electronics

MDPI

Article

LiDAR Point Clouds Semantic Segmentation in Autonomous Driving Based on Asymmetrical Convolution

Xiang Sun [1], Shaojing Song [2,*], Zhiqing Miao [3], Pan Tang [4] and Luxia Ai [5]

[1] School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai 201209, China; 20211510167@stu.sspu.edu.cn
[2] School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China
[3] School of Communication and Electronic Engineering, East China Normal University, Shanghai 200062, China; 52275904009@stu.ecnu.edu.cn
[4] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; 20191510017@stu.sspu.edu.cn
[5] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China; luxiaai@hust.edu.cn
* Correspondence: sjsong@sspu.edu.cn

Abstract: LiDAR has become a vital sensor for autonomous driving scene understanding. To meet the accuracy and speed of LiDAR point clouds semantic segmentation, an efficient model ACPNet is proposed in this paper. In the feature extraction stage, the backbone is constructed with asymmetric convolutions, so the skeleton of the square convolution kernel is enhanced, which leads to greater robustness to target rotation. Moreover, a contextual feature enhancement module is designed to extract richer contextual features. During training, global scaling and global translation are performed to enrich the diversity of datasets. Compared with the baseline network PolarNet, the mIoU of ACPNet on the SemanticKITTI, SemanticPOSS and nuScenes datasets are improved by 5.1%, 1.6% and 2.9%, respectively. Meanwhile, the speed of ACPNet is 14 FPS, which basically meets the real-time requirements in autonomous driving scenarios. The experimental results show that ACPNet significantly improves the performance of LiDAR point cloud semantic segmentation.

Keywords: LiDAR point clouds; semantic segmentation; deep learning; asymmetric convolution; contextual feature enhancement

Citation: Sun, X.; Song, S.; Miao, Z.; Tang, P.; Ai, L. LiDAR Point Clouds Semantic Segmentation in Autonomous Driving Based on Asymmetrical Convolution. *Electronics* **2023**, *12*, 4926. https://doi.org/10.3390/electronics12244926

Academic Editor: Jose Eugenio Naranjo

Received: 18 October 2023
Revised: 1 December 2023
Accepted: 4 December 2023
Published: 7 December 2023

1. Introduction

Scene understanding is one of the most critical tasks in autonomous driving. With the challenges introduced by recent technologies such as autonomous driving, a detailed and accurate understanding of the road scene has become a main part of any outdoor autonomous robotic system in recent years. Although semantic segmentation of 2D images is crucial to attaining scene understanding, there are still some limitations to visual sensors, such as the inefficiency of acquiring information under insufficient light, lack of depth information and limited field of view. In contrast, LiDAR can obtain accurate depth information with higher density and wider viewing field regardless of lighting conditions, which makes it a more reliable source of information for environmental perception. Therefore, the scene understanding of LiDAR point clouds with semantic segmentation has become a focal point in autonomous driving.

According to point clouds' encoding methods, the current LiDAR point clouds semantic segmentation methods can be divided into three categories: point-based methods, voxel-based methods, and projection-based methods. In terms of speed, there is a lot of computation and memory consumption in point-based and voxel-based methods, which makes it difficult to achieve real-time effects with the on-board computing platform. A higher priority should be placed on real-time performance when it comes to autonomous driving than segmentation accuracy. In contrast, projection-based methods are lightweight

and fast, so real-time effects can be achieved during deployment. In terms of segmentation accuracy, the projection-based method has shown some success. However, since the point cloud information is not fully utilized during feature extraction, there is still room for improving segmentation accuracy.

When achieving real-time effects, it is of great relevance to improve the segmentation accuracy in autonomous driving scenarios. To meet segmentation accuracy and speed, an efficient real-time network ACPNet (Asymmetric Convolution based on PolarNet) is proposed in this paper. PolarNet [1] is the baseline network of ACPNet, which encodes point clouds through polar bird's-eye-view (BEV) representation. BEV is the abbreviation for Bird's Eye View, which is a perspective that views an object or scene from above, just like a bird looking down at the ground in the air. Also known as God's perspective, which is a perspective or coordinate system used to describe the perception of the world. The using of polar BEV has some advantages: First, in terms of point allocation within grid cells, the polar BEV method will assign point clouds to their respective grid cells more evenly. Second, since the partitioning method brings about a more balanced distribution of points, the theoretical upper limit of prediction accuracy for the semantic classification of point clouds will be increased, thereby improving the performance of downstream semantic segmentation models [1]. In ACPNet, the encoded point cloud features are fed into an Asymmetric Convolution Backbone Network (ACBN) for feature extraction. Then, the features extracted by the backbone are input to the Contextual Feature Enhancement Module (CFEM) for further mining of contextual features. Moreover, global scaling and global translation are used as Enhanced Data Augmentation (EDA) while ACPNet is being trained. Experiments are conducted on the SemanticKITTI [2], SemanticPOSS [3] and nuScenes [4] dataset to verify the validity and generalization of our method. The main contributions of this paper can be summarized as follows:

- An Asymmetric Convolution Backbone Network is proposed. Asymmetric convolutions are used in the backbone to enhance the skeleton of square convolution kernels and reduce interference caused by target rotations.
- A Contextual Feature Enhancement Module is proposed, which can fully extract the contextual feature by decomposing and aggregating the features.
- Enhanced Data Augmentation methods of global scaling and global translation are used to enrich the diversity of the dataset samples. Thus, the generalization capability of the model is further improved without increasing the computational cost.

2. Related Works

Due to the sparsity and disorderliness of point clouds, encoding the input point cloud is a crucial issue when using convolutional neural networks for semantic segmentation of 3D point clouds. According to the encoding methods for point clouds, existing point cloud encoding methods can be divided into three categories: Point-based Methods, Voxel-based Methods, and Projection-based Methods.

