Multiscale Adjacency Matrix CNN: Learning on Multispectral LiDAR Point Cloud via Multiscale Local Graph Convolution

Jian Yang, Binhan Luo, Ruilin Gan, Ao Wang, Shuo Shi, *Member, IEEE*, and Lin Du

***Abstract*—Multispectral LiDAR can rapidly acquire 3D and spectral information of objects, providing richer features for point cloud semantic segmentation. Despite the remarkable performance of existing graph neural networks in point cloud segmentation, extracting local features still poses challenges in multispectral LiDAR point cloud scenes due to the uneven distribution of geometric and spectral information. To address the prevailing challenges, cutting-edge research predominantly focuses on extracting multiscale local features, compensating for feature extraction shortcomings. Thus, we propose a multiscale adjacency matrix convolutional neural network (MS-AMCNN) for multispectral LiDAR point cloud segmentation. In the MS-AMCNN, a local adjacency matrix convolution module was first proposed to efficiently leverage the point cloud's topological relationships and perceive local geometric features. Subsequently, a multiscale feature extraction architecture was adopted to fuse local geometric features and utilize a global self-attention module to globally model the semantic features of multiscale. The network effectively captures global and local representative features of the point cloud by harnessing the capabilities of convolutional neural networks in local feature modeling and the self-attention mechanism in global semantic feature learning. Experimental results on the Titan dataset demonstrate that the proposed MS-AMCNN network achieves a promising multispectral LiDAR point cloud segmentation performance with an overall accuracy of 94.39% and a mean intersection over union (MIoU) of 86.57%. Compared with other state-of-the-art methods, such as DGCNN, which achieved an MIoU of 85.43%, and RandLA-net, with an MIoU of 85.20%, the proposed approach achieves optimal performance in segmentation.**

***Index Terms*—Deep learning, graph convolution, multiscale structure, multispectral LiDAR, point cloud segmentation, self-attention mechanism.**

Manuscript received 10 September 2023; revised 20 September 2023 and 11 November 2023; accepted 17 November 2023. Date of publication 21 November 2023; date of current version 6 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 42271388, Grant 42171347, and Grant 41801268, in part by the Fundamental Research Funds for the Central Universities under Grant 2042022kf1200, in part by the Special Fund of Hubei Luojia Laboratory under Grant 220100034, and in part by the LIESMARS Special Research Funding. *(Corresponding author: Jian Yang.)*

Jian Yang, Binhan Luo, Ruilin Gan, Ao Wang, and Lin Du are with the School of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China (e-mail: yangjian@cug.edu.cn; luobinhan@cug.edu.cn; grl_rs@cug.edu.cn; wangao1999@cug.edu.cn; dulin@cug.edu.cn).

Shuo Shi is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China (e-mail: shishuo@whu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3335300

I. Introduction

With the increasing improvement of devices such as 3D laser scanners and depth sensors, the application scope of 3D data is becoming increasingly wide. Due to affluent geometric shape information, these data play an important role in fields such as autonomous driving [1], [2], urban modeling [3], [4], and engineering surveying [5], [6]. As a typical form, point cloud data contains spatial coordinates and related attribute information and plays a crucial role in the recognition and segmentation of 3D scenes. Achieving accurate point cloud semantic segmentation enables a more detailed understanding and description of scenes, thereby improving scene perception capabilities. In order to enhance the discernment and segmentation capabilities of objects, initial single-wavelength light detection and ranging (LiDAR) systems have progressively evolved into multispectral LiDAR systems, aiming to obtain more comprehensive spectral information. Due to the lack of spectral information, traditional single-wavelength LiDAR cannot comprehensively describe the features of point cloud scenes and obtain satisfactory segmentation results. In contrast, multispectral LiDAR and hyperspectral LiDAR can simultaneously obtain spectral information from multiple wavelengths, thereby achieving more accurate point cloud segmentation results [7], [8]. In real point cloud scenes, multispectral LiDAR data is voluminous, and the feature selection is complex, making the task of multispectral LiDAR semantic segmentation challenging and valuable.

In recent years, two mainstream methods exist for multispectral LiDAR point cloud segmentation: the image-based approach [9], [10] and the point cloud-based approach [7], [11]. The image-based approach primarily converts the multispectral intensity, echo, and elevation information of the multispectral LiDAR into 2-D raster data for classification. Utilizing the constructed multispectral imagery, a subsequent procedure is executed to extract 2-D spectral attributes [12], texture characteristics [13], and normalized vegetation indices [13], among other representative features. Subsequently, a classic machine learning methodology is applied to discern various land cover types. This method simplifies the complexity of data processing. However, converting 3D point clouds into 2-D image data results in the loss of spatial 3D information and spectral information, thereby decreasing the accuracy of semantic segmentation. With the advancement of computer technology and

hardware, more research has shifted towards directly processing multispectral LiDAR point clouds. The point cloud-based segmentation method, similar to the image-based method, utilizes intensity, elevation, and designed spatial descriptors to classify the 3D point cloud, thereby accurately segmenting spatial information. Luo et al. [7] employed the random forest algorithm to classify Titan's multispectral LiDAR point cloud directly, thus verifying the potential of spectral information derived from multispectral LiDAR for land cover classification. Shi et al. [14] performed feature selection on the spectral and contextual attributes of the multispectral LiDAR point cloud using an equalization-based optimization algorithm, followed by classification of the 3-D point cloud using a support vector machine. Their approach achieved higher classification accuracy than solely utilizing the raw coordinate information. However, designing optimal spatial features is a cumbersome task in large-scale and complex scenes, which may lead to unreliable accuracy in point cloud semantic segmentation.

Nowadays, deep learning techniques have experienced significant advancements in various fields, including speech recognition, natural language processing, and computer vision. Deep learning has facilitated an extensive exploration of the abundant spectral characteristics and the potential for point cloud segmentation in multispectral LiDAR. In the domain of deep learning for point clouds, PointNet [15], as a pioneering end-to-end model for point cloud deep learning, directly utilizes raw point clouds as input to extract point features through multilayer perceptrons (MLP) and has demonstrated outstanding performance in tasks such as point cloud classification and semantic segmentation. To address the issue of poor performance in capturing local structural information in PointNet, researchers have proposed a series of improvement methods. DGCNN [16] employs edge convolution operations to replace the stacked MLPs in PointNet in order to preserve permutation invariance while extracting local geometric features from point clouds. On the other hand, HDGCN [17] introduces the graph convolution operator, DGConV block, which aggregates local neighborhood features within the graph and propagates them to the neighboring points. By utilizing a hierarchical structure of DGConV blocks, the network achieves local and global feature extraction from point clouds. As for DDGCN [18], it constructs a similarity matrix in the local graph, which incorporates both point cloud distances and orientations, thereby enabling the extraction of local features in a dynamic neighborhood graph. Graph-based methods model point clouds as the topological structure of graphs and design corresponding convolutional operators, utilizing graph convolutional neural networks (CNNs) for feature extraction and classification. These methods [16], [17], [18], [19], [20] have demonstrated exemplary performance in semantic segmentation. However, most graph convolution-based methods only consider the relationship between the central point and its neighboring points in the local graph while neglecting the importance of relationships among neighboring points. Moreover, in methods based on graph CNNs, most approaches only extract local geometric features from a single scale, neglecting the multiscale neighborhood structure information. This limitation results in a restricted capability of the network to describe scene features. The research [8], [21] on deep learning-based segmentation of multispectral LiDAR point clouds encompasses a comprehensive exploration of global and local representative feature acquisition. Nevertheless, in the context of multispectral LiDAR point cloud scenes, the necessity of achieving multiscale adaptive point cloud feature extraction becomes remarkably prominent, owing to the uneven distribution of geometric spatial patterns and spectral information.

To address these issues, this study proposes a multiscale adjacency matrix CNN (MS-AMCNN) for multispectral LiDAR point cloud analysis. Specifically, we design a local adjacency feature convolution (LAF-ConV) block to extract local features by constructing a graph. This block builds an adjacency matrix between the central point and its neighboring points in the local graph, effectively utilizing spatial relationships in the 3D point cloud to encode local features. Next, we design a multiscale feature extraction block (MSFE block) to aggregate multiscale local features, enriching the structural characteristics of the point cloud. Finally, leveraging the powerful global feature learning capability of the transformer, we further extract multiscale features to establish the contextual information of the point cloud.

In summary, our work contributes to the following aspects.

1) MS-AMCNN is an improved approach based on graph CNNs that aim to input raw data from a multispectral LiDAR and directly process the irregular and unstructured point cloud data. This approach preserves all 3D spatial information while simultaneously ensuring the permutation invariance of the network.
2) We propose a novel algorithm called local adjacency feature convolution. This algorithm constructs multiple local graphs through sampling and performs self-attention convolution on the adjacency matrices formed by the sets of points in the graphs, effectively extracting local information from multispectral LiDAR point clouds.
3) Based on the LAF-ConV block, we introduce a multiscale feature extraction (MSFE) and fusion framework suitable for multispectral LiDAR point clouds. We leverage a global self-attention (GSA) mechanism to perform global feature learning on the fused features, enhancing the expressive capability of the model.

II. Related Works

Semantic segmentation methods for 3D point clouds based on deep learning include four different approaches: projection-based methods, voxel-based methods, point-based methods, and graph-based methods.

A. Projection-Based Method

Projection-based methods are closely related to 2D image processing, where the fundamental idea is to project 3D point cloud data onto a 2D plane or utilize multiple-view images and then process them using 2D CNNs. As a pioneering work in this approach, MVCNN [22] aggregates multiview image features into a global feature descriptor, enhancing segmentation accuracy and precision by observing visual information from different object viewpoints. To improve network robustness and the

accuracy of multiview fusion features, several variant methods [23], [24], [25] have been proposed based on MVCNN. GVCNN [26] groups different visual descriptors extracted by CNNs under different viewpoints based on discriminative scores. It then aggregates the visual feature operators of each group through global pooling to obtain corresponding segmentation results. In contrast to previous methods, View-GCN [27] adopts a graph convolutional network structure. It converts multiview point cloud data into a View-Graph, which is used to aggregate node features of multiple views for learning global shape descriptors. In summary, projection-based methods integrate the projection information of multiple viewpoints or multiple point clouds to enhance the expressive power of point clouds. However, this method often sacrifices the spatial 3D features of point clouds, and the extensive use of projections brings higher time costs and memory consumption.

