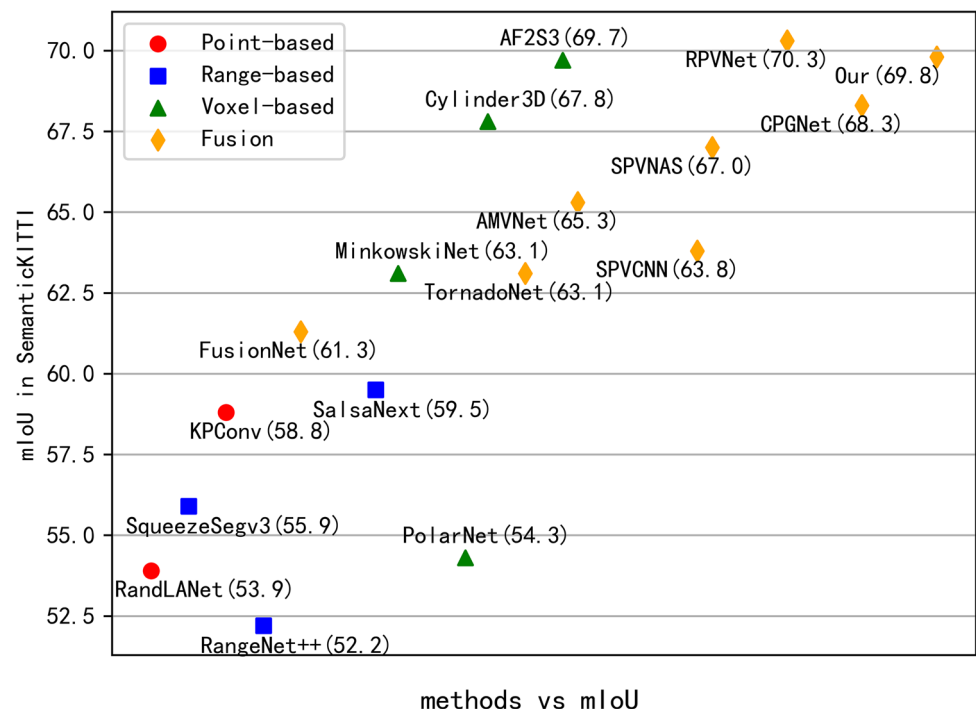
Some parallel works [5–8] followed a spherical projection scheme, i.e., range-based view, as shown in Fig. 1(c), which transformed the 3-dimensional information of the points into 2-dimensional information, and then utilized a well-established convolutional neural network for feature extraction, which had the advantage of densely representing the point cloud and utilizing the convolution kernel to gather a larger sensory field on the image, greatly eased the point sparsity problem. However, due to the spherical projection, the depth information of the points is partially lost

and the size and distance of the objects are changed, so the objects in a dense and cluttered scene may overlap with each other severely.

In conjunction with the results presented in Fig. 2, for the segmentation of large-scale point clouds, we find that: 1) the voxel-based approach is to outperform the point-based and range-based approaches, and the best performance achieved by the point-based and range-based ones is about the same; 2) the fusion-based approach is generally much higher than the point-based, range-based and voxel-based ones.

Out of instinct, it is natural to think of combining different views together to utilize complementary information, retaining strengths and reducing weaknesses. Recent research has attempted an approach called PVCNN [9], which employed a fusion strategy of point clouds and voxels. In this approach, voxels were used to provide coarse-grained local feature information, while the point clouds retained fine-grained geometric features by applying a simple MLP at each point. Whereas this approach provided a promising perspective, the simple additive fusion of point clouds and voxels does not result in significant performance gains, as shown in Fig. 2. Therefore, more sophisticated and effective fusion strategies are needed to further improve the performance. RPVNet [10] was the first network to fuse three views, which leveraged a

Fig. 2 Performance distribution of different methods on the SemanticKITTI dataset



hashing mechanism to record the location information of the same points in different views and performed feature fusion across views. This offers a promising prospect.

In this thesis, we propose a deeply adaptive range-point-voxel network that is intended to come to synergise the presentation of three views. See Fig. 4 for more details, where we design an RPVLayer with the original point cloud as input, and transform the point cloud into ranges, points and voxels, respectively, which are all mapped back to the features of the points after the features are extracted. In order to minimize the loss of point quality, the final feature alignment and fusion is performed at the point level after feature re-inference for anomalous points. Unlike previous approaches [11–15] that fused upstream and downstream of the network, we consider that less information is lost by convergence in the middle of the network. Our network can not only reduce the problems caused by range branch and voxel branch in collaboration with the FRM module, depicted in Fig. 6, but also efficiently capture fine-grained information and structural features at the point level from different views at different levels through the continuous iteration of the FGA module. The details of each module of the FGA will be described in Section 3.1.

Our main contributions are as follows:

- We design a novel range-point-voxel interactive structure which overcomes the drawbacks of the point-based, voxel-based and range-based single view approaches by allowing different views to compensate and enhance each other in a softer way.
- Considering the excellent performance of Convolutional Neural Networks in the field of 2D images, we propose a novel extractor for modeling contextual modules called Dynamic Pyramid Feature Extractor. We construct a dynamic skeleton network in the range branch, specifically designed for processing spherical distance images LiDAR data, which extracts features from the range branch at different scales suitable for semantic segmentation.
- We propose a Feature Refinement Module, which divides the points into abnormal points (points with low confidence) and normal points (points with high confidence). For the points with low confidence, we use their own features, the features of the point branch and the features of the neighboring points to reason jointly, and then finally perform the calculation of the self-attention. The problem of losing the quality of points when restoring features back to the point level in range view and voxel view is solved.
- We propose a novel fusion mechanism, the channel-based multiple fine-grained self-attention mechanism, which contains only one attention layer who captures and fuses different fine-grained features of the same object from different views.
- We conduct detailed experiments, the experimental results show that our method achieves 69.8% mIoU and 77.1% mIoU on the SemanticKITTI [16] dataset and nuScenes

[17] dataset, respectively, which outperforms most state-of-the-art algorithms.

2 Related work

In this section, we present the exploration of attention mechanisms in the field of point cloud segmentation, along with the four conventional research directions.

2.1 Self-attention in point cloud segmentation

Self-Attention (SA) is a very powerful neural network operation for capturing complex dependencies among elements in sequential or ensemble data. In the field of point cloud learning, SA has been widely used as a core module. Some early works [18, 19] used SA as an auxiliary module to advance the development of point cloud learning.

Recent studies have shown that point cloud learning methods based on Transformer models have great potential. PT (Point Transformer) [20] and PCT (Point Cloud Transformer) [21] were two purely Transformer models that have achieved significant performance in point cloud task improvement. In particular, the performance of PT [20] far surpassed the previous methods [22, 23], which suggested that the Transformer model has a great advantage in modelling non-locality of point clouds and multi-scale feature fusion in point cloud learning. Subsequent works, such as PCTMA-Net [24] and Hierarchical Transformer [25], further improved the performance of the point cloud Transformer by introducing the standard multi-head attention mechanism and shift window mechanism.

However, all of the above methods were based on point-level self-attention mechanisms, and the complexity of the point cloud increased quadratically with the number of input points, which led to the limitation of the local receptive field. Therefore, many subsequent research works [26–29] have adopted voxels as attention and proposed voxel-level self-attention mechanisms. These methods usually used a de-voxelization operator to obtain a point-by-point feature representation.

In addition, PatchFormer [30] proposed a method to compute the attention graph between points and patches by directly aggregating the patch features for each point. These methods have made significant progress in considering multi-scale features, but they still faced the challenge of size-aware feature learning, i.e., how to effectively capture and utilise the different scale information in the input data. In point cloud tasks, size-aware feature learning is important for dealing with point cloud objects of different sizes and complexities, thus this remains an area of intensive research. To the best of our knowledge, we are the first network to fuse three views at the point level with the help

of self-attention. We propose a range-point-voxel multifine-grained self-attention in RPVLayer, which differs from the voxel attention described above in that our range-point-voxel multifine-grained self-attention directly captures the relationship among range-point-voxels. Furthermore, fine-grained and coarse-grained labelling is preserved in the attention layer, enabling the free unravelling of multi-scale features in integrated point cloud learning.

2.2 Point-based segmentation

PointNet [2] was the first attempt to directly process point cloud data through a network based on MLP. Although subsequent studies [3, 4] demonstrated the effectiveness on indoor point cloud data, most of these methods could not be easily scaled up to process large-scale outdoor data due to computational and memory constraints. RandLA-Net [31] employed random sampling and local feature aggregation to reduce the information loss introduced by random sampling, which provided a feasible method to accelerate point cloud processing, but it still could not overcome the accuracy loss caused by sampling. KPConv [4] achieved the best performance of current point-based methods by introducing spatial kernel-based point convolution. However, faced with the same problem of dealing with large scene data, it is still unable to train directly using all of the data. To solve this problem, a method of balanced sampling by radius classification has been proposed to reduce the amount of data. However, this sampling method may destroy some of the information inherent in the point cloud. Although point-based methods usually have fewer parameters [31], they inevitably involve inefficient local neighbor search operations, which limits their efficiency and performance when dealing with large-scale scenes.