2.1. Point-Based Methods

PointNet [5] is a point-wise learning method for point cloud features, and max pooling is used to integrate global features. PointNet++ [6] is an extension to PointNet, and the ability to extract local information of different scales is strengthened. A spatially continuous convolution is proposed in PointConv [7], which reduces the memory consumption of the algorithm effectively. For semantic segmentation in large-scale point clouds scenarios, the point clouds are represented as interconnected superpoint graphs in SPG [8], and then PointNet was used to learn the features of the superpoint graph. An attention-based module was designed in RandLA-Net [9] to integrate local features, achieving efficient segmentation in large-scale point clouds. Segmentation performance was further improved in KPConv [10] with a novel spatial kernel-based point convolution. Lu et al. [11] suggested the use of distinct aggregation strategies for both within-category and between-category data. Employing aggregation or enhancement techniques on local features [12]

can effectively enhance the perception of intricate details. Furthermore, to effectively learn features from extensive point clouds encompassing diverse target types, Fan et al. [13] introduced the SCF-Net. This network incorporates a dual-distance attention mechanism and global contextual features to enhance semantic segmentation performance.

Point-based methods directly work on the raw point clouds without excessive initialization transformation steps. However, when handling expansive point cloud scenes, the local nearest neighbor search is inevitably involved, which is computationally inefficient. Thus, there is still clearly room for improvement in point-based methods.

2.2. Voxel-Based Methods

Point clouds are regularly divided into 3D cubic voxels, and Voxel-based methods employ 3D convolution for the extraction of features. SEGCloud [14] is one of the earlier methods for semantic segmentation based on voxel representation. In order to utilize 3D convolution efficiently and expand the receptive field, 3D sparse convolution [15] is used in Minkowski CNN [16], which reduces the computational complexity of convolution. In pursuit of higher segmentation accuracy, a neural architecture search (NAS) based model SPVNAS [17] is proposed, which trades high computational cost for accuracy. In order to fit the spatial distribution of the LiDAR point clouds, a cylinder voxel division method is proposed in Cylinder3D [18], which makes it obtain high accuracy. In order to streamline computations and enhance the intricacies of smaller instances, an attention-focused feature fusion module and an adaptive feature selection module are proposed by Cheng et al. [19]. To improve the speed of voxel-based networks, a method of knowledge distillation from point to voxel is proposed in PVKD [20] to achieve model compression.

High segmentation accuracy is typically achieved in voxel-based methods. However, 3D convolution is inevitably used, resulting in significant memory occupation and high computational consumption.

2.3. Projection-Based Methods

The basic concept behind projection-based methods is to transform point clouds into images that can undergo 2D convolution operations. The SqueezeSeg [21–23] series of algorithms based on SqueezeNet [24] perform semantic segmentation after projecting point clouds. RangeNet++ [25] implements semantic segmentation based on the backbone network of DarkNet53 [26], and a K-Nearest Neighbor (KNN) algorithm is proposed to improve segmentation accuracy. 3D-MiniNet [27] is based on a lightweight backbone to build the network, achieving a faster speed. A polar BEV representation method is proposed in PolarNet [1], which uses a simplified version of PointNet to encode the point clouds of each polar coordinate grid to obtain a pseudo image, and KNN post-processing operation is no longer needed. Peng et al. [28] introduced a multi-attention mechanism to enhance the understanding of driving scenes, specifically focused on dense top-view semantic segmentation using sparse LiDAR data. SalsaNext [29] introduced a new context module, which replaces the ResNet encoder blocks with a residual convolution stack that has increasing receptive fields. Additionally, it incorporated a pixel-shuffle layer into the decoder. MINet [30] employed multiple paths with varying scales to effectively distribute computational resources across different scales. FIDNet-Point [31] designed a fully interpolation decoding module that directly upsamples the multi-resolution feature maps using bilinear interpolation. CENet+KNN [32] incorporated convolutional layers with larger kernel sizes, replacing MLP, and integrated multiple auxiliary segmentation heads into its architecture.

There are obvious advantages in computational complexity and speed in projection-based methods. Therefore, it is significant to improve the segmentation accuracy of projection-based methods for practical application in autonomous driving.

3. Methodology

The overall framework of ACPNet is shown in Figure 1. First, the raw point clouds are encoded using a polar BEV encoder. Then, the encoded point cloud features are input into the ACBN constructed with asymmetric convolutions for feature extraction. Next, the features extracted by the backbone are inputted into the CFEM for further mining contextual features. Finally, the output features are processed by the semantic head to acquire the semantic segmentation results.

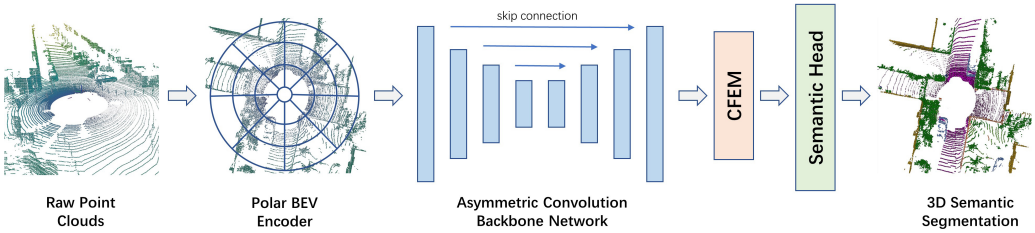


Figure 1. The overall framework of ACPNet.

