applied sciences

Article

Real-Time Semantic Segmentation of Point Clouds Based on an Attention Mechanism and a Sparse Tensor

Fei Wang, Yujie Yang, Zhao Wu, Jingchun Zhou and Weishi Zhang *

College of Information Science and Technology, Dalian Maritime University, Dalian 116026, China
* Correspondence: teesiv@dlmu.edu.cn

Abstract: A 3D point cloud is one of the main data sources for robot environmental cognition and understanding. Due to the limited computation and memory capacities of the robotic platform, existing semantic segmentation models of 3D point clouds cannot meet the requirements of real-time applications. To solve this problem, a lightweight, fully convolutional network based on an attention mechanism and a sparse tensor is proposed to better balance the accuracy and real-time performance of point cloud semantic segmentation. On the basis of the 3D-Unet structure, a global feature-learning module and a multi-scale feature fusion module are designed. The former improves the ability of features to describe important areas by learning the importance of spatial neighborhoods. The latter realizes the fusion of multi-scale semantic information and suppresses useless information through the task correlation learning of multi-scale features. Additionally, to efficiently process the large-scale point clouds acquired in real time, a sparse tensor-based implementation method is introduced. It is able to reduce unnecessary computation according to the sparsity of the 3D point cloud. As demonstrated by the results of experiments conducted with the SemanticKITTI and NuScenes datasets, our model improves the mIoU metric by 6.4% and 5%, respectively, over existing models that can be applied in real time. Our model is a lightweight model that can meet the requirements of real-time applications.

Keywords: 3D point cloud; attention mechanism; semantic segmentation; sparse tensor

Citation: Wang, F.; Yang, Y.; Wu, Z.; Zhou, J.; Zhang, W. Real-Time Semantic Segmentation of Point Clouds Based on an Attention Mechanism and a Sparse Tensor. *Appl. Sci.* **2023**, *13*, 3256. https://doi.org/10.3390/app13053256

Academic Editor: Luoyi Fu

Received: 17 February 2023
Revised: 28 February 2023
Accepted: 1 March 2023
Published: 3 March 2023

1. Introduction

With the rapid development of deep learning technology, computer vision, a popular research direction in the artificial intelligence field, is now widely used in areas such as driverless and unmanned ground vehicles (UGV). The operation of intelligent systems such as unmanned vehicles relies heavily on sensors to capture information about the environment around the vehicle. The semantic segmentation of point clouds is a key aspect of scene understanding [1] for intelligent systems. It enables the segmentation of different objects in a scene by predicting the label of each point. Semantic segmentation [2] not only helps to identify and localize dynamic objects and determine road conditions, but it can also be used to construct a 3D semantic map of the environment [3]. However, the amount of point clouds collected by LiDAR (light detection and ranging) in real time is extremely large and spatially unevenly distributed. With the limited computing power and memory of unmanned vehicle systems, the existing segmentation models fail to effectively balance the relationship between model accuracy, computational efficiency, and real-time performance. The realization of real-time semantic segmentation of point clouds still faces great challenges.

Currently, semantic segmentation methods are classified into three types depending on the way the point cloud data are represented: the raw point-based method [4,5], the multi-view-based method [6,7], and the voxelization-based method [8,9]. The raw point-based method takes point cloud data directly as input and processes irregular point clouds by designing special convolution operations [5]. However, this method is less computationally efficient and is generally used for small-scale point cloud computation in indoor settings; it is not considered effective for segmenting large-scale point clouds for outdoor scenes. Although the multi-view-based method meets the real-time needs of the system, the process of projecting 3D information into 2D can result in a significant loss of the original information and affect the segmentation accuracy. The voxelization-based approach divides the disordered point cloud into an ordered raster representation. This method has achieved better performance in terms of 3D data. Due to the sparsity of the point cloud data, it is inefficient to apply the voxelated point cloud data directly to dense convolutional neural networks. For this reason, many models have introduced sparse operations based on voxelization methods. Examples include SSCN [10], LatticeNet [11], and Minkowski-Net [8], but the segmentation accuracy of these models is limited by the real-time nature of the system. Therefore, current point cloud semantic segmentation models are unable to simultaneously meet the system's real-time requirements and increase the model accuracy.

To achieve the real-time semantic segmentation of unmanned vehicle systems, we propose a lightweight, fully convolutional network (LFNet) based on an attention mechanism and a sparse tensor to process voxelized point cloud data. First, we propose a global feature-learning module (GFLM) based on the 3D-Unet structure, which improves the ability of features to describe important regions by learning the importance of spatial domains. In addition, we propose a multi-scale feature fusion module (MFFM), which enables the fusion of multi-scale semantic information by learning features at different scales. Additionally, to handle the large amount of point cloud information acquired by the intelligent system in real time and to reduce the model computation, we introduce a sparse tensor-based implementation. Finally, our model improves the mIoU (mean intersection over union) index by 6.4% and 5%, respectively, for the SemanticKITTI and NuScenes datasets compared with the existing models that can be applied in real time. The main contributions of this work are as follows:

- A lightweight full convolutional network based on an attention mechanism and a sparse tensor is proposed to solve the real-time point cloud semantic segmentation problem for unmanned vehicle systems;
- We propose a global feature-learning module and a multi-scale feature fusion module. The former is used to learn global contextual information, and the latter is used to achieve the effective fusion of feature information at different scales;
- As demonstrated by the experimental results for the SemanticKITTI and NuScenes datasets, our model improves the mIoU metric by 6.4% and 5%, respectively, over existing models that can be applied in real time. The comparative efficiency analysis shows that our model is able to meet the real-time requirements.

The rest of this paper is organized as follows. Section 2 summarizes the existing semantic segmentation methods. Section 3 describes the structure of LFNet and the two modules in detail. Section 4 gives the experimental results of LFNet on different datasets. Section 5 provides a summary of the work.

2. Related Works

2.1. Point Cloud Semantic Segmentation

The semantic labeling of preprocessed [12] disorganized point cloud data acquired by sensors [13] is the goal of semantic segmentation of point clouds. In general, we can classify the existing methods into four categories, namely, raw point-based, projection-based, graph model-based, and voxelization-based methods.

Raw point-based methods are usually performed directly on irregular point clouds. A representative work is PointNet [4], which uses MLPs to extract features of points and global features through symmetric functions. Thomas et al. [5] proposed kernel point convolution (KPConv) to directly process point clouds, where the convolution kernel of KPConv consists of a series of kernel points with weights, each with an influence distance. Fang et al. [14] proposed a spatially transformed point convolution (STPC) to achieve the anisotropic convolutional filtering of point clouds and introduced a spatial direction dictionary to capture the geometric structure of the data.

