

Article

Point Cloud Segmentation Network Based on Attention Mechanism and Dual Graph Convolution

Xiaowen Yang^{1,2,3,*} , Yanghui Wen^{1,2,3}, Shichao Jiao^{1,2,3}, Rong Zhao^{1,2,3}, Xie Han^{1,2,3} and Ligang He⁴

¹ School of Computer Science and Technology, North University of China, Taiyuan 030051, China; s202107012@st.nuc.edu.cn (Y.W.); 20230006@nuc.edu.cn (S.J.); b1907073@st.nuc.edu.cn (R.Z.); hanxie@nuc.edu.cn (X.H.)

² Shanxi Key Laboratory of Machine Vision and Virtual Reality, Taiyuan 030051, China

³ Shanxi Province's Vision Information Processing and Intelligent Robot Engineering Research Center, Taiyuan 030051, China

⁴ Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK; ligang.he@warwick.ac.uk

* Correspondence: wenyang1314@nuc.edu.cn

Abstract: To overcome the limitations of inadequate local feature representation and the underutilization of global information in dynamic graph convolutions, we propose a network that combines attention mechanisms with dual graph convolutions. Firstly, we construct a static graph based on the dynamic graph using the K-nearest neighbors algorithm and geometric distances of point clouds. This integration of dynamic and static graphs forms a dual graph structure, compensating for the underutilization of geometric positional relationships in the dynamic graph. Next, edge convolutions are applied to extract edge features from the dual graph structure. To further enhance the capturing ability of local features, we employ attention pooling, which combines max pooling and average pooling operations. Secondly, we introduce channel attention modules and spatial self-attention modules to improve the representation ability of global features and enhance semantic segmentation accuracy in our network. Experimental results on the S3DIS dataset demonstrate that compared to dynamic graph convolution alone, our proposed approach effectively utilizes both semantic and geometric relationships between point clouds using dual graph convolutions while addressing limitations related to insufficient local feature extraction. The introduction of attention mechanisms helps mitigate underutilization issues with global information, resulting in significant improvements in model performance.

Keywords: point cloud; semantic segmentation; dynamic graph convolution; dual graph convolution; attention mechanism



Citation: Yang, X.; Wen, Y.; Jiao, S.; Zhao, R.; Han, X.; He, L. Point Cloud Segmentation Network Based on Attention Mechanism and Dual Graph Convolution. *Electronics* **2023**, *12*, 4991. <https://doi.org/10.3390/electronics12244991>

Academic Editors: Chiman Kwan and Yoichi Hayashi

Received: 31 October 2023

Revised: 27 November 2023

Accepted: 8 December 2023

Published: 13 December 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

With the increasing maturity of laser scanning technology [1,2], the efficiency and quality of point cloud acquisition have significantly improved. Three-dimensional point cloud semantic segmentation technologies have crucial applications in fields such as autonomous driving, smart cities, and virtual reality [3], making it a prominent research area in computer vision [4]. In comparison to two-dimensional images, 3D point clouds offer more comprehensive information, including scene depth and three-dimensional object shapes, enabling machines to better comprehend scenes and recognize the real world. However, 3D point clouds possess disadvantages such as non-structural characteristics, unordered data points, and sparsity. Traditional convolution cannot be directly applied to point cloud tasks; thus, automatically and efficiently utilizing point cloud semantic segmentation for understanding the objective world remains an immensely challenging task [5].

Traditional point cloud segmentation algorithms, such as edge-based, model-based, and region-based methods, heavily rely on manually extracted features. This dependence

results in poor generalization and high computational complexity, leading to low efficiency when dealing with large-scale data [6]. In contrast, deep learning-based algorithms offer efficient computations and strong generalization capabilities that enable them to handle large-scale point clouds effectively. Consequently, they have gradually become dominant in the field of point cloud semantic segmentation [7,8]. In recent years, various deep learning-based segmentation networks have been proposed by researchers, which can be categorized into three types of methods: projection-based [9–13], voxelization-based [14–18], and point-based [19–23]. However, projection-based and voxelization-based methods may suffer from information loss due to the transformation of point clouds into other forms of data that could potentially impact segmentation accuracy. In contrast, point-based methods enable direct feature learning on point clouds without additional transformations. In 2017, Qi et al. [19] introduced the pioneering PointNet network, which utilized multi-layer perceptron to extract per-point features and employed max pooling to aggregate global information. However, it neglected the learning of local features. To address this issue, Qi et al. [23] proposed PointNet++, which adopted a hierarchical approach by sampling points at each layer and using PointNet to learn features within the local neighborhood, thus enlarging the receptive field and aggregating local features. Despite its advancements, PointNet++ failed to consider inter-point relationships resulting in insufficient learning of local features. In 2019, Wang et al. [21] introduced a Dynamic Graph Convolutional Neural Network (DGCNN), which represented point clouds as graph structures and utilized edge convolutions to capture local relationships between center points and their neighbors, effectively extracting local features from point clouds. In 2020, Zhai et al. [24] proposed a multi-scale graph convolutional network architecture that used graph convolutional networks of different scales to extract and aggregate multi-scale features in groups, improving model robustness, while Zhang et al. [25], in 2021, connected the output of dynamic graphs for each layer effectively solving gradient disappearance problems; they fixed feature extractor after training then repeatedly trained classifier improving network performance.

However, current dynamic graph convolutional networks face two challenges: (1) the construction of the graph structure based on feature distances partially loses the geometric relationships between point clouds, leading to a loss of intrinsic connections between the data and insufficient extraction of local features and (2) the use of max pooling functions for obtaining global information results in excessive information loss and insufficient representation of global information. To address these issues and further explore the semantic and geometric relationships between point clouds, this paper proposes a Dual Graph Convolutional Neural Network (DualGraphCNN). (1) To address the issue of insufficient local feature extraction capability in dynamic graph convolutions, we introduce a dual graph convolution approach. While constructing the dynamic graph, a static graph independent of the network layers is constructed based on spatial geometric distances. The combination of the dynamic and static graphs forms a dual graph structure. A multi-layer perceptron is utilized to extract edge features on the dual graph structure and aggregate neighborhood features of point clouds. The dual graph convolution not only retains the characteristics of non-local diffusion of information in dynamic graph convolution but also considers the geometric structure of point clouds to a certain extent, which enhances the ability of the network to capture the internal relationship between point clouds. (2) In order to enhance the network's ability to extract global information, we propose a channel attention module and a spatial self-attention module. The channel attention module selectively enhances informative channels while suppressing irrelevant ones by learning how to utilize global information. Meanwhile, the spatial self-attention module models long-range dependency relationships between point clouds, thereby enhancing the representation of global information in point cloud features.

To summarize, our main contributions include the following.