3.1. Asymmetric Convolution Backbone Network (ACBN)

Objects such as vehicles, riders, and pedestrians are the main detection targets in autonomous driving scenarios. These objects will be presented in a small rectangular area after the BEV projection. Furthermore, it is common for objects to rotate in the horizontal direction. Horizontal rotation refers to the rotation angle of objects on the road compared to the front of the LiDAR sensor. When an object is not directly in front of the LiDAR, horizontal rotation occurs. Recent studies [33] also indicate that the central crisscross weights play a more significant role in the square convolution kernel.

Asymmetric convolution is a type of convolutional operation used in convolutional neural networks. Unlike square convolutions that use local convolution blocks with equal length and width, asymmetric convolutions use rectangular blocks with unequal length and width. These convolutions are characterized by their ability to extract different global features depending on the orientation of the rectangular block. When the length is greater than the width, the convolution can extract more global features in the vertical direction, resulting in a larger receptive field or attention range. Conversely, when the length is less than the width, the convolution can extract more global features in the horizontal direction. By combining asymmetric convolutions with different orientations, the weights in the horizontal and vertical directions can be overlaid to enhance the weights at the center cross position of the square convolutional kernel.

As shown in Figure 2, when the feature map is flipped left-right or up-down, the information extracted by the original square kernel will change. But at the same time, if there are horizontal kernels or vertical kernels in the convolution combination, some of the kernels will still get the same output as the original feature map in the axially symmetric locations. From this, it can be seen that asymmetric convolution can still extract correct features when dealing with rotational distortions, thus it will enable the model to generalize better on the unseen rotated samples and show robustness.

To enhance the horizontal and vertical responses, we introduce the Asymmetric Convolutions Block as a means to achieve this objective and it can improve the robustness of the model for certain transformations, such as target rollover and rotation in BEV. Inspired by the observation and subsequent conclusion in [33], asymmetric convolutions of 1×3 and 3×1 are used to build the asymmetric convolutions, which strengthen the skeleton of the square convolution kernel while weakening the corner. Moreover, the receptive field of the combination composed of 1×3 and 3×1 asymmetric convolutions are the same as 3×3 square convolutions.

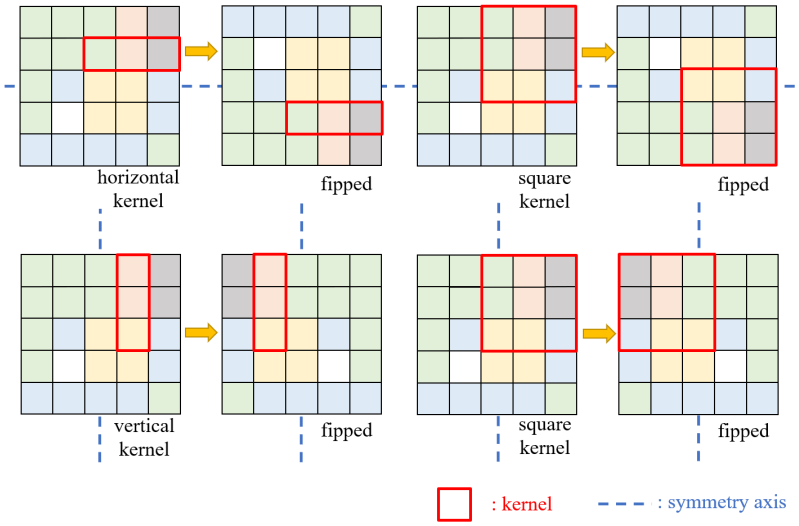


Figure 2. In contrast to square kernels, horizontal and vertical kernels demonstrate greater resilience against flipping.

As shown in Figure 3, the ACBN consists of four downsampling asymmetric convolution blocks and four upsampling asymmetric convolution blocks. In addition, three skip connections (the dotted line in Figure 3) are also employed to merge the low-level and high-level features within the network, thereby enhancing the capability of network for detailed learning.

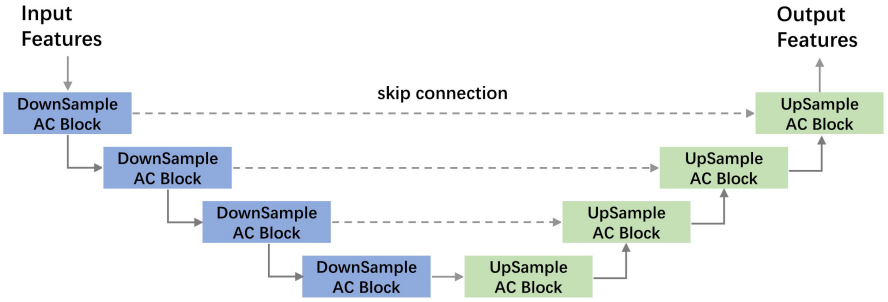


Figure 3. Asymmetric Convolution Backbone Network.

The Downsample Asymmetric Convolution Block is shown in Figure 4, in which a square convolution with the stride of 2 is operated on the features. After that, two asymmetric convolution combinations are operated separately, and the summed results are output. In these asymmetric convolution combinations, the kernels are 3×1, 1×3, and 1×3, 3×1, respectively.

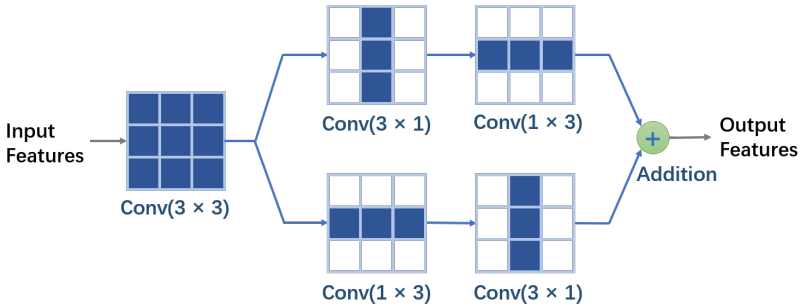


Figure 4. Downsample Asymmetric Convolution Block.