Projection-based methods typically project 3D data onto 2D images and then perform subsequent segmentation using image processing methods [15–18]. However, this method cannot take full advantage of the structural information of the 3D data. To solve the problem of the uneven distribution of LiDAR points in space, Zhang et al. [6] performed end-to-end learning of 2D semantic segmentation models via polarized bird's-eye-view representations. To solve the problem of large differences in the distribution of features at different image locations, Xu et al. [7] proposed a spatially adaptive convolution (SAC).

Regarding graph model-based methods, Li et al. [19] used point convolution and point pooling methods to learn the high-level features of point clouds and then combined point convolution, point pooling, and a graph structure to propose a point cloud feature-learning network for the feature learning of point clouds. To solve the hyperspectral image problem, Ding [20] et al. proposed the feature fusion network to learn the spectral features of nodes and cluster the hyperspectral images.

Due to the sparse nature of the point cloud data, the method of voxelizing the point cloud into a dense grid is less efficient. For this reason, Choy et al. [8] proposed Minkowski-Net and generalized sparse convolution algorithms for processing 3D videos. For better recognition of small elements, such as pedestrians and bicycles, Tang et al. [9] proposed sparse point voxel convolution and 3D-NAS for 3D scene understanding. Zhu et al. [21] proposed a LiDAR segmentation framework consisting of cylindrical coordinate voxel partitioning and the asymmetric 3D convolutional network, Cylinder3D.

2.2. *Attention Mechanism*

The introduction of attention mechanisms was mainly influenced by the ability of humans to find the most important parts in various complex scenarios and to ignore the unimportant parts. Currently, there are many kinds of attention mechanisms, and we focus on the attention mechanisms relevant to this paper, which mainly include channel attention, spatial attention, and self-attention.

Channel attention focuses on the channel parts of different feature maps and selects the most important parts. The idea of channel attention was first proposed by Hu et al. [22], and SENet networks were designed to obtain the relationships between channels. To achieve information interactions between different features, Zhou et al. [23] presented a channel and spatial attention-based network.

Spatial attention is mainly concerned with the selected spatial region. Feng et al. [24] presented a local attention-edge convolution network to construct a local graph of neighboring points based on a multi-directional search and learn local features by spatial attention. To effectively obtain the correlation between the global and local information of point clouds, Chen et al. [25] presented PointSCNet, which uses channel attention and spatial attention to refine point cloud features. Chen et al. [26] introduced a spatial attention module between different domains to obtain contextual information.

The self-attentive mechanism focuses on the underlying relationships among the input information and uses the relationships to assign weights to the input information. Sun et al. [27] presented an autoregressive model, PointGrow, to enhance the model by introducing a self-attentive module, thus making it possible to generate diverse and realistic point cloud samples based on the semantic context. Due to the unstructured nature of the point cloud, it is difficult to use deep neural networks directly on the original data. For this reason, Wang et al. [28] proposed a crossed self-attentive network with permutation invariance that can learn both the coordinates and features of the point cloud. Wen et al. [29] presented the feature of the context fusion network CF-SIS by introducing a self-attentive mechanism to fuse contextual features.

2.3. Sparse Convolution in Point Cloud Processing

Due to the sparse nature of the data, the direct application of convolutional networks is often inefficient, and sparse convolution is therefore used to process point cloud data. Graham et al. [10] proposed the SSCN to efficiently process spatially sparse data by introducing a new sparse convolutional operation, which greatly saves computational resources. Choy et al. [8] proposed Minkowski-Net along with a generalized sparse convolution of sparse tensors and a self-differentiating library. For the efficient processing of large amounts of data, Rosu et al. [11] proposed LatticeNet, which back-projects grid features onto point clouds via a data-dependent interpolation module, DeformsSlice. To reduce the calculation effort, Su et al. [30] proposed a sparse grid network based on bilateral convolutional layers to compute hierarchical and spatially perceptive features. To extract structural information from the point cloud, Gu et al. [31] proposed DownBCL, Up-BCL, and CorrBCL based on bilateral convolutional layers (BCL).

In summary, the original point-based method is usually applied to indoor point clouds, which is difficult to adapt to large-scale data. Although the graph model-based method has a higher segmentation accuracy, the computational efficiency cannot meet the requirements of real-time applications of the system. Although the projection-based method has a higher computational efficiency, some spatial information will be lost in the projection transformation. The voxelization-based method can extend the 2D model to 3D space, but the corresponding sparse operations need to be designed to solve the computational efficiency problem. Therefore, to achieve real-time semantic segmentation for unmanned vehicle systems with limited computing power and memory capacities, we designed two attention modules based on the voxelization method. A sparse tensor-based implementation is also introduced to improve the accuracy and real-time performance of semantic segmentation.

3. Methods

The proposed attention-based mechanism and sparse tensor-based lightweight full convolutional network model (LFNet) are described in detail in this section. Our LFNet consists of a 3D-Unet model with a feature-learning and fusion module. In Part 1, we present the overall structure and implementation of the LFNet model. Then, the model accuracy is improved by GFLM and MFFM in Parts 2 and 3. Furthermore, to achieve real-time semantic segmentation, we present the sparse tensor-based model implementation method adopted by LFNet in Part 4.

3.1. Overview

Due to the memory limitation of intelligent systems, existing semantic segmentation models are difficult to achieve the requirements of real-time applications for unmanned vehicle systems. Therefore, based on the 3D-Unet [32] model, a real-time lightweight, fully convolutional network, LFNet, based on an attention mechanism and a sparse tensor, is proposed (as shown in Figure 1). Our model is a lightweight, fully convolutional neural network implemented using a sparse tensor. The left half of the figure shows the feature extraction network, which uses multiple stacked convolution blocks and pooling operations to learn point cloud features at various scales. The right half is the feature fusion network where feature fusion is achieved through transposed convolutional operations and the MFFM.

The GFLM and MFFM are designed. Our model improves the feature representation of important regions by acquiring contextual information and learning the importance of spatial neighborhoods through a global feature-learning module. Our model learns the weight distribution of features at different scales through the MFFM, thus enhancing the fusion of multi-scale semantic information. To improve the calculation efficiency of the model, we use a sparse tensor representation of the rasterized point cloud data. The sparse approach only stores the coordinates and feature information of the raster, thus reducing the memory footprint. Convolution, pooling, normalization, and other operations in the model are implemented based on the sparse tensor in order to reduce the unnecessary calculation quantity.