B. Voxel-Based Method

Voxel-based methods are primarily based on dividing the point cloud into multiple regular 3D grids and then utilizing deep learning models to segment each voxel. Among these methods, VoxNet [28] is one of the earliest point cloud segmentation networks that introduced voxelization. VoxNet directly processes sparse 3D point clouds and effectively captures their shape information by incorporating voxel feature encoding. These networks transform unstructured point cloud data into structured voxel grids and employ CNNs for learning. However, these networks frequently struggle to establish high-resolution voxelized models. To address this issue, OctNet [29] introduces an octree structure, which efficiently handles and represents 3D point clouds with irregular distributions and nonuniform densities, thereby enhancing network performance and efficiency. Additionally, PointGrid [30] adopts space-filling curves to map point cloud data onto a 3D voxel grid. It better learns local geometric feature details by performing convolutions and pooling operations on the voxels. It is worth noting that although voxel-based methods have performed well in point cloud segmentation, the voxelization process sacrifices specific spatial details. Moreover, constructing and storing high-resolution voxel grids requires substantial memory resources, resulting in typically lower computational efficiency.

C. Point-Based Method

Point-based methods directly process 3D point clouds without voxelization or projection. As a pioneering work, PointNet [15] introduces a method based on MLP that can directly perform deep learning on unstructured point clouds while ensuring permutation and rotation invariance in the results. To address the inability of PointNet to capture local features, PointNet++, proposed by Qi et al. [31], models multiscale and multilevel local regions through processes such as hierarchical sampling, local feature extraction, and feature aggregation to capture a broader range of contextual information. The PointNet series networks have demonstrated exemplary performance in tasks such as point cloud classification and segmentation, and many networks [32], [33], [34] have been developed based on this foundational framework. Recent research has discovered that multiscale features exhibit robustness in capturing density variations within point clouds and in aggregating geometric information from different scales. To achieve this objective, 3DMAX-Net [35] has designed a multiscale contextual feature learning block that combines upsample and downsample unit blocks to obtain rich contextual features from point clouds at different scales. Inspired by global feature aggregation algorithms in image processing, 3D-PSPNet [36] adopts a pyramid structure. At each scale of the pyramid, local contextual information within subscenes is independently obtained through grid pooling, and global features are obtained through feature aggregation, thereby enhancing the interaction between large-scale contextual information and small-scale local details. MS-PCNN [37] employs a U-shaped structure of upsampling and downsampling to utilize multiscale global and local features. MNFEAM [38] accelerates semantic abstraction and aggregation of features in the network by capturing semantic relationships in the local feature space with different receptive fields from multiscale neighboring point sets. Enhancing the receptive field of networks to obtain rich local and global contextual information and improve the expressive capability of point cloud features has become a hot research topic in the current field.

D. Graph-Based Method

With the advancement of graph neural networks, graph-based methods have been widely used for mining unstructured data. GACNet network proposed by Wang et al. [39] utilizes the graph attention convolution block to establish local graphs between points and their neighboring points. By employing attention mechanisms to compute edge weights between the central point and its neighbors, the relationships between different nodes are weighted, enabling the superior propagation of important node information. Similarly, DGCNN [16] utilizes EdgeConv to construct dynamic local graphs to extract and learn local semantic features efficiently. DGCNN variants have been developed to enhance the capability of extracting local features and improve overall performance. LGGCM [40] leverages local spatial attention convolution and global spatial attention module to capture geometric features of local point cloud spaces and global contextual information. AGConv [41] dynamically learns point clouds of different semantic parts, generating adaptive graph convolutional kernels, thereby enhancing the flexibility of local convolutions. 3D-GCN [42] introduces a deformable kernel in 3-D space for extracting point cloud features at multiple scales. By establishing topological structures and fully considering the interrelations between points, graph-based methods have proven effective for 3D point cloud data.

Most existing graph-based methods typically construct local graphs using ball queries or k-nearest neighbor searches and utilize MLPs to extract point features. However, these approaches only consider the pairwise relationships between the center point and its neighborhood in the local graph while neglecting the inherent connections among the neighborhood points. This limitation leads to insufficient network capturing of

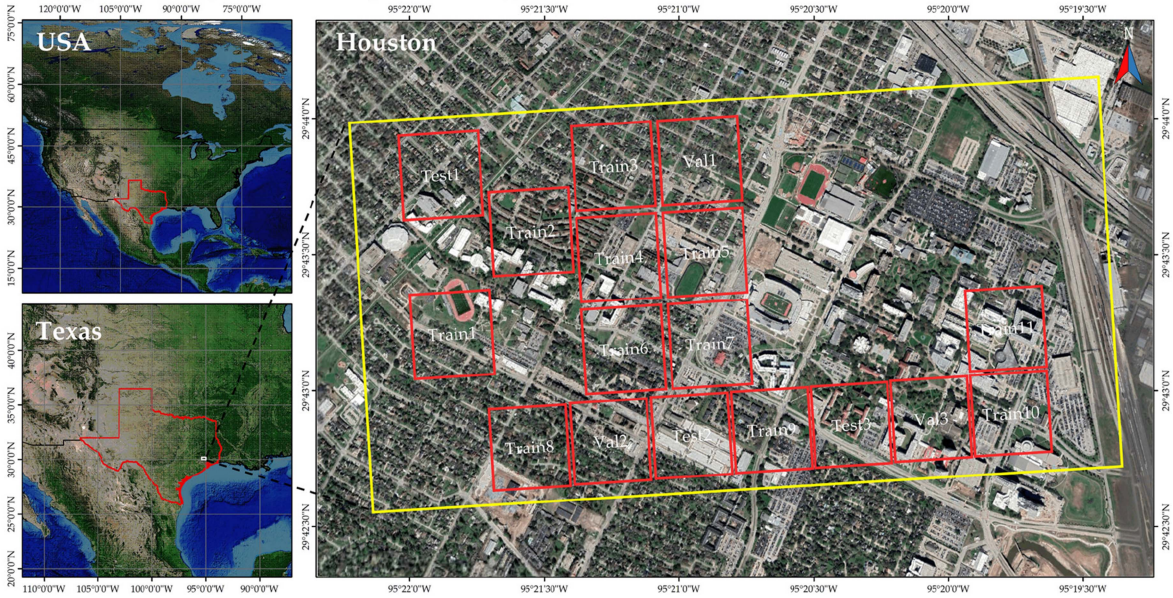


Fig. 1. Study area overview and multispectral lidar point cloud dataset partitioning.

local geometric feature correlations. Additionally, relying solely on single-scale local features makes it challenging to describe the shape of objects in the scene comprehensively. The most designed local convolutional kernels struggled to adapt to varying point densities in multispectral LiDAR point cloud scenes. Therefore, the aggregation of multiscale features is necessary to compensate for the loss of structural information.

Currently, most multispectral LiDAR point cloud segmentation methods rely on classical machine learning algorithms [7], [9], [43], while there is limited research on deep learning-based approaches. Inspired by DGCNN and transformer, this article proposes a graph-based semantic segmentation network called MS-AMCNN. The network includes a local feature extraction block that extracts local features by establishing an adjacency matrix between the central point and its neighboring points in local graphs. Moreover, a GSA is introduced to learn the fused local features in a multiscale structure. Experimental results demonstrate that our network performs satisfactorily in real-world multispectral LiDAR semantic segmentation scenarios. The evaluation metrics reach state-of-the-art levels, providing valuable insights for further research.

III. Study Area and Dataset

A. Study Area

The study area is located near the University of Houston, USA. The data were acquired on February 16, 2017, using an Optech Titan MW (14SEN/CON340) LiDAR system. Optech Titan is a multispectral LiDAR system containing three bands (1550, 1064, and 532 nm) with a pulse repetition frequency of 175 kHz per channel (525 kHz total) and a scan angle of $\pm 26°$. The average flight height during scanning is 500 m above ground level, and the average point density of the multispectral lidar point cloud after data preprocessing is 11.2 points/m^2. The whole dataset covers 4167 m $\times$ 1200 m, and a total of 14 LAS datasets were obtained, which include 20 land cover classes, e.g., healthy grass, artificial turf, evergreen trees, deciduous trees, bare earth, residential buildings, roads, and cars. In this study, we manually selected 17 sample scenes from a pool of preprocesses 14 multispectral LiDAR point cloud LAS datasets. These scenes covered 27 20 000 m^2 and were partitioned into training, validation, and testing sets according to the methodology illustrated in Fig. 1. According to the study of relevant LiDAR data classification, we mainly considered six classes of land cover, i.e., impervious ground, grass, buildings, trees, cars, and powerlines. The samples for impervious surfaces, grass, buildings, and trees have sufficient numbers, while the samples for cars and powerlines are only one-eighth the size of the trees class.

B. Multispectral LiDAR Data Processing

Optech Titan is not strictly a multispectral LiDAR system. The three channels of laser beams have different downward tilting angles, so not every point has intensity data for all three channels [11]. To consolidate the intensity values of the three channels in Titan point cloud data onto a single point, the point cloud data from the three independent channels were merged into a single point cloud data using the method described in [7], based on the principle that adjacent points have correlated intensity information. Only the Titan dataset's first channel (1550 nm) return points were the sole determining factor for inclusion. Any point lacking intensity return values from other channels at that position was excluded.

