

Article

FGCN: Image-Fused Point Cloud Semantic Segmentation with Fusion Graph Convolutional Network

Kun Zhang ¹, Rui Chen ¹, Zidong Peng ², Yawei Zhu ¹ and Xiaohong Wang ^{1,*}

¹ College of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China; zhangkun@hebest.edu.cn (K.Z.); cheerui@stu.hebest.edu.cn (R.C.); zhuyawei@stu.hebest.edu.cn (Y.Z.)

² College of International Education, Guangxi University of Science and Technology, Liuzhou 545006, China

* Correspondence: wangxiaohong@hebest.edu.cn; Tel.: +86-13832159723

Abstract: In interpreting a scene for numerous applications, including autonomous driving and robotic navigation, semantic segmentation is crucial. Compared to single-modal data, multi-modal data allow us to extract a richer set of features, which is the benefit of improving segmentation accuracy and effect. We propose a point cloud semantic segmentation method, and a fusion graph convolutional network (FGCN) which extracts the semantic information of each point involved in the two-modal data of images and point clouds. The two-channel k-nearest neighbors (KNN) module of the FGCN was created to address the issue of the feature extraction's poor efficiency by utilizing picture data. Notably, the FGCN utilizes the spatial attention mechanism to better distinguish more important features and fuses multi-scale features to enhance the generalization capability of the network and increase the accuracy of the semantic segmentation. In the experiment, a self-made semantic segmentation KITTI (SSKIT) dataset was made for the fusion effect. The mean intersection over union (MIoU) of the SSKIT can reach 88.06%. As well as the public datasets, the S3DIS showed that our method can enhance data features and outperform other methods: the MIoU of the S3DIS can reach up to 78.55%. The segmentation accuracy is significantly improved compared with the existing methods, which verifies the effectiveness of the improved algorithms.

Keywords: point clouds; graph attention convolution; multi-modal data; multi-scale features; FGCN



Citation: Zhang, K.; Chen, R.; Peng, Z.; Zhu, Y.; Wang, X. FGCN: Image-Fused Point Cloud Semantic Segmentation with Fusion Graph Convolutional Network. *Sensors* **2023**, *23*, 8338. <https://doi.org/10.3390/s23198338>

Academic Editor: Denis Laurendeau

Received: 28 August 2023

Revised: 24 September 2023

Accepted: 29 September 2023

Published: 9 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

At present, semantic segmentation of point clouds has emerged as a highly significant research topic across various domains, including machine vision, artificial intelligence, photogrammetry, remote sensing, etc. [1–4]. Three-dimensional semantic segmentation is one of the crucial areas within the field.

Nowadays, the objective world is filled with various types of data, such as texts, images, sounds, point clouds, etc. However, in the field of computer vision, we give image and point cloud data special consideration. Point clouds provide rich geometric structures, while images contain texture information, both of which are beneficial for acquiring semantic information. Some early 3D semantic segmentation approaches only utilized single-modal data, which are point clouds. This approach was unable to differentiate between objects of the same category based on their colors or textures. As we all know, different modalities of data can provide more diverse and complementary information. Therefore, we have integrated point clouds with images, enabling the input data to encompass both depth and color information simultaneously [5–7].

However, there are various methods to fuse different modalities of data. Common fusion strategies include fusion strategies based on discernible units (FSBDU), fusion strategies based on complementary features (FSBCF), fusion strategies based on target attributes (FSBTA), and fusion strategies based on multi-source decision (FSBMD) [8]. In our approach, we adopted the FSBCF method for data fusion. As shown in Figure 1,

the original data were obtained from different modalities as inputs, features from each modality were extracted using separate feature extraction networks, the extracted features were processed, and the fusion was then performed.

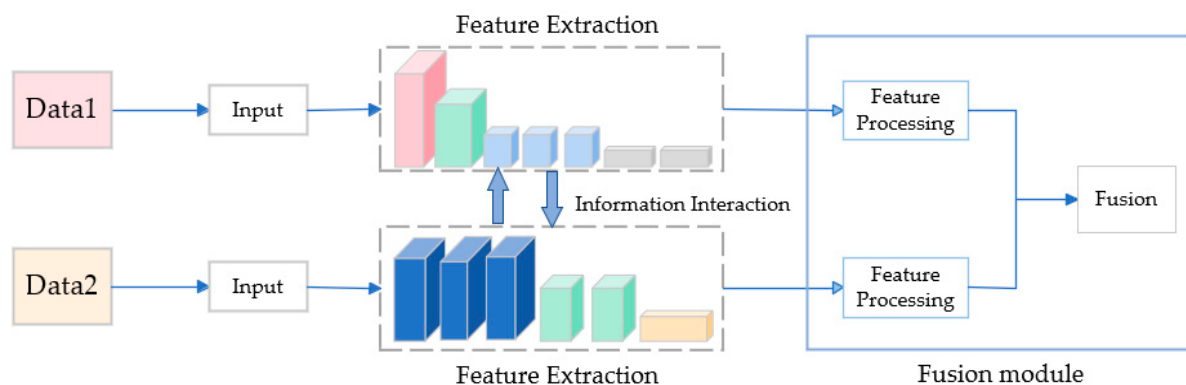


Figure 1. Fusion strategy based on complementary features process.

A semantic segmentation network, called the FGCN, was suggested by us. The FGCN is intended to analyze the fused data from the point clouds and picture modalities and was inspired by FSBCF. The following can be said about our contribution:

(1) **Multi-modal topological map.** Data from the two modalities are combined using an image-guided structural representation of the topological map. The alignment algorithm of the image and point clouds is used to obtain multi-modal data. A two-channel KNN algorithm is used to obtain local information and construct a local topological structure map; (2) **Spatial attention mechanism.** For the extraction of more significant features, a spatial attention mechanism is employed. Average pooling and maximum pooling are performed in channel units and concatenate the pooled content. The spatial attention mechanism can assist the model in acquiring object features more effectively. It achieves this by introducing weights to the input data, and allowing the model to prioritize specific areas or objects of interest. (3) **Multi-modal SSKIT dataset.** An outdoor scene dataset called SSKIT was produced by us, and experimental validation was performed. The log_softmax classifier was added to the FGCN network and the semantic segmentation task was implemented using the SSKIT dataset and the S3DIS dataset, which validated the merits of the FGCN segmentation method proposed in this paper. The MIoU achieved on the SSKIT dataset was 88.06%; while on the S3DIS dataset, the MIoU reached 78.55%. Rigorous testing on the two reference datasets showed that our technique performs better than others. When this paper is published, we would like to open the dataset.

2. Related Work

In this section, we review point cloud semantic segmentation works. Three categories of segmentation methods are concerned with the perspective of different modal data. In the early days, semantic segmentation was implemented on the single-modal data of point clouds. Subsequently, for the use of semantic information in images, the two-modal data of images and point clouds were applied in segmentation mainly using fusion strategy. In this paper, the fusion strategies fall into two major categories: point clouds projected onto the image methods, and image-guided point clouds featuring learning methods.

2.1. Point Clouds-Only Methods

Many semantic segmentation methods use point cloud data exclusively. These methods can be roughly categorized into three groups. The first category is based on projection methods [9], which involve transforming 3D point clouds into 2D grids using techniques like spherical [10] or bird's-eye view [11] projections. This approach enables the utilization of powerful existing 2D segmentation models [12]. However, due to this transformation, the point clouds' geometric properties in three dimensions can be degraded. The second

category involves the direct processing of original 3D point clouds [13,14]. Methods in this group often summarize the input point clouds using a set of sparse key points. PointNet [15] and PointNet++ [16] are representative examples of this approach. The third category comprises voxelization techniques [17,18], which involve grouping points within the point clouds into clusters that are treated as individual units. The original point clouds are processed and transformed into a voxel grid.

2.2. Point Clouds Are Projected onto Image Methods

Point clouds have rich geometry features, and images have rich texture features. The spatial properties of 3D point clouds are merged into 2D images to enhance feature representation and provide the network the ability to learn object distance information. D. K. Kim et al. [19] proposed a framework for projecting the features of 3D point clouds in 2D images to achieve robust pixel semantic segmentation of off-road terrain. The multi-scale data of the 3D point clouds and the image were trained separately by F. Yang et al. [20]. Multi-scale sparse point clouds are projected onto the 2D images, allowing the network to perform better feature learning. The final road detection model can eliminate shadows, curbs, and other interferences in complex scenes. To achieve high-accuracy road detection, L. Caltagirone et al. [21] projected a 3D point cloud onto the 2D plane of the camera and learned dense 2D image features with spatially encoded information by upsampling. Ye Chao et al. [22] improved the accuracy of semantic segmentation by transforming scattered point cloud features into densely distributed 2D spherical features through spherical mapping and spatial modules. Occlusion is the source of low-target detection accuracy; H. Wang et al. [23] employed grid projection and the fusion method to find obstacles in the 3D point cloud data. Compared with YOLO, the average accuracy of the road detection was improved by 17%.