2.3 Voxel-based segmentation

In early voxel-based approaches [32, 33], the point cloud data was converted into voxel grids of points and a common 3D convolution operation was subsequently applied for semantic segmentation. However, recent research efforts [34, 35] have been devoted to accelerating the 3D convolution operation while improving the performance and reducing the computational cost. Meanwhile, there were some variant approaches [36, 37] involving the segmentation of 3D space. Among them, Cylinder3D [37] introduced asymmetric residual blocks to reduce the computational burden and ensure that features associated with rectangular objects are captured. AF2S3Net [38], on the other hand, achieved an advanced level of performance based on previous work [33] by introducing two novel attentional modules, the Attentional Feature Fusion Module (AF2M) and the Adaptive Feature Selection Module (AFSM). These modules efficiently

learnt local and global contextual information and emphasized fine detail information. In addition, the method used a hybrid loss function with geometric-aware anisotropy [39] to recover fine detail information. Conventional voxel methods may face a serious problem of information loss when the resolution is reduced. However, our method employs a multi-view approach to compensate for this shortcoming, which improves the recovery of detailed information.

2.4 Range-based segmentation

Range-based point cloud segmentation methods [6, 7, 23, 40, 41] utilized 2D Convolutional Neural Networks (CNNs) by mapping 3D point clouds into dense 2D spherical meshes. For example, RangeNet++ [6] employed the DarkNet backbone network from YOLOv3 [42] as a feature extractor with efficient K Nearest Neighbour (KNN) post-processing. SalsaNext [7] used Salnet [43] as a baseline and also introduced an uncertainty-aware mechanism for point feature learning. In addition, KPRNet [40] stood out in this class of methods by employing a powerful ResNeXt-101 backbone network and an Astrous Spatial Pyramid Pooling layer to achieve state-of-the-art segmentation performance, as well as the innovative use of KPConv [4] in place of the less efficient KNN as a segmentation head method. Although range-based methods can leverage well-established 2D image segmentation techniques, mapping 3D point clouds to 2D spherical meshes introduces the problem of distortion of physical dimensions. This problem is avoided in our proposed RPV-CASNet, which employs a multi-view interactive learning approach to process 3D point cloud data more accurately, alleviating the size distortion of objects by a large extent.

2.5 Multi-view fusion

Given that there are some limitations with single view, recent research approaches [44–49] have attempted to fuse information from two or more different views to improve point cloud segmentation performance. As an example, the approach in [11] combined point-level information from bird's eye view and range images, which were then passed to a subsequent network for early fusion. AMVNet [13], on the other hand, devised a late fusion strategy by calculating the uncertainty in the outputs of different views and further refining the results with the help of an additional network. FusionNet [14] proposed a point-voxel interaction MLP for aggregating features between neighbourhood voxels and corresponding points, which reduced the time cost of neighbourhood search and maintained satisfactory accuracy when processing large-scale point clouds. In particular, PVCNN [9] proposed an effective point-voxel fusion method in which the voxels provide coarse-grained local feature information while the points retain fine-grained geometric features through a sim-

ple MLP. [50, 51] fused RGB pictures and range images, but it needed to fuse images and point clouds in a hard correlation manner with the help of a calibration matrix, and thus was susceptible to calibration errors [52]. A simple additive fusion strategy for RPVNet [10] in multi-view cross-fertilisation ignored the fact that the noise introduced by the points in the mapping for the Range view is also incorporated into the other modalities, which can severely handicap the performance of the network. The aforementioned methods face the same problem i.e. the fusion strategy is also relatively simple, e.g. using additive operations, which limits the performance of the network. However, our method is able to utilize the information provided by multiple views more efficiently and has the ability to select and fuse the most helpful parts for point cloud segmentation. This means that our approach has greater potential and flexibility in improving the performance of point cloud segmentation.

3 FGA transformer block

In this section, we first introduce our overall network architecture, and then introduce the components of this network, exemplified by the RPVLayer in the FGA module, which consists of three parts: the Dynamic Feature Pyramid Extractor (DFPE), the Feature Refinement Module (FRM), and the Multi-Fine-Grained Feature Self-Attention Module (MFGFSAM). Finally, an improved and efficient data enhancement method we proposed will also be presented.

3.1 Framework overview

Our proposed algorithm, named RPV-CASNet, is built on top of the FGA Transformer Block, whose core idea is to introduce effective multiscale features for each attentional layer and allow each point to adapt to its attentional domain. Therefore, our network completely relies on the FGA block to learn and interact with channel features from different viewpoints to achieve fusion of multi-fine-grained features from objects of different scales. The details of the FGA block are shown in Fig. 3(b). In order to efficiently capture features from different views and scales in each attention layer, the FGA must overcome three challenges: loss of point quality, integration of features from distant regions, and accurate discrimination of multiscale features from objects at different scales. To address the first challenge, drawing on Video Anomaly Detection (VAD) ideas, we design a FRM that handles anomalies and re-inferences point features in a more subtle way. To address the second challenge, we design a new range-point-voxel interaction structure on which we use a novel channel-based Multi-Fine-Grained Attention Mechanism for planning the learning of effective multi-scale features for the point cloud. And to overcome the third challenge, we intro-

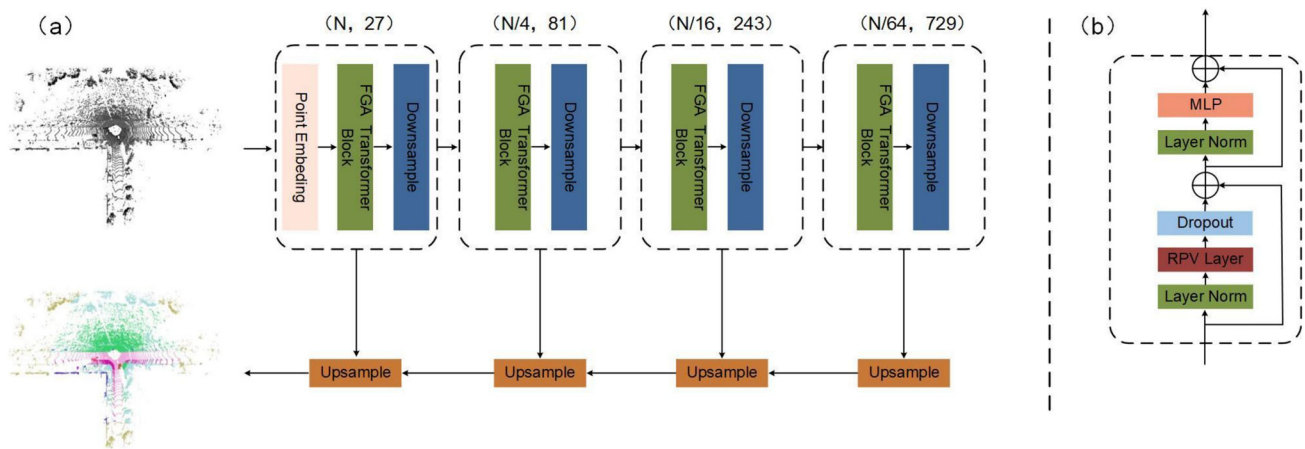


Fig. 3 (a)Overview (b) FGA Transformer Block

duce a range-point-voxel triple decoupling strategy based on the standard multi-head self-attention mechanism. We also propose a simple but effective DFPE, and use it to dynamically build our DFPE module for extracting features at different scales of the range branch.

3.2 Rang-point-voxel cross adaptive layer

Our RPVLayer is shown in Fig. 4, it contains three main parts, which are Feature Extraction, FRM and MFGFSAM. Specifically, the input point cloud first passes through a point embedding layer with linear transformation, followed by feature extraction using a simple MLP, which will then be

transformed into three branches. For the range branch we use a DFPE as a backbone network. For the point branch we deploy the lightweight PointNet++, and in particular, for the voxel branch we use the lightweight Spvans network. Finally we obtain the channel features for the three sequences of points, which will be fused in our FRM and MFGFSAM.

3.2.1 Dynamic feature pyramid extractor

Here we introduce an extractor for modeling the context module, called “Dynamic feature pyramid extractor”, or DFPE for short, which aims to handle the mapping of the point cloud into “image” features in the range branch. DFPE uses regular

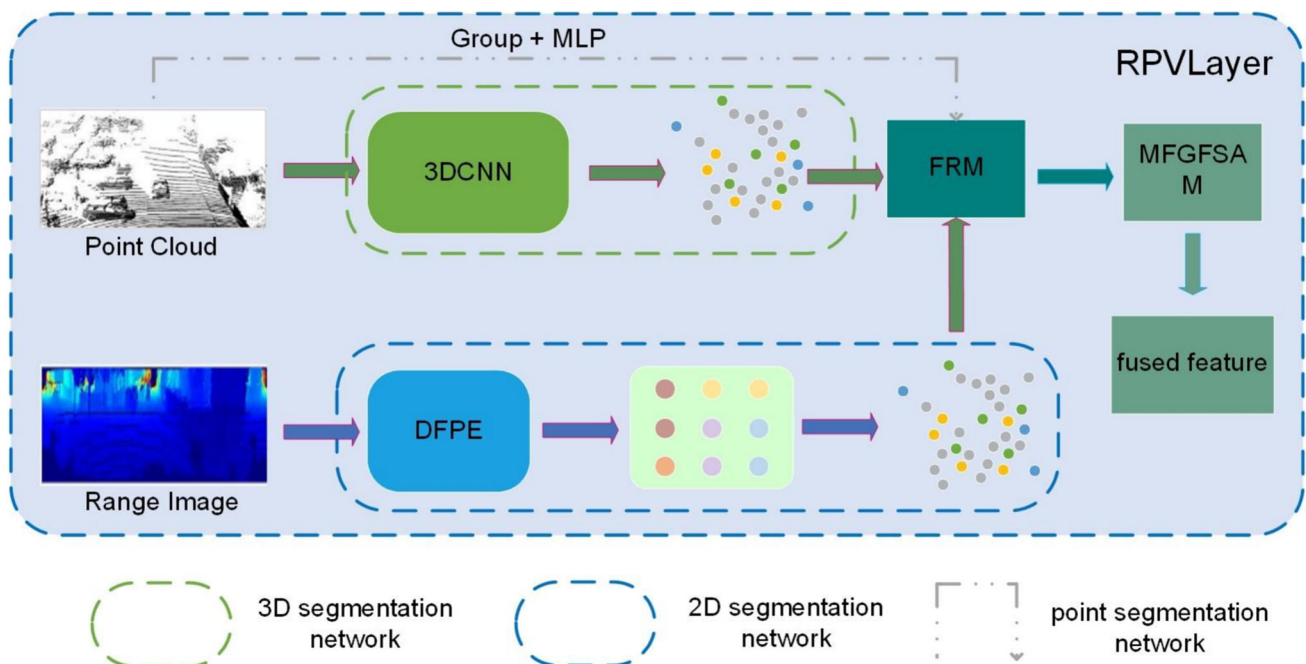


Fig. 4 Rang-Point-Voxel Cross Adaptive Layer

2D convolutional operations and provides contextual information at different scales through efficient computational dynamics, which can be used to extend DFPE to a wide range of models non-destructively. The following figure illustrates a generic block diagram of DFPE (see Fig. 5).