To obtain the ground truth for the scene, the multispectral LiDAR point cloud is manually labeled point by point with corresponding class labels. Due to the limited capacity of the GPU, the whole sample area cannot be directly input into the network. Therefore, we constructed the multispectral LiDAR

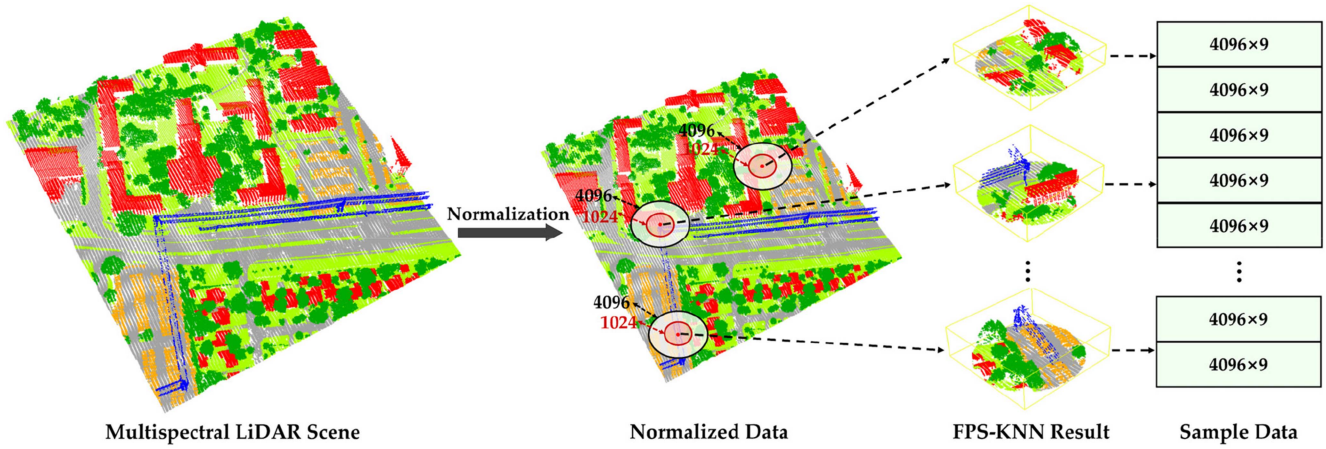


Fig. 2. Proposed sampling strategy for multispectral LiDAR point cloud samples.

dataset for the study area by following the S3DIS [43] and Semantic3D [44] datasets. Typically, when using the S3DIS dataset for deep learning-based point cloud methods, the number of points input into the network is fixed. Meanwhile, considering the point density of the Titan dataset in this area, we have improved the farthest point sampling and K-nearest neighbor (FPS-KNN) sampling method [44] to quickly obtain a fixed number of training samples while preserving the integrity of the scene. Fig. 2 shows the process of our sampling strategy.

1) Data Normalized: In order to accelerate the convergence of the network, normalization of the original data is necessary. A 9-D vector, including $X, Y, Z, R, G, B, X', Y', Z'$ represent each point in the data. Where XYZ represent the position coordinates of each point in the scene ranging from $[-1,1]$; RGB represent the intensity values of the 1550, 1064, and 532 nm channels of the Titan point cloud ranging from [0,1]; $X'Y'Z'$ represent the position coordinates of each point relative to its location in the scene, ranging from [0,1]. Titan multispectral point cloud data are normalized using the mapminmax function after removing the offset. The data normalization is implemented as follows:

$$F_{out} = \frac{(\max - \min) \times (F_{in} - F_{in\_\min})}{(F_{in\_\max} - F_{in\_\min})} + \min \quad (1)$$

where F_{out} is the result of feature normalization, $[\min, \max]$ is the range of values after feature normalization. F_{in}, $F_{in\_\min}$, and $F_{in\_\max}$, respectively, represent the original input feature and its minimum and maximum values. Individually the 9-D features of every point in the original data are normalized using (1).

2) Optimized FPS-KNN: Uniform sampling [15], voxel sampling [45], and block sampling [46] are standard point cloud sampling methods. The FPS-KNN sampling method for multispectral LiDAR point cloud data can better preserve the integrity of objects and effectively generate samples that cover the whole scene, compared with the above-mentioned methods. KNN compensates for some pointwise spatial relationships lost in downsampling by acquiring a fixed number of neighboring points in FPS. Considering the density of multispectral LiDAR point cloud and data augmentation in this study area, we improved the FPS-KNN method, and the flowchart is as follows.

1) Randomly select one point from the input multispectral point cloud scene as the initial point. Then, use this point as the center of KNN to search for k_1 and k_2 nearest neighbors in its neighborhood. To minimize the loss of multispectral LiDAR point cloud scene features during the sampling process, we establish a reasonable number of sampling points based on the density of the multispectral LiDAR point cloud in the dataset. The rationality of this parameter setting has been validated through ablation experiments to ensure that the sampling points adequately represent the characteristics of the entire multispectral LiDAR scene. In this experiment, k_1 is set to 4096, and k_2 is set to 1024, both including the point itself. Use the k_1 nearest neighbors as a training sample and remove the k_2 nearest neighbors from the point cloud scene. During the generation process of multispectral LiDAR point cloud samples, it is essential to record the index number of each point in the scene to determine the presence of duplicate regions within the samples.
2) Calculate the 3D distance from the previous seed point to the remaining point cloud scene, and designate the point with the farthest spatial distance as the next seed point. Repeat the operation in step 1) to obtain another sample.
3) Iterate the operation in step 2) until the sample covers the entire point cloud scene and obtain a fixed number of output samples.

For point cloud samples in overlapping areas, the method uses the class with the highest predicted count for each duplicated point as the final segmentation result. Compared with the original FPS-KNN sampling method, this method can effectively perform data augmentation and expand the scene's sample size.

IV. Methodology

Point clouds have plenty of 3D spatial features, which can intuitively describe the characteristics of natural spatial objects. However, point clouds have the characteristics of discrete and

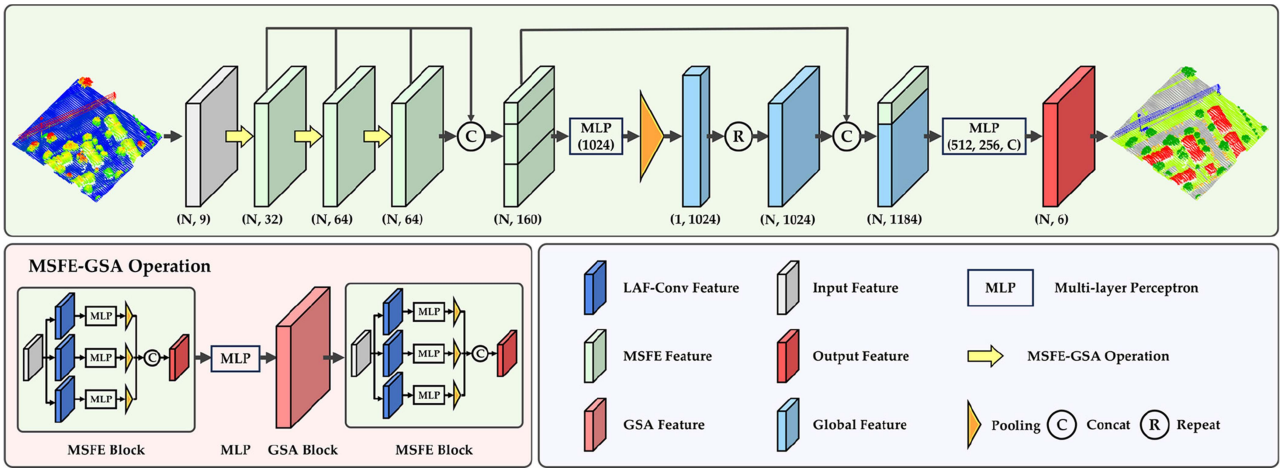


Fig. 3. Proposed MS-AMCNN.

uneven distribution, which leads to the need for a topological relationship between points. Standard convolution operators are powerless to deal with disordered point cloud features. Inspired by previous related research, we proposed a novel approach to more effectively obtain topological relationships between point clouds and better perceive 3D spatial information and local semantic features of points. This network enhances the topological information of point clouds by modifying the structure of local adjacency graphs. Furthermore, it utilizes a multiscale structure to improve point cloud segmentation efficiency and the ability to obtain local geometric features. The designed network mainly consists of three key components: (1) A point cloud local adjacency matrix feature convolution that fully uses the spatial relationship between 3D points to extract local features effectively. (2) An MSFE block that performs feature extraction on points at diverse levels and aggregates multidimensional features to further encode local features. (3) A GSA block that utilizes the excellent global feature learning ability of the self-attention mechanism to enhance the global feature from the multiscale feature block.

A. Network Architecture

This study highlights the importance of selecting the optimal segmentation scale in the multispectral LiDAR point cloud scene, where objects exist at multiple scales. To address this challenge, we designed the detailed architecture of MS-AMCNN, as depicted in Fig. 3. The network comprises multiple MSFE blocks, GSA blocks, and MSFE-GSA operations stacked together to output the point cloud's semantic segmentation results end-to-end during training.

MSFE-GSA operation effectively combines the proposed MSFE and global self-attention mechanism. Adopting a multiscale local graph feature extraction method containing LAF-ConV can effectively extract the local features of the multispectral LiDAR point cloud. The MSFE block aggregates the multiscale features, while the GSA block enhances global contextual features. Following multiple MSFE-GSA operations, extraction of global features occurs through the maxpooling layer. After that, global features and local features are fused and passed through fully connected layers to output each point's label. The network's details are described in the following.

First, a multispectral LiDAR point cloud sample of dimensions $N \times d$ (excluding batch dimension) is inputted into the network, where N represents the number of points in the sample and d represents the initial feature dimension of the input points. The initial features undergo processing by the MSFE-GSA operation layer, resulting in local features of dimensions $N \times 32$. Within the framework of the MSFE-GSA operation, regarding the input multispectral LiDAR point cloud samples, a multiscale local graph is established through KNN nearest neighbor search. For each local graph, local adjacency matrices are constructed utilizing the LAF-ConV approach to acquire local features. Using MLP and pooling operations, the multiscale local features are connected to generate the output MSFE features. Subsequently, these features are fed into the GSA block for global feature modeling, thereby obtaining the output results of the graph convolutional layers. Subsequently, the features extracted by the previous MSFE-GSA operation layer are used as inputs to the subsequent MSFE-GSA operation layer, resulting in two levels of local features, each with dimensions of $N \times 64$. Like a CNN network, the three local features are fused and inputted into an MLP layer, resulting in a global feature of dimensions $N \times 1024$, further enhanced by a max-pooling layer to obtain a global descriptor.

Subsequently, the 1-D global descriptor is repeated to expand to each point, resulting in a new feature of dimensions $N \times 1024$. The global descriptor is then concatenated with the previous three local features to obtain the fused global and local features with dimensions of $N \times 1184$.

Finally, the fused features are transformed into multispectral LiDAR point cloud class labels using a fully connected layer.