The calculation of the Downsample Asymmetric Convolution Block, as demonstrated in Equation (1):

$$F_{out} = C_{3\times1}(C_{1\times3}(C_{3\times3}(F_{in}))) + C_{1\times3}(C_{3\times1}(C_{3\times3}(F_{in}))) \quad (1)$$

where F_{in} and F_{out} represent input features and output features, respectively, $C_{3\times3}$, $C_{1\times3}$ and $C_{3\times1}$ represent 3 × 3, 1 × 3 and 3 × 1 convolution, respectively.

Figure 5 illustrates the Upsample Asymmetric Convolution Block which makes use of bilinear interpolation, and then low-level features are concatenated from skip connections. Lastly, an asymmetric convolution combination consisting of 1 × 3 and 3 × 1 kernels is performed.

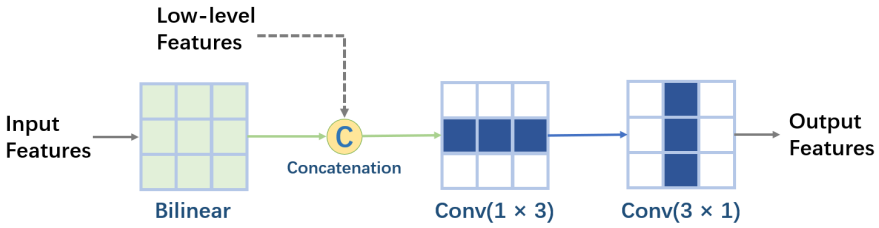


Figure 5. Upsample Asymmetric Convolution Block.

The calculation of the upsample asymmetric convolution block as shown in Equation (2):

$$F_{out} = C_{3\times1}(C_{1\times3}(\Delta(B(F_{in}), F_{low}))) \quad (2)$$

where F_{low} represents low-level features, Δ represents feature concatenation, and B represents the bilinear interpolation operation.

3.2. Contextual Feature Enhancement Module (CFEM)

One of the primary challenges of semantic segmentation is the lack of contextual features in the whole network, so exploring the global contextual features of different scales is crucial in learning the complex correlations among classes. Recently, Studies regarding the semantic segmentation of 3D point clouds also pay attention to the extraction of global contextual features [12,34] and achieved good results. Constructing high-rank global context features directly is challenging due to the need for sufficient capacity to capture extensive contextual variations [35]. To simplify high-rank feature extraction, the Contextual Feature Enhancement Module is proposed. We utilize the tensor decomposition theory [36] to construct the high-rank contextual feature by combining low-rank tensors. This involves using two rank-1 kernels to generate the low-rank features, which are then aggregated to produce the ultimate global context.

As shown in Figure 6, rank-1 kernels are first used to decompose high-rank contextual features based on dimension, which generate low-rank encodings. Next, the active values of the Sigmoid function are added as output. Finally, the current features are multiplied with the input features to acquire the enhanced high-rank contextual features. The decomposition and aggregation strategy is used here to avoid the difficulties of direct high-rank feature extraction.

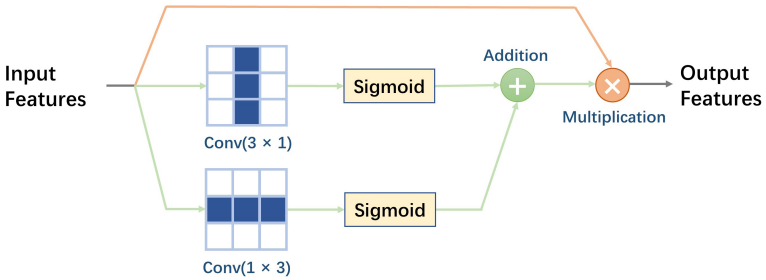


Figure 6. Contextual Feature Enhancement Module.

The calculation of the CFEM as shown in Equation (3):

$$F_{out} = F_{in} \cdot (Sig(C_{3\times1}(F_{in})) + Sig(C_{1\times3}(F_{in}))) \tag{3}$$

where F_{in} and F_{out} represent the input feature and output feature, respectively. Sig represents the logistic Sigmoid function, while $C_{3\times1}$ and $C_{1\times3}$ represent the 3 × 1 and 1 × 3 convolution, respectively.

3.3. Enhanced Data Augmentation (EDA)

Inspired by [37], the global scaling and global translation are employed in training to provide more sample information and improve the model's generalization ability. For global scaling, this method increases the diversity of sample scales in the training data by randomly magnifying and shrinking the global original point cloud information and label information, thereby adding different scale information to the dataset. Moreover, global translation enriches the dataset samples by randomly translating all points in each frame of the point cloud, from the perspective of transforming the distance between the targets and the sensor. Implementation details are shown in Figure 7.

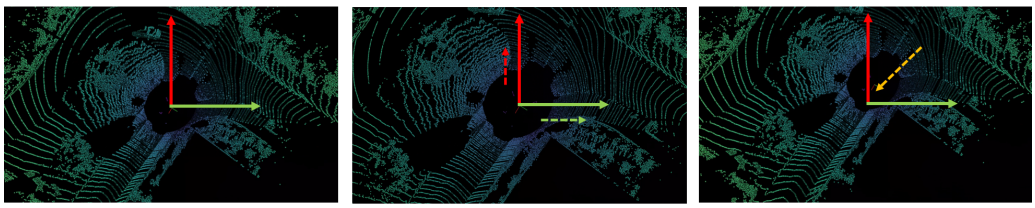


Figure 7. Visualization of the original scene and enhanced data augmentation methods (shown in BEV). (**a**) Original scene, (**b**) Scene augmented by global scaling, (**c**) Scene augmented by global translation.