2.3. Image-Guided Point Clouds Feature Learning Method

The precision of image feature learning networks fall short of requirements, which can be overcome using the image-guided point cloud feature learning approach. A 3D point cloud guided image fusion module was proposed by T. Huang et al. [24] to determine the semantic features of images. This module adaptively enhanced the characteristics of point cloud data and achieved good results on KITTI and SUN-RGBD datasets. A. S. Mohamed et al. [25] studied pre-fusion, mid-fusion, and post-fusion of images and point clouds. It was verified that mid-term fusion enabled better feature learning and is most effective on semantic segmentation tasks. V. Poliyapram et al. [26] proposed a multi-modal feature learning method that can act directly on the end-to-end structure of the point clouds. Image features are integrated into the 3D point cloud data to improve the modeling effect. L. Deng et al. [27] used a superpoint approach for the feature learning of images and point clouds, which showed some improvements in the semantic segmentation task. For the feature-level fusion of 2D feature maps and 3D point cloud aerial views, J. H. Yoo et al. [28] transformed 2D map features into smooth spatial features and achieved STOA performance in the KITTI benchmark. The SqueezeSeg method proposed by K. El Madawi et al. [29] added image RGB information to point clouds and showed good performance in the field of semantic segmentation. The network verified that adding RGB information leads to better feature learning, and there was almost no change in the time complexity of the network.

3. Methodology

We employ an image-guided point cloud feature learning method to fuse the original data. To construct a multi-modal topological map that can contain both point clouds and image information [30], 2D image data and 3D point cloud data are jointly feature represented F_{mp} .

Two-dimensional images are acquired by the camera and adjusted by the camera's internal parameters. The image is rich in texture information, so the image m_i can consist of pixel points (u, v) with three channels of r, g, and b for each pixel point. The 3D point cloud

data acquired by the Velodyne HDL-64E rotating 3D laser scanner acquisition consists of the spatial coordinates x, y, z , and the intensity information i . The image and point cloud representation are shown in the Equation:

$$\begin{cases} m_i = (R(u, v), G(u, v), B(u, v)) \\ p_i = \{x_i, y_i, z_i, i_i\} \end{cases}, \quad (1)$$

where m_i is the feature representation of the image and p_i is the feature representation of the point clouds. (u, v) represents the pixel coordinates. R, G , and B are the color channel values, x_i, y_i and z_i are the point transport coordinates, and i_i is the point cloud reflection intensity information.

A graph convolutional network named FGCN is proposed. As shown in Figure 2, the FGCN consists of five core modules, which are data alignment, data enhancement, building topological maps, spatial attention mechanism, and multi-layer perceptron.

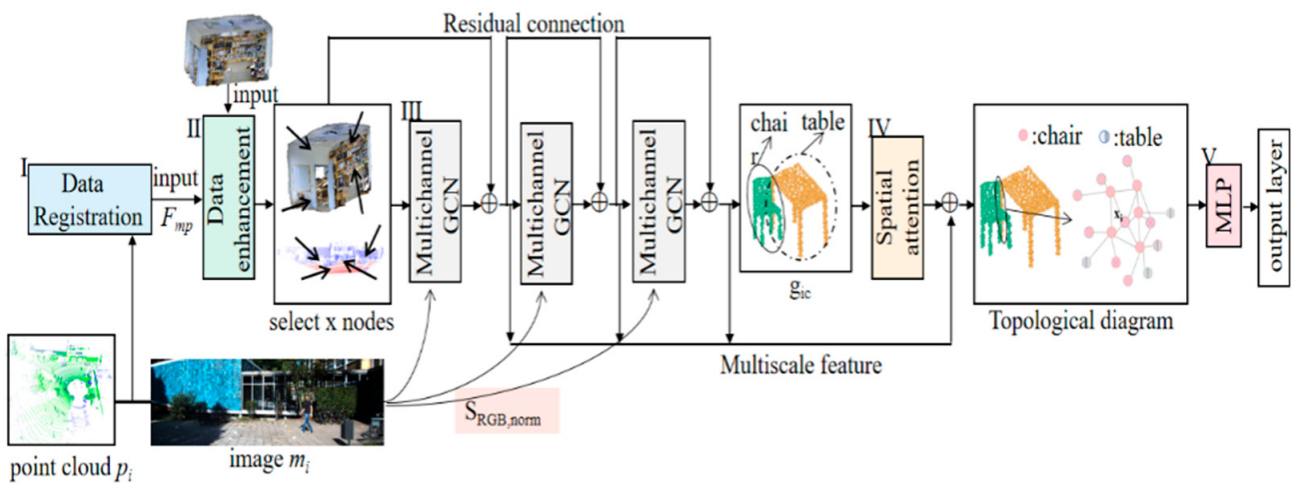


Figure 2. FGCN network structure. I–V are the operating procedures of the FGCN network.

The 3D point clouds are registered with the 2D image data using camera and radar parameters. A unique pixel point correspondence is found for each point cloud and the color information is added to the point cloud to obtain p_i , $p_i = \{x_i, y_i, z_i, r_i, g_i, b_i\}$. Data enhancement is performed by Module II. The multi-channel graph convolutional network of Module III constructs an image-guided topological graph structure, and the graph is constructed by combining chromaticity values $S_{RGB, norm}$. The feature learning is enhanced by adding residual connections to the multilayer graph convolution to obtain the local feature map, g_{ic} . The data features are learned after the spatial attention of Module IV and then aggregated into a multi-mode topological diagram. The prediction labels for each point are obtained after the Module V MLP and output layers.

3.1. Image-Guided Multi-Channel Graph Convolution

In the field of computer vision, spatial graph convolution networks for feature extraction can optimize the problems of inadequate local information extraction and limited regional information merging. Therefore, this paper uses graph convolutional networks. It is necessary to convert 3D point clouds into vectors before entering them into the network. A sequence of points $\{x_1, x_2, x_3, \dots, x_n\}$ is written as a multidimensional tensor using the mapping function γ , as shown in the Equation:

$$f(x_1, x_2, x_3, \dots, x_n) = \gamma_{i=1}^n(x_i), \quad (2)$$

where γ is the mapping function.

Since the KNN algorithm focuses more on the distance between nodes and the graph convolution focuses more on the edge information, we use images to guide the graph creation. The image's color information is extracted as a compositional attribute, and the color information is added to the composition after the KNN aggregates the neighborhood features. The RGB feature set can be represented as S_{RGB} , $S_{RGB} = \{R, G, B\}$. In the multi-channel KNN module, the RGB features are processed as a chromaticity-valued feature set $S_{RGB,norm}$ to represent the color features, as shown in the Equation:

$$S_{RGB,norm} = \left\{ \frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B} \right\} = \{r', g', b'\}, \quad (3)$$

where $S_{RGB,norm}$ is the chromaticity value of the color image, and r', g' and b' are the values after the color channel operation.

Relying on color invariance in the form of normalized colors, this color is insensitive to the illumination intensity. The new point cloud representation is shown in the Equation:

$$p_i = \{p_i | x_i, y_i, z_i, r', g', b'\}. \quad (4)$$

The process of building the graph structure is shown in Figure 3, where the image-guided topological graph structure is built around the input sequence of central nodes. The two-channel KNN structure extracts the features of the central node x_i and uses different K values to ensure that the local features are fully learned.

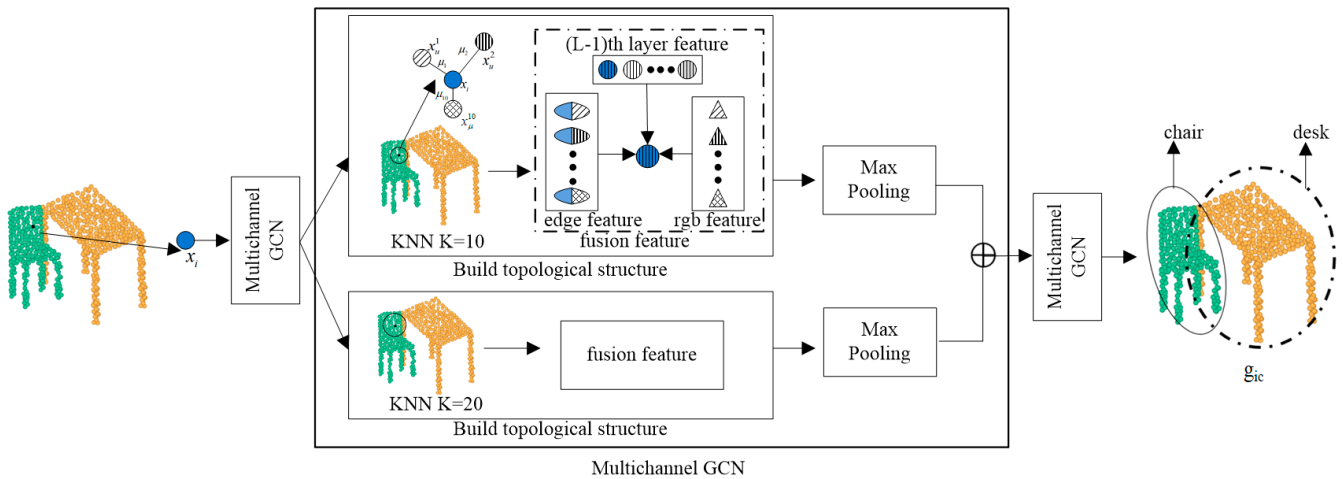


Figure 3. Image-guided graph structure building.