B. Local Adjacency Feature Convolution

We build the local graph of each point in the scene by searching the point cloud using KNN for the input samples. As shown in Fig. 4, consider the local graph $G(V, E)$ consisting of the set of points $P_i = \{p_i, p_{i1}, p_{i2}, \ldots, p_{iK}\} \epsilon \mathbb{R}^{3+C}$, where p_i is the central point of the point set P_i, and p_{ij} denotes

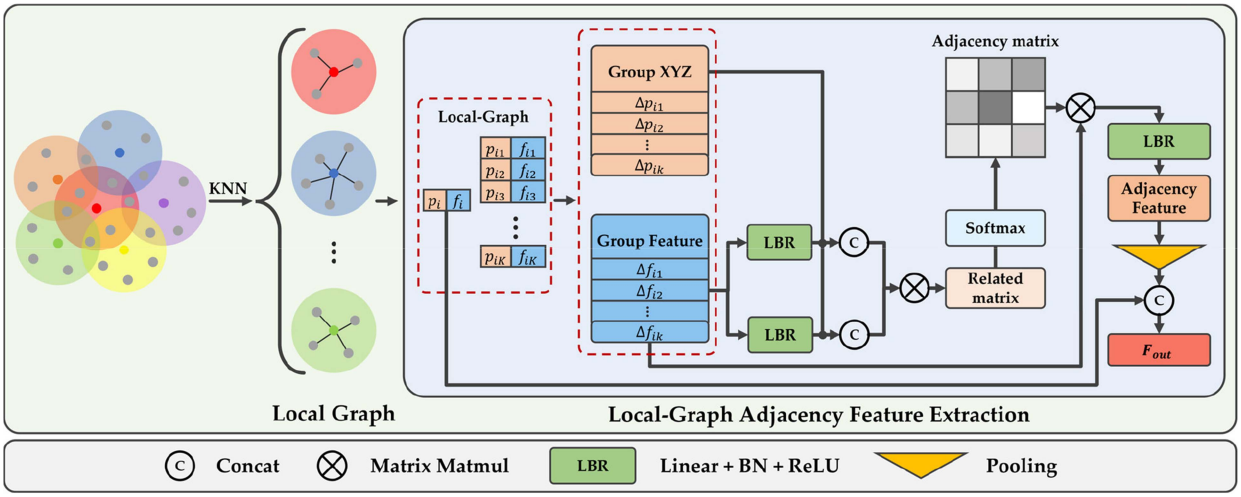


Fig. 4. Local adjacency feature convolution block on the local graph.

the K neighboring points of p_i. Here, 3 and C, respectively, stand for the spatial coordinate dimension and spectral feature dimension of the multispectral LiDAR point cloud. For the local graph, $V_i = \{1, 2, \ldots, K\}$ and $E_i \subseteq V_i \times \bar{V}_i$ represent the set of vertices and edges, respectively.

Let $F_i = \{f_i; f_{i1}; f_{i2}; \ldots; f_{iK}\}$ be the feature set corresponding to P_i. The correlation between neighboring nodes is calculated by constructing the adjacency matrix of the local graph using a self-attentive mechanism. In order to capture feature differences in the local neighborhood, the relative positional coordinates $\Delta P_i = \{\Delta p_{i1}, \Delta p_{i2}, \ldots, \Delta p_{iK}\}$ and feature differences $\Delta F_i = \{\Delta f_{i1}, \Delta f_{i2}, \ldots, \Delta f_{iK}\}$ of all adjacent points p_{ij} to the central point p_i and their corresponding features f_{ij} are calculated. The implementation is as follows:

$$\Delta p_{ij} = p_{ij} - p_i \tag{2}$$

$$\Delta f_{ij} = f_i - f_{ij}. \tag{3}$$

Subsequently, a self-attention mechanism is employed to generate the adjacency matrix of the neighboring points, which further obtains the relationship between adjacent nodes in the local graph. This method satisfies the point cloud permutation invariance [47]. We obtain the local autocorrelation matrix by concating the relative positional coordinates and the differences in high-dimensional features. The related matrix $\mathcal{R}$ measures the high-dimensional spatial relationship between features of adjacent nodes, and it is defined as follows:

$$\mathcal{R} = [\Delta p_{ij} || \gamma (\Delta f_{ij})]^T \times [\Delta p_{ij} || \theta (\Delta f_{ij})]. \tag{4}$$

In the equation, γ and θ represent two different MLPs with nonlinear activation functions. The symbols $||$ and $\times$ denote the connect operation and matrix multiplication. We utilize Softmax to diminish the redundant features among different nodes in the related matrix and generate the adjacency matrix A. Each element of A is defined as follows:

$$A_{ij} = \frac{\exp(\mathcal{R}_{ij})}{\sum_{j=1}^{k} \exp(\mathcal{R}_{ij})}. \tag{5}$$

A_{ij} and $\mathcal{R}_{ij}$ are the elements of adjacency matrix A and correlation matrix $\mathcal{R}$, respectively. The central node features are updated by multiplying the adjacency matrix with the original features. The formula is as follows:

$$\tilde{f}_i = maxpooling(A_{ij} \Delta f_{ij}). \tag{6}$$

Finally, by incorporating the captured global shape structural information f_i with the local neighborhood information $\tilde{f}_i$, the output features of the central point p_i in the local graph are obtained. The output features of the LAF-Conv are as follows:

$$F_{LAF-Conv} = LAF - Conv\left(\tilde{f}_i || f_i\right). \tag{7}$$

C. MSFE Block

We have designed a multiscale structure to improve the accuracy of point cloud segmentation and enhance the diversity of local features. The effectiveness of this method has been well-established in previous studies [31], [45], [46]. Based on the locally self-attentive adjacency matrix, LAF-ConV is implemented to extract local features. In order to improve the efficiency of point cloud segmentation and the sensitivity of local feature extraction, we design an MSFE block to construct local graphs with different numbers of neighboring points and utilize LAF-ConV for feature extraction and multiscale feature fusion.

Previous studies have demonstrated the effectiveness of multiscale structures; therefore, we adopt the MSFE block architecture, as shown in Fig. 5. For the original training samples of the input network, we perform KNN search on different numbers of neighboring points to construct multiscale sampling results. We construct corresponding local graphs at different scales of neighboring points and use the LAF-ConV block for local self-attentive adjacency matrix feature extraction. For the extraction results of the LAF-ConV block at each scale, we use three different MLPs with nonlinear, learnable activation functions for feature transformation and pool the features through a pooling layer for feature fusion to obtain the output results. The entire

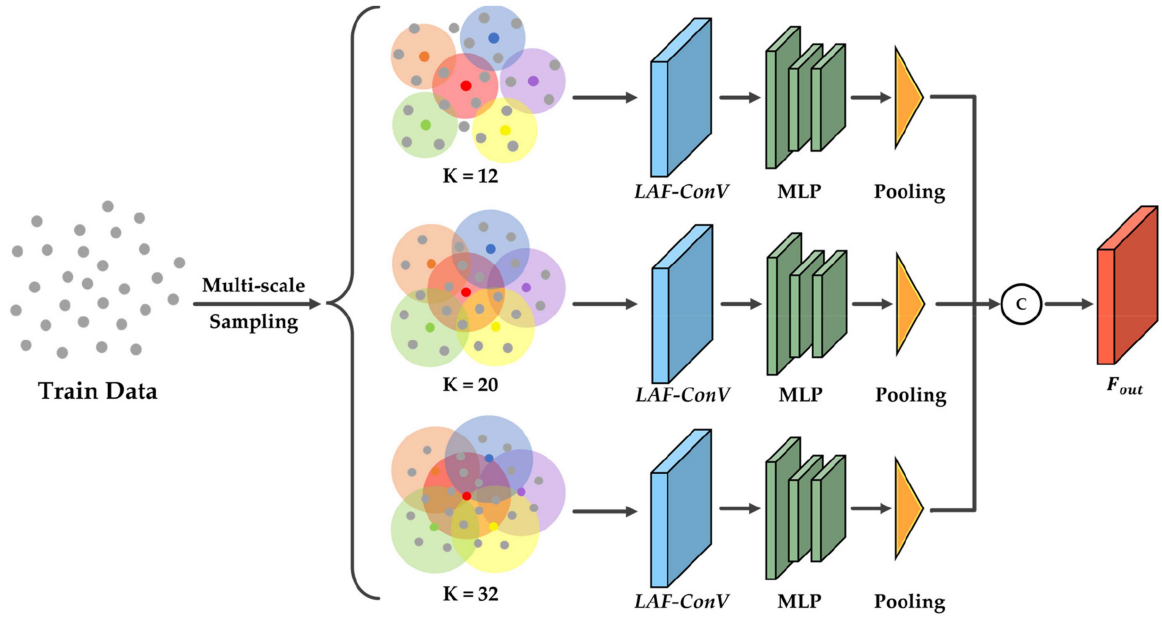


Fig. 5. MSFE block design.

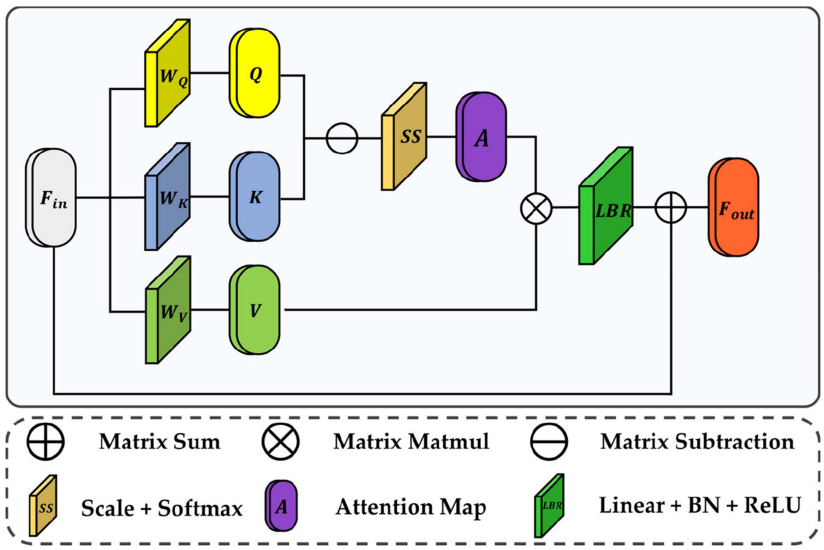


Fig. 6. GSA block structure.

process of the MSFE block is shown in the following:

$$F_{MSFE} = MSFE\left[\varphi(F^{12}_{LAF-Conv}) \,||\, \Theta(F^{20}_{LAF-Conv}) \,||\, \psi(F^{32}_{LAF-Conv})\right] \tag{8}$$

where $F^{12}_{LAF-Conv}$, $F^{20}_{LAF-Conv}$, and $F^{32}_{LAF-Conv}$ represent the features extracted by LAF-ConV from the local graphs constructed at scales of 12, 20, and 32 neighboring points, respectively. φ, Θ, and ψ are three independent MLPs, respectively.