As shown in Figure 7b, the global scaling is implemented by extracting the scalar s to scale the point $p(x, y, z) \in P$ in each direction from a uniform distribution $U(1-t, 1+t)$ with $t \in \{0.05, 0.1, 0.25\}$, so the randomly scaled point p^* can be represented as $p^*(s \cdot x, s \cdot y, s \cdot z)$. Also, each label a is scaled so that $a(x_c, y_c, z_c, w, l, h, \theta) \in A$ can be represented as $a^*(s \cdot x_c, s \cdot y_c, s \cdot z_c, s \cdot w, s \cdot l, s \cdot h, \theta) \in A^*$.

As shown in Figure 7c, the global translation is implemented by translating each point $p(x, y, z) \in P$, so each translated point p^* can be represented as $p^*(x + \Delta x, y + \Delta y, z + \Delta z)$. Also, each label $a(x_c, y_c, z_c, w, l, h, \theta) \in A$ is converted to the form $a^*(x_c + \Delta x, y_c + \Delta y, z_c + \Delta z, w, l, h, \theta) \in A^*$, where Δx, Δy and Δz are sampled independently from the normal distribution $N(0, \sigma^2)$ and σ takes values in the range $\sigma^2 \in \{0.1, 0.2, 0.4\}$.

Apart from the methods discussed above, Random Flip and Random Rotation in the baseline model are still preserved in the training of ACPNet.

3.4. Loss Function

The loss in ACPNet follows the existing models [19,29], the weighted cross-entropy loss and the Lovász-Softmax loss [38] are used to improve the accuracy of segmentation and the value of Intersection-over-Union (IoU), i.e., the Jaccard index.

The formula of weighted cross-entropy loss is shown in Equation (4):

$$L_{wce}(y, \hat{y}) = -\sum_i a_i P(y_i) \log P(\hat{y}_i), a_i = \frac{1}{\sqrt{v_i}} \tag{4}$$

where v_i is the frequency of each class, $P(y_i)$ and $P(\hat{y}_i)$ correspond to the ground truth probability and prediction probability of the model, respectively.

The formula of Lovász-Softmax loss as shown in Equation (5):

$$L_{ls} = \frac{1}{|C|} \sum_{c \in C} J(e(c)) \tag{5}$$

where J is the Lovász extension of the Jaccard index, C is the class number, $e(c)$ is the vector of errors for class c, $e(c) \in [0,1]^p$, and p is the number of pixels considered.

Therefore, the total loss of ACPNet is given by Equation (6):

$$L = L_{wce} + L_{ls} \tag{6}$$

4. Experiments

In order to evaluate the performance of ACPNet, experiments are conducted in this part. During training, the Adam optimizer is used to fit the parameters with a learning rate of 0.001 and a batch size of 2, and the maximum number of training epochs is 30. Moreover, the training process is conducted on a server with Intel Xeon Gold 5118 @ 2.30 GHz CPU and NVIDIA RTX 3090 GPU.

4.1. Dataset and Metric

SemanticKITTI [2] is a LiDAR point clouds segmentation dataset for large-scale outdoor scenes, which is made based on the KITTI Vision Odometry Benchmark [39]. SemanticKITTI provides 22 sequences of dense point-level annotations, and 19 main classes are used for evaluation. Among all 22 sequences, sequences 00 to 10 are used as the training set (of which sequence 08 is the validation set), and sequences 11 to 21 are used as the test set.

SemanticPOSS [3] is a challenging benchmark created by Peking University, comprising 2988 intricate LiDAR scenes with a large number of sparse dynamic instances, such as people and riders. It is smaller and sparser compared to other benchmarks, making it more challenging. The dataset is divided into six sequences, with sequence 2 designated as the test set and the remaining sequences used for training.

nuScenes [4] is a large-scale autonomous driving dataset created by Motional. It consists of 1000 scenes, each 20 s in duration and captured using a 32-beam LiDAR sensor. In total, the dataset comprises 40,000 frames. They also formally divided the data into training and validation sets. Following the consolidation of similar classes and removal of infrequent ones, a total of 16 classes remain for the purpose of LiDAR semantic segmentation.

The mean Intersection-over-Union (mIoU) [40] over all classes is used as the primary evaluation metric. The formula of mIoU is as shown in Equation (7):

$$mIoU = \frac{1}{n} \sum_{c=1}^{n} \frac{TP_c}{TP_c + FP_c + FN_c} \tag{7}$$

where TP_c, FP_c and FN_c represent the predictions of True Positive, False Positive and False Negative of each class c, respectively, and n is the number of classes.

FPS is also used as an evaluation metric, and the FPS is measured on a single NVIDIA RTX 2080Ti GPU. Note that the speed of the LiDAR semantic segmentation model is considered real-time when it reaches 10 FPS on NVIDIA RTX 2080Ti. That is because the computing power of this GPU is comparable to that of current mainstream on-board computing platforms, and the acquisition frequency of the Velodyne-HDLE64 LiDAR used in the SemanticKITTI dataset is 10 Hz.

4.2. Results on SemanticKITTI, SemanticPOSS and nuScenes

In order to verify the effectiveness of ACPNet, the performance is evaluated on SemanticKITTI, SemanticPOSS and nuScenes datasets. In this section, ACPNet is compared with several other current mainstream methods.