The DFPE module consists of three parallel dynamic residual blocks (shown as dashed boxes). Inspired by [15], in order to better introduce contextual information and optimise the feature extraction process, we use different sizes of convolutional kernels to redesign our DFPE module. In the actual deployment, the dynamic residual block mainly consists of Conv 3×3 ($S=1, P=1$), Conv 5×5 ($S=1, P=2$), Conv 7×7 ($S=1, P=3$) and Conv 1×1 2D convolution kernels. Each convolutional layer is followed by a Batch Normalisation layer and a ReLU activation layer, except for the 1×1 convolutional layer in the dynamic feature pyramid extractor (no ReLU activation layer here). Also, in order to preserve local features, we introduce a skip connection with a 2D convolution after the first layer convolution within each residual block. To further reduce the computational overhead, we use Dwconv with the same convolutional kernel size. Finally, the three parallel convolutional blocks are subjected to a Conv 1×1 operation, thus generating the final feature tensor for further processing. Instead of following the complex deployment of networks such as RangNet++, we view that:

1. A portion of the points will be covered or occluded after mapping, and these points are not taken into account by the network rely.
2. Our extractor is very efficient although it contains only three sub-blocks. With the coordinated iteration of the downsampling operation and the FGA block, the number of points in the network decreases, and then after mapping,

the occluded points are exposed to some extent and captured by our dynamic extractor. 3. If a dedicated mapping-based network (encoder and decoder) is deployed, it will introduce additional memory as well as computational overhead. We use the same architecture in both datasets. Details about the deployment will be presented in Section 4.2.

3.2.2 Feature refinement module

The point cloud introduces noise or loses some information after mapping and voxelization operations, which is an important challenge in multi-view fusion. Motivated by VAD, we propose a module called FRM, which is placed before the MFGFSAM. The exact deployment is shown in Fig. 6.

Specifically, the inputs $R \{R_{x_m}\}_{m=1}^N / V \{R_{x_n}\}_{n=1}^N$ and $P \{R_{x_k}\}_{k=1}^N$ to our FRM module are the sequential sets of three point features in the feature extraction stage, which are rfeature/vfeature collections from range branch / voxel branch and pfeature collections from point branch as inputs. We select the rfeature/vfeature set and the pfeature set as inputs, and after multiplication of vectors and sigmoid activation function, we aim to screen out the point set n ($n < N$) as anomalous points. For the anomalous point sequence n in range branch/voxel branch, we embed the features of range branch/voxel branch, features of point branch and coordinate information of anomalous points together with the help of point branch lossless point features and threshold α with joint different weights S and $(1 - S)$. With the help of KDTree we can easily obtain the feature infor-

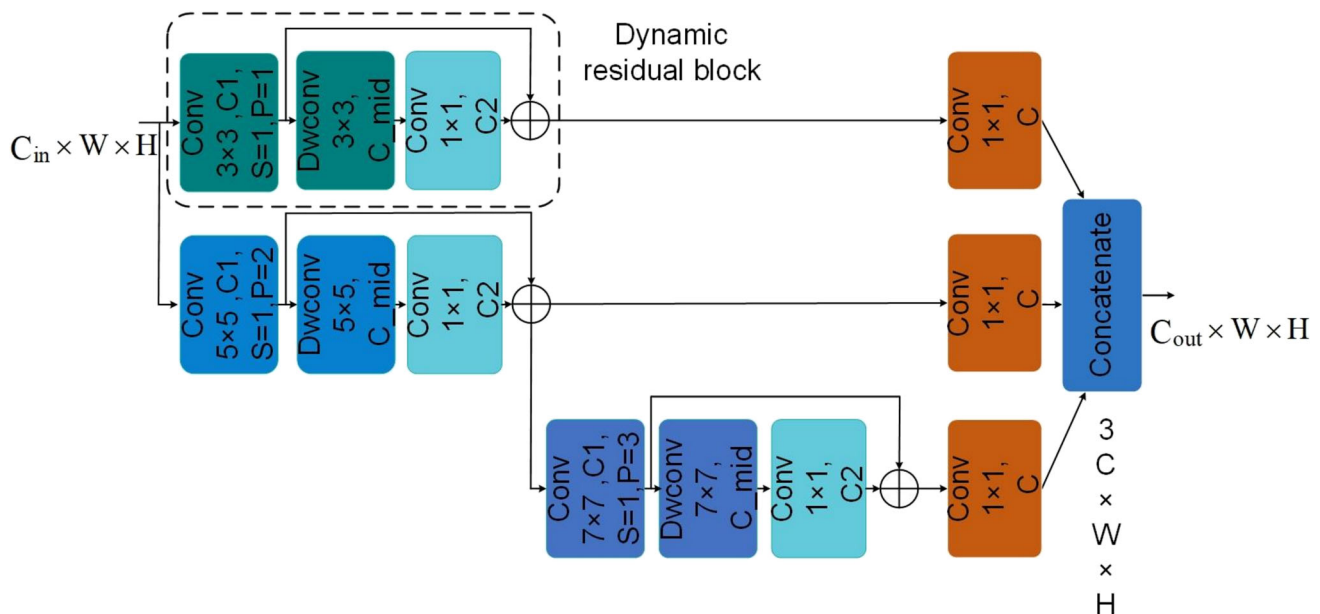
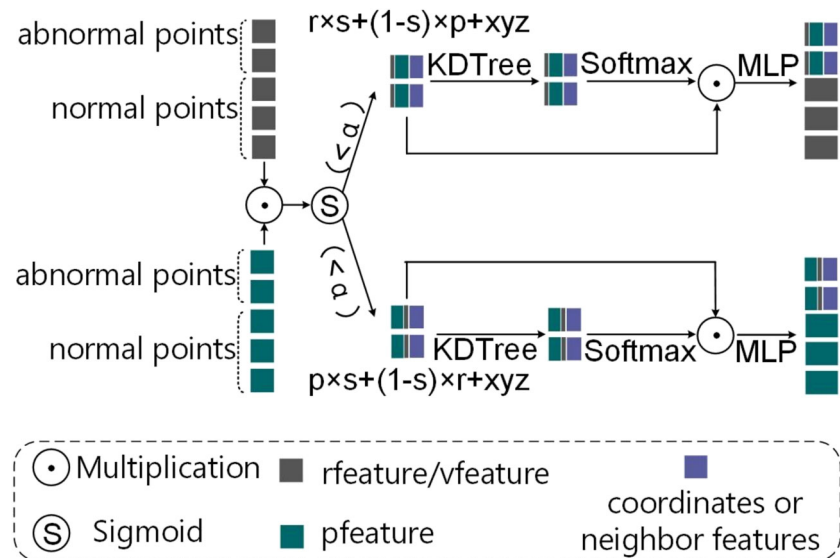


Fig. 5 Dynamic feature pyramid extractor

Fig. 6 Feature Refinement Module

mation of the neighbouring points around the anomalous point, we select 16 neighbouring points, and finally splice the three together according to the number of channels, and then multiply them with the original features as the final re-representation of the anomalous point features after obtaining the corresponding weight matrices through a linear layer and a Softmax layer. On the contrary, we simply “copy” the normal points. Similarly, we perform the same operation for the anomalies in the point branch. One of our

flashpoints is that when the sequence of anomalies is separated, both the range branch / voxel branch and point branch perform a re-inference representation of the anomalies, with the benefit of preventing the network from over-relying on the point branch. Finally, we get the re-inference representation of the feature set at the point level of R, P and V, which are $R \left\{ \{R_{x_m}\}_{m=1}^a, \{R_{x_m}\}_{m=a}^N \right\}$, $V \left\{ \{R_{x_n}\}_{n=1}^b, \{R_{x_n}\}_{n=b}^N \right\}$, $P \left\{ \{R_{x_k}\}_{k=1}^c, \{R_{x_k}\}_{k=c}^N \right\}$, where $\{R_{x_m}\}_{m=a}^N$, $\{R_{x_n}\}_{n=b}^N$ and