D. Global Self-Attention Mechanism

The self-attention mechanism possesses the capability to capture long-range dependencies between features dynamically. Moreover, it can adaptively adjust weights based on the information from different positions in the input model, thereby better modeling contextual relationships and significantly enhancing the model's generalization ability [48].

Although LFA-ConV and MSFE block can extract local structural features of point clouds through neighborhood graphs at different scales, they still need global contextual information of the scene. Therefore, we introduced a GSA block to learn the fusion of local point features at multiple scales to perceive global contextual relationships sufficiently and improve the accuracy and generalization ability of point cloud scene understanding. GSA block as shown in Fig. 6.

The output features of the MSFE block are used as the embedding layer input of the GSA block. Specifically, the input

TABLE I
AVERAGE ACCURACY EVALUATION METRICS FOR DIFFERENT METHODS UNDER THE THREE SCENARIOS

Networks	OA(%)	AP(%)	AR(%)	A-F1(%)	MIou(%)	Kappa(%)
PointNet++	90.50	77.21	74.96	73.99	66.94	86.31
GACNet	87.07	76.33	81.57	78.80	65.09	91.68
RandLA-Net	90.28	92.25	91.29	90.13	85.20	85.94
DGCNN	94.27	90.55	93.39	**91.91**	85.43	91.37
LDGCNN	93.65	**92.52**	90.89	89.46	84.27	90.81
PointTransformer	88.75	92.31	90.32	91.30	84.57	83.91
Our model	**94.39**	91.85	**94.73**	91.12	**86.57**	**91.80**

The bold values represent the highest value of the accuracy evaluation index in each column.

features $Y = \{y_1; y_2; y_3; \ldots; y_S\}$. The following formula can express the computation of the GSA block:

$$GSA\,(Q, K, V) = \text{Attenction}\left(QW^Q, KW^K, VW^V\right). \quad (9)$$

Among them, Q, K, and V represent query, key, and value vectors, respectively. Where W_i^Q, W_i^K, and W_i^V are the linear learnable transformation matrices for the query, key, and value, respectively. In the GSA block, the self-attention mechanism is used to calculate attention, which can be expressed as follows:

$$\text{Attenction}\ (Q, K, V) = \sigma\left(\text{Softmax}\left(Q \ominus K\right) \cdot V\right). \quad (10)$$

Here, the Softmax function is used for normalization to calculate the weight of each key vector, which is then multiplied by the value vector to obtain the self-attention result of that head, and using σ for linear transformation of the results. Finally, the self-attention results obtained from all heads are weighted and fused and added to the original embedding features to obtain the output of the GSA block. The GSA block can be represented as a whole as follows:

$$F_{out} = GSA\ (Y) = GSA\,(Q, K, V) + Y. \quad (11)$$

E. Implementation Details

All experiments were conducted on a workstation with 64 GB of memory, an Intel Core i7-12700k processor, and an NVIDIA GeForce RTX 3090. In the experiments, we utilized cross-entropy as the loss function and Adam [49] as the optimizer, with an initial learning rate set to 0.001, and trained the model for 500 iterations. During the model training process, we saved and evaluated the best-performing model on the test data. To quantitatively assess the performance of the proposed network, we employed six commonly used metrics: overall accuracy (OA), average precision (AP), average recall (AR), average F1-score (A-F1), mean intersection over union (MIoU), and Kappa coefficient [50].

V. EXPERIMENTS

A. Overall Performance

Due to the limited research on point-based deep learning methods for multispectral LiDAR, some classic point cloud deep learning semantic segmentation algorithms were selected as comparison methods in this study, including PointNet++ [31], GACNet [39], RandLA-Net [51], DGCNN [16], LDGCNN [52], and PointTransformer [53]. These methods have achieved significant results in the point cloud and have been widely used for semantic segmentation tasks with point cloud data.

The average accuracy evaluation metrics table and the segmentation comparison chart of several methods are shown in Table I and Fig. 7, respectively. In general, the network architecture proposed in this study achieved state-of-the-art performance in most metrics. Compared with other models, it improved by 2 to 3 percentage points. Our model accurately delineated the contour shapes of different categories in the multispectral LiDAR point cloud scenes, effectively extracting the boundaries of objects when compared with the ground truth. Due to the sparse point cloud density in the multispectral LiDAR point cloud scenes considered in this study, PointNet++ and GACNet, which are suitable for indoor point cloud scene semantic segmentation, struggled to effectively extract local information in complex urban scenes, making the extraction and recognition of categories such as power lines, cars, buildings, and trees difficult. RandLA-Net and PointTransformer methods involve downsampling operations on the sample point clouds and, while compensating for key point information loss by designing local feature aggregation modules or self-attention layers, still experience some degree of local and global information loss. These two networks exhibited limited recognition capabilities for small impervious ground and grassland regions in the scene, making them inadequate for comprehensive semantic feature modeling of multispectral LiDAR scenes. Due to the similarity of spectral features between impervious ground and grassland categories, as well as the similarity in height between buildings and trees, many of the comparative methods exhibited lower accuracy when classifying these categories. Conversely, graph-based methods such as DGCNN, LDGCNN, and our proposed model performed well in the semantic segmentation of scenes.

Our model showed higher consistency with the ground truth and fewer misclassified outlier points compared with the other two graph-based methods, demonstrating its strong ability to extract geometric features. Our network achieved average OA and MIoU of 94.39% and 86.57% on three test datasets, respectively. Compared with other methods, our approach achieved the best segmentation performance, demonstrating our method's superiority.

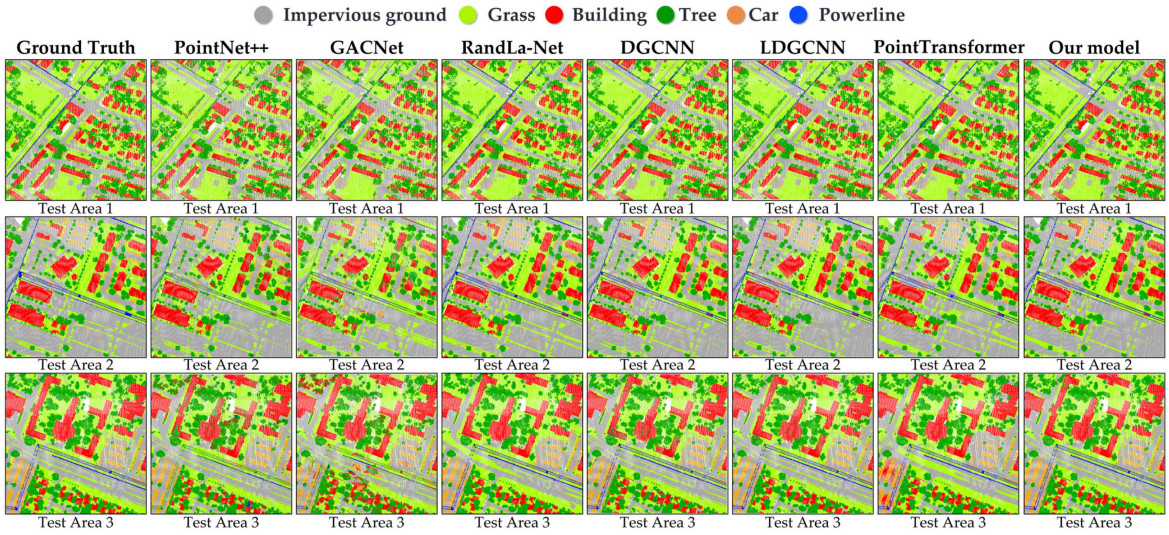


Fig. 7. Results of different segmentation methods in the context of testing scenarios.

TABLE II
SEMANTIC SEGMENTATION RESULTS ON THE MULTISPECTRAL LIDAR DATASET EVALUATED

Networks	OA (%)	MIoU (%)	Impervious ground(%)	Grass (%)	Building (%)	Tree (%)	Car (%)	Power line(%)
PointNet++	90.50	66.94	78.71	78.80	77.44	78.55	55.87	32.35
GACNet	87.07	65.09	77.39	76.76	77.84	81.54	46.12	32.94
RandLA-Net	90.28	85.20	82.83	79.92	92.24	92.25	80.95	83.02
DGCNN	94.27	85.43	85.98	87.33	**92.45**	94.65	77.36	74.84
LDGCNN	93.65	84.27	84.89	85.51	91.70	94.42	75.89	73.22
PointTransformer	88.75	84.57	77.36	78.43	91.87	93.30	**81.70**	**84.80**
Our model	**94.39**	**86.57**	**86.12**	**87.45**	92.26	**94.93**	76.25	82.44
Ranking	**1\7**	**1\7**	**1\7**	**1\7**	**2\7**	**1\7**	4\7	3\7

The bold values represent the highest value of the accuracy evaluation index in each column.

In particular, Table II depicts the intersection over union (IoU) evaluation results of our proposed method for various classes in TestArea3. Within multispectral LiDAR point cloud scenes, our approach exhibits remarkable segmentation accuracy for grass, buildings, trees, and impervious surfaces compared with previous methods. It is worth noting that our method achieves an impressive OA of 95.82%. However, our model's identification of cars and powerlines falls short of ideal performance. This can be attributed to the diverse shapes observed in the powerline category within multispectral LiDAR scenes and the substantial discrepancies in their spectral characteristics and uneven distribution, resulting in segmentation ambiguity. Generally, our model performs exceptionally well in the IoU rankings for several categories, effectively enabling the segmentation and recognition of objects in multispectral LiDAR point cloud scenes.