- We propose a dual-graph convolution module, which can make full use of the geometric and semantic information of point clouds, capture the internal relationship between point clouds, and realize the effective extraction of local features of point clouds.
- We propose a channel attention module and a spatial self-attention module to enhance the network's ability to extract global information in both spatial and channel dimensions, thereby optimizing the final segmentation results.
- We apply the proposed methodology to conduct quantitative experiments and various ablation studies on the challenging Stanford Large-Scale 3D Indoor Space (S3DIS) dataset. The experimental results verify the rationality and effectiveness of our method.

2. Methods

2.1. Network Structure Design

The overall structure of the DualGraphCNN network is illustrated in Figure 1. The input to this network consists of point clouds with a size of $N \times D$, where n represents the number of input point clouds and D denotes the feature dimension of each point cloud. These input point clouds undergo a series of stacked three-layer dual graph convolution modules to extract local neighborhood information effectively. To emphasize important features at the channel level, suppress irrelevant features, and enhance the representation of point cloud features, a layer of channel attention module is connected after each dual graph convolution module. Subsequently, the extracted point cloud features from each layer are concatenated and further processed by a shared multi-layer perceptron for deep feature extraction. Global features are obtained via max pooling applied on these processed features. Furthermore, these global features are concatenated with the output features from the first three layers of the double graph convolution module again, followed by another round of feature learning using three shared multi-layer perceptron. Finally, $N \times D$ segmentation results for point clouds are generated as output. Additionally, a spatial self-attention module is employed between these three shared multi-layer perceptrons to model long-range dependencies among point clouds and enhance global information representation.

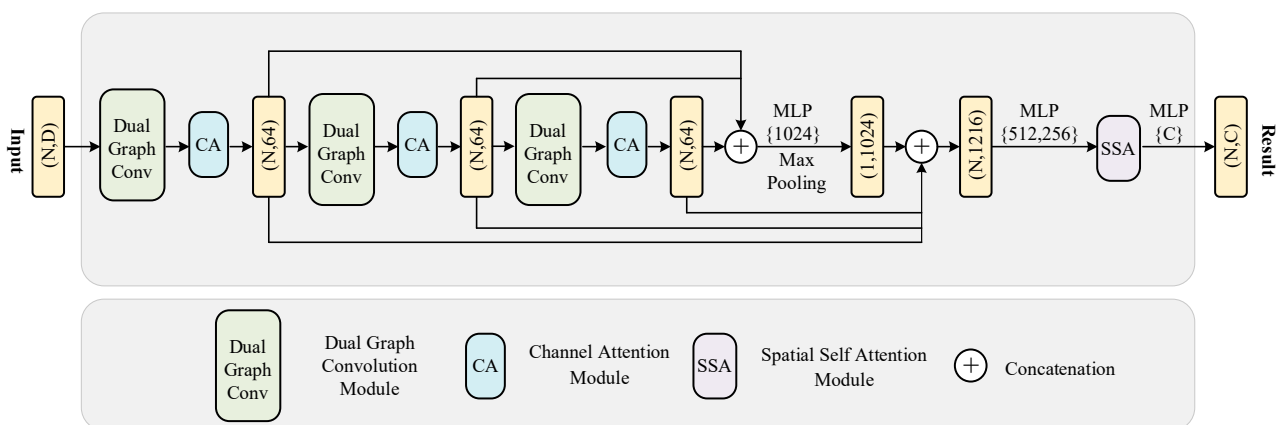


Figure 1. Overall network structure.

2.2. Dual Graph Convolution Module

The data are organized into a graph structure consisting of vertices and edges, and the convolution operation performed on the graph data is referred to as graph convolution. DGCNN employs dynamic graph convolution, which leverages the distances between point cloud features to compute the K-nearest neighbors for each point. As point cloud features evolve during network extraction, the K-nearest neighbor points vary across different network layers, resulting in dynamically updated graph structures for each layer; hence, it is termed dynamic graph convolution. Generally, closer spatial distances indicate higher similarity in feature information and stronger internal relationships. However, in

dynamic graph convolution, neighborhood point cloud sets are constructed solely based on the distance between point cloud features, leading to a partial loss of geometric positional relationships among point clouds and diminishing the network's capacity to capture fine-grained local features.

To address the issue of insufficient utilization of geometric relationships in dynamic graph convolution, this paper proposes a dual-graph convolution method that leverages spatial geometric information to enhance local feature extraction from point clouds by constructing a double graph structure. As illustrated in Figure 2, the dual-graph convolution for feature extraction involves two steps: (1) employing the K-nearest neighbor algorithm to construct the dual-graph structure and (2) utilizing MLP to extract edge features in the graph structure, which are then aggregated by the feature aggregation module to generate local features of the point cloud, C^* represents a higher feature dimension.

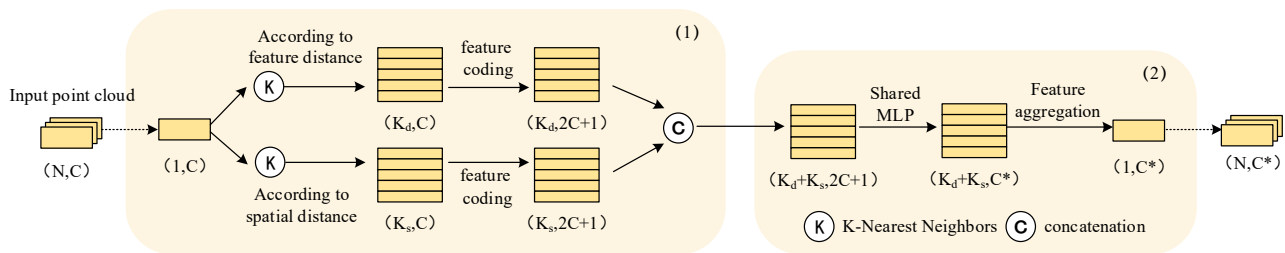


Figure 2. Dual graph convolution module.