As shown in Table 1, these are the results of the SemanticKITTI test set, compared to the baseline model PolarNet, there is a significant performance improvement in ACPNet. In particular, the IoU is improved in 17 out of 19 classes, with improving by more than 5% in traffic participants such as trucks, buses, motorcycles, motorcyclists and bicyclists. Furthermore, the mIoU over all classes is increased by 5.1%, reaching 59.4%. Besides, the comparison of ACPNet and other methods is presented in Table 1, there are advantages in seven classes, and the IoU of the car is outstanding. Regarding speed, the running speed of ACPNet exceeds 14 FPS, meeting the demand for real-time autonomous driving.

Table 1. Evaluation Results of ACPNet and existing methods on the SemanticKITTI Test Set.

		Per Class IoU (%)																		
Methods	**mIoU (%)**	**Car**	**Bicycle**	**Motorcycle**	**Truck**	**Bus**	**Person**	**Bicyclist**	**Motorcyclist**	**Road**	**Parking**	**Sidewalk**	**Other-Ground**	**Building**	**Fence**	**Vegetation**	**Trunk**	**Terrain**	**Pole**	**Traffic-Sign**	**FPS (Hz)**
TangentConv [16]	35.9	86.8	1.3	12.7	11.6	10.2	17.1	20.2	0.5	82.9	15.2	61.7	9.0	82.8	44.2	75.5	42.5	55.5	30.2	22.2	0.3
RangeNet53++ [25]	52.2	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	91.8	65.0	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9	12
LatticeNet [41]	52.9	92.9	16.6	22.2	26.6	21.4	35.6	43.0	46.0	90.0	59.4	74.1	22.0	88.2	58.8	81.7	63.6	63.1	51.9	48.4	7
RandLA-Net [9]	53.9	94.2	26.0	25.8	40.1	38.9	49.2	48.2	7.2	90.7	60.3	73.7	20.4	86.9	56.3	81.4	61.3	66.8	49.2	47.7	22
PolarNet [1]	54.3	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5	16
MINet [30]	55.2	90.1	41.8	34.0	29.9	23.6	51.4	52.4	25.0	90.5	59.0	72.6	25.8	85.6	52.3	81.1	58.1	66.1	49.0	59.9	24
3D-MiniNet [27]	55.8	90.5	42.3	42.1	28.5	29.4	47.8	44.1	14.5	91.6	64.2	74.5	25.4	89.4	60.8	82.8	60.8	66.7	48.0	56.6	28
SqueezeSegV3 [23]	55.9	92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9	6
CNN-LSTM [42]	56.9	92.6	45.7	49.6	48.6	30.2	53.8	74.6	9.2	90.7	23.3	75.7	17.6	90.0	51.3	87.1	60.8	75.4	63.9	41.5	11
ACPNet (ours)	59.4	95.2	39.3	41.7	41.8	37.7	55.2	48.1	33.7	91.3	66.0	74.9	14.2	90.5	61.5	84.4	67.6	68.2	57.5	59.9	14

As shown in Table 2, these are the results of the SemanticPOSS test set, ACPNet outperforms the compared methods significantly in terms of mIoU. Additionally, ACPNet has achieved the highest results in seven classes, including rider, plants, traffic sign, etc.

As shown in Table 3, these are the results on the NuScenes validation set, ACPNet has achieved a mIoU metric of 72.8%, which is 2.9% higher than the baseline model PolarNet. Besides, the IoU of ACPNet is improved in 15 out of 16 classes, and improvements are obtained in traffic participants classes such as car, bus, bicycle and motorcycle.

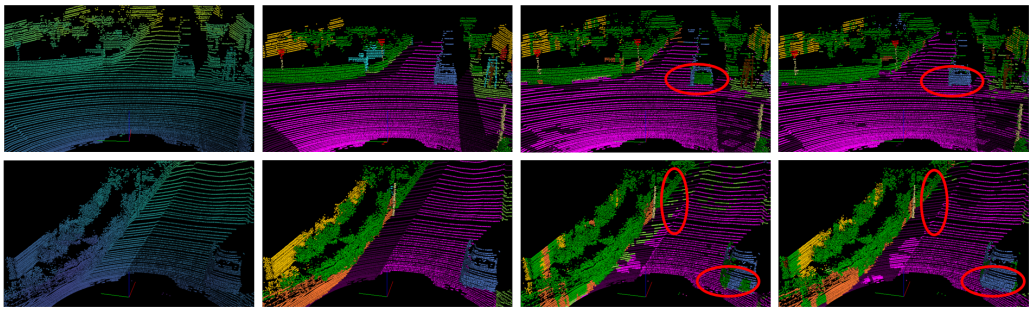
Table 2. Evaluation Results of ACPNet and existing methods on the SemanticPOSS test set.

		Per Class IoU (%)												
Methods	**mIoU (%)**	**People**	**Rider**	**Car**	**Truck**	**Plants**	**Traffic Sign**	**Pole**	**Trashcan**	**Building**	**Cone/Stone**	**Fence**	**Bike**	**Ground**
SqueezeSeg [21]	18.9	14.2	1.0	13.2	10.4	28.0	5.1	5.7	2.3	43.6	0.2	15.6	31.0	75.0
SqueezeSegV2 [22]	30.0	48.0	9.4	48.5	11.3	50.1	6.7	6.2	14.8	60.4	5.2	22.1	36.1	71.3
RangeNet53++ [25]	30.3	55.7	4.5	34.4	13.7	57.5	3.7	6.6	23.3	64.9	6.1	22.2	28.3	72.9
MINet [30]	42.7	61.8	12.0	63.3	22.2	68.1	16.3	29.3	28.5	74.6	25.9	31.7	44.5	76.4
FIDNet-Point [31]	45.8	71.6	22.7	71.7	22.9	67.7	21.8	27.5	15.8	72.7	31.3	40.4	50.3	79.5
CENet + KNN [32]	50.3	75.5	22.0	77.6	25.3	72.2	18.2	31.5	48.1	76.3	27.7	47.7	51.4	80.3
PolarNet [1]	52.4	72.3	31.1	72.7	29.8	74.0	27.0	33.4	48.8	79.1	41.4	39.6	53.7	78.0
ACPNet (ours)	54.0	72.8	31.2	75.0	29.1	74.4	28.9	35.3	45.6	80.0	51.5	46.0	55.6	76.9

Table 3. Evaluation Results of ACPNet and existing methods on the nuScenes validation set.