B. Specific Scenario

As shown in Fig. 8, to better showcase the model's performance, we selected four typical subscenes from three test sets. We conducted a comparative analysis to highlight the distinctions between our proposed method and DGCNN and PointTransformer. DGCNN is a graph-based approach, whereas our method was developed based on a variant. On the other hand, PointTransformer is a Transformer-based method. Both of these approaches demonstrate remarkable semantic segmentation accuracy. In Scene A, the area represents a grass region in a parking lot. Compared with DGCNN, our model can extract the grass area more accurately, mainly due to constructing the adjacency autocorrelation matrix in the local neighborhood graph, effectively allocating weights to capture the relevance between neighboring points. Similarly, in Scene B, the area consists of complex impervious ground and grassland. DGCNN's EdgeConV block, in constructing the local graph, needs a fixed number of points in its KNN search, resulting in relatively limited local geometric features that fail to fully reflect the actual characteristics of the impervious ground and grassland in this scene. In Scene C, the area is mainly composed of buildings and trees. The buildings are surrounded and obscured by trees with uneven distribution. Our model performs better segmentation in recognizing edge points among different object types. For example, DGCNN misclassifies several building edges as tree points in this area, struggling to distinguish building points close to trees effectively. Our model's MSFE block successfully integrates neighborhood graph features extracted at multiple scales by LAF-ConV, enhancing the interclass separability of our model. In Scene D, the area represents a parking lot with

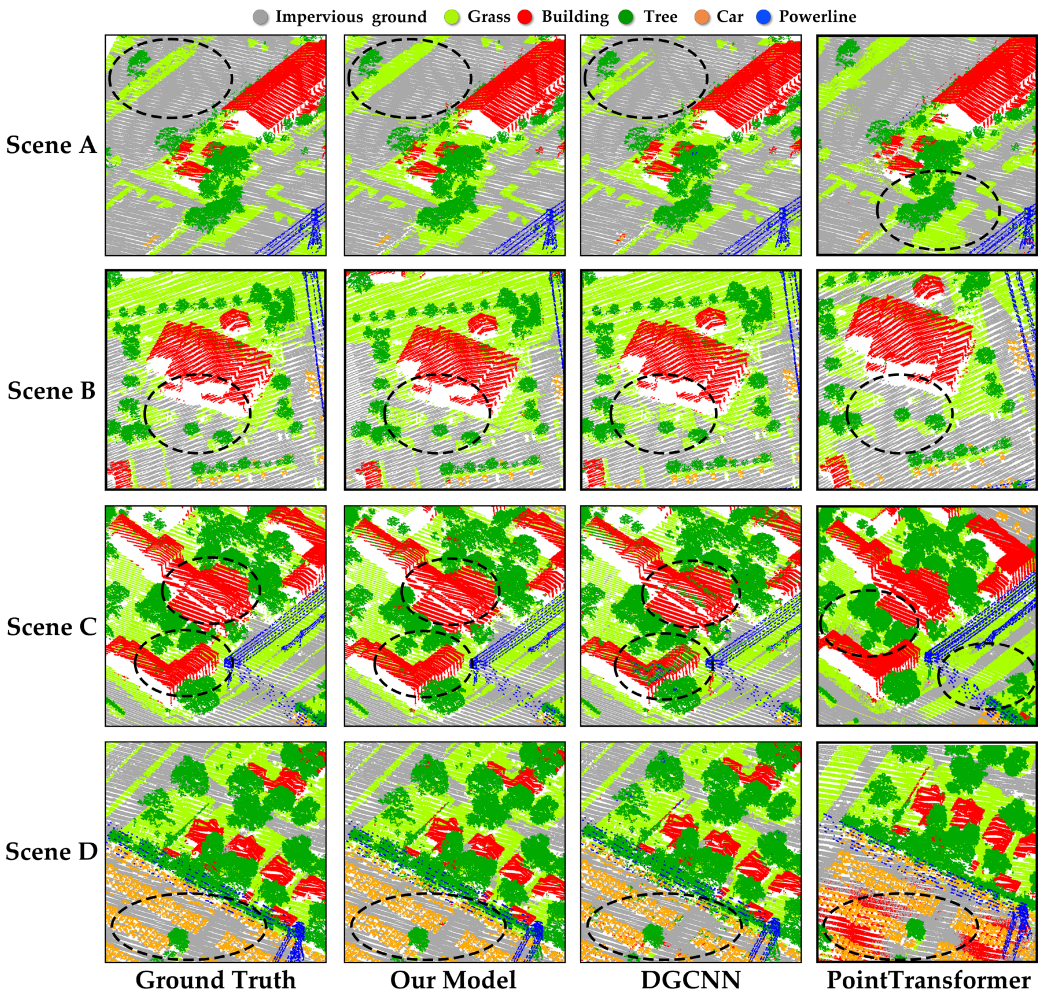


Fig. 8. Comparison of segmentation methods in different scenes (highlighted in black box for differences).

many cars. Similarly, although our model still faces challenges in recognizing and extracting car and impervious ground edge categories in this area, where some boundaries that belong to the impervious ground are misclassified as buildings, our model still outperforms DGCNN. It should be noted that PointTransformer exhibits unsatisfactory performance in differentiating between impervious surfaces and grass, particularly in scenarios A and B, with the issue being most pronounced in scenario C. Similarly, in scenario D, PointTransformer wrongly classifies impervious surfaces in parking lots as building categories. These segmentation disparities are validated in Table II. Due to the lower point density generated by airborne multispectral LiDAR and the point cloud downsampling operations utilized in PointTransformer, RandLa-Net, and similar networks, the learned features fail to capture local semantic information adequately. These detailed visualization results further validate the effectiveness of our model in capturing spatial geometric structures and extracting local features.

C. Analysis of Computational Cost

We compared our proposed method with other networks regarding network parameter count and computational complexity on the multispectral LiDAR dataset. Table III shows that our method has a higher number of parameters compared with PointNet++, GACNet, RandLA-Net, DGCNN, and LDGCNN networks. Additionally, our method also exhibits relatively higher FLOPs (floating-point operations). Despite having a larger parameter count, our method achieves the highest accuracy regarding OA and MIoU within acceptable hardware constraints, striking a good balance between accuracy and computational cost.

On the other hand, when comparing the approaches with and without the MSFE block and GSA block, it is evident that the majority of model parameters are concentrated in the MSFE block. This is primarily due to the increased computational cost of the multiscale local LAF-ConV.

D. Ablation Experiments

1) Effectiveness of Optimized FPS-KNN: Different quantities of training samples from multispectral LiDAR point clouds reflect varying scene semantics, object continuity, and integrity. To validate the effectiveness of the improved FPS-KNN method, we tested the model's performance under different training sample configurations. Considering the limitations of GPU memory, we set the maximum sample quantity to 4096, consistent with the settings of other datasets such as ModelNet40. Additionally, we

TABLE III
FLOPs AND PARAMETERS OF DIFFERENT METHODS ON MULTISPECTRAL LIDAR DATASETS ("M" AND "G" FOR MEGABYTES AND GIGABYTES)

Networks	FLOPs(G)	Parameters(M)	OA(%)	mAcc(%)	MIoU(%)
PointNet++	8.097	0.96	90.50	77.21	66.94
GACNet	5.344	0.97	87.07	76.33	65.09
RandLA-Net	14.547	4.41	90.28	92.25	85.20
DGCNN	8.363	1.04	94.27	90.55	85.43
LDGCNN	11.667	1.27	93.65	**92.52**	84.27
PointTransformer	**20.317**	**5.37**	88.75	92.31	84.57
Ours (without MSFE)	7.443	1.97	93.33	91.28	83.47
Ours (without GSA)	11.179	2.10	93.17	90.11	84.32
Ours (with GAC[39])	8.975	1.77	94.13	89.15	82.43
Ours (with AdaptConv[20])	9.318	2.41	94.38	89.66	83.85
Ours (with EdgeConv[16])	12.315	2.67	**94.49**	92.30	85.32
Ours	13.085	4.72	94.39	91.85	**86.57**

The bold values represent the highest value of the accuracy evaluation index in each column.

TABLE IV
EVALUATION METRICS FOR DIFFERENT PARAMETERS OF THE MSFE BLOCK ON TEST AREA 3

Number of LAF-ConV	Parameter	OA(%)	mAcc(%)	MIoU(%)	Times(ms)
One layers	K=8	94.36	89.73	81.94	**1609.6**
	K=12	94.89	91.21	84.03	1626.5
	K=16	94.95	91.99	84.50	1701.8
	K=20	**95.35**	92.70	**86.39**	1707.2
	K=32	94.76	**93.41**	85.94	1796.1
Two layers	K=8, K=16	94.05	89.96	83.92	**1742.1**
	K=8, K=32	94.77	92.73	84.75	1883.6
	K=12, K=20	95.71	90.53	86.77	1791.0
	K=12, K=32	95.54	**93.62**	87.45	1774.0
	K=16, K=32	95.06	90.36	85.72	1810.2
	K=20, K=32	**95.80**	92.42	**88.12**	1903.6
Three layers	K=8, K=12, K=16	94.21	90.70	81.07	**1814.9**
	K=8, K=16, K=32	94.99	93.57	87.47	1940.2
	K=12, K=16, K=20	95.45	94.94	86.70	1907.6
	K=12, K=16, K=32	94.34	92.57	84.20	1929.5
	K=12, K=20, K=32	**95.82**	**94.98**	**90.38**	1961.4
	K=16, K=20, K=32	95.73	93.96	88.68	2124.8
Four layers	K=8, K=12, K=16, K=20	94.86	91.89	85.27	**2247.4**
	K=8, K=16, K=20, K=32	95.35	92.35	86.07	2457.5
	K=12, K=16, K=20, K=32	**95.60**	**94.17**	**88.37**	2763.8

The bold values represent the highest value of the accuracy evaluation index in each column.

compared the two sample generation methods: random sampling [51] and FPS-KNN [21].

From Table V, as the sample quantity increases, models trained with samples generated using the same sampling strategy exhibit an upward trend in accuracy. When the sample point quantity is set to 4096, the accuracy improves by approximately 0.5% compared with the experiments with 1024 and 2048 points. In other words, the size of the multispectral LiDAR point cloud samples is positively correlated with the accuracy of scene segmentation. On the other hand, when the sample points are all set to 4096, our proposed sampling method achieves the highest semantic segmentation accuracy for multispectral LiDAR compared with Random Sampling and FPS-KNN. This can be primarily attributed to our method's ability to expand the sample quantity, enhance the expression of scene semantics in multispectral LiDAR point clouds, and better match the density of airborne multispectral LiDAR point clouds.