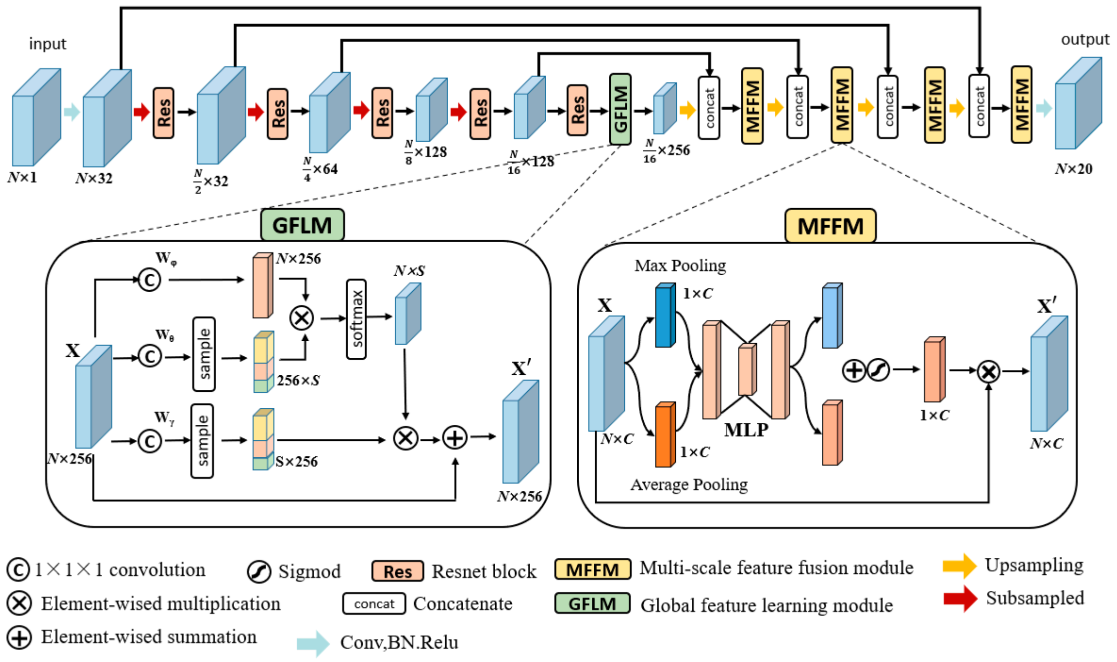

Figure 1. Real-time point cloud semantic segmentation network model structure diagram.

3.2. Global Feature-Learning Module

To obtain global contextual information and to distinguish the importance of different spatial location information in the input features, our model adds the GFLM to the deep features of the subsampling module to obtain a spatial weight matrix through training. The specific structure is described below. First, the input feature $X \in R^{(N \times C)}$ is passed through three 1×1 convolutions, W_φ, W_θ, and W_γ, to obtain the features $\varphi \in R^{(N \times C')}$, $\theta \in R^{(N \times C')}$, and $\gamma \in R^{(N \times C')}$, respectively:

$$\varphi = W_\varphi(X) \tag{1}$$

$$\theta = W_\theta(X) \tag{2}$$

$$\gamma = W_\gamma(X) \tag{3}$$

A sampling module is added after the features θ and γ. In this module, 3 pooling operations of 2×2, 4×4, and 8×8 are performed, respectively, and the results are concatenated and used as inputs for the next layer. The sampled results are denoted as $\theta_p \in R^{(S \times C')}$ and $\gamma_p \in R^{(S \times C)}$ where S is the number of anchor points sampled. The similar matrix V_p between φ and θ_p is then calculated as follows:

$$V_p = \varphi \cdot \theta_p{}^T \tag{4}$$

where $V_p \in R^{(N \times S)}$, which after a normalization function, is obtained as $V_p{}'$. The attention matrix is obtained as:

$$Q_p = Vp\prime \cdot \gamma_p \tag{5}$$

Finally, this module sums the attention map with the input features to obtain the feature $X' \in R^{(N \times C)}$. X' is used as the final output feature with the following equation:

$$X' = W_Q(Q_p) + X \tag{6}$$

where W_Q is the 1×1 convolution for recovering the channel number C' to C.

In addition, in the process of designing the structure of the global feature-learning module, the idea of a lightweight design was introduced to reduce the calculation quantity of the model. The amount of information, such as coordinates and features, in the point cloud data is extremely large. If the dot-product operation is performed directly on the features φ and θ, the model requires a huge amount of computation. To enhance the efficiency of the model calculations, our model adds a sample module consisting of several pooling operations after the 1×1 convolution, W_θ and W_γ. The sample module samples several sparse anchor points to avoid selecting all spatial points. As a result, the computational complexity is significantly reduced. The data are shown in Table 1. Whether or not the sampling module is added to the global feature-learning module has little effect on the calculation of the number of parameters, which are all 0.2 M. In addition, the FLOPs (floating-point operations per second) were 1.14 G before the sample module was added to this global feature-learning module. With the addition of the sample module, the model FLOPs are reduced by 0.2 G. Therefore, this operation effectively reduces the calculation quantity of the model.

Table 1. Comparison of calculation volumes for the lightweight design of a global feature-learning module.

Method	Params/M	FLOPs/G
Sampling model	0.2	0.94
No sampling model	0.2	1.14

3.3. Multi-Scale Feature Fusion Module

Our model incorporates the MFFM into the feature fusion network in order to distinguish the importance of different features. The channel attention map is generated using the relationship between feature channels, and it is multiplied with the original input feature element to obtain a new feature element. To efficiently compute features at different scales, the spatial dimension of the input feature map is compressed in our model. First, this module aggregates the spatial information of the input features $X \in R^{(N\times C)}$ by global average pooling and maximum pooling. Additionally, the features, $X_{avg} \in R^{(1\times C)}$ and $X_{max} \in R^{(1\times C)}$, are obtained by the above pooling layers. X_{avg} and X_{max} are formulated as follows:

$$X_{avg} = \mathrm{AvgPool}(X) \tag{7}$$

$$X_{max} = \mathrm{MaxPool}(X) \tag{8}$$

Then, they are fed separately into a two-layer multilayer perceptron (MLP). To reduce the parameters and computational overheads, the number of channels in the first layer of the MLP is set to C/r, where r is the reduction rate. In addition, the activation function is ReLU, and the second layer channel number is C. Afterwards, the outputs of the features from the two pooling layers through the MLP are summed by element-wise multiplication and afterwards by the sigmoid activation layer to generate the final attention feature map M_c. M_c is calculated as follows:

$$M_c = \sigma\left(\mathrm{MLP}\left(X_{avg}\right) + \mathrm{MLP}\left(X_{max}\right)\right) \tag{9}$$

where the activation function sigmoid is denoted by σ, and the two input features, X_{avg} and X_{max}, share the weights of the MLP. Finally, the output feature X_{out} is obtained by multiplying the attentional feature map M_c and the input features X. X_{out} is formulated as follows:

$$X_{out} = M_c \cdot X \tag{10}$$

3.4. Sparse Operation

As point clouds are sparse in terms of spatial distribution, a sparse tensor-based model implementation method is used for our model to avoid the large number of invalid calculations performed in three-dimensional space by voxelization methods.