As shown in Figure 3, The topological map structure aggregates the central node information, edge information, color information, and node information aggregated in the previous layer. The spatial and color features of each point cloud are fully utilized to speed up learning. The aggregation process of neighboring node features x'_{ic} of node x_i is shown in the Equation:

$$\begin{cases} x'_i = \sum_{j:(i,j) \in \mathcal{E}} h_{\Theta}(x_i, x_j) \\ x'_{ic} = \Lambda_{\Theta}(x'_i, x_i^c) \end{cases}, \quad (5)$$

where (x_i, x_j) is the edge features between nodes x_i and x_j . $h_{\Theta} \Lambda_{\Theta}$ are nonlinear functions with learnable parameters of the layer. x'_i and x_i^c are edge features and color features of x_i . Respectively, x'_{ic} is a node of aggregated color features, where the label c represents the color. The node attribute feature $\vec{x}_i = \{x, y, z, r', g', b', i\}$ forms the local feature graph $r', g',$ and b' attribute information can enhance the graph structure building.

After constructing the graph structure, the edge features of the center point and the selected point are fused to achieve the aggregation of local and global information

within the neighborhood graph. The l th layer features of x'_{ic} after fusion are shown in the Equation:

$$h_{ij}^l = \Lambda_{\Theta}(x'_{ic}, x'_{jc} - x'_{ic}). \quad (6)$$

After aggregating the features, max-pooling is used to filter the features. The max-pooling operation replaces an area's output with the highest value found in the nearby rectangular region at a position. The max-pooling operation is shown in the Equation:

$$x''_i = \max_{j:(i,j) \in E} h_{ij}^l. \quad (7)$$

Inspired by the jump connection network, this paper connects the local features with residuals after computing them. To improve graph structure perception using nearby nodes, the input and output of the current layer are used as the input of the following layer. The residual connectivity is shown in the Equation:

$$h_{i+1} = h_i + \mathfrak{S}(h_i, W_i), \quad (8)$$

where h_i is the feature at layer i , h_{i+1} is the feature at layer $i + 1$, W_i is the convolution operation at layer i , and $\mathfrak{S}$ is the residual function.

The FGCN uses a three-layer graph convolution structure, and the central node x_i aggregation is the last layer of features. Taking the l th layer graph convolution as an example, the nodes are first embedded, and the neighborhood information of the nodes is aggregated using the two-channel KNN algorithm. All of the neighborhood node information, edge information between nodes, and aggregated node information from the $l-1$ th layer are contained in the l th layer features. To create a local neighborhood feature graph with color information, the layer $l-1$ feature graph is first enhanced with color information. The features from the previous layers, color features, and edge features are all simultaneously aggregated when the layer 1 graph convolution process is carried out. The local feature map g_{ic} is created by combining the features of all the nodes in layer 3, as shown in the Equation:

$$\begin{cases} h_v^{l-1} = f\left(w^{l-1}x_i + \sum_{u \in N(v)} \mu_v x_u\right) \\ h_v^l = f\left(w^l x_j + \sum_{i=1}^l \sum_{u \in N(v)} \eta^l h_u^{(l-1)}\right) \end{cases}, \quad (9)$$

it is the aggregation mode of layer $l-1$ features and layer l features of graph convolution. Where h_v^{l-1} is the feature of layer $l-1$, w, η, μ are the learnable parameters, x_u is one of the neighborhood nodes. h_v^l is the feature of layer l , and $N(v)$ is the set of neighborhood nodes of node v .

The sense field is expanded using the structure of multi-layer graph convolution. All node features are aggregated to the centroid, which eventually forms a local neighborhood graph g_{ic} .

3.2. Spatial Attention Module

The multi-channel graph convolutional network learns local features, and use of the spatial attention mechanism may concentrate on crucial areas in the local features and handle the scene's critical elements.

As shown in Figure 4, first, each channel of the feature x_i is subjected to mean operation and max operation respectively. The x_{mean} obtained by the mean operation contains the basic shape and overall distribution of the features in the scene, and the x_{max} obtained by the max operation contains the salient features of each object. Then, x_{max} and x_{mean} are stitched to get x_i' , and the attention weights are obtained by convolution and pooling layers.

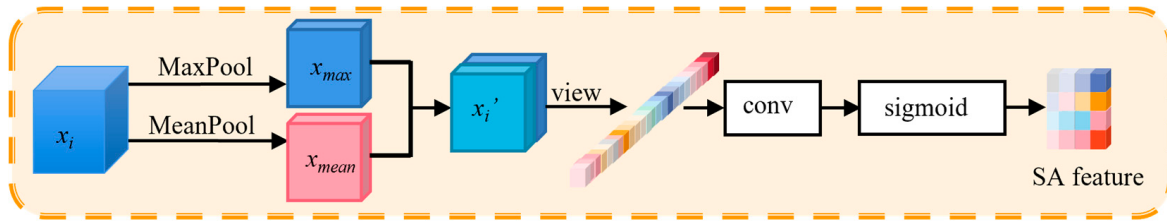


Figure 4. Spatial attention module.

Spatial attention is shown in the Equation:

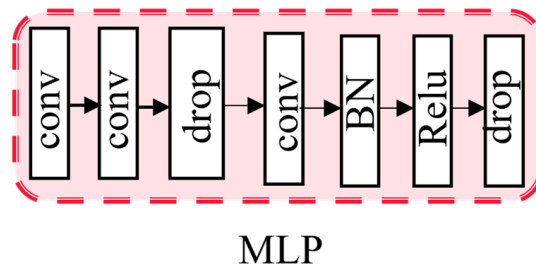
$$x_i' = \sigma \left(f^{7 \times 7}([MaxPool(x_i); MeanPool(x_i)]) \right) = \sigma \left(f^{7 \times 7}([x_{max}; x_{mean}]) \right), \quad (10)$$

where *MaxPool* is the maximum pooling function and x_{max} is the maximum pooling result of the channel. *MeanPool* is the average pooling and x_{mean} is the average pooling result of the channel. $f^{7 \times 7}$ is the convolution function. σ is the nonlinear activation function.

Spatial attention allows for the analysis of the details between objects at a local scale and uses residual structures to accelerate feature transfer. This module extracts attention information and makes the weights non-negative using sigmoid. The information about the same object in the immediate area can be filtered using the spatial attention module, while the information about interference can be ignored. The spatial attention mechanism can concentrate on essential information within the data, aiding the model to acquire pertinent features more efficiently and enhancing the model's performance. The spatial attention mechanism plays a crucial role in enabling the model to better manage noise and interference. In our ablation study, it becomes evident that the removal of the spatial attention module leads to a significant drop in both the mean accuracy (MAcc) and overall accuracy (OAcc) of the model, indicating that point cloud noise has a detrimental impact on segmentation performance. Additionally, we noted that when the point cloud becomes excessively sparse, the spatial attention mechanism's ability to handle noise diminishes.

3.3. Output Layer

After obtaining spatial attention weights, deeper and more semantic information is multi-scale fused with shallow features. As shown in Figure 5, MLP is used to help the network converge. A nonlinear factor is introduced through the activation function $f(x_i)$. $f(x_i) = \max(0, x_i)$. The nonlinear activation function Relu is used in MLP, which allows the neural network to learn and represent complex linear relationships. Batch normalization [31] is also used, which helps to alleviate the vanishing gradient problem and accelerates the convergence of the model by normalizing the input of each layer to have zero mean and unit variance.



MLP

Figure 5. MLP module.

After the convolution operation, the labels and categories are processed into a dictionary form for the statistics and output of the results; the Equation is as follows:

$$f_{dic}^{i=c}(num_i, l_i) = \{num_1 : l_1, num_2 : l_2, \dots, num_c : l_c\}, \quad (11)$$

where num_i is the category, l_i is the label, and f_{dic} is the dictionary.

As shown in Figure 6, the Softmax function makes one prediction for all categories of each point cloud x_k . These predictions are given different weight values, and the prediction with the largest weight is taken as the final result. The use of $\log_softmax$ can prevent data overflow and thus improve data stability. Speeding up the derivation in the training process makes the backpropagation operation faster, as is shown in the Equation:

$$\begin{cases} \sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^n e^{x_k}} \\ \log_Softmax = \log_e |\sigma(x)_j| \end{cases} \quad (12)$$

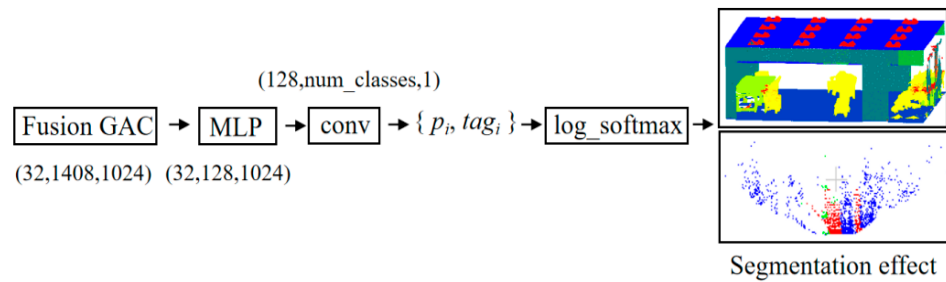


Figure 6. Output layer.