(1) Construction of Dual Graph Structure

The input point clouds are defined as $P = \{p_i | i = 1, \dots, N\}$, N is the number of point clouds, p_i represents the i -th point of the input, $x_i \in \mathbb{R}^{N \times 3}$ represents the 3D coordinate vector of point p_i , $f_i \in \mathbb{R}^{N \times C}$ represents the feature vector of point p_i , and C is the feature dimension. For an input point p_i , the construction process of the dual graph structure is shown in Figure 3. The red circle indicates the center point, the blue circle indicates the nearest neighbor point searched by the feature distance, and the orange circle indicates the nearest neighbor point searched by the geometric distance. Firstly, p_i is regarded as the center point, and the feature distance between the center point and the remaining points is calculated by the K-nearest neighbor algorithm. The K_d nearest points are selected as the nearest neighbor points, $p_{ij} (j = 1, 2, \dots, K_d)$, and the edge features between the nearest neighbor points and the center point are calculated, as shown in Equation (1):

$$e_{ij} = (f_i, f_i - f_{ij}, |f_i - f_{ij}|) \quad (1)$$

where f_{ij} represents the feature vector of the neighboring point p_{ij} and $|f_i - f_{ij}|$ represents the feature distance of two points. The center point, neighbor point, and edge feature constitute the dynamic graph structure. Since the point cloud features constantly change with the extraction of the network, the dynamic graph constructed according to the feature distance will be dynamically updated with different layers. Secondly, the geometric distance between the center point and the remaining points is calculated by the K-nearest neighbor algorithm, and the K_s nearest points are selected as the nearest neighbor points: $p'_{ij} (j = 1, 2, \dots, K_s)$, and the edge features between the nearest neighbor points and the center point are calculated, as shown in Equation (2):

$$e'_{ij} = (f_i, f_i - f'_{ij}, |x_i - x'_{ij}|) \quad (2)$$

where f'_{ij} represents the feature vector of the nearest neighbor point p'_{ij} , and x'_{ij} represents the 3D coordinate vector of the nearest neighbor point p'_{ij} . The center point, the nearest neighbor point, and the edge feature constitute the static graph structure. The position of the point clouds in space is fixed, so the vertices of the static graph need to be calculated

only once, and the edge features need to be updated each time, which does not cause too much additional performance loss and can compensate for the lack of geometric position relationship in the dynamic graph structure. Finally, the dynamic graph and the static graph are combined to obtain the double graph structure. The dual graph structure preserves both points with similar features and points with related geometric positions, which is helpful for the extraction of local fine-grained features at the center point.

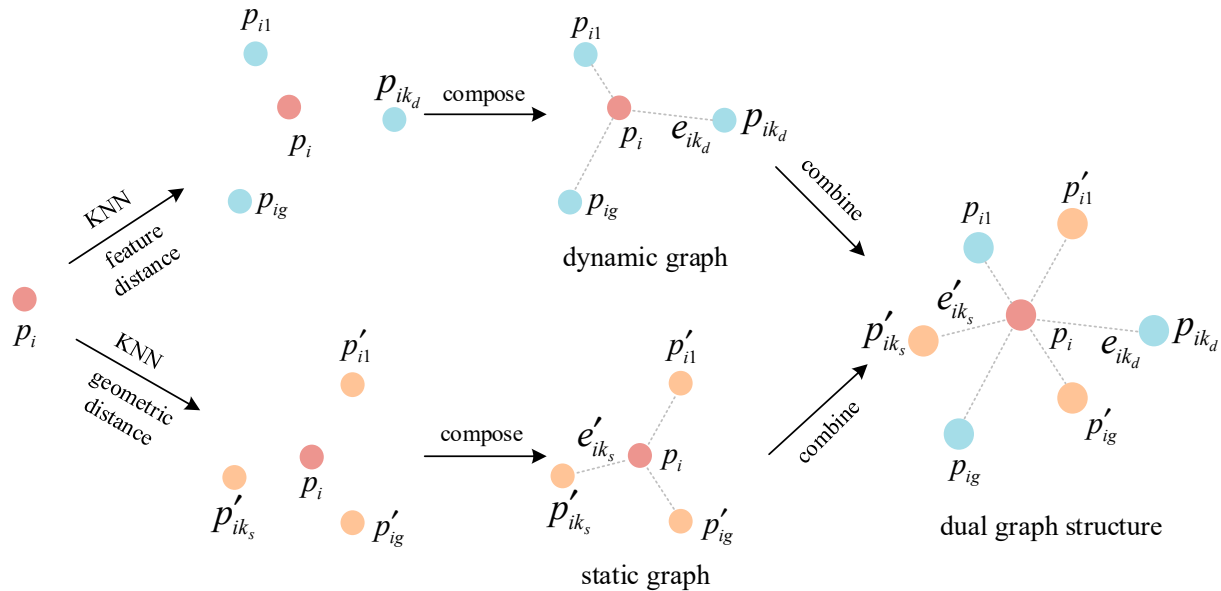


Figure 3. Construction process of dual graph structure.

(2) Extraction of Local Features

In this paper, we employ a shared multi-layer perceptron to extract edge features in the dual graph structure. The encoding process of the edge features constructs feature relationships between the center point and its neighbors, while the multi-layer perceptron further extracts and abstracts high-level semantic features. Specifically, a two-layer MLP is utilized for extracting edge features in the dual-graph convolution module, as shown in Equation (3):

$$\tilde{e}_{ij} = \sigma(bn(MLP(\sigma(bn(MLP(e''_{ij})))))) \quad (3)$$

where σ represents the activation function, bn represents the batch normalization, and e''_{ij} represents the edge features of the dual graph structure. Subsequently, the local feature extraction of point clouds is realized via feature aggregation operation, as shown in Equation (4):

$$\tilde{f}_i = \square_{j:(i,j) \in E_i} \tilde{e}_{ij} \quad (4)$$

where E_i represents the set of edges of the dual graph structure, $\tilde{e}_{ij}$ represents edge features of the dual graph structure after feature extraction, $\tilde{f}_i$ represents the local feature of point p_i , and $\square$ represents the feature aggregation operation. For feature aggregation, the common approach is to use max or average pooling to extract the maximum or average value of local neighborhood features as a representation. However, simple pooling operations can result in a significant loss of information from the neighborhood. To ensure that local neighborhood information is transmitted as much as possible, we propose a feature aggregation module that combines attention mechanisms with both average and max pooling methods. This allows for automatic focus on important local neighborhood information while considering all features within the area. The feature aggregation module is shown in Figure 4, which corresponds to the feature aggregation process in Figure 2.

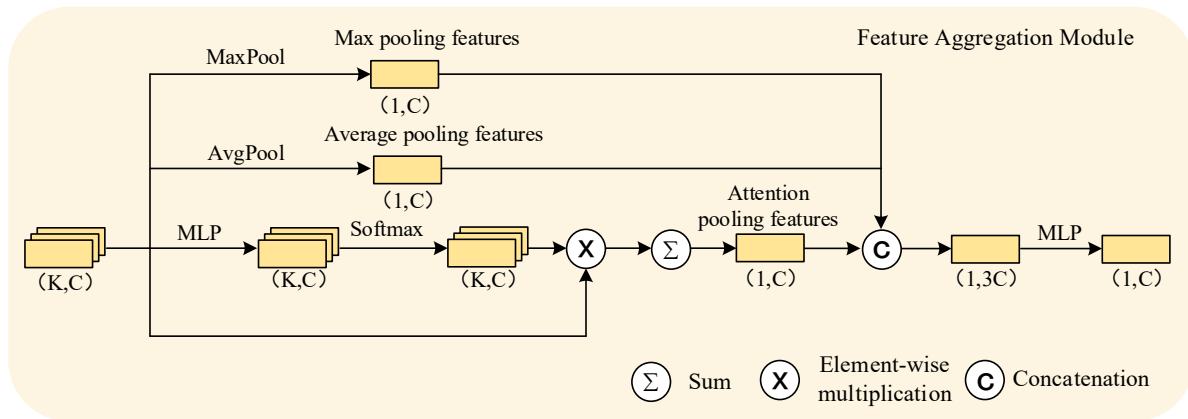


Figure 4. Feature aggregation module.