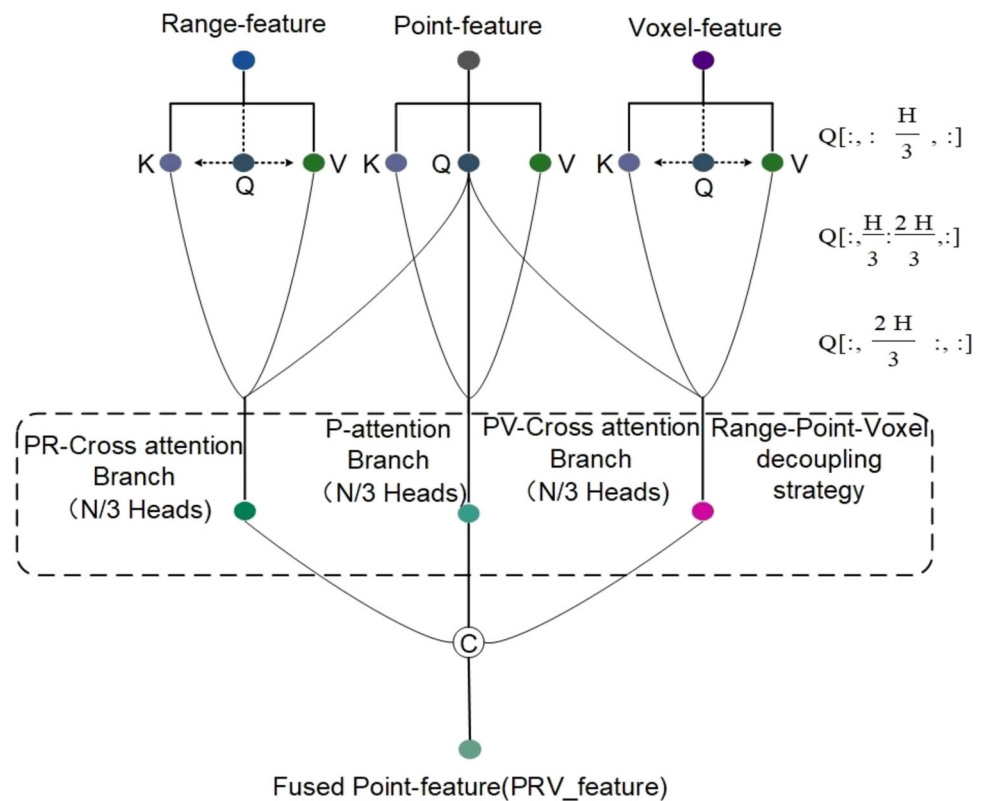
Fig. 7 Multi-Fine-Grained Feature Self-Attention Module

Table 1 Detailed deployment of the network

	Output Size	Layer Name	SemanticKITTI	nuScenes
Stage 1	27	FGAblock	$r_i = \begin{cases} 2, & i \leq \frac{Head}{3} \\ 4, & \frac{Head}{3} < i < \frac{2Head}{3} \\ 8, & i \geq \frac{2Head}{3} \end{cases}$ $H = 3, V = 0.04, N = 2,$ $h \times w = 64 \times 2048$	$r_i = \begin{cases} 2, & i \leq \frac{Head}{3} \\ 4, & \frac{Head}{3} < i < \frac{2Head}{3} \\ 8, & i \geq \frac{2Head}{3} \end{cases}$ $H = 3, V = 0.04, N = 2,$ $h \times w = 32 \times 2048$
Stage 2	81	FGAblock	$r_i = \begin{cases} 2, & i \leq \frac{Head}{3} \\ 4, & \frac{Head}{3} < i < \frac{2Head}{3} \\ 8, & i \geq \frac{2Head}{3} \end{cases}$ $H = 9, V = 0.06, N = 6,$ $h \times w = 64 \times 1024$	$r_i = \begin{cases} 2, & i \leq \frac{Head}{3} \\ 4, & \frac{Head}{3} < i < \frac{2Head}{3} \\ 8, & i \geq \frac{2Head}{3} \end{cases}$ $H = 9, V = 0.06, N = 4,$ $h \times w = 32 \times 1024$
Stage 3	243	FGAblock	$r_i = \begin{cases} 1, & i \leq \frac{Head}{3} \\ 2, & \frac{Head}{3} < i < \frac{2Head}{3} \\ 4, & i \geq \frac{2Head}{3} \end{cases}$ $H = 27, V = 0.08, N = 6,$ $h \times w = 64 \times 512$	$r_i = \begin{cases} 1, & i \leq \frac{Head}{3} \\ 2, & \frac{Head}{3} < i < \frac{2Head}{3} \\ 4, & i \geq \frac{2Head}{3} \end{cases}$ $H = 27, V = 0.08, N = 4,$ $h \times w = 32 \times 512$
Stage 4	729	FGAblock	$r_i = \begin{cases} 1, & i \leq \frac{Head}{3} \\ 2, & \frac{Head}{3} < i < \frac{2Head}{3} \\ 4, & i \geq \frac{2Head}{3} \end{cases}$ $H = 81, V = 0.1, N = 2,$ $h \times w = 64 \times 256$	$r_i = \begin{cases} 1, & i \leq \frac{Head}{3} \\ 2, & \frac{Head}{3} < i < \frac{2Head}{3} \\ 4, & i \geq \frac{2Head}{3} \end{cases}$ $H = 81, V = 0.1, N = 2,$ $h \times w = 32 \times 256$

In each stage, H, V, N, h, w are the number of heads, voxel size, number of blocks and size of the range image, respectively. r_i is the downsampling rate of the i th head in the multi-head self-attention layer. The bold text indicates the best performance in terms of a category or module or an evaluation metric during the course of the experiment.

Table 2 Evaluation on the SemanticKITTI dataset. Our results use GANMix (see Section 3.3), where ‡ indicates that GANMix is used for data enhancement

Methods	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road
PointNet [2]	14.6	46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6
RandLANet [31]	53.9	94.2	26.0	25.8	40.1	38.9	49.2	48.2	7.2	90.7
RandLANet ‡ [31]	55.4	94.2	38.2	41.8	40.1	38.9	49.2	48.2	7.2	90.7
KPConv [4]	58.8	96.0	30.2	42.5	33.4	44.3	61.5	61.6	11.8	88.8
SqueezeSegv3 [5]	55.9	92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7
RangeNet++ [6]	52.2	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	91.8
SalsaNext [7]	59.5	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7
PolarNet [36]	54.3	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8
Cylinder3D [37]	67.8	97.1	67.6	64.0	59.0	58.6	73.9	67.9	36.0	91.4
Cylinder3D ‡ [37]	69.5	97.1	81.8	82.6	59.0	58.6	73.9	67.9	36.0	91.4
AF2S3 [38]	69.7	94.5	65.4	86.8	39.2	41.1	80.7	80.4	74.3	91.3
FusionNet [14]	61.3	95.3	47.5	37.7	41.8	34.5	59.5	56.8	11.9	91.8
TornadoNet [15]	63.1	94.2	55.7	48.1	40.0	38.2	63.6	60.1	34.9	89.7
AMVNet [13]	65.3	96.2	59.9	54.2	48.8	45.7	71.0	65.7	11.0	90.1
SPVNAS [34]	67.0	97.2	50.6	50.4	56.6	58.0	67.4	67.1	50.3	90.2
RPVNet [10]	70.3	97.6	68.4	68.7	44.2	61.6	75.9	74.4	73.4	93.4
CPGNet [58]	68.3	96.7	62.9	61.1	56.7	55.3	72.1	73.9	27.9	92.9
Our	69.8	97.4	68.9	67.8	44.0	62.2	76.4	68.4	33.4	93.1

Table 2 continued

Methods	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign
PointNet [2]	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7
RandLANet [31]	60.3	73.7	20.4	86.9	56.3	81.4	61.3	66.8	49.2	47.1
RandLANet ‡ [31]	60.3	73.7	20.4	86.9	56.3	81.4	61.3	66.8	49.2	47.1
KPConv [4]	61.3	72.7	31.6	90.5	64.2	84.8	69.2	69.1	56.4	47.4
SqueezeSegv3 [5]	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9
RangeNet++ [6]	65.0	75.2	17.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9
SalsaNext [7]	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	62.1
PolarNet [36]	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5
Cylinder3D [37]	65.1	75.5	32.3	91.0	66.5	85.4	71.8	68.5	62.6	65.6
Cylinder3D ‡ [37]	65.1	75.5	32.3	91.0	66.5	85.4	71.8	68.5	62.6	65.6
AF2S3 [38]	68.8	72.5	53.5	87.9	63.2	70.2	68.5	53.7	61.5	71.0
FusionNet [14]	68.8	77.1	30.8	92.5	69.4	84.5	69.8	68.5	60.4	66.5
TornadoNet [15]	66.3	74.5	28.7	91.3	65.6	85.6	67.0	71.5	58.0	65.9
AMVNet [13]	71.0	75.8	32.4	92.4	69.1	85.6	71.7	69.6	62.7	67.2
SPVNAS [34]	67.6	75.4	21.8	91.6	66.9	86.1	73.4	71.0	64.3	67.3
RPVNet [10]	70.3	80.7	33.3	93.5	72.1	86.5	75.1	71.7	64.8	61.4
CPGNet [58]	68.0	78.1	24.6	92.7	71.1	84.6	72.9	70.2	64.5	71.9
Our	71.2	78.6	32.7	92.7	71.3	86.3	74.8	70.5	64.4	72.1

The bold text indicates the best performance in terms of a category or module or an evaluation metric during the course of the experiment

$\{R_{x_k}\}_{k=c}^N$ are the “anomaly” feature after re-inferred representation in the corresponding set.

Regarding the division of anomalies normal points are set as follows:

$$\psi = \begin{cases} 0 & , \text{ if } \alpha < \delta \\ 1 & , \text{ if } \alpha > \delta \end{cases} \quad (1)$$

We make the distinction between abnormal and normal points by using agent task ψ . Specifically, if the value of α is less than δ , the agent task ψ treats the point features that are less than δ as abnormal features and then further extracts the features of abnormal and abnormal points. Conversely, they are considered normal points.