3.4.1. Sparse Tensor

In the voxelization method, a sparse tensor is used to represent the voxelized point cloud data to reduce the memory footprint. The sparse tensor ζ is a high-dimensional extension of the sparse matrix, which contains two parts. The first part is a set of coordinates C. The coordinates are a matrix of $N \times 3$, where N is the number of nonzero elements in the sparse tensor, which is used to indicate that there are N data points per frame. In addition, the number 3 shows that each point contains x, y, and z coordinate information. The second part is the associated feature $F \in R^{(N\times C)}$ in which C is the number of channels. The features of the input sparse tensor are initialized to one, which is an $N \times 1$ matrix where all the data are 1 s. The sparse tensor is represented as follows:

$$C = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 \\ \vdots & \vdots & \vdots \\ x_N^1 & x_N^2 & x_N^3 \end{bmatrix},\ F = \begin{bmatrix} f_1^T \\ \vdots \\ f_N^T \end{bmatrix} \tag{11}$$

$$\zeta\left[x_i^1, x_i^2 \cdots, x_i^3\right] = \begin{cases} f_i & \text{if}\left(x_i^1, x_i^2, \cdots, x_i^3\right) \in C \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

3.4.2. Submanifold Convolution Operation

The vast majority of the model's calculation quantity comes from convolution operations. The D-dimensional traditional convolution formula is as follows:

$$x_u^{out} = \sum\nolimits_{i\in V^D(k)} W_i x_{i+u}^{in},\ \text{for } u \in Z^D \tag{13}$$

where k denotes the convolution kernel size, x denotes the feature vector, W denotes the convolution kernel weight, and N^D (k) denotes a set of offsets that defines the shape of the convolution kernel.

However, the proportion of nonzero values in the point cloud data is very small. As a result, it is inefficient to apply conventional convolution to sparse data. For this reason, submanifold convolution based on a sparse tensor is used as the implementation for all convolution layers in the model. Submanifold convolution can reduce the calculation effort by avoiding calculations in the blank space of the point cloud. In addition, the number of valid points is the same for each layer of convolution.

In the submanifold convolution operation, the nonzero values of the input data are first preserved by indexing and associating features. The convolution output is only calculated when the valid input points are covered by the convolution kernel's center, that is, the input data contain five valid rasters and the output data also contain five valid rasters. Taking the $3 \times 3 \times 3$ convolution operation as an example, the ordinary convolution performs 27 multiplications. However, in submanifold convolution, the actual number of operations is much lower than 27 because the input features contain a large number of zero elements.

3.4.3. Other Operations

In addition to the convolution operation, the model contains structures such as normalization layers and activation layers. The computation process of these structures does not change the index matrix of the sparse tensor. Taking the activation function as an example, the sigmoid function is formulated as follows:

$$\sigma(x) = \frac{1}{(1 + e^{-x})} \tag{14}$$

The sigmoid function is only applied to the valid raster of the input data; therefore, only the associated feature F is changed, and it does not affect the calculation of the index matrix C. The sigmoid function in sparse operations is expressed as:

$$\sigma(\zeta)= \zeta(\mathrm{C}, \sigma(\mathrm{F})) \tag{15}$$

4. Results

The public point cloud datasets SemanticKITTI and NuScenes were selected as the experimental data. The LFNet model is compared with existing models of the same category in a series of experiments. This section affirms the validity of our model through qualitative analysis and quantitative comparison.

4.1. Datasets

We selected the public datasets SemanticKITTI and NuScenes in the field of autonomous driving as our experimental data.

SemanticKITTI: The semantic category of this dataset contains 20 objects divided into the categories of moving and nonmoving objects, including traffic participants such as cyclists, motorcyclists, and pedestrians, but also nonmoving objects such as car parks, pavements, and roads. The dataset consists of 43,551 frames of LiDAR scans from 22 sequences. Each sequence is scanned by LiDAR in different street scenes and contains between 200 and 4500 frames of point cloud data. The scene data from sequence 00 to sequence 10 were used as the experimental data. To reduce data repetition and avoid overfitting, the experimental data were selected from one frame from five consecutive frames. The selected data were divided into a training set (about 60%), a validation set (about 20%), and a test set (about 20%).

NuScenes: This dataset collects information from 1000 driving scenarios in different cities and manually selects scenarios of 20 s duration. The complete dataset contains 390,000 LiDAR scans and 40,000 keyframes. The dataset contains 32 semantic labels in the LiDAR semantic segmentation task. In the experiments, 850 scenes with semantic annotations were used as the experimental data. These selected data were also divided into a training set (about 60%), a validation set (about 20%), and a test set (20%).

4.2. Experimental Setup

During the training of the LFNet model, two data enhancement methods were adopted to enhance the robustness of the model. First, the point cloud was scaled along the x, y, and z dimensions with a scaling factor in the range of [0.95, 1.05]. In addition, the point cloud was randomly rotated by $\alpha \in [-\pi/6, \pi/6]$ degrees about the x and y axes and by $\alpha \in [-\pi, \pi]$ degrees about the z axis.

The model was parameter optimized using the Adam optimizer and set to an initial learning rate of 0.0001, a weight decay of 0.0001, a momentum of 0.9, a batch size of 16, and a maximum number of 300 iterations.

Our model was implemented using the programming interface provided by Minkowski Engine. The server hardware environment was an Inter Core i9-10980XE (3.0 GHz), 32 GB RAM, and an Nvidia GeForce3090 (24 GB). The operating system was Ubuntu 18.04. Other software used for the experiments included Pytorch 1.8, CUDA Toolkit 11.1, and CUDNN 8.0.