4. Results

4.1. Experimental Data and Environment

4.1.1. Datasets

(1) The S3DIS contains 3D scanned point clouds of 6 interior regions, with a total of 271 point cloud rooms; each point belongs to one of 13 semantic categories, such as wood panels, bookshelves, chairs, ceilings, and other miscellaneous objects. There are 19,898 point cloud files in the training set and 12 rooms in each region in the test set, for a total of 48 rooms. The data are shown in Figure 7a. At the same time, we also used the public outdoor datasets nuScenes for the evaluation;

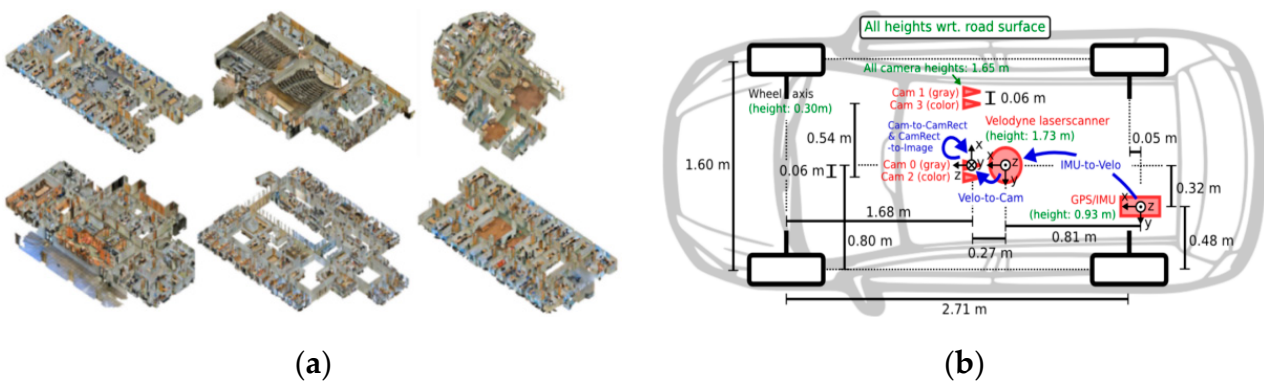


Figure 7. Public datasets: (a) S3DIS datasets; and (b) KITTI datasets.

(2) At present, the largest international dataset for autonomous driving situations is KITTI. With up to 15 vehicles and 30 pedestrians per image, as well as varying degrees of occlusion and truncation, KITTI comprises real image data gathered from urban, rural, and highway situations. It can be used for target detection and data multimodality studies. The data acquisition is shown in Figure 7b;

(3) SSKIT is a bimodal semantic segmentation dataset obtained by registration and annotation using KITTI data in this paper. SSKIT contains point clouds and image data of 984 outdoor urban scenes with 9 semantic categories, such as pedestrian, car, bicycle,

road, background, etc. There are 778 scene samples in the training set and 206 scene data in the validation set. To obtain better semantic segmentation data and to better validate the effectiveness of this representation, KITTI data were registered and labeled. The data were roughly divided into 12 scenes according to scenes, such as city walking street, forest road, other scenes, pedestrian-only compounds, and city highway. We selected appropriate training and testing sets in different scenarios to prevent data imbalances from affecting the experimental results.

The SSKIT registration and labeling process is as follows:

Using the image, the point clouds, and the camera's parameter matrix, the point cloud data are projected into the 2D image, and the reflectance value is changed to one. The data are expressed as a vector and the 3D point cloud data x_i are projected into the 2D image to obtain y'_i . The projection uses the left color camera projection matrix P_{rect}^2 and the left laser to the camera coordinate system transformation matrix T_2 . Figure 8 shows the effect of the projection on the 2D image. The transformation matrix is shown in the Equation:

$$y'_i = P_{rect}^2 R_{rect}^0 T_2 x_i. \quad (13)$$

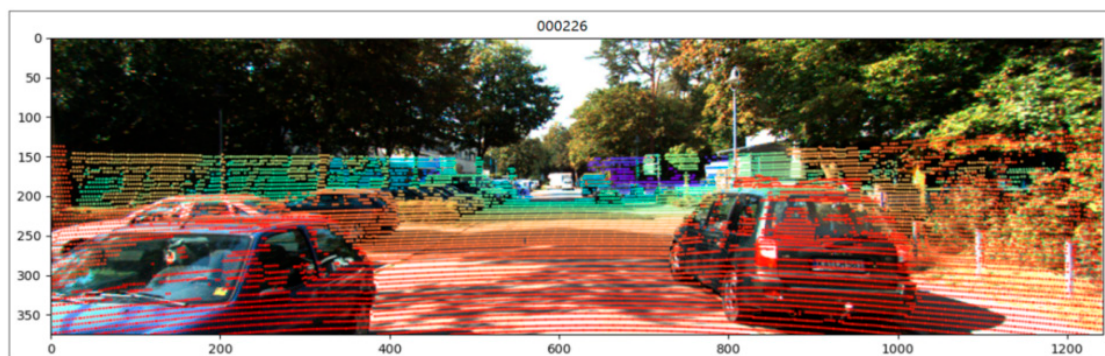


Figure 8. 3D point clouds projection to 2D image.

The point clouds without point-by-point labeling are cropped to obtain 3D data of the same general size as the image data, and all information on the road surface is labeled. The visualization of the points in the data processing is shown in Figure 9.

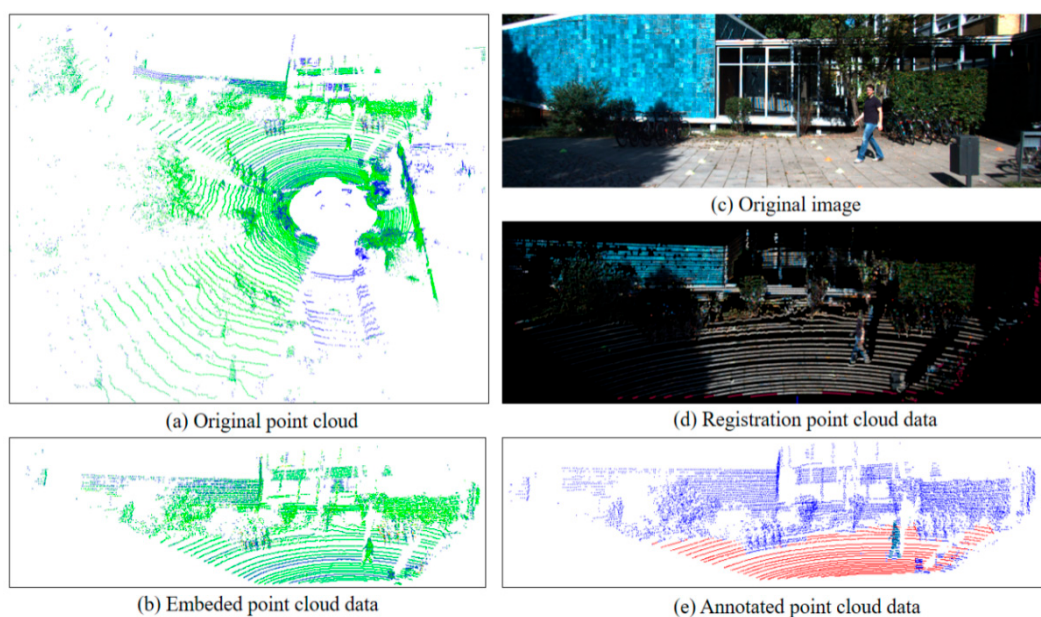


Figure 9. Presentation of data for each phase: (a) original point clouds; (b) embedded point cloud data; (c) original image; (d) registration point cloud data; and (e) annotated point cloud data.

Data Enhancement. In general, the more sufficient the number of samples in the training data, the greater the model's capacity for generalization and the better the trained model effect. In order to better identify the scene data, a data enhancement code module was added to the dataset in this paper. There are two types of data augmentation, one is for the S3DIS dataset, which rotates the data randomly around the z-axis and keeps the z-axis unchanged. Because objects such as tables and chairs can reduce the recognition effect after being flipped up and down, using the data enhancement method that keeps the z-axis constant can better train the data. The second data augmentation type is for the small point cloud scene dataset produced in this paper, and the data are shifted randomly in the x, y, and z axes to obtain more training data. As only one frame of the data is used, rotation does not apply to this dataset. Keeping the randomly panned data to data augmentation, the training can better learn the features. A schematic of the data enhancement module is shown in Figure 10.

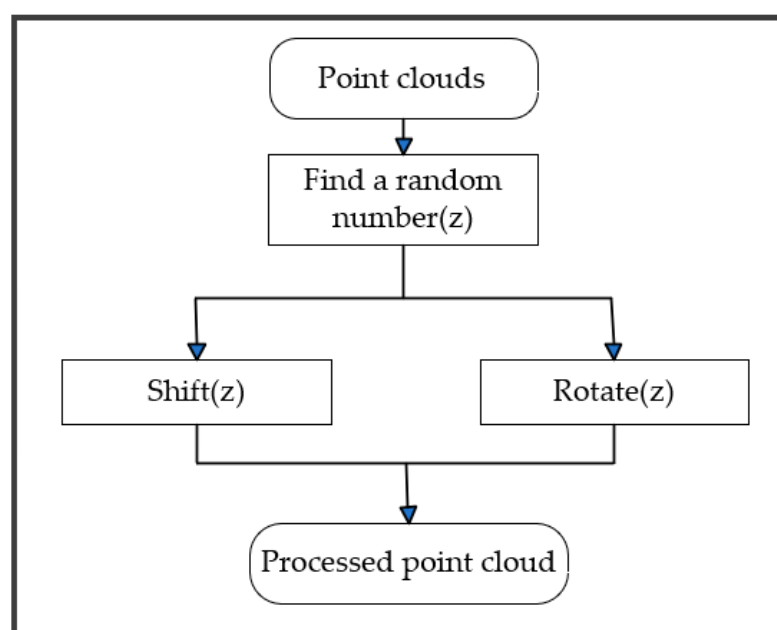


Figure 10. Data enhancement module.