3.2.3 Multi-fine-grained feature self-attention module

MFGFSAM does not choose to be at the voxel level but chooses to compute the channel attention from point sequences, voxel sequences and range sequences directly at the point level, where the attention is computed as the channel feature of the query sequence from the point sequence, the branch that has pure features, and the key and value are derived from the channel features obtained from itself and the other two branches, respectively. Thus our MFGFSAM is more flexible. As shown in the Fig.7.

The inputs to this module are the range features $R_f \in R^{D_{nr} \times (D_h \times D_d)}$, D_{nr} is the number of points in the range branch in the nth FGA transformation block, and $(D_h \times D_d)$ is the dimension of the points obtained by going through rang to point in it. Similarly we have $P_f \in R^{D_{np} \times (D_h \times D_d)}$, D_{np} is the number of points in the point branch in the nth FGA transformation block, and $(D_h \times D_d)$ is the feature dimension of the point obtained therein. $V_f \in R^{D_{nv} \times (D_h \times D_d)}$, D_{nv} is the number of points in the voxel branch in the nth FGA transformation block, and $(D_h \times D_d)$ is the channel feature dimensions of the points obtained from the points that pass through the voxel to point in it. In particular, in order to facilitate the subsequent channel self-attention computation, $D_{nr} = D_{np} = D_{nv}$ in the nth conversion block. The RPVLayer in the nth FGA conversion block can be computed in accordance with the following (2)-(9):

$$R_f, P_f, V_f = LN(R_f), LN(P_f), LN(V_f) \quad (2)$$

$$Q_R, K_R, V_R = \psi(R_f | r_i)W_1^Q, \psi(R_f | r_i)W_1^K, \psi(R_f | r_i)W_1^V \quad (3)$$

$$Q_V, K_V, V_V = \psi(V_f | r_i)W_2^Q, \psi(V_f | r_i)W_2^K, \psi(V_f | r_i)W_2^V \quad (4)$$

$$K_R, V_R = Q_R + K_R, Q_R + V_R$$

Table 3 Evaluation on the nuScenes dataset. Our results do not use any GANMix-like trick

Methods	mIoU	barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other flat	sidewalk	terrain	manmade	vegetation
RangeNet++[6]	65.5	66.0	21.3	77.2	80.9	30.2	66.8	69.6	52.1	54.2	72.3	94.1	66.6	63.5	70.1	83.1	79.8
SalsaNext[7]	71.0	74.7	28.2	85.3	90.9	35.1	77.5	71.3	58.8	57.4	76.1	96.5	71.1	74.7	74.0	87.3	85.7
PolarNet [36]	72.2	74.8	34.1	85.9	88.4	42.2	72.4	72.2	63.1	61.3	76.5	96.0	70.8	71.2	71.5	86.7	84.4
Cylinder3D[37]	76.1	79.8	32.4	82.2	86.4	62.5	81.9	75.3	72.3	83.5	65.1	97.4	67.0	78.8	74.6	90.8	87.9
AMVNet [13]	76.1	76.4	40.3	91.2	93.8	51.3	78.0	78.9	64.9	62.1	84.4	96.8	71.6	76.4	75.4	90.5	87.4
RPVNet[10]	77.6	78.2	43.4	92.7	93.2	49.0	85.7	80.5	66.0	66.9	84.0	96.9	73.5	75.9	76.0	90.6	88.9
Our	77.1	78.0	44.2	92.3	93.4	51.9	80.9	80.7	65.1	65.4	85.0	96.1	72.2	74.6	75.6	90.1	88.1

The bold text indicates the best performance in terms of a category or module or an evaluation metric during the course of the experiment

$$K_V, V_V = Q_V + K_V, Q_V + V_V \quad (5)$$

$$Q_p^1, Q_p^2, Q_p^3 = P_f W_1^Q, P_f W_2^Q, P_f W_3^Q \quad (6)$$

$$F_{coarse1}^{PR}, F_{coarse2}^{PV} = MSA_{\frac{H}{3}}(Q_p^1, K_R, V_R), \\ MSA_{\frac{H}{3}}(Q_p^2, K_V, V_V) \quad (7)$$

Where LN is layer normalization, W is a linear matrix, this MSA is channel-based multi-head self-attention, and $\psi(V_f | r_i)$ is the downsampling rate of the feature in the i th head. We reshape the input features (B, N, C) to (B, H, W, C) , and then use a convolution kernel with step size r_i to implement the downsampling layer. $F_{coarse1}^{PR}, F_{coarse2}^{PV} \in R^{D_{nr} \times (D_{\frac{H}{3}} \times D_d)}$, all linear matrices W are used to project input vectors at the channel level onto output vectors with only one-third of the input dimension. In the specific deployment, the MSA is performed in parallel. In addition, we further reduce the computational consumption by reducing the query, key, and value dimensions to one-third of the original ones through the linear layer. Our RPYLayer can directly compute the attention maps between range, point and voxel, which makes our model well modeled between different fine-grains of range-point-voxel by channel features.

3.2.4 Range-point-voxel triple decoupling strategy

Shown in Fig. 7, MFGFSAM contains point-attention branch, point-range attention branch and point-voxel attention branch. For point-range attention branch and point-voxel attention branch, they model coarse fine-grained features with larger sense fields, and for the point-attention branch it can better capture fine-grained features at a micro-fine scale. Inspired by Shunted Transformer [53], similarly we use a channel-based multi-head self-attention mechanism in RPYLayer to combine multi-scale rough features from point-range attention branch, point-voxel attention branch and fine-grained features from point attention branch. In particular, our RPYLayer can be computed as:

$$K_p, V_p = P_f W_3^K, P_f W_3^V \quad (8)$$

$$F_{fine3}^P = MSA_{\frac{H}{3}}(Q_p^3, K_p, V_p) \quad (9)$$

$$F' = \text{Concat}(F_{coarse1}^{PR}, F_{coarse2}^{PV}, F_{fine3}^P) \quad (10)$$

Where $F' \in R^{D_{nr} \times (D_h \times D_d)}$, D_{nr} is the output of the final RPYLayer layer, our RPYLayer is able to learn multiscale features from among different objects in an unmerged manner on a single attention layer. With the help of RPYLayer, FGA can combine a range of sensory field sizes from different objects at different levels.

Table 4 The comparisons between efficiency (run-time) and accuracy (mIoU) on the SemanticKITTI val set

view	#params(M)	latency(ms)	mIoU
RandLANet [31]	1.2	416	53.9
SqueezeSegv3 [5]	26.2	113	55.9
Cylinder3D [37]	56.3	75.1	65.9
SPVCNN [34]	21.8	150	63.8
CPGNet [58]	—	43	68.3
RPVNet † [10]	24.8	168	70.3
RPV-CASNet	28.2	115	69.8

Note that RPV-CASNet latency is tested on a single Nvidia GeForce RTX 3090 GPU, while † indicates the use of an NVIDIA TESLA V100 GPU

The bold text indicates the best performance in terms of a category or module or an evaluation metric during the course of the experiment

3.3 GANMix

Although some data enhancement methods [54, 55] have achieved good results in point cloud segmentation indoors, relatively few studies [56, 57] have been conducted for outdoor scenes. To address the problem of category imbalance in LIDAR semantic segmentation, we propose a data enhancement method called GANMix.

Empirically, by increasing the number of samples in a few categories, the network can predict less frequently occurring objects more accurately. Inspired by this, we extract object instances from scarce categories, such as bicycles or vehicles, from each frame in the training set and put them into a small sample pool. Unlike simply copying and pasting the same object over and over again, we generate a certain amount of realistic samples with the help of an Generative Adversarial Network (GAN) to provide greater robustness and realism. During training, we select samples uniformly at random from a small sample pool to keep the categories balanced. Then, we perform random operations such as scaling and rotation on these samples. In order to better match with the real environment, we randomly place these objects on the points of the ground category. Finally, we obtain some new sparse objects

Table 5 Effect of different view combinations on network performance

Views	mIoU(%)	
	SemanticKITTI	nuScenes
RP	47.6	56.4
PV	59.9	69.6
RV	60.4	69.9
RPV	62.9	72.2

The bold text indicates the best performance in terms of a category or module or an evaluation metric during the course of the experiment

Table 6 Impact of different modules on network performance

Method	Three Views	FRM	MFGFSAM	GANMix	mIoU(%)	
					SemanticKITTI	nuScenes
(a)	✓				60.8	68.4
(b)	✓	✓			64.3	73.2
(c)	✓	✓	✓		68.0	77.1
(d)	✓	✓	✓	✓	69.8	—

The bold text indicates the best performance in terms of a category or module or an evaluation metric during the course of the experiment

from other scenes and “paste” them into the current training scene to simulate objects in various environments.

This approach aims to address the problem of category imbalance in point cloud semantic segmentation and to improve the performance of the model by generating realistic samples.

4 Experiments and results

In the experimental part, firstly, we introduce the two datasets and the evaluation metrics in Section 4.1. Next, Section 4.2 describes the network training strategy and the detailed deployment, after that, the experimental results are given in Section 4.3. Finally, the ablation experiments and the visualisation demonstrations are described in Sections 4.5 and 4.7 respectively.

4.1 Datasets and evaluation metrics

We evaluate our RPV-CASNet on the SemanticKITTI dataset [16] and the nuScenes dataset [17].