For attention pooling, the input neighborhood features are first passed through a shared multi-layer perceptron to learn the potential activation levels of each feature, and then the softmax function is applied to calculate an attention score. This learned score acts as a soft mask that automatically selects important features. The attention score is element-wise multiplied by the input features and then summed to obtain an attention-pooled feature. Next, these focused features are concatenated with saliency features from max pooling output and global features from average pooling output. Information interaction and feature redistribution between these three types of pooled outputs are achieved via a multi-layer perceptron network. Finally, local details-rich features are produced.

2.3. Channel Attention Module

The channel attention mechanism is an effective deep learning technology, which can be used to improve the learning ability of the model for key information in the input features [26–28]. It can adaptively adjust the weight of different channels, reduce the weight of unimportant channels, and increase the weight of useful information channels so that the model pays more attention to important features. To this end, we design a channel attention module to highlight important feature channels of point clouds, improve the representation ability of the network by modeling the interdependence between feature channels, and learn to use global information to selectively strengthen useful channels and suppress irrelevant channels.

The channel attention module is shown in Figure 5. Firstly, the spatial information of the input point clouds feature f is aggregated via the average pooling and max pooling operations to generate two different spatial context representative features f_{avg} and f_{max} , which represent the average level and maximum level of the input point clouds in the channel dimension, respectively. f_{avg} and f_{max} are input into a shared multi-layer perceptron to further capture the channel dependency and the two output feature vectors are summed by vector addition, and then the channel attention score cas is obtained by the sigmoid function, as shown in Equation (5):

$$cas = \sigma(MLP(MaxPool(f)) + MLP(AvgPool(f))) \quad (5)$$

where σ denotes the sigmoid function, and the MLP is shared with respect to the parameters of the inputs f_{avg} and f_{max} . Finally, the input feature f and cas are multiplied element-by-element to obtain the enhanced feature after channel attention screening, and the idea of residual connection is used to add the input feature and the enhanced feature to obtain the final output result f_{out} . The specific calculation is shown in Equation (6):

$$f_{out} = f \oplus f \otimes cas \quad (6)$$

where $\oplus$ represents element-wise addition and $\otimes$ represents element-wise multiplication.

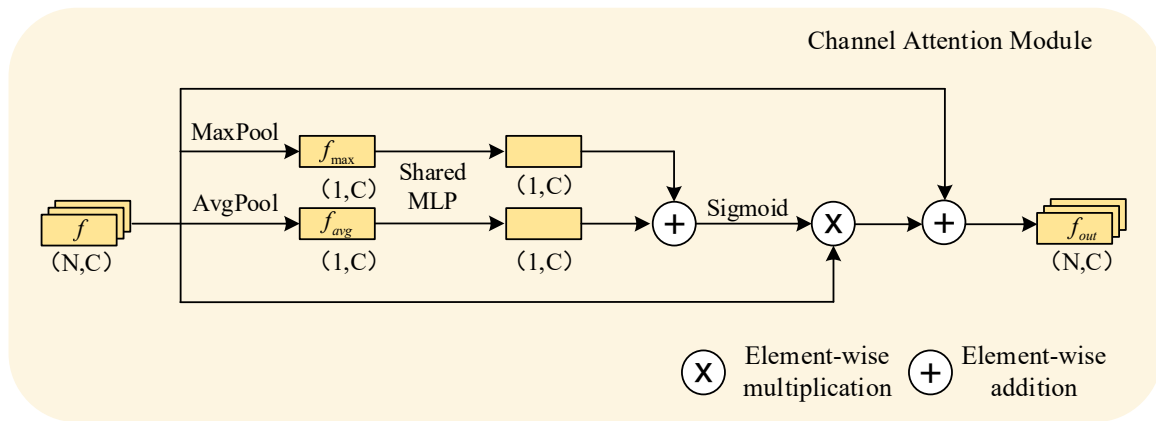


Figure 5. Channel attention module.

2.4. Spatial Self-Attention Module

In semantic segmentation tasks, the global information of the point clouds is very important for the final semantic class prediction because two points with large spatial distance differences may belong to the same class, and their feature representations can be considered at the same time to enhance each other. In the field of 2D images, many works [29,30] adopt the self-attention mechanism to model the long-range dependencies between pixels and obtain global information between points. To effectively capture the global contextual information within each point cloud, we propose a spatial self-attention module that facilitates the modeling of interrelationships among point clouds. For any given point in space, we utilize a weighted sum approach to aggregate all point cloud features and update its feature representation based on similarity with other points. This enables mutual enhancement even when points are spatially distant but exhibit similar features.

The specific operation is shown in Figure 6; given the input point cloud feature vector $F_{in} \in R^{N \times C}$, it is fed into two convolutional layers to generate two new feature vectors $F_A, F_B \in R^{N \times C}$, respectively. Matrix multiplication operation is performed on the transpose of F_A and F_B , and softmax function is applied to calculate the spatial similarity matrix $S \in R^{N \times N}$, and the calculation formula is shown in Equation (7):

$$s_{ij} = \frac{\exp(F_A^i \cdot F_B^j)}{\sum_{i=1}^N \exp(F_A^i \cdot F_B^j)} \quad (7)$$

where s_{ij} represents the influence of the j -th point on the i -th point, that is, the correlation between two points. At the same time, F_{in} is fed into the convolution layer to generate a new feature vector $F_C \in R^{N \times C}$, and then the similarity matrix S from Equation (7) is multiplied with F_C , and the operation result is summed element-wise with F_{in} to obtain the final output result F_{out} . The calculation formula is shown in Equation (8).

$$F_{out}^i = \sum_{j=1}^N (s_{ij} \cdot F_C^j) + F_{in}^i \quad (8)$$

The output feature F_{out}^i of each point cloud is the weighted sum of the original feature and all point cloud features, which has global context information and selectively aggregates the global information according to the similarity matrix. The semantic features of similar points achieve mutual gain, thereby enhancing intra-class aggregation and semantic consistency.