4.3. Evaluation Criterion

Accuracy (Acc) and the mIoU were used to evaluate the precision of the semantic segmentation results. The accuracy rate is the proportion of all correct judgments in the classification model to the total observed samples. mIoU is calculated as the ratio of the intersection and the union of the two sets of true and predicted values. The formulas are as follows:

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FT} + \mathrm{FN}} \tag{16}$$

$$\text{mIoU} = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{17}$$

where TP is the true example of a correct prediction, FP is the false positive example of an incorrect prediction, FN is the false negative example of an incorrect prediction, TN is the true negative example of a correct prediction, k is the number of categories, j is the predicted value, i denotes the true value, and p_{ij} denotes the prediction of i to j.

4.4. Qualitative Comparison

In this experiment, 3D-Unet was chosen as the baseline model. Typical semantic segmentation results of the LFNet and the 3D-Unet model for the SemanticKITTI dataset are shown in Figure 2. The 3D-Unet model's predictions are consistent with the true value category for most objects. For example, the 3D-Unet model is able to identify the "car" and separate it from the "ground". However, the model's predictions are relatively coarse as it only detects the general outline of the object and is unable to distinguish well between the edge boundaries of the object. For example, some of the "sidewalk" in the 3D-Unet model is incorrectly identified as "terrain", and some of the 'buildings' are incorrectly identified as 'fences', etc. Compared with the 3D-Unet model, the LFNet focuses on instances where the boundary lines and contours are not clear enough to learn the details of the object. The segmentation results show that the LFNet is able to accurately identify boundary information, such as sidewalks, street building outlines, etc.

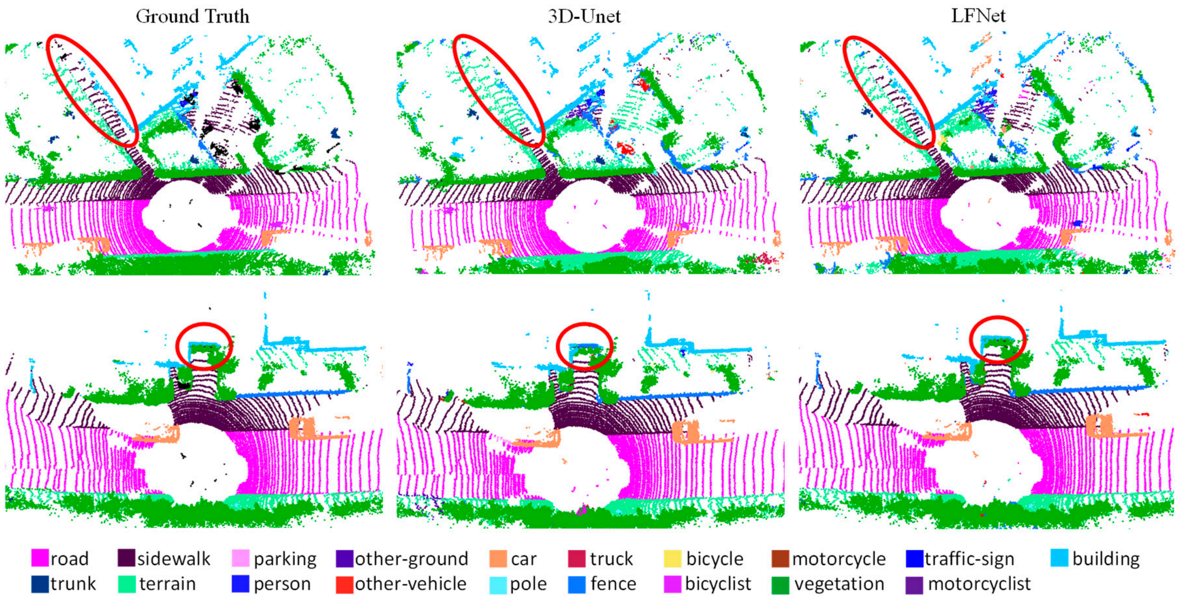

Figure 2. LFNet visualization results for the SemanticKITTI dataset.

Typical semantic segmentation results of the LFNet and the 3D-Unet model for the NuScenes dataset are shown in Figure 3. The 3D-Unet model is able to distinguish moving objects such as cars and trucks to a certain extent, but its predictions are not accurate, and there are many misclassifications. For example, the 3D-Unet model does not accurately identify the outlines of vehicles, and the model confuses "trucks" with "trailers". In addition, the model is not accurate in identifying pedestrians and cannot identify all pedestrians in outdoor scenes. The majority of LFNet's predictions are consistent with the category of ground truth, and it can accurately identify pedestrians, vehicle contours, etc.

From the segmentation results, it is clear that the LFNet model is able to focus on small objects as well as boundary information through the GFLM and the MFFM. Thus, it learns more detailed features in large scenes and eventually obtains better semantic segmentation results.