4.1.2. Experimental Environment

Hardware environment: We use an Intel CORE i7 CPU(Integrated Electronics Corporation) with 12G, NVIDIA GeForce GTX 2080(NVIDIA Corporation) for computing, CUDA11.1, and CuDnn11.1 for the GPU acceleration environment.

Software environment: We use Windows 10 OS with Python 3.7 + Pytorch 1.7.

4.2. Registration Experiments

The single-frame data acquired by laser radar are sparse, and more dense data are obtained by registering two consecutive frames of the point cloud. For the problem of unclear edges of many objects, the addition of texture information helps to distinguish the objects. In this experiment, lidar data and color image data were collected separately, registered, and fused. The two frames of point clouds are registered, and the superimposed registered data are then fused with the color images to form a dense color 3D point cloud.

As shown in Figure 11, the results of the fusion of the two acquisitions were compared with the results of the fusion of the single-acquisition data, and it was found that the registered data were clearer.



Figure 11. Comparison of data-fusion results: (a) the image shows the fusion effect before point cloud alignment; and (b) the image shows the fusion effect after point cloud alignment.

As shown in Table 1, before the registration of point cloud data, the number of colored point clouds in the scene is 3054. After the registration and fusion of 2D and 3D data, the number of colored point clouds is 5427. The aligned data have more complete spatial information.

Table 1. Number of color point clouds in the scene.

	Pre-Registration	Post-Registration
Number	3054	5427

4.3. Indoor and Outdoor Scene Segmentation Experiment

In this paper, experimental validation was performed on the SSKIT dataset and the S3DIS dataset, and the point cloud CNN deep learning network and graph convolutional network were selected for experimental comparison, respectively.

4.3.1. Experimental Results

The experimental results using the SSKIT dataset on PointNet++ [16], DGCNN [32], and the model in this paper are shown in Table 2. The experimental results show that the model in this paper is more accurate and has a high MIoU bias.

Table 2. Comparison of semantic segmentation OAcc (%) and MIoU (%) on the SSKIT dataset.

Method	MIoU	OAcc
PointNet++ [16]	83.40	92.78
PointNet++ [16] +RGB	85.49	93.20
DGCNN [32]	86.40	93.54
DGCNN [32] +RGB	87.22	93.55
Ours+NoRGB	86.70	94.11
Ours	88.06	93.72

First, it should be noted that, after the fusion of the point cloud and two-dimensional image, the data have six dimensions: x , y , z , R , G , B . The term NoRGB means that the R , G , and B columns of the fused data are hidden. The experiments were conducted using the S3DIS dataset and compared with the 3D point cloud semantic segmentation network models of PointNet++ [16], KVGCN [33], and Point Transformer [34]. The experimental results are shown in Table 3. The proposed model has more prominent improvement, and the OAcc of the FGVCN is 4.18% higher than that of the KVGCN [33], which is a graph convolution network with better experimental results. The OAcc of the FGVCN is 9.38% higher than that of PointNet++ [16], which is a stable CNN network, and 1.38% higher than that of Point Transformer [34], which is a relatively new network. The MIoU results in this paper were 17.65% higher than KVGCN [33], 27.75% higher than PointNet++ [16], and 5.05% higher than Point Transformer [34], which indicates that the proposed model shows a stable improvement in all the evaluation criteria.

Table 3. Comparison of MAcc (%), MIoU (%), and OAcc (%) on the S3DIS dataset.

Method	MAcc	MIoU	OAcc
PointNet [15]	66.20	47.60	78.50
PointNet++ [16]	66.85	50.80	82.20
DGCNN [32]	-	56.10	84.10
KVGCN [33]	72.30	60.90	87.40
Point Transformer [34]	81.90	73.50	90.20
Segcloud [35]	57.35	48.92	80.80
SPG [36]	64.40	54.10	82.90
ShellNet [37]	-	66.80	-
SPH3D-GCN [38]	67.06	57.20	86.94
PCT [39]	67.65	61.33	-
Ours	87.50	78.55	91.58

As can be seen in Table 4, the accuracy of the FGCN is the highest in some test categories, with more prominent improvements, especially in the challenging categories of the clutter, window, door, column, beam, and column segmentation. Among the categories where other models have lower segmentation results, the window reaches 85.3%, the board reaches 76.1, the column reaches 79.7%, and the door reaches 84.2%. The beam has an extremely low MIoU in other network models, and the model in this paper reaches 73.3%. The accuracy rate of the bookcase category is lower than that of PointTransformer [34], but it is also higher than that of the other networks. We hypothesize that the reason behind the subpar segmentation results for these home furnishings is the uneven distribution of tables, chairs, and sofas within the S3DIS dataset. During the model training process, there are a relatively limited amount of data available for this specific furniture category, leading to inadequate feature learning by the model.

Table 4. IoU (%) for selected categories in the S3DIS dataset.

Method	PointNet [15]	PointNet++ [16]	KVGCN [33]	Point Transformer [34]	Segcloud [35]	SPG [36]	SPH3D-GCN [38]	PCT [39]	Ours
ceiling	88.6	93.1	94.5	94.0	90.1	89.9	92.3	92.54	96.3
floor	97.3	97.6	94.1	98.5	96.1	95.1	98.2	98.42	98.3
wall	69.8	77.2	79.5	86.3	69.9	72.0	71.9	80.62	87.0
beam	0.05	00.0	53.4	0.0	0.0	62.8	0.03	0.00	73.3
column	3.92	16.2	36.3	38.0	18.4	47.1	17.6	19.37	79.7
window	46.3	66.1	56.8	63.4	38.4	55.3	46.8	61.64	85.3
door	10.8	58.1	63.2	74.3	23.1	60.0	43.8	48.00	82.4
table	52.6	62.3	64.3	89.1	70.4	69.2	71.0	85.20	73.5
chair	58.9	65.7	67.5	82.4	78.6	73.5	79.7	76.58	78.0
sofa	5.9	35.1	54.3	74.3	40.9	45.9	50.3	67.71	55.9
bookcase	40.3	45.2	23.6	80.2	58.4	3.2	32.0	46.22	68.8
board	26.4	50.4	43.1	76.0	13.0	8.7	25.8	67.93	76.1
clutter	33.2	50.4	53.2	59.3	41.1	52.9	52.7	52.29	65.5

Furthermore, we conducted experiments using the nuScenes outdoor dataset and compared these with the 3D point cloud semantic segmentation network models of Cylinder3D [14], Cylinder3D++ [14], and SPVCNN++ [40]. The experimental results are shown in Table 5. Our findings indicate that the FGCN outperforms the most advanced models, particularly in transportation categories such as cars, public bicycles, and trucks. Specifically, FGCN achieved outstanding results in these categories, with buses achieving an accuracy of 93.5%, cars an accuracy of 92.6%, motorcycles an accuracy of 85.2%, and trucks an accuracy of 78.2%. While certain metrics may show slightly lower values compared to other networks, the overall performance of the FGCN remains at a high level.

Table 5. IoU (%) for selected categories in the nuScenes dataset.

Method	Cylinder3D [14]	Cylinder3D++ [14]	SPVNAS [40]	SPVCNN++ [40]	AMVNet [41]	PolarNet [42]	JS3C-Net [43]	2D3DNet [44]	Ours
barrier	82.8	82.8	80	86.4	80.6	72.2	80.1	83	83.1
bicycle	29.8	33.9	30	43.1	32	16.8	26.2	59.4	35.9
bus	84.3	84.3	91.9	91.9	81.7	77	87.8	88	93.5
car	89.4	89.4	90.8	92.2	88.9	86.5	84.5	85.1	92.6
construction_vehicle	63	69.6	64.7	75.9	67.1	51.1	55.2	63.7	77.8
motorcycle	79.3	79.4	79	75.7	84.3	69.7	72.6	84.4	85.2
pedestrian	77.2	77.3	75.6	83.4	76.1	64.8	71.3	82	76.5
traffic_cone	73.4	73.4	70.9	77.3	73.5	54.1	66.3	76	78.6
trailer	84.6	84.6	81	86.8	84.9	69.7	76.8	84.8	88
truck	69.1	69.4	74.6	77.4	67.3	63.5	71.2	71.9	78.2
driveable_surface	97.7	97.7	97.4	97.7	97.5	96.6	96.8	96.9	97.1
other_flat	70.2	70.2	69.2	71.2	67.4	67.1	64.5	67.4	73.2
sidewalk	80.3	80.3	80	81.1	79.4	77.7	76.9	79.8	80.6
terrain	75.5	75.5	76.1	77.2	75.5	72.1	74.1	76	76.8
manmade	90.4	90.4	89.3	91.7	91.5	87.1	87.5	92.1	93.5
vegetation	87.6	87.6	87.1	89	88.7	84.5	86.1	89.2	88.5

4.3.2. Visualization Display

Two scene segmentation results were selected from the SSKIT dataset for visualization and analysis. Figure 12 illustrates how the model used in this study has a superior segmentation effect on the specifics of various road and pavement objects.