TABLE V
TEST RESULTS OF MS-AMCNN WITH DIFFERENT SAMPLING METHODS ON TEST AREA 3

Sampling methods	Sample size	OA(%)	mAcc(%)	MIoU(%)
Random sampling	1024	94.35	92.58	86.68
Random sampling	2048	94.93	93.54	86.88
Random sampling	4096	95.44	92.87	87.62
FPS-KNN	1024	94.68	92.86	86.56
FPS-KNN	2048	95.41	94.73	88.44
FPS-KNN	4096	95.53	94.01	89.82
Ours	4096	**95.82**	**95.30**	**90.38**

The bold values represent the highest value of the accuracy evaluation index in each column.

2) Effectiveness and Performance of LAF-ConV: To further extract localized geometric and spectral information from multispectral LiDAR point clouds, we propose the LAF-ConV method based on extracting features from the localized adjacency matrix. As presented in Table III, we investigate the effectiveness of the LAF-ConV model by substituting the conventional convolutional kernels in our model with alternative graph convolutional kernels from established literature.

Compared with GAC [39], AdaptConv [20], and EdgeConV [16], our LAF-ConV model demonstrates superior performance in accuracy evaluation metrics, with the highest achieved MIoU. Hence, our proposed LAF-ConV method facilitates improved capture of the local characteristics of multispectral LiDAR by the network, thereby enhancing its robustness.

3) Effectiveness and Performance of MSFE Block: In order to fuse multiscale local geometric features, we designed an MSFE structure to enhance the diversity of local features. In the MS-AMCNN model, the number of LAF-ConV blocks in the MSFE block and the number of KNN neighbor points in each local graph are two key parameters. In our model, the default number of LAF-ConV blocks in the MSFE block is set to 3, corresponding to neighborhood point numbers 12, 20, and 32, respectively. To set these parameters reasonably, we conducted a series of comparative experiments on Test Area 3 to verify the effectiveness of multiscale structure in improving network accuracy. The specific evaluation metrics are shown in Table IV.

For the number of LAF-ConV blocks in the MSFE block, we can observe that when the number of LAF-ConV blocks is set to 1 or 2, the network's performance is poor compared with the MSFE block with three layers of LAF-ConV blocks, especially in terms of MIoU. However, as the LAF-ConV blocks increase to three layers, there is a significant improvement in the fusion of local features and perception capability. This increase also allows for a more diverse range of geometric information among different objects without significantly increasing time consumption. However, when the number of LAF-ConV layers increases to 4, the network's performance decreases with the increase in layers. Therefore, moderately integrating multiple LAF-ConV blocks can increase the model's receptive field. However, excessive stacking may negatively impact the model's performance. Consequently, the optimal setting that balances performance and effectiveness is determined to be three layers.

4) Number of Neighboring Points in LAF-ConV: The number of selected neighboring points in the LAF-ConV block significantly impacts the extraction of local features in multispectral LiDAR data. In the MSFE block, we observed that the network's performance improves as the number of neighborhood points in the LAF-ConV blocks increases beyond 8. However, when the number of neighborhood points reaches 16, there is no significant improvement in network accuracy compared with when the number of neighborhood points is 12. Instead, it increases both time and memory consumption. Since the geometric information constructed by the two different neighborhood point settings may be similar, we chose the smaller option of 12 neighborhood points to minimize the cost. Additionally, when the number of neighborhood points is set to 20 or 32, there is a significant improvement in network performance. Therefore, based on the MSFE block with three layers of LAF-ConV blocks, we select 12, 20, and 32 as the KNN search points for the three local neighborhood graphs, respectively.

5) Effectiveness and Performance of GSA Block: Despite integrating multiscale local geometric features in the MSFE block, we introduce the GSA block to perceive global contextual semantic information more comprehensively. We investigate the impact of different global feature learning methods based on self-attention mechanisms on network accuracy. As shown in Fig. 9, we compare and analyze the self-attention mechanism modules in P-A [54], A-SCN [55], and PCT [56] through experiments. This examination explores their contributions to global feature learning and demonstrates, through experimental evidence, the influence of various global feature learning approaches on model accuracy.

As shown in Fig. 10, when the network does not utilize a self-attention mechanism, although it does not directly affect the segmentation's OA, there are more mis-segmentation results in the model, resulting in an MIoU of only 86.90%. Therefore, it is crucial to employ a self-attention mechanism when fusing local geometric information for global modeling in the MSFE block. When attempting to replace the GSA block with other self-attention mechanisms, the model's performance further declined, with both the OA and MIoU of the segmentation results being lower than the results obtained using the GSA block, which achieved an OA of 94.52% and an MIoU of 90.38%. These results confirm the significant advantage of the transformer-based GSA block in modeling global contextual information.

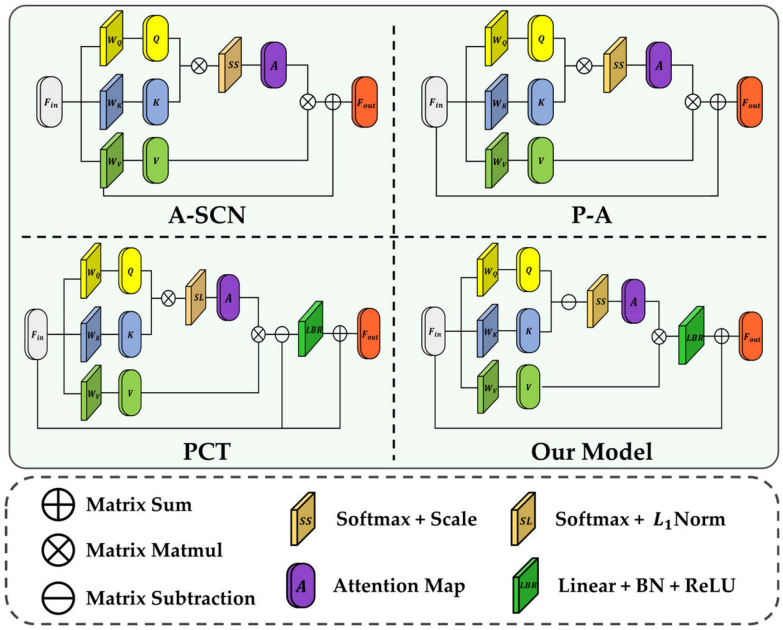


Fig. 9. Architecture of various self-attention mechanisms in 3D point cloud processing.

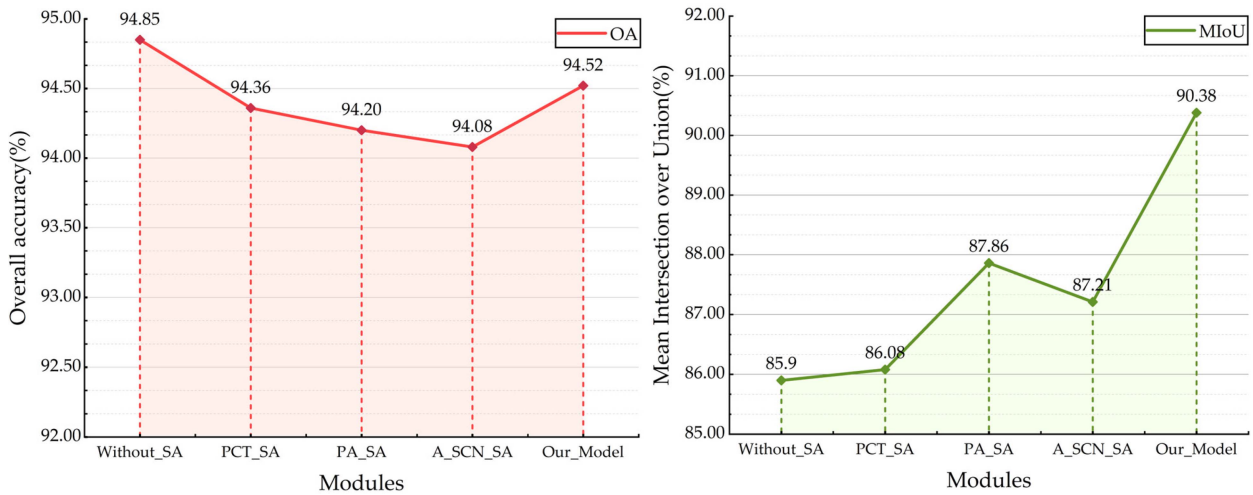


Fig. 10. Comparison of model impact evaluation metrics using different self-attentive mechanisms.

VI. Conclusion

This article proposes an MS-AMCNN for multispectral LiDAR point cloud segmentation scenes. To effectively utilize the local topological structure of the point cloud, we design a local adjacency feature convolution (LAF-ConV) block to better encode and capture local features by establishing a local adjacency self-attention matrix. Subsequently, we introduce an MSFE block for fusing multiscale local neighborhood features and utilize a GSA block based on the self-attention mechanism for global semantic contextual modeling. By applying MS-AMCNN to the Titan dataset, we achieve excellent multispectral LiDAR point cloud segmentation performance with an OA of 94.39%, an MIoU of 86.57%, and a Kappa of 91.80%, demonstrating the exceptional semantic segmentation capability of our network on multispectral LiDAR. Compared with other state-of-the-art models, our approach demonstrates a more comprehensive capability in extracting object outlines for multispectral LiDAR classification. Furthermore, the results of comparative experiments revealed that our model could effectively capture the local features of multispectral lidar point clouds and acquire multiscale contextual semantic information, demonstrating superior performance in classifying edge points of various terrains.

Although this study achieves satisfactory segmentation accuracy on multispectral LiDAR, the network's computational efficiency and memory consumption are relatively high due to the involvement of multiscale processing and self-attention mechanism learning. Therefore, future directions include designing a lightweight, high-precision network to handle

large-scale multispectral LiDAR point cloud scenes. Additionally, this experiment is limited to the Titan multispectral LiDAR dataset, which includes only three spectral channels, far fewer than the corresponding high spectral imaging channels. Further exploration of the potential to effectively classify scenes using the rich spectral information of hyperspectral LiDAR is yet to be developed.