Datasets SemanticKITTI [16], derived from the KITTI Vision Benchmark [15], is a large-scale dataset for the task of point cloud segmentation in driving scenarios. It consists of 43551 LiDAR scans from 22 sequences collected from German cities. Equipped with a Velodyne-HDLE 64 LiDAR, each scan has about 120 k points. These 22 sequences were divided into 3 groups, i.e., the training set (00 to 10, except 08, 19, 130 scans), the validation set (08, 4071 scans), and the test set (11 to 21, 20,351 scans.) SemanticKITTI provides up to 28 categories, but the official evaluation ignores categories with only a few points and merges categories with different movement states and categories, so a set of 19 valid categories was used.

nuScenes [17] is a newly released dataset for semantic segmentation in LiDAR containing 1000 scenes collected from different areas in Boston and Singapore. Each scene is 20 seconds long and is sampled at 20 Hz using a Velodyne HDL-32 E sensor, resulting in a total of 40,000 frames for nuScenes. It used 8130 samples for training, 6019 for validation and 6008 for testing. After merging similar classes and

removing rare classes, a total of 16 classes were retained for LiDAR semantic segmentation.

Evaluation metric Following the official recommendations of [16, 17], we use the mean intersection of all classes (mIoU) as the evaluation metric. mIoU can be formulated as:

$$mIoU = \frac{1}{C} \sum_{i=1}^C \frac{TP_C}{TP_C + FP_C + FN_C} \quad (11)$$

Where TP_C , FP_C , FN_C denote the true positive, false positive and false negative predictions for category c . C is the number of categories.

4.2 Training strategy and network deployment

In the training phase, we deploy a cross-entropy loss function. For both the SemanticKITTI dataset and the nuScenes dataset, we use the ADAM optimizer, and our learning rate has an initial trial value of 0.006 and is gradually reduced through multiple steps of decay. The number of epochs of training is 60 and 80 respectively. The training time of our network on Nvidia GeForce RTX 3090 take about a week. During training, we utilize widely used data enhancement strategies, including global scaling with a random scaling factor sampled from [0.95, 1.05], and global rotation around the Z-axis at a random angle. We also incorporate our proposed GANMix method in the last ten epochs of the training phase to further fine-tune the network. The detailed network deployment is shown in Table 1.

4.3 Results on SemanticKITTI and nuScenes

In this experiment, we compare the results of our method with most existing state-of-the-art LiDAR segmentation methods on the SemanticKITTI test set in Table 2 and on the nuScenes validation set in Table 3. Our method outperforms most models, in terms of mIoU metrics.

Our method, RPV-CASNet, has an overall segmentation performance almost equal to that of RPVNet [10] on both data, and the segmentation accuracy of some classes even exceeds that of the former. Due to the strong robustness of our multi-view interaction learning, RPV-CASNet is more

Table 7 The impact of different fusion methods on networks

Method	mIoU(%)	
	SemanticKITTI	nuScenes
Addition	64.3	71.8
Concatnation	65.5	73.2
MFGFSAM	68.0	77.1
Replacing MFGFSAM with ordinary multi head self attention	66.2	74.9

The bold text indicates the best performance in terms of a category or module or an evaluation metric during the course of the experiment

accurate than most state-of-the-art networks in terms of small target segmentation accuracy. On the SemanticKITTI dataset, RPV-CASNet improves segmentation accuracy by 6.0, 6.7, 6.9, 4.3, 5.5 and 0.3 percentage points compared to CPGNet [58] in the small target categories of bicycles, motorcycle, other vehicles, person, motorcyclist, and traffic signs, respectively. Compared to SPVNAS [34] in bicycle, motorcycle, other vehicles, person, bicyclist, pole, traffic signs also improved by 18.3, 17.4, 4.2, 9, 1.3, 0.1 and 4.8 percentage points respectively. The segmentation accuracy of the four small target classes such as bicycle, other vehicles, pedestrians, and traffic signs even exceeds that of RPVNet [10] by 0.5, 0.6, 0.5 and 0.7 percentage points, respectively. On the nuScenes dataset, RPV-CASNet improves the segmentation accuracy over AMVNet [13] by 3.9, 11.8 percentage points, over Cylinder3D [37] by 1.8, 5.4 percentage points, and over RPVNet [10] by 0.8 and 0.2 percentage points for the two small-target categories of bicycles and pedestrians. These results show that our model achieves significant improvements on the small target segmentation task.

4.4 Model appraisal

We provide a comparison of efficiency and accuracy as shown in Table 4. When parameters and latency are compared with other methods, although we reduce the complexity of the model when self-attention is computed, we still have a higher number of parameters, but our model is lower than SPVCNN, RandLANet and RPVNet in latency.

Table 8 Threshold setting for hyperparameter δ

	SemanticKITTI				nuScenes			
	Stage 1	Stage 2	Stage 3	Stage 4	Stage 1	Stage 2	Stage 3	Stage 4
PR_{δ}	0.45	0.45	0.35	0.35	0.45	0.45	0.40	0.40
PV_{δ}	0.40	0.40	0.35	0.35	0.40	0.40	0.35	0.35

The bold text indicates the best performance in terms of a category or module or an evaluation metric during the course of the experiment

4.5 Ablation studies

Performance of different view combinations We explore the effect of different view combinations on the network, including RP, RV, PV and RPV. We use a quarter of the training set and conduct validation on a validation set of two datasets. The results in Table 5 show that: 1. P-branch contributes the least to the performance of the whole network while V-branch contributes the most to the network. 2. P-branch and R-branch contribute almost equally to the network. 3. The best performance is achieved based on the three views of RPV.

Effects of different modules As shown in Table 6, we provide an exhaustive analysis of the overall performance of the network by each module of the network, where our FRM module has 3.5% and 4.8% improvement in the mIoU values on the two datasets respectively. While MFGFSAM possesses 3.7% and 3.9% enhancement on both the datasets respectively.

Variants of fusion style In the paper, we explore the impact of different fusion methods on the network performance in Table 7. We can see that our multi-fine-grained feature fusion module, on the SemanticKITTI dataset, gains 3.7 points more than using the addition and on the nuScenes dataset gains 3.9 points more than the Concatnation's method. We also compare our proposed channel-based Multi-Fine-Grained Self-Attention Mechanism with the normal multi-head self-attention mechanism, also with 1.8 and 2.2 points of improvement, respectively.

Threshold setting for hyperparameter δ We conduct a lot of experiments to determine the threshold of hyperparameter δ on different datasets, the experiments reveal that about 10% – 25% of the features are classified as anomalies. PR_{δ} denotes the hyperparameter setting situation between point view and range view, and PV_{δ} denotes the hyperparameter setting situation between point view and voxel view. See Table 8 for details.

Comparison of effects with different similarities We compare different methods for separating anomalies in FRM, where our method point multiplication and sigmoid on the SemanticKITTI dataset outperforms calculating the Euclidean distance between the two by 1.6 percentage points,

Table 9 Comparison of different methods of finding anomalies

Method	mIoU(%)	
	SemanticKITTI	nuScenes
Cosine similarity	64.3	73.1
Euclidean distance	66.4	75.3
Point Multiplication and Sigmoid	68.0	77.1

The bold text indicates the best performance in terms of a category or module or an evaluation metric during the course of the experiment

and calculating the cosine similarity has the lowest performance of 64.3 percentage points, with a difference of up to 3.7 percentage points between the two. In the same case, on the nuScenes dataset, our proposed method outperforms calculating the cosine similarity and Euclidean distance between

the two by 4.0 and 1.8 percentage points, respectively, as shown in Table 9. It can be insightful that it is very important to seek the same semantic features of the same object in different modalities, this is because the features of different modalities are difference in feature space, and the simple calculation of cosine similarity or Euclidean distance sometimes does not satisfy the needs of some complex tasks. This provides a brilliant prospect for our future work.

4.6 Scalability of GANMix

Due to restricted code availability, we use our GANMix on two algorithms, Cylinder3D and RandLANet, for the small target classes bicycle and motorcycle. As shown in Table 2, there is a dramatic improvement of 14.2, 18.6 and 12.2, 16.0 points in the segmentation accuracy for the bicycle