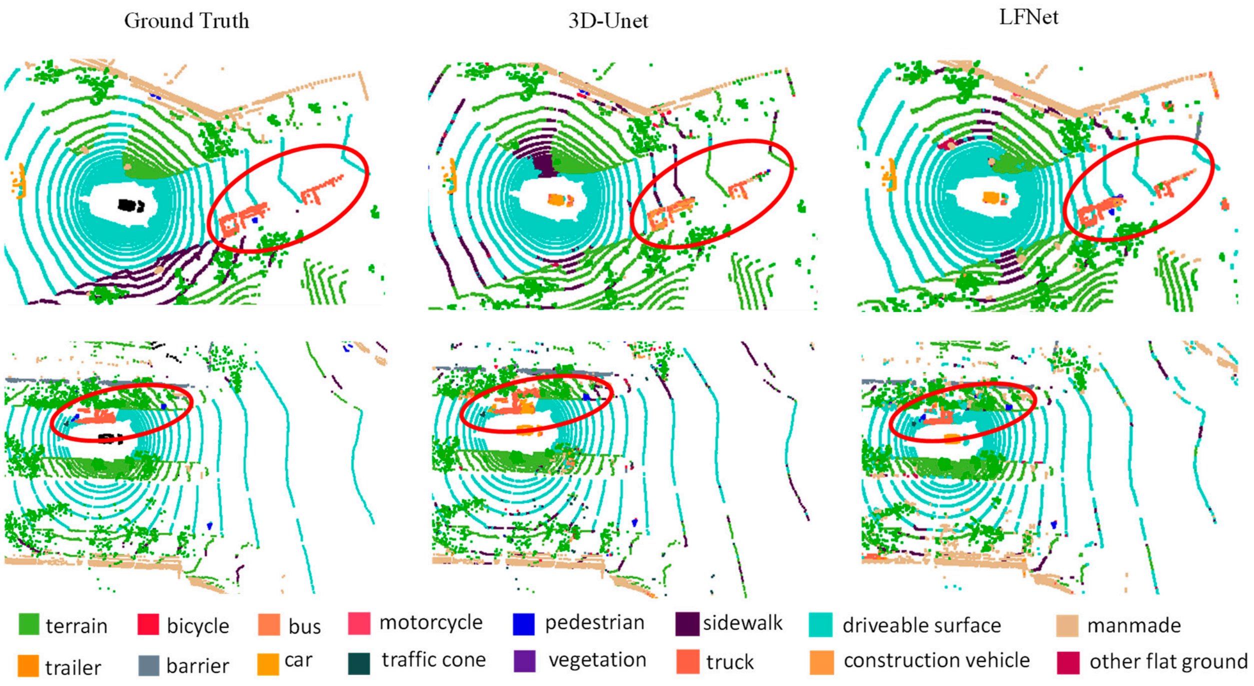

Figure 3. LFNet visualization results for the NuScenes dataset.

4.5. Quantitative Comparisons

SemanticKITTI dataset: This experiment compares LFNet with existing models for the SemanticKITTI public dataset, and the statistical results of each method are shown in Table 2. LFNet outperforms most existing models in terms of mIoU metrics. Compared to projection-based 2D spatial methods (including SqueezeSegv3 and PolarNet), our model achieves a 16.9–18.5% performance improvement. Compared with the STPC method based on raw points, the mIoU of LFNet improves by 18.2%. Compared to Cylinder3D, SPVNAS, and FusionNet, which are based on voxel partitioning and 3D convolution, LFNet improves by 5–11.5% in terms of mIoU.

LFNet improved by only 2.5% in mIoU compared to RPVNet, which incorporates all three model structures. The RPVNet model uses three types of input data: voxelized point clouds, raw point clouds, and depth maps. It performs feature learning and information fusion on different forms of data through the three networks. It is able to fully mine semantic information from multiple data sources using a more complex fusion network. However, due to its high degree of complexity and computational costs, the RPVNet model cannot achieve the real-time operation of a system.

NuScenes dataset: In addition, LFNet is compared with the existing models for the NuScenes dataset, and the statistical results of each method are shown in Table 3. LFNet outperforms most existing models in terms of mIoU. Compared with the fusion methods based on raw points and voxelization (including JS3C-Net and AF2S3Net), LFNet improves by 6.2–17.6% in terms of mIoU. Compared to multi-view-based methods (including PolarNet and SalsaNext), LFNet improves by 8.8–21.3% in terms of mIoU. Compared to Cylinder3D and SPVNAS, which are based on voxel partitioning and 3D convolution, LFNet improves by 3.7–5% in terms of mIoU.

Compared to LidarMultiNet, LFNet has a 1.6% reduction in mIoU. LidarMultiNet extracts both 2D and 3D information through different branches and processes the semantic segmentation results of the first stage through a second-stage refinement module. However, due to the complexity of the model structure, LidarMultiNet cannot be applied to real-time semantic segmentation.

Table 2. Results of LFNet and existing models for the SemanticKITTI dataset.

Method	**Real Time**	**Acc/%**	**mIoU/%**	**Car**	**Bicycle**	**Motorcycle**	**Truck**	**Other-Vehicle**	**Person**	**Bicyclist**	**Motorcyclist**	**Road**	**Parking**	**Sidewalk**	**Other-Ground**	**Building**	**Fence**	**Vegetation**	**Trunk**	**Terrain**	**Pole**	**Traffic-Sign**
RPVNet [33]	no	-	70.3	**97.6**	**68.4**	**68.7**	44.2	61.1	**75.9**	**74.4**	**73.4**	93.4	70.3	80.7	33.3	**93.5**	72.1	86.5	**75.1**	71.8	**64.8**	61.4
Cylinder3D [21]	no	-	67.8	97.1	67.6	64.0	50.8	58.6	73.9	67.9	36.0	91.4	65.1	75.5	32.3	91.0	66.5	85.4	71.8	68.5	62.6	65.6
FusionNet [34]	no	91.8	61.3	95.3	47.5	37.7	41.8	34.5	59.5	56.8	11.9	91.8	68.8	77.1	30.8	92.5	69.4	84.5	69.8	68.5	60.4	66.5
STPC [14]	no	-	54.6	94.7	31.1	39.7	34.4	24.5	51.1	48.9	15.3	90.8	63.6	74.1	5.3	90.7	61.5	82.7	62.1	67.5	51.4	47.9
SPVNAS [9]	**yes**	-	66.4	97.3	51.5	50.8	59.8	58.8	65.7	65.2	43.7	90.2	67.6	75.2	16.9	91.3	65.9	86.1	73.4	70.0	64.2	**66.9**
SqueezeSegV3 [7]	**yes**	88.6	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
PolarNet [6]	**yes**	90.0	54.3	83.8	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
LFNet (ours)	**yes**	**93.3**	**72.8**	92.5	43.8	63.6	**84.1**	**75.8**	41.9	69.2	52.6	**95.8**	**77.7**	**85.9**	**77.6**	91.6	**79.2**	**90.4**	59.7	**84.5**	61.1	56.6

The bolded value is the best result for that column.

Table 3. Results of LFNet and existing models for the NuScenes dataset.

Method	**Real Time**	**mIoU/%**	**Barrier**	**Bicycle**	**Bus**	**Car**	**Construction Vehicle**	**Motorcycle**	**Pedestrain**	**Traffic cone**	**Trailer**	**Truck**	**Driveable Surface**	**Other Flat Ground**	**Sidewalk**	**Terrain**	**Manmade**	**Vegetation**
LidarMultiNet [35]	no	**81.4**	80.4	48.4	**94.3**	90	71.5	**87.2**	**85.2**	**80.4**	86.9	74.8	**97.8**	67.3	80.7	76.5	92.1	89.6
Cylinder3D [21]	no	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
SPVNAS [9]	**yes**	74.8	78.1	31.2	89.2	88.0	65.4	69.1	81.2	53.4	72.7	70.4	94.9	61.6	76.8	79.7	**93.8**	**90.7**
JS3C-Net [36]	**yes**	73.6	80.1	**59.4**	88.0	85.1	63.7	84.4	82.0	76.0	84.8	71.9	96.9	67.4	79.8	76.0	92.1	89.2
AF2S3Net [37]	**yes**	62.2	60.3	12.6	82.3	80.0	20.1	62.0	59.0	49.0	42.2	67.4	94.2	68.0	64.1	68.6	82.9	82.4
PolarNet [6]	**yes**	71.0	74.7	28.2	85.3	90.9	35.1	77.5	71.3	58.8	57.4	76.1	96.5	71.7	74.7	74.0	87.3	85.7
SalsaNext [38]	**yes**	58.8	56.6	4.7	77.1	81.0	18.4	47.5	52.8	43.5	38.3	65.7	94.2	60.0	68.9	70.3	81.2	80.5
LFNet (ours)	**yes**	79.8	**86.2**	50.3	92.4	58.7	**79.0**	79.0	78.9	63.8	**87.0**	83.4	96.6	**76.8**	**81.5**	**82.4**	90.9	90.0

The bolded value is the best result for that column.