		Per Class IoU (%)															
Methods	**mIoU (%)**	**Barrier**	**Bicycle**	**Bus**	**Car**	**Construction**	**Motorcycle**	**Pedestrian**	**Traffic_Cone**	**Trailer**	**Truck**	**Driveable**	**Other_Flat**	**Sidewalk**	**Terrain**	**Manmade**	**Vegetation**
RangeNet53++ [25]	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
PolarNet [1]	69.9	69.7	20.2	87.2	84.7	34.6	76.5	71.1	54.3	58.5	78.7	95.5	69.2	73.1	73.3	86.2	85.2
Salsanext [29]	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
ACPNet (ours)	72.8	73.2	22.5	87.7	90.9	47.0	77.1	70.5	58.1	64.2	82.3	96.2	73.4	74.5	73.7	87.9	85.8

The experimental results show that our method effectively performs semantic segmentation in LiDAR point clouds and outperforms other methods. Part of the visualizations of prediction results on the SemanticKITTI dataset are shown in Figure 8.



(a) raw data (b) ground truth (c) PolarNet (d) ACPNet(ours)

Figure 8. Visualization on SemanticKITTI validation set. Where (**a**,**b**) are LiDAR raw data and ground truth of semantic segmentation, (**c**,**d**) are predictions of this frame for PolarNet and our method. The areas circled by the red circles represent the different properties of the segmentation results.

4.3. Ablation Studies

To investigate the individual contribution of each module over the baseline model PolarNet [1], ablation studies are conducted on the validation set within the SemanticKITTI dataset (seq 08). The studied modules include the Contextual Feature Enhancement Module (CFEM), the Asymmetric Convolution Backbone Network (ACBN), and the Enhanced Data Augmentation (EDA). GS and GT, respectively, stand for global scaling and global translation. The results of the ablation experiments are presented in Table 4.

Table 4. Ablation studies for network components on SemanticKITTI Validation Set (seq 08).

Baseline	**CFAM**	**ACBN**	**EDA_GS**	**EDA_GT**	**mIoU (%)**
✓					56.4
✓	✓				58.0
✓	✓	✓			59.1
✓	✓	✓	✓		59.4
✓	✓	✓		✓	59.3
✓	✓	✓	✓	✓	59.6

By adding the CFEM, the mIoU of the model is improved by 1.6%. This result points out that the module is able to mine and extract contextual features, avoiding the difficulty of directly extracting high-ranking features.

By adding the ACBN, the mIoU of the model is improved by 1.1%. The skeleton of the square convolution kernel is strengthened due to the asymmetric convolutions.

By adding the EDA, the richness of training samples is increased by global scaling and global translation. The mIoU is improved by another 0.5% on the SemanticKITTI validation set, reaching 59.6%. The refined ablation experiment results show that the effect of global scaling is basically consistent with that of global translation, but the effect of global scaling is slightly stronger than that of global translation.

From the results of ablation experiments, it can be concluded that the methods proposed in this paper all lead to gains in performance.

4.4. Influence of Grid Density

In this section, the influence of grid density on the model is analyzed. When partitioning the original point clouds, the segmentation accuracy and speed are affected by grid density. To verify whether higher speed can be achieved by sacrificing some accuracy, ACPNet-mini is designed by varying the grid density. The grid sizes of ACPNet and ACPNet-mini are 480 × 360 × 32 and 320 × 240 × 32, respectively, where the three dimensions represent radius, angle and height.

According to Table 5, ACPNet-mini sacrifices 1.9% of the mIoU by reducing the computation, resulting in a 33.3% improvement in running speed. Besides, it can be found that ACPNet achieves a real-time effect without introducing additional computation while having a large improvement in mIoU compared to the baseline model.

Table 5. Experiments with different grid sizes on SemanticKITTI Validation Set (seq 08).

	Grid Size	**mIoU (%)**	**Params (M)**	**FPS (Hz)**
Baseline	480 × 360 × 32	56.4	13.6	16
ACPNet (ours)	480 × 360 × 32	59.6	10.3	14
ACPNet-mini (ours)	320 × 240 × 32	57.7	10.3	19

5. Conclusions

An efficient real-time LiDAR point clouds semantic segmentation model ACPNet is proposed in this paper. Asymmetric Convolution Backbone Network and Contextual Feature Enhancement Module are proposed to improve the feature extraction ability of the model, and Enhanced Data Augmentation methods are used to enrich the diversity of training samples. Compared with the baseline network PolarNet, the mIoU of ACPNet on the SemanticKITTI, SemanticPOSS and nuScenes datasets are improved by 5.1%, 1.6% and 2.9%, respectively. Meanwhile, the speed of ACPNet is 14 FPS, which basically meets the real-time requirements in autonomous driving scenarios. Besides, ACPNet-mini is designed by reducing the grid density in the point clouds encoding stage, significantly increasing the speed at the expense of smaller segmentation accuracy. In summary, ACPNet essentially satisfies the demands of real-time semantic segmentation of LiDAR point clouds for autonomous driving.

6. Discussion

In the future, we will continue to investigate more general and effective methods to enhance performance. Additionally, we plan to expand our approach to achieve end-to-end 3D panoptic segmentation on LiDAR point clouds for autonomous driving.