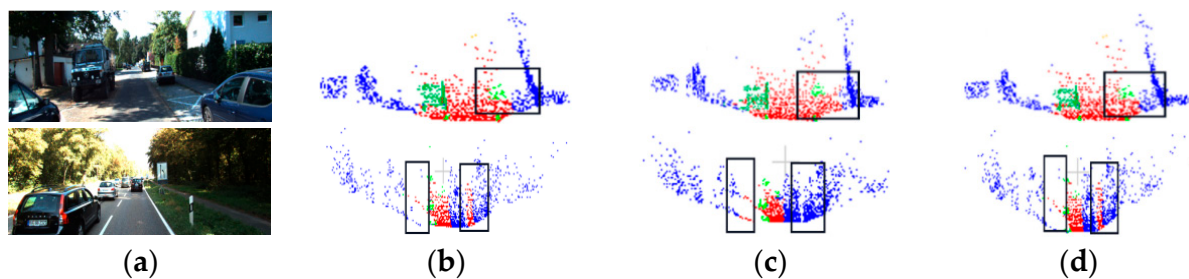


Figure 12. Visualization results of SSKIT with color information: (a) original image; (b) visualization of DGCNN; (c) visualization of PointNet++; and (d) visualization of FGCM.

Figure 13 displays the outcomes of the two rooms of the S3DIS dataset that the model selected for visualization. We can observe from the visualization results that the proposed model has an excellent effect on segmenting windows, walls, and doors, etc.

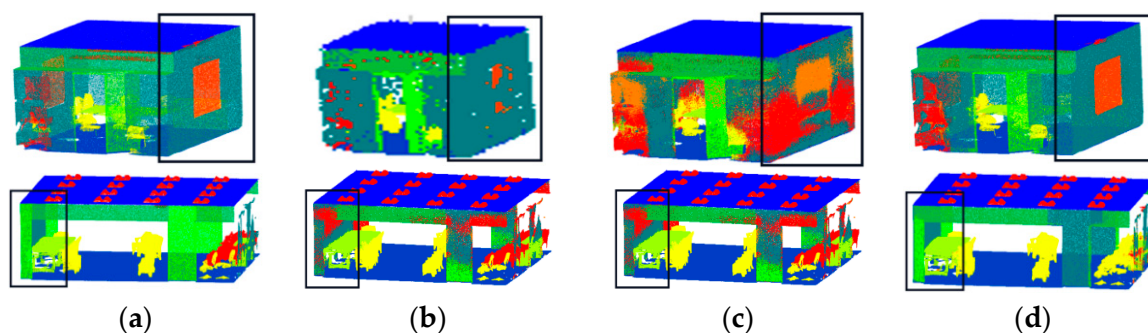


Figure 13. Visualization results of S3DIS: (a) visualization of point clouds with original labels; (b) visualization of DGCNN; (c) visualization of PointNet++; and (d) visualization of FGCM.

5. Discussion

5.1. RGB Feature Set Analysis

In order to test the effect of different texture information feature sets on the experimental results, an experimental comparison between the RGB feature set and the chroma value feature set on the S3DIS dataset was conducted.

As shown in Figure 14, the training speed was improved for the chroma value feature set S_{RGB} compared to the RGB feature set $S_{RGB,norm}$. There was also an improvement in the training accuracy after processing, as shown in Table 6.

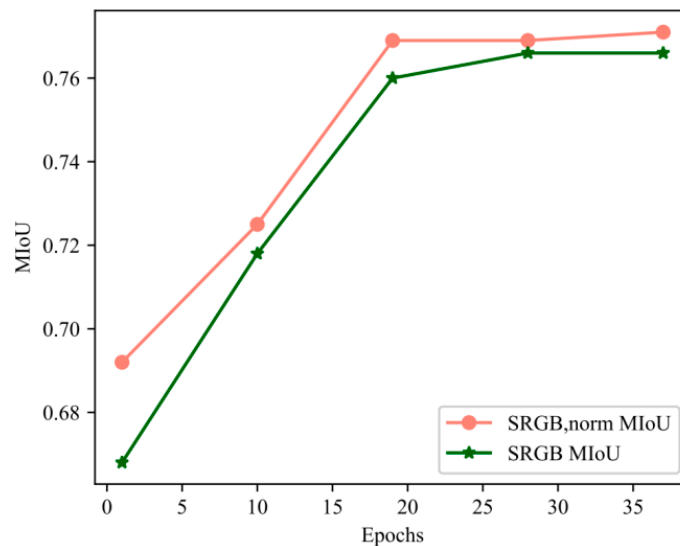


Figure 14. Training curve comparison.

Table 6. MIoU for selected categories in the S3DIS dataset (%).

Method	MIoU	OAcc
S_{RGB}	87.44	91.34
$S_{RGB,norm}$	87.50	91.58

5.2. The Effect of Dual-Channel KNN on Speed

The two-channel KNN can extract local features more comprehensively. As shown in Figure 15, the two-channel KNN made the network reach the optimal state more quickly. At the 14th epoch, the MIoU value of the dual-channel KNN reached the value of the single-channel KNN at the 35th epoch.

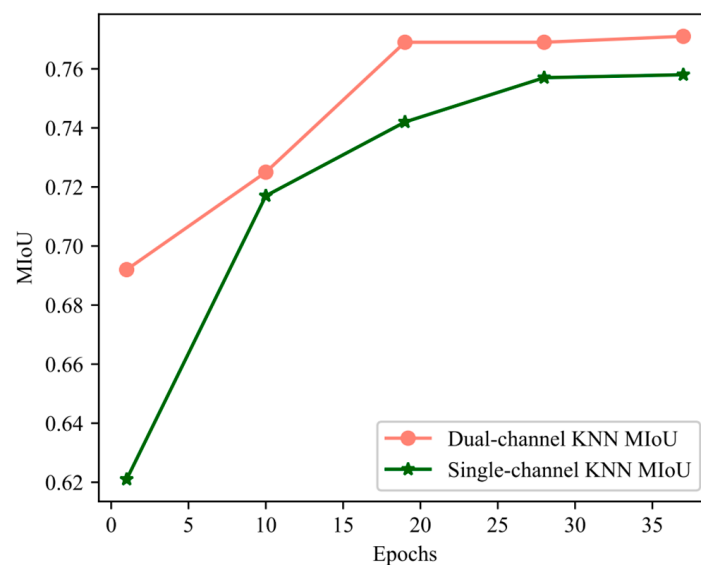


Figure 15. Training curve of MIoU.

5.3. Robustness Analysis

To test the robustness [45,46] of the model, Gaussian noise [47] was added to some files of the semantic segmentation dataset for testing.

As shown in Table 7, after adding Gaussian noise, the segmentation effect of indoor scene 1 and scene 2 was comparable to that before the noise was added. This indicates the greater robustness of our model on the S3DIS indoor dataset. The segmentation effect of outdoor scenes 3 and 4 decreased. The visualization comparison result of scene 3 in Figure 16 demonstrates that despite the accuracy being reduced, the road surface is almost completely segmented, and some unlabeled background information is segmented into means of transportation such as cars. But in scene 4, the leftmost parked bicycle is not segmented, and the people on the road are not segmented. The reason for this is that the point clouds of these scenes are sparse, and the accuracy was reduced due to the noise effect after adding Gaussian noise.

Table 7. Adding Gaussian noise test MIoU (%) comparison.

Scene	Ours	Adding Gaussian Noise
1	71.18	71.70
2	84.15	84.05
3	46.82	44.26
4	72.03	67.88