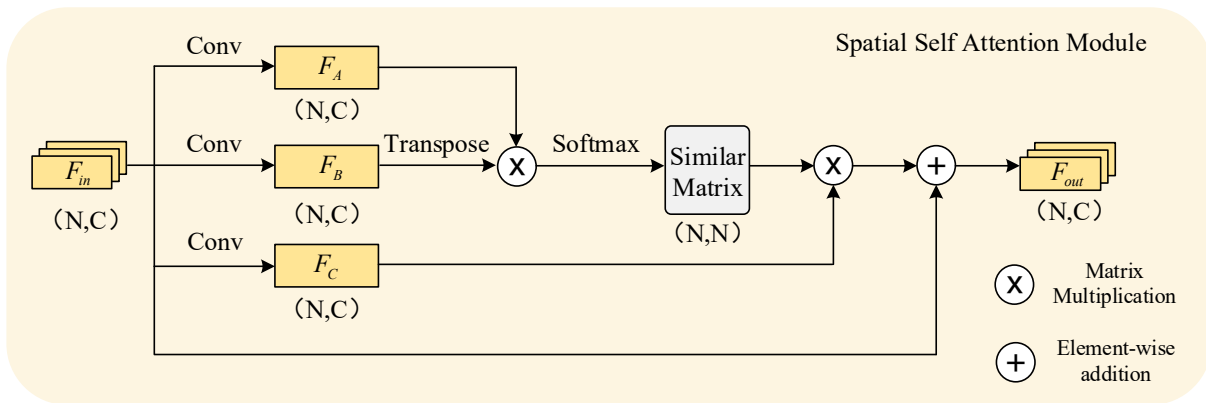


Figure 6. Spatial self-attention module.

3. Experiments and Discussions

3.1. Experimental Dataset

In order to evaluate the segmentation performance of the proposed algorithm model, the public data set S3DIS [31] is selected for training and testing. The S3DIS dataset is a semantic dataset with pixel-level annotations developed by Stanford University. The dataset is a large indoor point cloud dataset containing six large indoor areas consisting of 271 rooms. The areas include office areas, educational and exhibition spaces, meeting rooms, personal offices, restrooms, open spaces, lobbies, staircases, and corridors. The dataset consists of about 273 million scanned points. Each point contains XYZ coordinates and RGB color information and has an explicit semantic label. The S3DIS dataset contains a total of 13 labeled classes, which are ceiling, floor, wall, beam, column, window, door, table, chair, sofa, bookcase, board, and clutter.

The performance evaluation metrics of semantic segmentation are overall accuracy (OA) and Mean Intersection over Union ($mIoU$), which can be defined as follows:

$$OA = \frac{\sum_{i=0}^{K-1} p_{nn}}{\sum_{n=0}^{K-1} \sum_{m=0}^{K-1} p_{nm}} \quad (9)$$

$$mIoU = \frac{1}{K} \sum_{n=0}^{K-1} \frac{p_{nn}}{\sum_{m=0}^{K-1} p_{nm} + \sum_{m=0}^{K-1} p_{mn} - p_{nn}} \quad (10)$$

In Equations (9) and (10), K is the number of categories, n is the true category, m is the predicted category, p_{nn} is the number of correct category predictions, p_{nm} represents the number of false negative examples, and p_{mn} represents the number of false positive examples.

3.2. Network Parameter Setting

The experimental setup for the algorithm in this paper is as follows: the hardware used was an RTX3090 24GB GPU with 128 GB of running memory, while the software employed included UBUNTU20.04 operating system, CUDA11.1, and Python 3.7; Pytorch 1.8.2 served as the deep learning framework utilized in this study. Network training parameters were set to include SGD optimizer with momentum of 0.9, initial learning rate of 0.1, and cosine annealing algorithm to decay it to a value of 0.001. The batch size is set to 12, the number of point clouds fed into the network at each time is about 4096 points, and the network is trained for 100 epochs. The nearest neighbor K_d size is set to 20 for dynamic graphs, and the nearest neighbor K_s size is set to 20 for static graphs.

3.3. Contrast Experiment

The proposed model is evaluated using the six-fold cross-validation method on the S3DIS dataset, where each time, five different regions are selected as the training set and

one region is used as the test set. This process is repeated six times to cover the entire dataset. The input data for each room is divided into blocks with an area of $1\text{ m} \times 1\text{ m}$, and 4096 points are sampled from each block. All point cloud data are used for testing, and overall accuracy and mean intersection over union ratio serve as metrics. Table 1 presents a comparison of metrics for semantic segmentation in S3DIS data between the proposed method and other methods. The results demonstrate that DualGraphCNN, which incorporates dual graph convolution, outperforms the DGCNN network by 7.1% and 3.7% in terms of *mIoU* and *OA*, respectively, indicating its effectiveness in aggregating local features and improving segmentation accuracy. Moreover, compared with PointNet, PointNet++, Point-PlaneNet, KVGCN, and DBAN networks, our model achieves optimal *mIoU* and *OA* scores, demonstrating its superior performance in fine-grained segmentation of complex scenes.

Table 1. Comparison of segmentation accuracy of different methods on S3DIS dataset.

Methods	<i>mIoU</i> /%	<i>OA</i> /%
PointNet [19]	47.6	78.6
PointNet++ [23]	54.5	81.0
DGCNN [21]	56.1	84.1
Point-PlaneNet [32]	54.8	83.9
KVGCN [33]	60.9	87.4
DBAN [34]	60.9	86.1
DualGraphCNN	63.2	87.8

Table 2 presents a comparison of segmentation results for various object types between the proposed network and other models under Region 5. Compared to the DGCNN network, except for beam and window, there are varying degrees of improvement in IoU for other categories, particularly sofa, bookshelf, and wood board, which increased by 15.9%, 12.4%, and 17.3%, respectively. The experimental findings demonstrate that incorporating dual graph convolution and attention mechanisms enhances the model's ability to capture local and global information from point clouds while improving segmentation accuracy across different object types.

Table 2. Semantic segmentation results of Area 5 in S3DIS dataset (unit: %).