REFERENCES

[1] Y. Cui et al., "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 722–739, Feb. 2022, doi: 10.1109/TITS.2020.3023541.

[2] X. Yue et al., "A LiDAR point cloud generator: From a virtual world to autonomous driving," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2018, pp. 458–464.

[3] N. Haala et al., "Mobile LiDAR mapping for 3D point cloud collection in urban areas—A performance test," *Int. Arch. Photogrammetry, Remote Sens., Spatial Inf. Sci.*, vol. 37, pp. 1119–1127, 2008.

[4] L. Zhang et al., "Large-scale urban point cloud labeling and reconstruction," *ISPRS J. Photogrammetry Remote Sens.*, vol. 138, pp. 86–100, 2018, doi: 10.1016/j.isprsjprs.2018.02.008.

[5] Q. Wang and M.-K. Kim, "Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018," *Adv. Eng. Inform.*, vol. 39, pp. 306–319, 2019, doi: 10.1016/j.aei.2019.02.007.

[6] M. Samie Tootooni et al., "Classifying the dimensional variation in additive manufactured parts from laser-scanned three-dimensional point cloud data using machine learning approaches," *J. Manuf. Sci. Eng.*, vol. 139, no. 9, 2017, Art. no. 091005, doi: 10.1115/1.4036641.

[7] B. Luo et al., "Target classification of similar spatial characteristics in complex urban areas by using multispectral LiDAR," *Remote Sens.*, vol. 14, no. 1, 2022, Art. no. 238, doi: 10.3390/rs14010238.

[8] D. Li et al., "AGFP-Net: Attentive geometric feature pyramid network for land cover classification using airborne multispectral LiDAR data," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 108, 2022, Art. no. 102723, doi: 10.1016/j.jag.2022.102723.

[9] S. Morsy, A. Shaker, and A. El-Rabbany, "Multispectral LiDAR data for land cover classification of urban areas," *Sensors*, vol. 17, no. 5, 2017, Art. no. 958, doi: 10.3390/s17050958.

[10] W. Y. Yan and A. Shaker, "Radiometric correction and normalization of airborne LiDAR intensity data for improving land-cover classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7658–7673, Dec. 2014, doi: 10.1109/TGRS.2014.2316195.

[11] V. Wichmann et al., "Evaluating the potential of multispectral airborne lidar for topographic mapping and land cover classification," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 2, pp. 113–119, 2015, doi: 10.5194/isprsannals-II-3-W5-113-2015.

[12] T.-A. Teo and H.-M. Wu, "Analysis of land cover classification using multi-wavelength LiDAR system," *Appl. Sci.*, vol. 7, no. 7, 2017, Art. no. 663.

[13] L. Matikainen et al., "Object-based analysis of multispectral airborne laser scanner data for land cover classification and map updating," *ISPRS J. Photogrammetry Remote Sens.*, vol. 128, pp. 298–313, 2017.

[14] S. Shi et al., "Land cover classification with multispectral LiDAR based on multi-scale spatial and spectral feature selection," *Remote Sens.*, vol. 13, no. 20, 2021, Art. no. 4118.

[15] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[16] Y. Wang et al., "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019, doi: 10.1145/3326362.

[17] Z. Liang et al., "Hierarchical depthwise graph convolutional neural network for 3D semantic segmentation of point clouds," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 8152–8158.

[18] L. Chen and Q. Zhang, "DDGCN: Graph convolution network based on direction and distance for point cloud learning," *Vis. Comput.*, vol. 39, no. 3, pp. 863–873, 2023, doi: 10.1007/s00371-021-02351-8.

[19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.

[20] H. Zhou, Y. Feng, M. Fang, M. Wei, J. Qin, and T. Lu, "Adaptive graph convolution for point cloud analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4945–4954.

[21] P. Zhao et al., "Airborne multispectral LiDAR point cloud classification with a feature reasoning-based graph convolution network," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 105, 2021, Art. no. 102634, doi: 10.1016/j.jag.2021.102634.

[22] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953.

[23] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5648–5656.

[24] A. Kanezaki, Y. Matsushita, and Y. Nishida, "Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5010–5019.

[25] C. Ma, Y. Guo, J. Yang, and W. An, "Learning multi-view representation with LSTM for 3-D shape recognition and retrieval," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1169–1182, May 2019, doi: 10.1109/TMM.2018.2875512.

[26] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: Group-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 264–272.

[27] X. Wei, R. Yu, and J. Sun, "View-GCN: View-based graph convolutional network for 3D shape analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1847–1856.

[28] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.

[29] G. Riegler, A. Osman Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6620–6629.

[30] T. Le and Y. Duan, "PointGrid: A deep network for 3D shape understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9204–9214.

[31] C. R. Qi et al., "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5105–5114.

[32] J. Li, B. M. Chen, and G. H. Lee, "So-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9397–9406.

[33] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5588–5597.

[34] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu, "PointSIFT: A sift-like network module for 3D point cloud semantic segmentation," 2018.

[35] Y. Ma et al., "3DMAX-Net: A multi-scale spatial contextual network for 3D point cloud semantic segmentation," in *Proc. 24th Int. Conf. Pattern Recognit.*, 2018, pp. 1560–1566.

[36] H. Fang and F. Lafarge, "Pyramid scene parsing network in 3D: Improving semantic segmentation of point clouds with multi-scale contextual information," *ISPRS J. Photogrammetry Remote Sens.*, vol. 154, pp. 246–258, 2019, doi: 10.1016/j.isprsjprs.2019.06.010.

[37] L. Ma, Y. Li, J. Li, W. Tan, Y. Yu, and M. A. Chapman, "Multi-scale point-wise convolutional neural networks for 3D object segmentation from LiDAR point clouds in large-scale environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 821–836, Feb. 2021, doi: 10.1109/TITS.2019.2961060.

[38] D. Li, G. Shi, Y. Wu, Y. Yang, and M. Zhao, "Multi-scale neighborhood feature extraction and aggregation for point cloud segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2175–2191, Jun. 2021, doi: 10.1109/TCSVT.2020.3023051.

[39] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10288–10297.

[40] Z. Du, H. Ye, and F. Cao, "A novel local-global graph convolutional method for point cloud semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2022.3155282.

[41] M. Wei et al., "AGConv: Adaptive graph convolution on 3D point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9374–9392, Aug. 2023, doi: 10.1109/TPAMI.2023.3238516.

[42] Z.-H. Lin, S.-Y. Huang, and Y.-C. F. Wang, "Learning of 3D graph convolution networks for point cloud analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4212–4224, Aug. 2022, doi: 10.1109/TPAMI.2021.3059758.

[43] J. C. Fernandez-Diaz et al., "Capability assessment and performance metrics for the Titan multispectral mapping lidar," *Remote Sens.*, vol. 8, no. 11, 2016, Art. no. 936, doi: 10.3390/rs8110936.
[44] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1534–1543.
[45] J. Wu et al., "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 82–90.
[46] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3D semantic segmentation of point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 716–724.
[47] D. Li et al., "Building extraction from airborne multi-spectral LiDAR point clouds based on graph geometric moments convolutional neural networks," *Remote Sens.*, vol. 12, no. 19, 2020, Art. no. 3186, doi: 10.3390/rs12193186.
[48] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 6000–6010.
[49] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–15.
[50] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.
[51] Q. Hu et al., "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11105–11114.
[52] K. Zhang et al., "Linked dynamic graph CNN: Learning on point cloud via linking hierarchical features," in *Proc. 27th Int. Conf. Mechatronics Mach. Vis. Pract.*, 2021, pp. 7–12, doi: 10.1109/M2VIP49856.2021.9665104.
[53] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16239–16248.
[54] M. Feng et al., "Point attention network for semantic segmentation of 3D point clouds," *Pattern Recognit.*, vol. 107, 2020, Art. no. 107446, doi: 10.1016/j.patcog.2020.107446.
[55] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional ShapeContextNet for point cloud recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4606–4615.
[56] M.-H. Guo et al., "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, pp. 187–199, 2021, doi: 10.1007/s41095-021-0229-5.

Jian Yang received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2017.

He is currently an Associate Professor with the School of Geography and Information Engineering, China University of Geosciences, Wuhan, China. His research interests include radiation transfer model, light detection, laser-induced fluorescence technology, and its application in quantitative monitoring.

Binhan Luo received the B.S. degree in remote sensing and technology in 2022 from China University of Geosciences, Wuhan, China, where he is currently working toward the master's degree in photogrammetry and remote sensing.

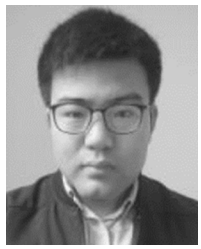
His research interests primarily focus on lidar data processing, multispectral laser radar applications, laser radar point cloud forest data processing, and point cloud deep learning.

Ruilin Gan received the B.S. degree in remote sensing and technology from Chang'an University, Xi'an, China, in 2022. He is currently working toward the master's degree in photogrammetry and remote sensing with the China University of Geosciences, Wuhan, China.

His research interests primarily focus on lidar radar data processing, multispectral laser radar applications, point cloud single tree segmentation, and forest biomass.

Ao Wang received the B.S. degree in remote sensing and technology from Southwest Jiaotong University, Chongqing, China, in 2021. He is currently working toward the master's degree in photogrammetry and remote sensing with the China University of Geosciences, Wuhan, China.

His research interests primarily focus on lidar data processing, point cloud fusion of hyperspectral images, point cloud single tree segmentation, and lidar hardware development.



Shuo Shi (Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2015.

He is currently an Associate Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University. He has been supported by the "Wuhan Morning Light Plan of Youth Science and Technology." He has authored more than 30 peer-reviewed research articles. His research interests include hyperspectral lidar, fluorescence lidar, true-color imaging, target classification, and vegetation quantitative remote sensing.

Lin Du received the Ph.D. degree in condensed matter physics from Wuhan University, Wuhan, China, in 2017.

He is currently an Associate Professor with the School of Geography and Information Engineering, China University of Sciences, Wuhan, China. His research interests include hyperspectral LiDAR detection and its application in vegetation monitoring, especially in the 3-D biochemical parameter reversion and reconstruction in canopy level.