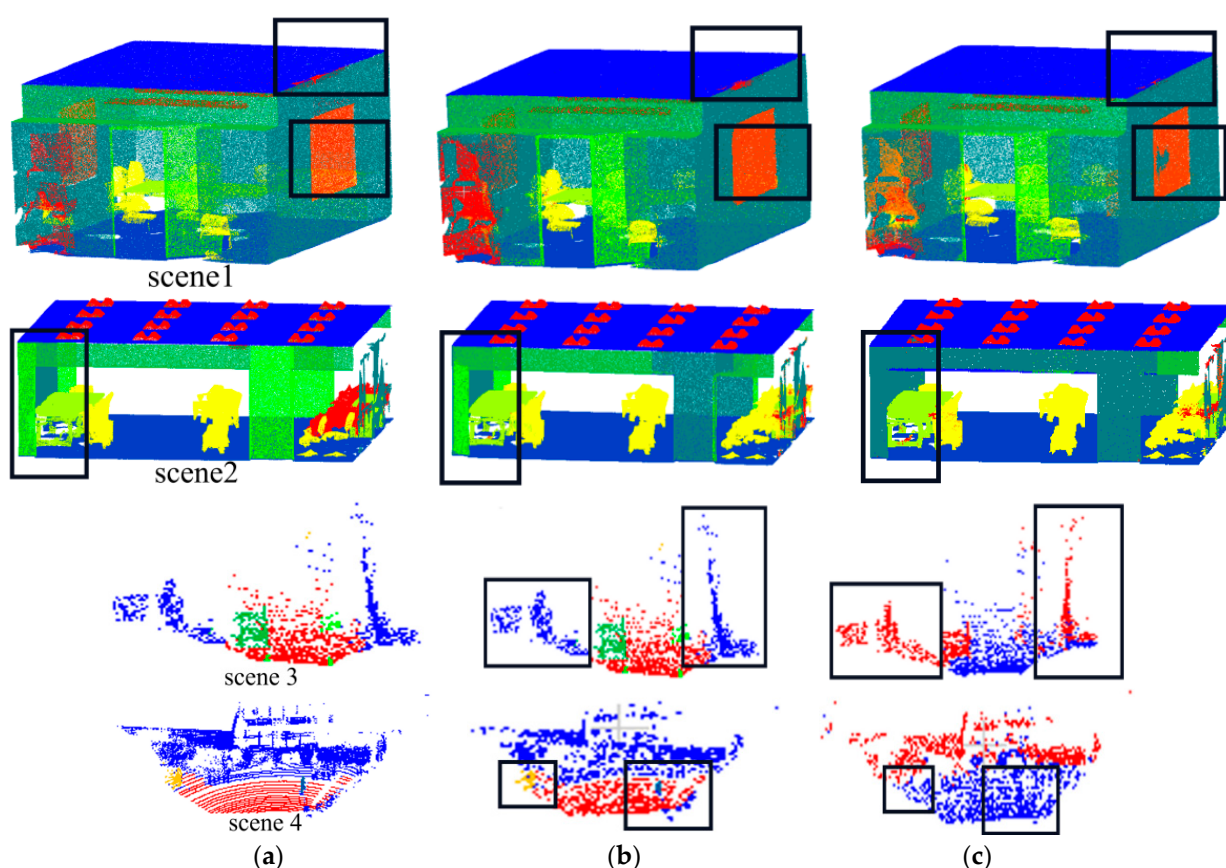


Figure 16. Adding Gaussian noise of SSKIT where: (a) is the real label visualization; (b) is the model test visualization in this paper; and (c) is the visualization of the model in this paper under the data with the addition of Gaussian noise.

To evaluate the performance of our model in scenarios with sparse point clouds, we downsampled the data in scene 1 from its original density to 1/2, 1/4, 1/6, and 1/8.

After analysis, we discovered that when the point cloud is dense, Gaussian noise has a minimal impact on the segmentation results. However, in cases where the point cloud is sparse, the introduction of Gaussian noise significantly disrupts the segmentation results and can even lead to scene failure, as shown in Table 8. In summary, it is evident that the presence of Gaussian noise reduces the segmentation performance of our model, particularly in outdoor environments with sparse point clouds. This indicates a certain level of sensitivity to noise, resulting in decreased robustness.

Table 8. Adding Gaussian noise after downsampling to test MIoU (%) comparison.

Sampling Ratio	Ours	Adding Gaussian Noise
1/2	70.65	70.32
1/4	71.05	70.95
1/6	66.34	50.85
1/8	63.88	30.88

In addition, we discussed potential model failures, which include the following:

(1) Through our experiments, we identified a scenario in which our network encounters difficulties. When objects in the scene are severely occluded, our network struggles to accurately identify the edges of these occluded objects, leading to inaccurate segmentation results. To address this issue, we devised our own solution. In future work, we plan to shift from the traditional perspective to the Bird's-Eye View (BEV) space. The BEV space eliminates the occlusion problem, allowing us to verify the generalization capability of the FGCN model;

(2) When there are significant changes in light intensity, it can impact the 2D image, affecting the fusion result of the image and the point cloud, resulting in reduced segmentation efficiency. However, it is important to note that the model can still operate under these conditions and will not completely fail.

5.4. Ablation Studies

The role of each module in the FGCN was demonstrated through ablation experiments. Table 9 shows the experimental results under the parameters of a batch size of 8 and an epoch of 40. We used the grid search method to determine the parameters. After multiple sets of verifications, we found that a batch size of 8 and an epoch of 40 is the best combination.

Table 9. Ablation experiments on S3DIS dataset and SSKIT dataset (%). Y means this module is included in this experiment, N means this module is not included.

Exp	Data Enhancement	Dual-Channel KNN	Spatial Attention	MAcc (%)	MIoU (%)	OAcc (%)
Exp.0	Y	Y	N	83.33	72.98	89.46
	Y	N	Y	85.64	73.65	90.27
	N	Y	Y	87.23	75.32	90.35
	Y	Y	Y	87.50	78.55	91.58
Exp.1	Y	Y	N	84.48	73.25	89.54
	Y	N	Y	85.98	74.23	90.88
	N	Y	Y	87.47	76.47	90.48
	Y	Y	Y	89.23	79.54	91.45

We chose the S3DIS dataset for evaluation and trained four different models to validate the effectiveness of data enhancement, dual-channel KNN, and spatial attention. The first model excluded the spatial attention module, the second model excluded the dual-channel module, the third model excluded the data augmentation module, and the fourth model included all three components of the FGCN. This set of experiments is denoted as Exp.0. Additionally, to assess the impact of our design on model generalization performance, we

conducted an ablation experiment labeled as Exp.1. Exp.1 also involved training four models, and we used the SSKIT dataset for these experiments. The aim of Exp.1 was to compare the experiment with Exp.0 and evaluate the generalization capability of the FGCN network across different environments. Through these two sets of experiments, it was evident that the three modules of the FGCN contribute significantly to improving experimental results, affirming the generalization ability of the FGCN model. It is noteworthy that the accuracy decreased the most when the spatial attention mechanism was not used. This can indicate that the spatial attention module is the key to improving accuracy.

Furthermore, we also considered the aspect of statistical significance. In this context, we refer to the hypothesis being tested as the null hypothesis, which states that 'there is no significant difference between our method and the missing module methods. At the significance level of $\alpha = 0.05$, if $p > 0.05$, we accept the null hypothesis, and if $p < 0.05$, we reject the null hypothesis. When comparing our method with the missing module methods on both the SSKIT and S3DIS datasets, the p -values were all less than 0.05, leading us to reject the null hypothesis. Consequently, we conclude that there is a significant difference between our method and the missing module methods.

6. Conclusions

In this paper, we proposed the FGCN network model that combines 3D point clouds and 2D image color information for feature learning. First, we studied the representation of 2D images and 3D point clouds. The local feature map was obtained by constructing an image-guided topological map structure using multi-channel map convolution. Second, we added a spatial attention mechanism to process the important parts in space to boost the effectiveness of feature learning. Finally, multi-scale feature fusion was used to learn global features, and MLP and log_softmax were used to enhance model stability.

In order to validate the proposed model, we performed an experimental validation on the S3DIS dataset, comparing the FGCN with popular networks. We demonstrated that the FGCN has some advantages for indoor semantic segmentation tasks. On the SSKIT and KITTI datasets, we performed experimental validation to show that the addition of color information to the point clouds has a positive effect on experimental outcomes for outdoor datasets. The experiments showed that the FGCN can perform feature extraction better than the other networks. Nevertheless, the FGCN faces the challenge of low efficiency when extracting neighbor node information during the two-channel graph convolution feature extraction process. Moreover, there is a need for further research on how to enhance the integration of the graph convolution and attention mechanism for improved performance. Graph-focused convolutional networks have great advantages in dealing with unstructured data. Therefore, our future research will focus on efficient feature extraction using graph structures. Simultaneously, we observed that the FGCN tends to fail when the measured object is significantly obstructed. In future work, we plan to shift our focus to the semantic segmentation of the Bird's-Eye View (BEV) space, where the occlusion problem can be largely mitigated.

Author Contributions: K.Z.: Conceptualization, Methodology, Funding acquisition, Writing—Original Draft, Funding acquisition, Validation. X.W.: Writing—Review and Editing, Supervision, Project administration. R.C.: Software, Formal analysis, Resources, Writing—Original Draft. Z.P.: Data Curation, Visualization. Y.Z.: Investigation, Formal analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Department of Education of Hebei Province under Grant (No. ZD2020176) and supported by Hebei Science and Technology Department project (No. 21470302D).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset related to this article can be found at: <https://goo.gl/forms/4SoGp4KtH1jfRqEj2> (accessed on 28 September 2023) https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d (accessed on 28 September 2023).

Acknowledgments: The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which will improve the presentation. At the same time, we would also like to thank Stanford University for the 3D Indoor Scene Dataset (S3DIS) dataset and the nuScenes dataset launched by nuTonomy. Thanks also to the Computer Vision Algorithm Dataset (KITTI) for the autonomous driving scenarios provided by Karlsruhe Institute of Technology, Germany and Toyota American Institute of Technology.