Methods	<i>mIoU</i>	Ceil	Floor	Wall	Beam	Col	Wind	Door	Table	Chair	Sofa	Book	Board	Clut
PointNet [19]	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	58.9	52.6	5.9	40.3	26.4	33.2
PointNet++ [23]	50.0	90.8	96.5	74.1	0.0	5.8	43.6	25.4	76.9	69.2	55.6	21.5	49.3	41.9
DGCNN [21]	47.1	92.4	97.5	76.0	0.4	12.0	51.6	27.0	68.6	64.9	43.8	7.7	29.4	40.8
DualGraphCNN	53.6	94.5	98.1	79.4	0.2	19.8	50.3	29.3	72.3	67.4	59.7	20.1	46.7	49.5

Figure 7 shows the visual segmentation result images of the proposed model and DGCNN network. Here, Figure 7a are the input point clouds, Figure 7b are the segmentation results of the DGCNN network, Figure 7c are the segmentation results of the proposed method, and Figure 7d are the reference standard. From top to bottom, they are room 1, room 2, and room 3. The red dashed line represents the region where the results of our method are compared with those of the dynamic graph convolutional network. It can be seen that compared with DGCNN, the proposed method achieves better segmentation results in the details of various objects. In room 1, for the junction of windows, walls, and columns, DGCNN confused their edges together, and the segmentation was not smooth enough. In room 2, DGCNN mistakenly divided a part of the blackboard into a beam with a similar shape, and the boundary between the door frame and the wall was not clearly divided, resulting in the loss of part of the door frame. In room 3, DGCNN failed to correctly distinguish walls with similar colors from white blackboards, and completely divided stacks of bookshelves and sundry objects into bookshelves. The method in this

paper achieves relatively good segmentation results at the above positions, indicating that the introduction of dual graph convolution and attention mechanism strengthens the local information mining ability of the network, improves the contour segmentation ability of various objects, and the segmentation of object connections is smoother.

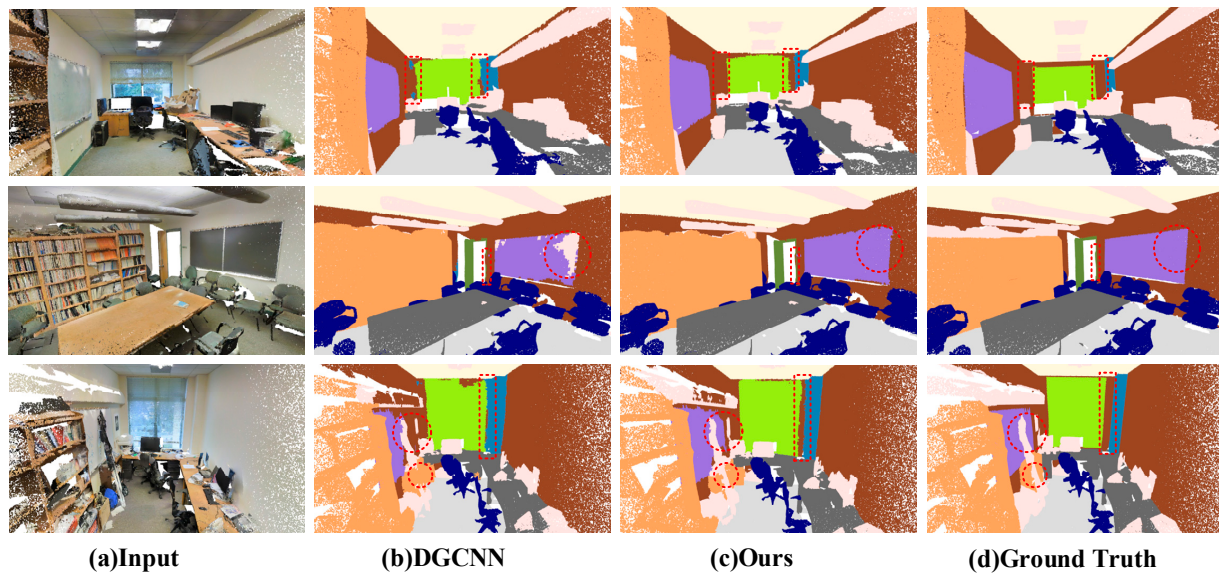


Figure 7. Visual comparison of segmentation results of S3DIS dataset.

3.4. Ablation Study

To validate the functionality and effectiveness of different modules in the network, we conducted an ablation study on various combinations of DualGraphCNN network modules. For this experiment, regions 1–5 from the S3DIS dataset were selected as the training set, while region 6 served as the test set. The results are presented in Table 3, $\checkmark$ means to use the corresponding module, $\times$ means that the corresponding module is not used. Method 1 represents the segmentation outcome of DGCNN. In method 2, we replaced the graph convolution module of DGCNN with a dual graph convolution module, resulting in a 3.8% increase in $mIoU$ for the network. This improvement can be attributed to how the dual graph convolution module compensates for geometric structure limitations present in dynamic graph convolutions by considering both geometric and feature connections between point clouds simultaneously. Additionally, when aggregating point cloud features, this module combines three pooling methods to further enhance local feature representation ability. Method 3 incorporates a spatial self-attention module (SSA) on top of method 2 and demonstrates a $mIoU$ improvement of 1.3% compared to method 2. This enhancement is due to SSA's capability to model long-range dependencies among point clouds and improve global information representation within point cloud features. Method 4 introduces a channel attention module (CA) based on method 2, which selectively strengthens useful channels using global information while suppressing irrelevant channels, thereby enhancing point cloud feature representation, consequently achieving a $mIoU$ that is higher by 0.6% than that achieved by method 2. In method 5, the simultaneous utilization of all modules further improves segmentation performance with an increased $mIoU$ of 5.8% compared to method 1, resulting in varying degrees of improvement compared to other methods.

Table 3. Ablation studies of different modules.

Methods	DualGrap	SSA	CA	<i>mIoU</i> /%	<i>OA</i> /%
1	×	×	×	70.3	88.7
2	✓	×	×	74.1	89.9
3	✓	✓	×	75.4	90.3
4	✓	×	✓	74.7	90.1
5	✓	✓	✓	76.1	90.6

3.5. Selection of Pooling Methods

During the aggregation process of local features, different pooling functions yield local features with distinct characteristics, thereby influencing the segmentation accuracy. To investigate the impact of pooling methods on network performance, combination tests were conducted using various pooling techniques. Region 6 was utilized as the test set, and the results are presented in Table 4, ✓ means to use the corresponding pooling method, × means that the corresponding pooling method is not used. Method A employed max pooling, B used average pooling, and C applied attention pooling, resulting in *mIoU* values of 75.1%, 74.7%, and 75.3%, respectively. It can be inferred that attention pooling produces focused features with superior effectiveness, while max pooling generates saliency-based features with a secondary effect; conversely, average pooling yields overall features with a comparatively lower efficacy. Method D combines max and average pooling to achieve better results than A and B alone, indicating complementarity between salient and overall features which enhances information transmission efficiency when combined together. Building upon method D's foundation, method E incorporates attention pooling to attain optimal *mIoU* scores by enriching information representation via the inclusion of important neighborhood-specific attributes within focused features.

Table 4. Combinatorial testing of different pooling methods.