Our model adds a feature-learning and fusion module to the 3D-Unet model. The LFNet performance is significantly improved. In the SemanticKITTI dataset, the segmentation performance of LFNet was optimal in 10 out of 19 categories compared to the existing models. In the NuScenes dataset, the segmentation performance of LFNet was optimal in 6 out of 16 categories. The experimental results demonstrate the effectiveness of the proposed LFNet for semantic segmentation.

4.6. Efficiency Comparisons

Since the vast majority of computation in the model comes from convolutional operations, this section only counts the calculated quantities of the convolutional layers in the model. Unlike traditional 3D convolution, we use the following formula to calculate the number of operations for sparse tensor-based convolution operations.

$$\mathrm{Flops} = 2 * (\mathrm{C_{in}} * \mathrm{K} - 1) * \mathrm{N} * \mathrm{C_{out}} \tag{18}$$

where the number of input and output channels of the convolution layer are denoted by C_{in} and C_{out}, respectively. For rasterized data, we express the size of the feature map in terms of the number of sparse rasters. Therefore, N is the number of output rasters in the convolution layer. K is the average amount of activation points. K is calculated by setting the convolution kernel parameter of each convolution layer to 1 and averaging the output to obtain the amount of activation points around each point.

In this section, the efficiency of LFNet is compared with existing models, and the specific statistics are shown in Table 4. Compared to the 3D-Unet model, LFNet introduces new convolutional layers due to the addition of the feature-learning and fusion modules, which increase the parameters of the model and require more computations. As a result, LFNet has a slight increase in FLOPs and single-frame prediction times, but the model still meets the requirements of real-time computing.

Table 4. Efficiency comparison of different methods.

Methods	Params/M	Flops/G	Device	Time/ms
RPVNet [33]	24.8	239	Tesla V100	168
SPVNAS [9]	**12.5**	147.6	GTX 1080Ti	259
FusionNet [34]	-	-	GTX 1080	900
SqueezeSegV3 [7]	26.2	1030.4	-	142
LFNet (ours)	17.7	≈**64**	GTX 3090	**94**

The bolded value is the best result for that column.

By analyzing the experimental data, it can be seen that LFNet increases the number of parameters by 5.2 M compared to the SPVNAS model, but it reduces the amount of computation by nearly 83.6 G. The sparsity of the point cloud is the main reason for this. LFNet uses sparse tensor-based implementation, meaning that the model is able to avoid performing a large number of invalid operations in blank areas and therefore effectively reduces computational effort. The results demonstrate that LFNet can reduce the model computation, improve the operation efficiency, and essentially meet the application requirements of UGV for real-time semantic segmentation.

4.7. Ablation Study

In order to verify the effectiveness of the GFLM and the MFFM, this section adds each of these two modules to the original 3D-Unet model. They are compared with the baseline model with respect to accuracy, mIoU, and the time to make predictions for each model in Table 5.

Compared with the 3D-Unet model, the addition of the GFLM increases the acc and mIoU of the model by 1.4% and 3.3%, respectively, and increases the model prediction time by 12.6/ms. The addition of the MFFM improves the accuracy and mIoU by 1.1% and 2.4%, respectively, and increases the model prediction time by 6.9/ms. With the addition of both modules, LFNet improved in accuracy and mIoU by 1.8% and 7.9%, respectively.

The experiments show that, although the GFLM and the MFFM increase the prediction time of the model, they both can effectively improve the prediction accuracy and segmentation effect of the model.

Table 5. Ablation experimental results for the SemanticKITTI dataset.

Method	Acc	mIoU	Time/ms
3D Unet	91.5%	64.9%	**67.8**
3D Unet + GFLM	92.9%	68.2%	80.4
3D Unet + MFFM	92.6%	67.3%	74.7
3D Unet + GFLM + MFFM (LFNet)	**93.3%**	**72.8%**	94.1

The bolded value is the best result for that column.

5. Conclusions

This paper investigated the semantic segmentation method of real-time 3D point clouds for unmanned vehicle systems. We proposed LFNet, a lightweight, fully convolutional network based on an attention mechanism and a sparse tensor, which learns global contextual features by designing GFML and achieves the effective fusion of feature information at different scales by designing MFFM. LFNet reduces unnecessary operations and memory occupation by using a sparse tensor. This improves the adaptability of the model to the limited computational resources of unmanned vehicle systems.

The experimental results showed that LFNet improves the mIoU evaluation metric by 6.4% and 5% over existing models for the SemanticKITTI and nuScenes datasets, respectively. For data acquired by HDL-64E LiDAR on the SemanticKITTI dataset, our LFNet efficiency can reach 10 fps on the GTX 3090 experimental platform. Thus, our model can meet the real-time application of unmanned vehicle systems. With our attention module, our method can significantly improve the accuracy of semantic segmentation.

Author Contributions: Conceptualization, W.Z.; methodology, F.W.; software, Y.Y.; validation, F.W., Z.W. and J.Z.; resources, J.Z.; data curation, Z.W.; writing—original draft preparation, F.W.; writing—review and editing, Y.Y. and W.Z.; supervision, W.Z.; project administration, F.W. and W.Z.; funding acquisition, F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Postdoctoral Science Foundation of China, grant number 2021M690502, and the Dalian Excellent Youth Talent Fund Project, grant number 2022RY23.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Achirei, S.-D.; Heghea, M.-C.; Lupu, R.-G.; Manta, V.-I. Human Activity Recognition for Assisted Living Based on Scene Understanding. *Appl. Sci.* **2022**, *12*, 10743. [CrossRef]
2. He, P.; Ma, Z.; Fei, M.; Liu, W.; Guo, G.; Wang, M. A Multiscale Multi-Feature Deep Learning Model for Airborne Point-Cloud Semantic Segmentation. *Appl. Sci.* **2022**, *12*, 11801. [CrossRef]
3. Kang, X.; Li, J.; Fan, X.; Jian, H.; Xu, C. Object-Level Semantic Map Construction for Dynamic Scenes. *Appl. Sci.* **2021**, *11*, 645. [CrossRef]
4. Qi, C.R.; Su, H.; Mo, K. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
5. Thomas, H.; Qi, C.R.; Deschaud, J.E. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 20–26 October 2019.
6. Zhang, Y.; Zhou, Z.; David, P. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, DC, USA, 14–19 June 2020.