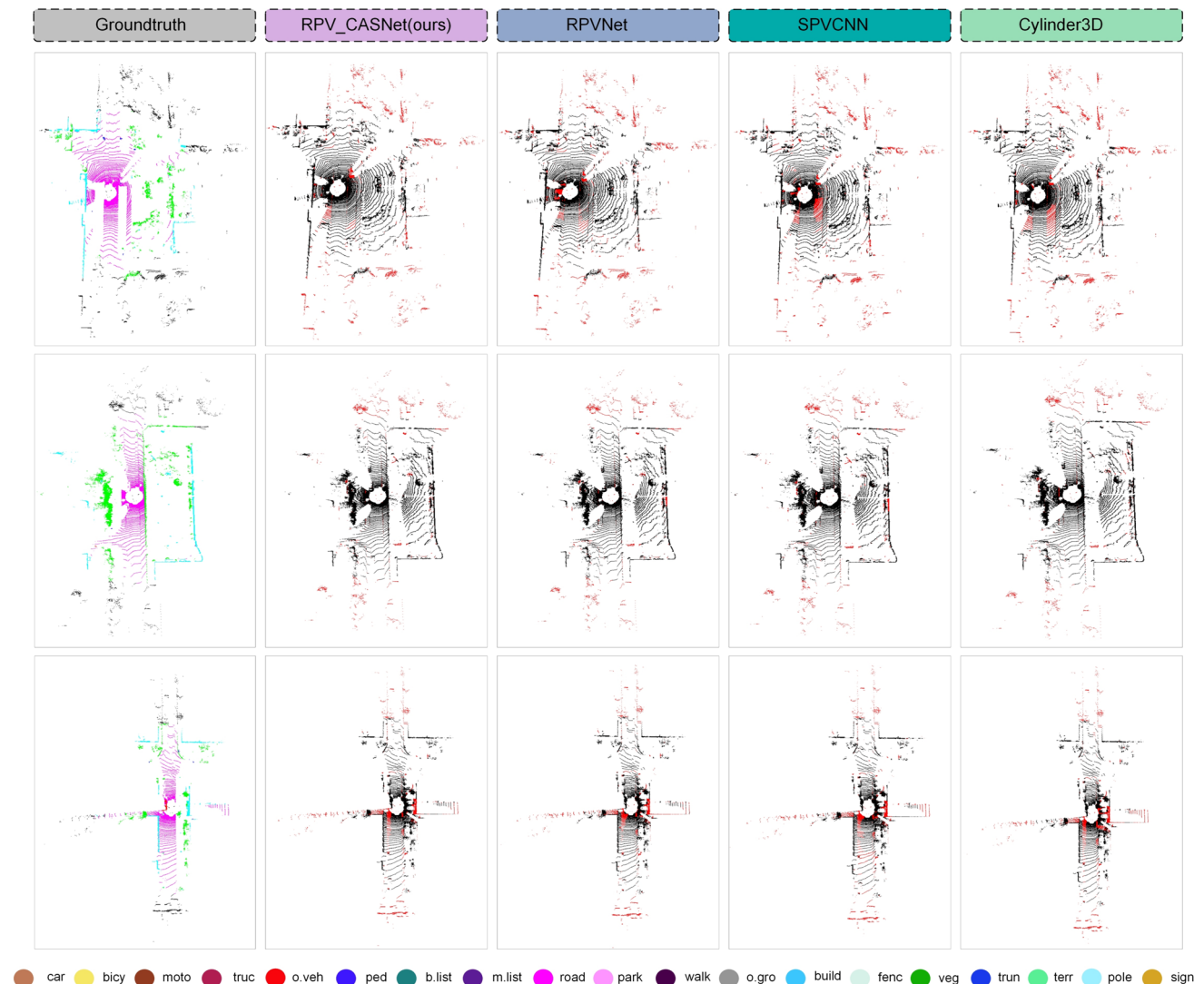


Fig. 8 Qualitative comparison with RPVNet [10], SPVCNN [34], Cylinder3D [37] on SemanticKITTI dataset. Segmentation errors are highlighted in red. Best viewed in colors

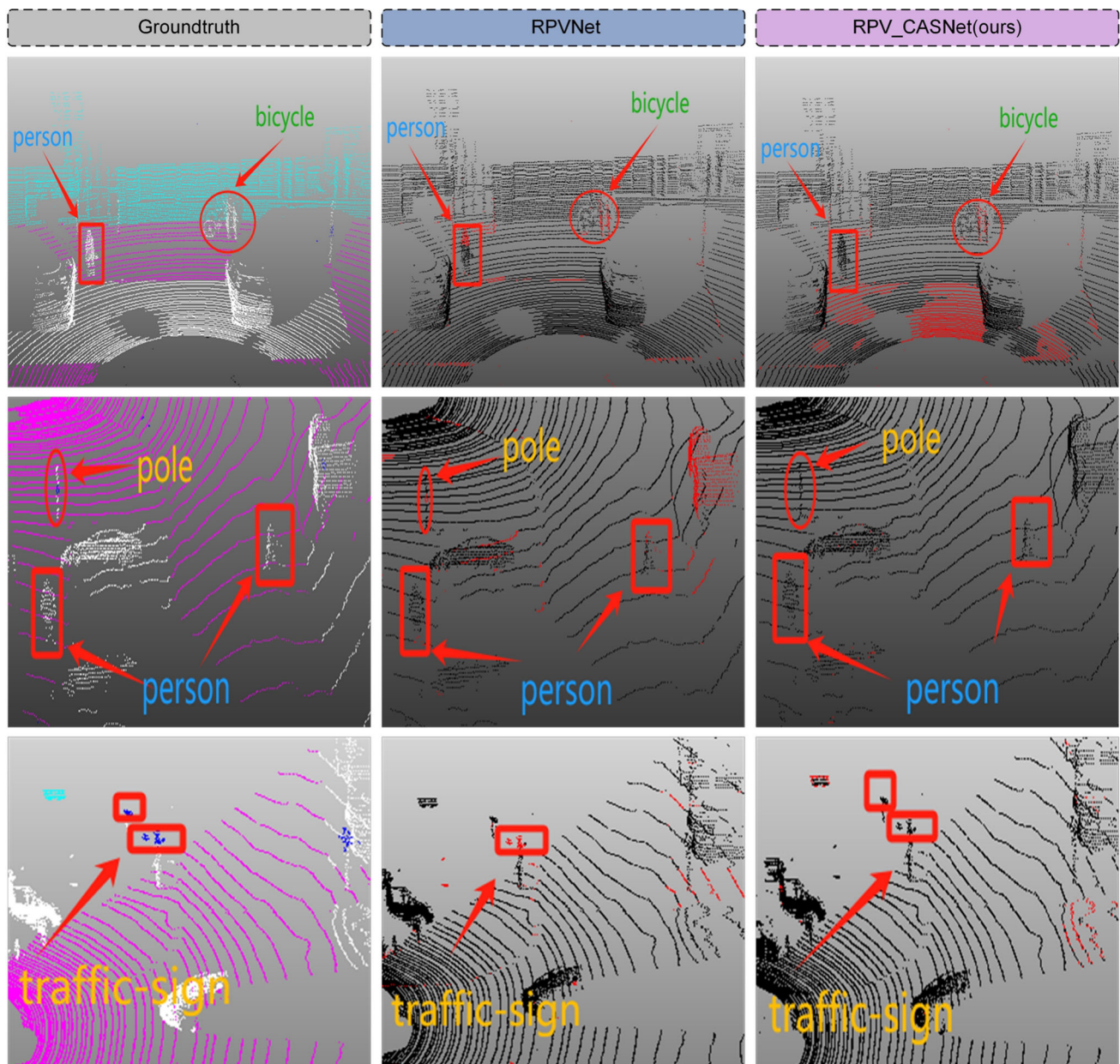


Fig. 9 Quantitative Comparison with RPNNet [10] in Selected Small Target Class Segmentation Accuracy on the SemanticKITTI dataset. Segmentation errors are highlighted in red. Best viewed in colors

and motorcycle categories, as well as an improvement of 1.7 percentage points and 1.5 percentage points in the overall segmentation accuracy. This shows that our designed GAN-Mix has strong horizontal scalability.

4.7 Visualization of results

We show the segmentation results of our proposed algorithm RPN-CASNet and the state-of-the-art algorithms on the SemanticKITTI dataset as shown in Fig. 8 in a visual way. We also provide the segmentation results of and RPNNet[10]

on some of the small target classes as shown Fig. 9, where the network segmentation errors are highlighted in red. Our network performance is even better.

5 Conclusion

This paper presents a new algorithm called RPN-CASNet for the task of fine-grained segmentation in point cloud multi-view fusion. The algorithm utilises a channel self-attention mechanism to fuse three different views, range, point and

voxel, and integrates them more finely through an interactive structure, the RPVLayer, to take full advantage of their differences. The RPVLayer consists of two key designs: the Feature Refinement Module (FRM) and the Multi-Fine-Grained Feature Self-Attention Module (MFGFSAM). The FRM allows for a more fine-grained segmentation of both range view and voxel view to re-correct anomalies introduced due to mapping and voxelisation operations, while MFGFSAM has the ability to efficiently integrate remote area tokens and maintain multi-scale features within a single attention layer. In addition, this paper also introduces a Dynamic Feature Pyramid Extractor (DFPE) for network deployment that extracts rich multi-scale features for range views. By conducting experiments on two publicly available large datasets, we demonstrate the efficiency and comparability of RPV-CASNet.

6 Discussion

From Table 2 to Table 3, we clearly demonstrate the advantages of our algorithm over most fusion-based state-of-the-art methods and achieve satisfactory performance. It is particularly noteworthy that the mIoU performance of our proposed RPV-CASNet on the SemanticKITTI dataset and the nuScenes dataset is very close to that of RPVNet [10], and it even exceeds the accuracy of RPVNet on five small target categories. This indicates that our network performs well in the task of fine-grained segmentation of radar point cloud data. However, we also realise that there are still some challenges that need to be further explored in our next work. Through in-depth reflection, we confirm that the self-attention mechanism has unrivalled advantages in capturing complex dependencies between elements, and is particularly suited to point cloud data. Yet, when performing self-attention computations in MFGFSAM, we found that about one-third of the features in the range branch and voxel branch were under-utilised, which may have limited the performance of our network. Therefore, we will continue to explore new efficient fusion mechanisms to address this issue. It is coming soon.

Acknowledgements Special thanks to Min Zhao and Chenyang Wang for their invaluable help during the experiments. In addition, I would like to thank Zitai Jiang for his insightful discussions. I would also like to thank the Image Engineering and Pattern Recognition Research Laboratory for their generous support in providing the necessary experimental equipment. Finally, I would like to thank all the viewers who contributed to this paper.

Author Contributions Jiajiong Li: Methodology, Software, Writing original draft and editing; Chuanxu Wang: Conceptualization, Funding acquisition; Chenyang Wang: Resources, Writing review; Min Zhao: Writing review; Zitai Jiang: Writing review.

Funding Research on Key Issues of Action Element Modeling and Deep Interaction Relationship Model Inference in Complex Group Behavior 61672305 (Ranked 1/9), National Natural Science Foundation of China.

Availability of data and materials The data generated and analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests The authors declare no conflict of interest.

Ethical and informed consent for data used We comply with ethical and informed consent for the data used.