Methods	Max Pooling	Average Pooling	Attention Pooling	<i>mIoU</i> /%	<i>OA</i> /%
A	✓	×	×	75.1	90.1
B	×	✓	×	74.7	90.0
C	×	×	✓	75.3	90.3
D	✓	✓	×	75.6	90.3
E	✓	✓	✓	76.1	90.6

3.6. K-Nearest Neighbor Size Setting in Dual Graph Convolution

The *K* value of the nearest neighbor plays a crucial role in capturing local information during the construction of dynamic and static graphs. To investigate its impact on network performance, various numerical values were employed for combinatorial testing, with region 6 serving as the test set. The results are presented in Table 5, where K_d represents the size of the *K* value for the dynamic graph, and K_s denotes the size of the *K* value for the static graph. Combination 1 corresponds to using the dynamic graph, similar to DGCNN solely, while combination 2 represents the complete utilization of the static graph. Combinations 3–9 involve employing dual graph structures. Comparing combinations 1 and 2 reveals that constructing a graph structure based on feature distance outperforms geometric distance-based construction. Furthermore, comparing combinations 1, 2, and 6 demonstrates that utilizing a dual graph structure yields superior results compared to using either dynamic or static graphs alone since it retains both points with similar features and those with related geometric positions; this facilitates extraction of fine-grained local features at central points. Additionally, by comparing combinations 3, 4, 5, and 6, it is observed that network performance consistently improves as both K_d and K_s increase due to insufficient acquisition of local information when using small *K* values. Conversely, comparing combinations 6, 7, 8, and 9 indicates that further increasing both K_d and K_s leads

to decreased network performance owing to the introduction of extra noise and redundant information associated with high values of K . Therefore, $K_d = 20$ and $K_s = 20$ are chosen as appropriate sizes for facilitating the extraction of local features in the proposed network.

Table 5. The K nearest neighbor size test.

Combination	K_d	K_s	$mIoU/\%$	$OA/\%$
1	20	0	73.6	89.7
2	0	20	73.1	89.5
3	10	10	72.5	89.2
4	20	10	75.4	90.3
5	10	20	75.1	90.2
6	20	20	76.1	90.6
7	25	20	75.6	90.3
8	20	25	75.5	90.3
9	25	25	75.0	90.1

3.7. Dual Graph Convolutional Module Layer Number Test

To further validate the impact of the number of layers in the dual graph convolution module on segmentation accuracy, we conducted experimental testing by stacking different numbers of layers in the dual graph convolution modules using region 6 as the test set. The results are presented in Table 6. It is evident that an increase in the number of layers in the dual graph convolution module leads to gradual improvements in $mIoU$ and OA . This improvement can be attributed to insufficient aggregation of local features when there are too few layers. However, when there are three layers, $mIoU$ and OA reach their optimum values. Further increasing the number of layers causes a decline in $mIoU$ and OA due to increased network complexity, higher risk of overfitting, training difficulties, and hindered extraction of high-dimensional features.

Table 6. Dual graph convolutional module layer number test.

Layer Number	$mIoU/\%$	$OA/\%$
1	69.5	88.4
2	73.8	89.8
3	76.1	90.6
4	75.4	90.3
5	75.3	90.2

3.8. Robustness Experiments

To evaluate the model's robustness to sparse point clouds, we trained it on regions 1–5 of the S3DIS dataset using randomly subsampled point clouds with 2048, 1024, 512, 256, and 128 points in addition to the original sampling of 4096 points. We then tested the Mean Intersection over Union ($mIoU$) of our model on region 6 and compared it with that of DGCNN. As shown in Figure 8, our proposed model achieved higher $mIoU$ at different sampling points than DGCNN did; this gap widened as the number of sampling points decreased. When only using a sample size of 128 points, our model outperformed DGCNN by a margin of up to 10.1%, indicating its superior robustness to sparse point clouds.

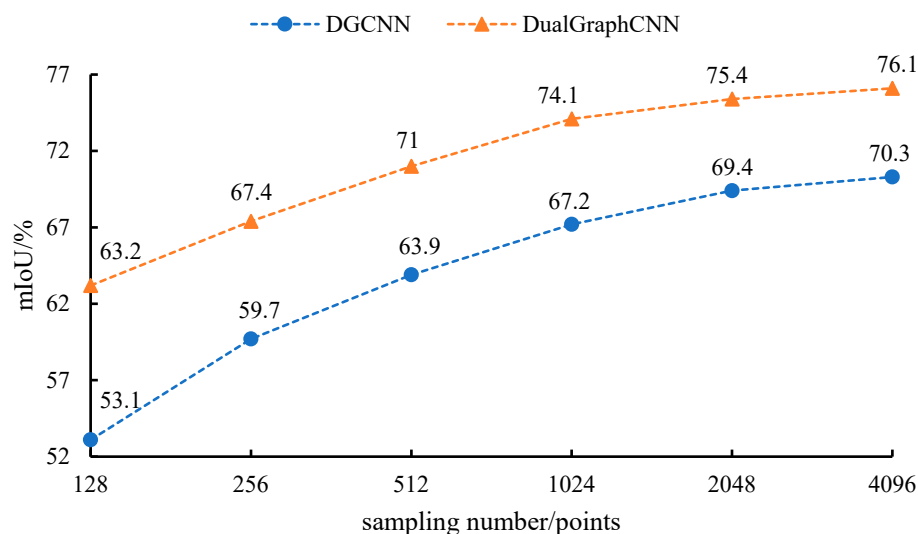


Figure 8. Network robustness testing.

4. Conclusions

The paper proposes a network model that integrates a Dual Graph Convolutional module, channel attention module, and spatial self-attention module to enhance the extraction of both local and global features. Firstly, a Dual Graph Convolutional module is constructed to extract fine-grained local features by effectively utilizing the geometric positional relationships among point clouds compared to dynamic graph convolutions. Secondly, the channel attention mechanism selectively enhances useful channels and suppresses irrelevant channels by learning from global information. This mechanism improves feature focus on relevant aspects and enhances the model's capability to extract discriminative information. Additionally, a spatial self-attention module is incorporated to capture long-range dependency relationships between point clouds, enhancing the representation of global information in point cloud features. This allows for capturing contextual information and considering interactions between points. Experimental results demonstrate that the proposed model enhances point interaction and better extracts local and global features of point clouds while achieving good performance.

Of course, the network model proposed in this paper still needs to be improved in many areas, and how to further simplify the model and reduce the complexity of the model is the focus of the next step. In addition, we will further extend our method to outdoor large-scale scenes for application in the field of autonomous driving, provide high-level semantic information for vehicles to understand the surrounding environment, and improve the environment perception and scene understanding ability of autonomous navigation.