7. Xu, C.; Wu, B.; Wang, Z. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In Proceedings of the 2020 European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
8. Choy, C.; Gwak, J.Y.; Savarese, S. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–21 June 2019.
9. Tang, H.; Liu, Z.; Zhao, S. Searching efficient 3d architectures with sparse point-voxel convolution. In Proceedings of the 2020 European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
10. Graham, B.; Engelcke, M.; Van, D.M.L. 3d semantic segmentation with submanifold sparse convolutional networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–21 June 2018.
11. Rosu, R.A.; Schütt, P.; Quenzel, J. Latticenet: Fast point cloud segmentation using permutohedral lattices. *arXiv* **2019**, arXiv:1912.05905.
12. Wang, X.; Jia, R.H.; Fu, L.Y. Online Spatial Crowdsensing with Expertise-Aware Truth Inference and Task Allocation. *IEEE J. Sel. Areas Commun.* **2021**, *40*, 412–427. [CrossRef]
13. Fan, G.Y.; Jin, H.M.; Fu, L.Y. Joint Scheduling and Incentive Mechanism for Spatio-Temporal Vehicular Crowd Sensing. *IEEE Trans Mob Comput.* **2019**, *20*, 1449–1464. [CrossRef]
14. Fang, Y.; Xu, C.; Cui, Z. Spatial transformer point convolution. *arXiv* **2020**, arXiv:2009.01427.
15. Zhou, J.C.; Pang, L.; Li, C.Y. Underwater image enhancement method by multi-interval histogram equalization. *IEEE J. Oceanic. Eng.* **2023**. [CrossRef]
16. Zhou, J.C.; Zhang, D.H.; Ren, W.Q. Auto Color Correction of Underwater Images Utilizing Depth Information. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
17. Zhou, J.C.; Zhang, D.H.; Zhang, W.S. Underwater image enhancement method via multi-feature prior fusion. *Appl. Intell.* **2022**, *52*, 16435–16457. [CrossRef]
18. Wu, H.Y.; Fu, L.Y.; Long, H. Unraveling the Detectability of Stochastic Block Model with Overlapping Communities. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1443–1455. [CrossRef]
19. Li, R.; Zhang, Y. PointVGG: Graph convolutional network with progressive aggregating features on point clouds. *Neurocomputing.* **2021**, *429*, 187–198. [CrossRef]
20. Ding, Y.; Zhang, Z.L.; Zhao, X.F. Self-Supervised Locality Preserving Low-Pass Graph Convolutional Embedding for Large-Scale Hyperspectral Image Clustering. *IEEE Trans Geosci Remote Sens.* **2022**, *60*, 1–16. [CrossRef]
21. Zhu, X.; Zhou, H.; Wang, T. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TX, USA, 19–25 June 2021.
22. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
23. Zhou, R.; Li, X.; Jiang, W. SCANet: A Spatial and Channel Attention based Network for Partial-to-Partial Point Cloud Registration. *Pattern Recognit. Lett.* **2021**, *151*, 120–126. [CrossRef]
24. Feng, M.; Zhang, L.; Lin, X. Point attention network for semantic segmentation of 3D point clouds. *Pattern Recogn.* **2020**, *107*, 107446. [CrossRef]
25. Chen, X.; Wu, Y.; Xu, W. PointSCNet: Point Cloud Structure and Correlation Learning Based on Space-Filling Curve-Guided Sampling. *Symmetry.* **2022**, *14*, 8. [CrossRef]
26. Chen, X.T.; Li, Y.; Fan, J.H. RGAM: A novel network architecture for 3D point cloud semantic segmentation in indoor scenes. *Inform. Sci.* **2021**, *571*, 87–103. [CrossRef]
27. Sun, Y.; Wang, Y.; Liu, Z. Pointgrow: Autoregressively learned point cloud generation with self-attention. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 2–5 May 2020.
28. Wang, G.; Zhai, Q.; Liu, H. Cross self-attention network for 3D point cloud. *Knowl. Based Syst.* **2022**, *247*, 108769. [CrossRef]
29. Wen, X.; Han, Z.; Youk, G. CF-SIS: Semantic-instance segmentation of 3D point clouds by context fusion with self-attention. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, DC, USA, 12–16 October 2020.
30. Su, H.; Jampani, V.; Sun, D. Splatnet: Sparse lattice networks for point cloud processing. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–21 June 2018.
31. Gu, X.; Wang, Y.; Wu, C. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–21 June 2019.
32. Çiçek, Ö.; Abdulkadir, A.; Lienkamp, S.S. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In Proceedings of the 2016 Medical Image Computing and Computer-Assisted Intervention, Istanbul, Turkey, 17–21 October 2016.
33. Xu, J.; Zhang, R.; Dou, J. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In Proceedings of the 2021 IEEE International Conference on Computer Vision, Montreal, QC, Canada, 11–18 October 2021.
34. Zhang, F.; Fang, J.; Wah, B. Deep FusionNet for Point Cloud Semantic Segmentation. In Proceedings of the 2020 European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
35. Ye, D.Q.Z.; Zhou, Z.X.; Chen, W.J. LidarMultiNet: Towards a Unified Multi-task Network for LiDAR Perception. *arXiv* **2022**, arXiv:2209.09385.
36. Yan, X.; Gao, J.T.; Li, J. Sparse Single Sweep LiDAR Point Cloud Segmentation via Learning Contextual Shape Priors from Scene Completion. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.

37. Cheng, R.; Razani, R.; Taghavi, E. (AF)2-S3Net: Attentive Feature Fusion with Adaptive Feature Selection for Sparse Semantic Segmentation Network. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
38. Cortinhal, T.; Tzelepis, G.; Aksoy, E.E. SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving. In Proceedings of the 2020 International Symposium on Visual Computing, San Diego, CA, USA, 5–7 October 2020.