Author Contributions: Conceptualization, X.S. and S.S.; methodology, X.S. and Z.M.; software, X.S. and Z.M.; validation, X.S., P.T. and L.A.; formal analysis, X.S. and P.T.; investigation, X.S. and L.A.; resources, X.S.; data curation, X.S.; writing—original draft preparation, X.S. and S.S.; writing—review and editing, X.S. and S.S; visualization, X.S. and S.S.; supervision, X.S.; project administration, S.S.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: The generous support of the Discipline Construction of Computer Science and Technology of Shanghai Polytechnic University under Grant B60KY150002-02 are gratefully acknowledged.

Data Availability Statement: The data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study. They can be requested from the author at e-mail (20211510167@stu.sspu.edu.cn) in the future.

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

References

1. Zhang, Y.; Zhou, Z.; David, P.; Yue, X.; Xi, Z.; Gong, B.; Foroosh, H. 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, Seattle, WA, USA, 13–19 June 2020; pp. 9601–9610.
2. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9297–9307.
3. Pan, Y.; Gao, B.; Mei, J.; Geng, S.; Li, C.; Zhao, H. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 687–693.
4. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
5. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
6. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5099–5108.
7. Wu, W.; Qi, Z.; Fuxin, L. Pointconv: Deep convolutional networks on 3d point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9621–9630.
8. Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4558–4567.
9. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11108–11117.
10. Thomas, H.; Qi, C.R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6411–6420.
11. Lu, T.; Wang, L.; Wu, G. Cga-net: Category guided aggregation for point cloud semantic segmentation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11693–11702.
12. Qiu, S.; Anwar, S.; Barnes, N. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1757–1767.
13. Fan, S.; Dong, Q.; Zhu, F.; Lv, Y.; Ye, P.; Wang, F.-Y. SCF-Net: Learning spatial contextual features for large-scale point cloud segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14504–14513.
14. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. Segcloud: Semantic segmentation of 3d point clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 537–547.
15. Graham, B.; Engelcke, M.; Van Der Maaten, L. 3d semantic segmentation with submanifold sparse convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9224–9232.
16. Choy, C.; Gwak, J.; Savarese, S. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3075–3084.
17. Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; Han, S. Searching efficient 3d architectures with sparse point-voxel convolution. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 685–702.
18. Zhou, H.; Zhu, X.; Song, X.; Ma, Y.; Wang, Z.; Li, H.; Lin, D. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv* **2020**, arXiv:2008.01550.
19. Cheng, R.; Razani, R.; Taghavi, E.; Li, E.; Liu, B. 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, Nashville, TN, USA, 20–25 June 2021; pp. 12547–12556.
20. Hou, Y.; Zhu, X.; Ma, Y.; Loy, C.C.; Li, Y. Point-to-voxel knowledge distillation for lidar semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8479–8488.

21. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1887–1893.
22. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4376–4382.
23. Xu, C.; Wu, B.; Wang, Z.; Zhan, W.; Vajda, P.; Keutzer, K.; Tomizuka, M. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 1–19.
24. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
25. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. Rangenet++: Fast and accurate lidar semantic segmentation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4213–4220.
26. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
27. Alonso, I.; Riazuelo, L.; Montesano, L.; Murillo, A.C. 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5432–5439. [CrossRef]
28. Peng, K.; Fei, J.; Yang, K.; Roitberg, A.; Zhang, J.; Bieder, F.; Heidenreich, P.; Stiller, C.; Stiefelhagen, R. MASS: Multi-attentional semantic segmentation of LiDAR data for dense top-view understanding. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 15824–15840. [CrossRef]
29. Cortinhal, T.; Tzelepis, G.; Erdal Aksoy, E. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In Proceedings of the Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, 5–7 October 2020; Proceedings, Part II 15; pp. 207–222.
30. Li, S.; Chen, X.; Liu, Y.; Dai, D.; Stachniss, C.; Gall, J. Multi-scale interaction for real-time lidar data segmentation on an embedded platform. *IEEE Robot. Autom. Lett.* **2021**, *7*, 738–745. [CrossRef]
31. Zhao, Y.; Bai, L.; Huang, X. Fidnet: Lidar point cloud semantic segmentation with fully interpolation decoding. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 4453–4458.
32. Cheng, H.X.; Han, X.F.; Xiao, G.Q. Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6.
33. Ding, X.; Guo, Y.; Ding, G.; Han, J. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1911–1920.
34. Qiu, S.; Anwar, S.; Barnes, N. Pnp-3d: A plug-and-play for 3d point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *45*, 1312–1319. [CrossRef] [PubMed]
35. Zhang, H.; Zhang, H.; Wang, C.; Xie, J. Co-occurrent features in semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 548–557.
36. Chen, W.; Zhu, X.; Sun, R.; He, J.; Li, R.; Shen, X.; Yu, B. Tensor low-rank reconstruction for semantic segmentation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XVII 16; pp. 52–69.
37. Hahner, M.; Dai, D.; Liniger, A.; Van Gool, L. Quantifying data augmentation for lidar based 3d object detection. *arXiv* **2020**, arXiv:2004.01643.
38. Berman, M.; Triki, A.R.; Blaschko, M.B. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4413–4421.
39. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
40. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. *arXiv* **2017**, arXiv:1704.06857.
41. Rosu, R.A.; Schütt, P.; Quenzel, J.; Behnke, S. Latticenet: Fast point cloud segmentation using permutohedral lattices. *arXiv* **2019**, arXiv:1912.05905.
42. Wen, S.; Wang, T.; Tao, S. Hybrid CNN-LSTM architecture for LiDAR point clouds semantic segmentation. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5811–5818. [CrossRef]