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

Abbreviations

The following abbreviations are used in this manuscript:

FGCN	Fusion Graph Convolutional Network
KNN	K-Nearest Neighbors
GCN	Graph Convolutional Network
MLP	Multi-Layer Perception
SSKIT	Semantic Segmentation KITTI
OAcc	Overall accuracy
MAcc	Mean Accuracy
BEV	Bird's-Eye View
MIoU	Mean Intersection over Union

References

1. Cui, W.; Yao, M.; Hao, Y.; Wang, Z.; He, X.; Wu, W.; Li, J.; Zhao, H.; Xia, C.; Wang, J. Knowledge and geo-object based graph convolutional network for remote sensing semantic segmentation. *Sensors* **2021**, *21*, 3848. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Dong, S.; Chen, Z. A multi-level feature fusion network for remote sensing image segmentation. *Sensors* **2021**, *21*, 1267. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Li, J.; Wang, H.; Zhang, A.; Liu, Y. Semantic segmentation of hyperspectral remote sensing images based on PSE-UNet model. *Sensors* **2022**, *22*, 9678. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Teixeira, M.A.S.; Nogueira, R.d.C.M.; Dalmedico, N.; Santos, H.B.; Arruda, L.V.R.d.; Neves-Jr, F.; Pipa, D.R.; Ramos, J.E.; Oliveira, A.S.d. Intelligent 3D perception system for semantic description and dynamic interaction. *Sensors* **2019**, *19*, 3764. [\[CrossRef\]](#)
5. Li, D.; Li, C.; Chen, C.; Zhao, Z. Semantic segmentation of a printed circuit board for component recognition based on depth images. *Sensors* **2020**, *20*, 5318. [\[CrossRef\]](#)
6. Memon, M.M.; Hashmani, M.A.; Junejo, A.Z.; Rizvi, S.S.; Raza, K. Unified DeepLabV3+ for Semi-Dark Image Semantic Segmentation. *Sensors* **2022**, *22*, 5312. [\[CrossRef\]](#)
7. Ton, B.; Ahmed, F.; Linssen, J. Semantic Segmentation of Terrestrial Laser Scans of Railway Catenary Arches: A Use Case Perspective. *Sensors* **2022**, *23*, 222. [\[CrossRef\]](#)
8. Wang, Z.; Wu, Y.; Niu, Q. Multi-sensor fusion in automated driving: A survey. *IEEE Access* **2019**, *8*, 2847–2868. [\[CrossRef\]](#)
9. Zhuang, Z.; Li, R.; Jia, K.; Wang, Q.; Li, Y.; Tan, M. Perception-aware multi-sensor fusion for 3D lidar semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 16280–16290. [\[CrossRef\]](#)
10. 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, 4–8 November 2019; pp. 4213–4220. [\[CrossRef\]](#)
11. 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. [\[CrossRef\]](#)
12. Benamer, S.; Mignotte, M.; Parent, S.; Labelle, H.; Skalli, W.; de Guise, J. 3D/2D registration and segmentation of scoliotic vertebrae using statistical models. *Comput. Med. Imaging Graph.* **2003**, *27*, 321–337. [\[CrossRef\]](#)
13. 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. [\[CrossRef\]](#)
14. Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Li, W.; Ma, Y.; Li, H.; Yang, R.; Lin, D. Cylindrical and asymmetrical 3D convolution networks for lidar-based perception. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 6807–6822. [\[CrossRef\]](#)

15. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660. [\[CrossRef\]](#)
16. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 4–9 December 2017; Advances in Neural Information Processing Systems 30; Neural Information Processing Systems Foundation, Inc.: South Lake Tahoe, NV, USA, 2017; Available online: <https://proceedings.neurips.cc/paper/2017/hash/d8bf84be3800d12f74d8b05e9b89836f-Abstract.html> (accessed on 15 October 2021).
17. Çiçek, Ö.; Abdulkadir, A.; Lienkamp, S.S.; Brox, T.; Ronneberger, O. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention, Athens, Greece, 17–21 October 2016; pp. 424–432. [\[CrossRef\]](#)
18. 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, 18–22 June 2018; pp. 9224–9232.
19. Kim, D.-K.; Maturana, D.; Uenoyama, M.; Scherer, S. Season-invariant semantic segmentation with a deep multimodal network. In Proceedings of the Field and Service Robotics: Results of the 11th International Conference, Zurich, Switzerland, 12–15 September 2018; pp. 255–270. [\[CrossRef\]](#)
20. Yang, F.; Yang, J.; Jin, Z.; Wang, H. A fusion model for road detection based on deep learning and fully connected CRF. In Proceedings of the 2018 13th Annual Conference on System of Systems Engineering (SoSE), Paris, France, 19–22 June 2018; pp. 29–36. [\[CrossRef\]](#)
21. Caltagirone, L.; Bellone, M.; Svensson, L.; Wahde, M. LIDAR–camera fusion for road detection using fully convolutional neural networks. *Robot. Auton. Syst.* **2019**, *111*, 125–131. [\[CrossRef\]](#)
22. Ye, C.; Pan, H.; Yu, X.; Gao, H. A spatially enhanced network with camera-lidar fusion for 3D semantic segmentation. *Neurocomputing* **2022**, *484*, 59–66. [\[CrossRef\]](#)
23. Wang, H.; Lou, X.; Cai, Y.; Li, Y.; Chen, L. Real-time vehicle detection algorithm based on vision and lidar point cloud fusion. *J. Sens.* **2019**, *2019*, 8473980. [\[CrossRef\]](#)
24. Huang, T.; Liu, Z.; Chen, X.; Bai, X. Epnnet: Enhancing point features with image semantics for 3D object detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 35–52, Part XV 16. [\[CrossRef\]](#)
25. Mohamed, A.S.; Abdelkader, A.; Anany, M.; El-Beahdy, O.; Faisal, M.; Hangal, A.; Eraqi, H.M.; Moustafa, M.N. An evaluation of rgb and lidar fusion for semantic segmentation. *arXiv* **2021**, arXiv:2108.07661.
26. Polyapram, V.; Wang, W.; Nakamura, R. A point-wise LiDAR and image multimodal fusion network (PMNet) for aerial point cloud 3D semantic segmentation. *Remote Sens.* **2019**, *11*, 2961. [\[CrossRef\]](#)
27. Deng, L.; Yang, M.; Liang, Z.; He, Y.; Wang, C. Fusing geometrical and visual information via superpoints for the semantic segmentation of 3D road scenes. *Tsinghua Sci. Technol.* **2020**, *25*, 498–507. [\[CrossRef\]](#)
28. Yoo, J.H.; Kim, Y.; Kim, J.; Choi, J.W. 3D-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3D object detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 720–736, Part XXVII 16. [\[CrossRef\]](#)
29. El Madawi, K.; Rashed, H.; El Sallab, A.; Nasr, O.; Kamel, H.; Yogamani, S. Rgb and lidar fusion based 3D semantic segmentation for autonomous driving. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 7–12. [\[CrossRef\]](#)
30. Gargano, M.; Longoni, M.; Pesce, V.; Palandri, M.C.; Canepari, A.; Ludwig, N.; Bruni, S. From Materials to Technique: A Complete Non-Invasive Investigation of a Group of Six Ukiyo-E Japanese Woodblock Prints of the Oriental Art Museum E. Chiossone (Genoa, Italy). *Sensors* **2022**, *22*, 8772. [\[CrossRef\]](#)
31. Zhang, Y.-D.; Satapathy, S.C.; Guttery, D.S.; Górriz, J.M.; Wang, S.-H. Improved breast cancer classification through combining graph convolutional network and convolutional neural network. *Inf. Process. Manag.* **2021**, *58*, 102439. [\[CrossRef\]](#)
32. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 146. [\[CrossRef\]](#)
33. Luo, N.; Yu, H.; Huo, Z.; Liu, J.; Wang, Q.; Xu, Y.; Gao, Y. KVGCN: A KNN searching and VLAD combined graph convolutional network for point cloud segmentation. *Remote Sens.* **2021**, *13*, 1003. [\[CrossRef\]](#)
34. Engel, N.; Belagiannis, V.; Dietmayer, K. Point transformer. *IEEE Access* **2021**, *9*, 134826–134840. [\[CrossRef\]](#)
35. 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. [\[CrossRef\]](#)
36. 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. [\[CrossRef\]](#)
37. Zhang, Z.; Hua, B.-S.; Yeung, S.-K. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1607–1616. [\[CrossRef\]](#)
38. Lei, H.; Akhtar, N.; Mian, A. Spherical kernel for efficient graph convolution on 3D point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3664–3680. [\[CrossRef\]](#) [\[PubMed\]](#)

39. Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R.R.; Hu, S.-M. Pct: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [[CrossRef](#)]
40. 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. [[CrossRef](#)]
41. Liong, V.E.; Nguyen, T.N.T.; Widjaja, S.; Sharma, D.; Chong, Z.J. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. *arXiv* **2020**, arXiv:2012.04934.
42. 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. [[CrossRef](#)]
43. Yan, X.; Gao, J.; Li, J.; Zhang, R.; Li, Z.; Huang, R.; Cui, S. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, USA, 2–9 February 2021; pp. 3101–3109. [[CrossRef](#)]
44. Genova, K.; Yin, X.; Kundu, A.; Pantofaru, C.; Cole, F.; Sud, A.; Brewington, B.; Shucker, B.; Funkhouser, T. Learning 3D semantic segmentation with only 2D image supervision. In Proceedings of the 2021 International Conference on 3D Vision (3DV), Online, 1–3 December 2021; pp. 361–372. [[CrossRef](#)]
45. Kim, J.-K.; Park, B.-S.; Kim, W.; Park, J.-T.; Lee, S.; Seo, Y.-H. Robust Estimation and Optimized Transmission of 3D Feature Points for Computer Vision on Mobile Communication Network. *Sensors* **2022**, *22*, 8563. [[CrossRef](#)] [[PubMed](#)]
46. Sun, L.; Ke, D.; Wang, X.; Huang, Z.; Huang, K. Robustness of Deep Learning-Based Specific Emitter Identification under Adversarial Attacks. *Remote Sens.* **2022**, *14*, 4996. [[CrossRef](#)]
47. Jung, W.; Hyeon, J.; Doh, N. Robust cuboid modeling from noisy and incomplete 3D point clouds using Gaussian mixture model. *Remote Sens.* **2022**, *14*, 5035. [[CrossRef](#)]

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