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

Funding: National Natural Science Foundation of China, Grant/Award Number: 62272426; Shanxi Province Science and Technology Major Special Project, Grant/Award Number: 202201150401021; Natural Science Foundation of Shanxi Province, Grant/Award Number: 202203021212138; Natural Science Foundation of Shanxi Province, Grant/Award Number: 202303021211153; Shanxi Province Science and Technology Achievements Transformation Guidance Special Project, Grant/Award Number: 202104021301055.

Data Availability Statement: The S3DIS dataset presented in this study is openly available on the website. Available online: <http://buildingparser.stanford.edu/dataset.html> (accessed on 11 December 2023).

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

References

1. Fu, X.-Y.; Chen, Z.-D.; Han, D.-D.; Zhang, Y.-L.; Xia, H.; Sun, H.-B. Laser fabrication of graphene-based supercapacitors. *Photonics Res.* **2020**, *8*, 577–588. [\[CrossRef\]](#)
2. Choe, J.; Park, C.; Rameau, F.; Park, J.; Kweon, I.S. (Eds.) Pointmixer: Mlp-mixer for point cloud understanding. In Proceedings of the Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Proceedings, Part XXVII; Springer: Berlin/Heidelberg, Germany, 2022.
3. Li, M.; Xie, Y.; Ma, L. Paying attention for adjacent areas: Learning discriminative features for large-scale 3D scene segmentation. *Pattern Recognit.* **2022**, *129*, 108722. [\[CrossRef\]](#)
4. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4338–4364. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Miao, Z.; Song, S.; Tang, P.; Chen, J.; Hu, J.; Gong, Y. MFFRand: Semantic Segmentation of Point Clouds Based on Multi-Scale Feature Fusion and Multi-Loss Supervision. *Electronics* **2022**, *11*, 3626. [\[CrossRef\]](#)
6. Shuai, H.; Xu, X.; Liu, Q. Backward Attentive Fusing Network With Local Aggregation Classifier for 3D Point Cloud Semantic Segmentation. *IEEE Trans. Image Process.* **2021**, *30*, 4973–4984. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Fan, S.; Dong, Q.; Zhu, F.; Lv, Y.; Ye, P.; Wang, F.-Y. SCF-Net: Learning Spatial Contextual Features for Large-Scale Point Cloud Segmentation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 14499–14508.
8. Zhang, J.; Li, X.; Zhao, X.; Zhang, Z. LLGF-Net: Learning Local and Global Feature Fusion for 3D Point Cloud Semantic Segmentation. *Electronics* **2022**, *11*, 2191. [\[CrossRef\]](#)
9. Ahn, P.; Yang, J.; Yi, E.; Lee, C.; Kim, J. Projection-based point convolution for efficient point cloud segmentation. *IEEE Access* **2022**, *10*, 15348–15358. [\[CrossRef\]](#)
10. Kellner, M.; Stahl, B.; Reiterer, A. Fused projection-based point cloud segmentation. *Sensors* **2022**, *22*, 1139. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Kundu, A.; Yin, X.; Fathi, A.; Ross, D.; Brewington, B.; Funkhouser, T.; Pantofaru, C. (Eds.) Virtual multi-view fusion for 3d semantic segmentation. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXIV 16. Springer: Berlin/Heidelberg, Germany, 2020.
12. Du, J.; Huang, X.; Xing, M.; Zhang, T. Improved 3D Semantic Segmentation Model Based on RGB Image and LiDAR Point Cloud Fusion for Automatic Driving. *Int. J. Automot. Technol.* **2023**, *24*, 787–797. [\[CrossRef\]](#)
13. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. (Eds.) Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019.
14. Meng, H.-Y.; Gao, L.; Lai, Y.-K.; Manocha, D. (Eds.) Vv-net: Voxel vae net with group convolutions for point cloud segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
15. Zhao, L.; Xu, S.; Liu, L.; Ming, D.; Tao, W. SVASeg: Sparse Voxel-Based Attention for 3D LiDAR Point Cloud Semantic Segmentation. *Remote Sens.* **2022**, *14*, 4471. [\[CrossRef\]](#)
16. Huang, M.; Wei, P.; Liu, X. An efficient encoding voxel-based segmentation (EVBS) algorithm based on fast adjacent voxel search for point cloud plane segmentation. *Remote Sens.* **2019**, *11*, 2727. [\[CrossRef\]](#)
17. Fang, Z.; Xiong, B.; Liu, F. Sparse point-voxel aggregation network for efficient point cloud semantic segmentation. *IET Comput. Vis.* **2022**, *16*, 644–654. [\[CrossRef\]](#)
18. Park, J.; Kim, C.; Kim, S.; Jo, K. PCSCNet: Fast 3D semantic segmentation of LiDAR point cloud for autonomous car using point convolution and sparse convolution network. *Expert Syst. Appl.* **2023**, *212*, 118815. [\[CrossRef\]](#)
19. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. (Eds.) 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.
20. Du, T.; Ni, J.; Wang, D. Fast Context-Awareness Encoder for LiDAR Point Semantic Segmentation. *Electronics* **2023**, *12*, 3228. [\[CrossRef\]](#)
21. 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*, 1–12. [\[CrossRef\]](#)
22. Wang, G.; Wang, L.; Wu, S.; Zu, S.; Song, B. Semantic Segmentation of Transmission Corridor 3D Point Clouds Based on CA-PointNet++. *Electronics* **2023**, *12*, 2829. [\[CrossRef\]](#)
23. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; p. 30.
24. Zhai, Z.; Zhang, X.; Yao, L. Multi-scale dynamic graph convolution network for point clouds classification. *IEEE Access* **2020**, *8*, 65591–65598. [\[CrossRef\]](#)

25. Zhang, K.; Hao, M.; Wang, J.; Chen, X.; Leng, Y.; de Silva, C.W.; Fu, C. (Eds.) Linked dynamic graph cnn: Learning through point cloud by linking hierarchical features. In Proceedings of the 2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Shanghai, China, 26–28 November 2021; IEEE: Piscataway, NJ, USA, 2021.
26. Hu, J.; Shen, L.; Sun, G. (Eds.) Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
27. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. (Eds.) Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
28. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. (Eds.) ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
29. Wang, X.; Girshick, R.; Gupta, A.; He, K. (Eds.) Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
30. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. (Eds.) Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
31. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. (Eds.) 3d semantic parsing of large-scale indoor spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
32. Peyghambarzadeh, S.M.; Azizmalayeri, F.; Khotanlou, H.; Salarpour, A. Point-PlaneNet: Plane kernel based convolutional neural network for point clouds analysis. *Digit. Signal Process.* **2020**, *98*, 102633. [[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. Zhu, G.; Zhou, Y.; Zhao, J.; Yao, R.; Zhang, M. Point cloud recognition based on lightweight embeddable attention module. *Neurocomputing* **2022**, *472*, 138–148. [[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.