References

1. Choy CB, Xu D, Gwak J, Chen K, Savarese S (2016) 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII* 14, pages 628–644. Springer
2. Qi CR, Yi L, Su H, Guibas LJ (2017) Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660
3. Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, Solomon JM (2019) Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)* 38(5):1–12
4. Thomas H, Qi CR, Deschaud JE, Marcotegui B, Goulette F, Guibas LJ (2019) Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420
5. Xu C, Wu B, Wang Z, Zhan W, Vajda P, Keutzer K, Tomizuka M (2020) Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII* 16, pages 1–19. Springer
6. Milioto A, Vizzo I, Behley J, Stachniss C (2019) Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE
7. Cortinhal T, Tzelepis G, Erdal Aksoy E (2020) Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II* 15, pages 207–222. Springer
8. Ando A, Gidaris S, Bursuc A, Puy G, Boulch A, Marlet R (2023) Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5240–5250
9. Liu Z, Tang H, Lin Y, Han S (2019) Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems* 32
10. Xu J, Zhang R, Dou J, Zhu Y, Sun J, Pu S (2021) Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16024–16033
11. Zhou Y, Sun P, Zhang Y, Anguelov D, Gao J, Ouyang T, Guo J, Ngiam J, Vasudevan V (2020) End-to-end multi-view fusion for

- 3d object detection in lidar point clouds. In *Conference on Robot Learning*, pages 923–932. PMLR
12. Wang Y, Fathi A, Kundu A, Ross DA, Pantofaru C, Funkhouser T, Solomon J (2020) Pillar-based object detection for autonomous driving. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII* 16, pages 18–34. Springer
13. Liong VE, Nguyen TN, Widjaja S, Sharma D, Chong ZJ (2020) Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. [arXiv:2012.04934](https://arxiv.org/abs/2012.04934)
14. Zhang F, Fang J, Wah B, Torr P (2020) Deep fusionnet for point cloud semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV* 16, pages 644–663. Springer
15. Gerdzhev M, Razani R, Taghavi E, Bingbing L (2021) Tornado-net: multiview total variation semantic segmentation with diamond inception module. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9543–9549. IEEE
16. Behley J, Garbade M, Milioto A, Quenzel J, Behnke S, Stachniss C, Gall J (2019) Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307
17. Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O (2020) nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631
18. Hertz A, Hanocka R, Giryas R, Cohen-Or D (2020) Pointgmm: A neural gmm network for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12054–12063
19. Yang X, Jin M, He W, Chen Q (2023) Pointcat: Cross-attention transformer for point cloud. [arXiv:2304.03012](https://arxiv.org/abs/2304.03012)
20. Zhao H, Jiang L, Jiang J, Torr PH, Koltun V (2021) Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268
21. Guo MH, Cai JX, Liu ZN, Mu TJ, Martin RR, Hu SM (2021) Pct: Point cloud transformer. *Computational Visual Media* 7:187–199
22. Yan X, Zheng C, Li Z, Wang S, Cui S (2020) Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5589–5598
23. Puy G, Boulch A, Marlet R (2023) Using a waffle iron for automotive point cloud semantic segmentation. [arXiv:2301.10100](https://arxiv.org/abs/2301.10100)
24. Lin J, Rickert M, Perzylo A, Knoll A (2021) Pctma-net: Point cloud transformer with morphing atlas-based point generation network for dense point cloud completion. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5657–5663. IEEE
25. Lai X, Liu J, Jiang L, Wang L, Zhao H, Liu S, Qi X, Jia J (2022) Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509
26. He C, Li R, Li S, Zhang L (2022) Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8417–8427
27. Mao J, Xue Y, Niu M, Bai H, Feng J, Liang X, Xu H, Xu C (2021) Voxel transformer for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3164–3173
28. Park C, Jeong Y, Cho M, Park J (2021) Efficient point transformer for large-scale 3d scene understanding
29. Zhang C, Wan H, Shen X, Wu Z (2022) Pvt: Point-voxel transformer for point cloud learning. *Int J Intell Syst* 37(12):11985–12008
30. Zhang C, Wan H, Shen X, Wu Z (2022) Patchformer: An efficient point transformer with patch attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11799–11808
31. Hu Q, Yang B, Xie L, Rosa S, Guo Y, Wang Z, Trigoni N, Markham A (2020) Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11108–11117
32. Zhou Y, Tuzel O (2018) Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499
33. Wang Z, Lu F (2019) Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes. *IEEE Trans Visual Comput Graphics* 26(9):2919–2930
34. Tang H, Liu Z, Zhao S, Lin Y, Lin J, Wang H, Han S (2020) Searching efficient 3d architectures with sparse point-voxel convolution. In *European conference on computer vision*, pages 685–702. Springer
35. Park J, Kim C, Kim S, Jo K (2023) Pscnet: Fast 3d semantic segmentation of lidar point cloud for autonomous car using point convolution and sparse convolution network. *Expert Syst Appl* 212:118815
36. Zhang Y, Zhou Z, David P, Yue X, Xi Z, Gong B, Foroosh H (2020) Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610
37. Zhu X, Zhou H, Wang T, Hong F, Ma Y, Li W, Li H, Lin D (2021) Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9939–9948
38. Cheng R, Razani R, Taghavi E, Li E, Liu B (2021) 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12547–12556
39. Li J, Liu Y, Yuan X, Zhao C, Siegwart R, Reid I, Cadena C (2019) Depth based semantic scene completion with position importance aware loss. *IEEE Robotics and Automation Letters* 5(1):219–226
40. Kochanov D, Nejadasl FK, Booi O (2020) Kprnet: Improving projection-based lidar semantic segmentation. [arXiv:2007.12668](https://arxiv.org/abs/2007.12668)
41. Jhaldiyal A, Chaudhary N (2023) Semantic segmentation of 3d lidar data using deep learning: a review of projection-based methods. *Appl Intell* 53(6):6844–6855
42. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)
43. Aksoy EE, Baci S, Cavdar S (2020) Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving. In *2020 IEEE intelligent vehicles symposium (IV)*, pages 926–932. IEEE
44. Cen J, Zhang S, Pei Y, Li K, Zheng H, Luo M, Zhang Y, Chen Q (2023) Cmdfusion: Bidirectional fusion network with cross-modality knowledge distillation for lidar semantic segmentation. *IEEE Robotics and Automation Letters* 9(1):771–778
45. Cheng HX, Han XF, Xiao GQ (2023) Transrvnet: Lidar semantic segmentation with transformer. *IEEE Transactions on Intelligent Transportation Systems*
46. Zhao L, Zhou H, Zhu X, Song X, Li H, Tao W (2023) Lif-seg: Lidar and camera image fusion for 3d lidar semantic segmentation. *IEEE Transactions on Multimedia*
47. Du J, Huang X, Xing M, Zhang T (2023) Improved 3d semantic segmentation model based on rgb image and lidar point cloud fusion for automatic driving. *Int J Automot Technol* 24(3):787–797
48. Chen D, Zhuang M, Zhong X, Wu W, Liu Q (2023) Rspmp: Real-time semantic perception and motion planning for autonomous navigation of unmanned ground vehicle in off-road environments. *Appl Intell* 53(5):4979–4995

49. Li M, Wang G, Zhu M, Li C, Liu H, Pan X, Long Q (2024) Dfamnet: dual fusion attention multi-modal network for semantic segmentation on lidar point clouds. *Applied Intelligence*, pages 1–12
50. Chen R, Liu Y, Kong L, Zhu X, Ma Y, Li Y, Hou Y, Qiao Y, Wang W (2023) Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030
51. Kong L, Liu Y, Chen R, Ma Y, Zhu X, Li Y, Hou Y, Qiao Y, Liu Z (2023) Rethinking range view representation for lidar segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 228–240
52. Kong L, Liu Y, Li X, Chen R, Zhang W, Ren J, Pan L, Chen K, Liu Z (2023) Robo3d: Towards robust and reliable 3d perception against corruptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19994–20006
53. Ren S, Zhou D, He S, Feng J, Wang X (2022) Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10853–10862
54. Li R, Li X, Heng PA, Fu CW (2020) Pointaugment: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6378–6387
55. Chen Y, Hu VT, Gavves E, Mensink T, Mettes P, Yang P, Snoek CGM (2020) Pointmixup: Augmentation for point clouds. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 330–345. Springer
56. Xiao A, Huang J, Guan D, Cui K, Lu S, Shao L (2022) Polarmix: A general data augmentation technique for lidar point clouds. *Adv Neural Inf Process Syst* 35:11035–11048
57. Kong L, Ren J, Pan L, Liu Z (2023) Lasermix for semi-supervised lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21705–21715
58. Li X, Zhang G, Pan H, Wang Z (2022) Cpgnet: Cascade point-grid fusion network for real-time lidar semantic segmentation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11117–11123. IEEE



Jiajiong Li graduated from Shangqiu Institute of Technology with a bachelor's degree in Computer Science and Technology in 2021. He is currently pursuing his master's degree in the School of Information Science and Technology, Qingdao University of Science and Technology. His research interest focuses on segmentation of outdoor LiDAR data.



Chuanxu Wang received a Ph.D. degree from the Ocean University of China and is currently a professor and master's tutor at the School of Qingdao University of Science and Technology, mainly engaged in research on computer vision and pattern recognition.



Chenyang Wang is currently a master student at Qingdao University of Science and Technology. His research interests include image data processing and analysis and computer vision.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Min Zhao graduated from Jinan University with a Bachelor's degree in Communication Engineering in 2021. She currently pursuing a master's degree at the School of Information Science and Technology, Qingdao University of Science and Technology. Her current research interests include computer vision and 3D point clouds.



Zitai Jiang obtained a bachelor's degree from the School of Information Science and Technology at Qingdao University of Science and Technology in 2022, and pursuing a master's degree at the same university, mainly engaged in computer vision research and 3D point cloud segmentation.