A Deep Neural Network Using Double Self-Attention Mechanism for ALS Point Cloud Segmentation

LILI YU, HAIYANG YU*, AND SHUAI YANG
School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China
Corresponding author: Haiyang Yu (yuhaiyang@hpu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1304402.

ABSTRACT Airborne laser scanning (ALS) point cloud segmentation is an essential procedure for 3D data understanding and applications. This task is challenging due to the unstructured, disordered, and sparse distribution of the point cloud. PointNet++ is a well known end-to-end learning network for point cloud segmentation without fully exploring the local and contextual features, which are less efficient and accurate in capturing the complexity of point clouds. On this basis, we design a novel encoder-decoder network architecture to obtain the semantic features of the ALS point cloud at different levels and achieve a better segmentation effect. The improved local feature aggregation module can merge the deep feature of the point cloud, combining local and global self-attention convolutional networks. It can adaptively explore the inherent semantics feature of points and capture more extensive context information of ALS point cloud, termed DSPNet++. Finally, the conditional random field optimization model can be used to refine the segmentation results. We evaluated the performance of our method on the Vaihingen dataset of the International Society for Photogrammetry and Remote Sensing (ISPRS) and the GML(B) 3D dataset. Experimental results show that our method fully exploits the semantic feature of the ALS point cloud and can achieve higher accuracy. A comparative study with established deep learning models also confirms that our proposed method has outstanding performance in the ALS point cloud segmentation task.

INDEX TERMS ALS, point cloud segmentation, PointNet++ network, encoder-decoder network, local and global self-attention convolutional network.

I. INTRODUCTION

With the continuous development and maturity of depth sensor technologies, such as LiDAR devices technologies, the quality, acquisition efficiency, and effectiveness of 3D point cloud data have improved continuously. As a result, point cloud data are widely used in uncrewed vehicles, smart cities, mapping, and remote sensing fields [1]–[4]. As one of the long-term research topics in computer vision, point cloud segmentation is the basis of 3D scene understanding and analysis for many vision tasks [5]. The irregular, non-uniform, and disordered 3D point clouds data makes automatic and accurate point clouds segmentation challenging [6], [7]. In the field of computer vision and remote sensing, there are three main technologies to obtain point cloud, such as binocular camera [8]. Based on light detection and ranging systems such as LiDAR [9], there is also point cloud acquisition based on RGB-D camera [10]. The characteristics and applications scope of data from different sources vary. LiDAR is a measurement and remote sensing technology, and it can be classified as Airborne LIDAR Scanning (ALS) [11], Terrestrial LIDAR Scanning (TLS) [12], Mobile LIDAR Scanning (MLS) [13], and Unmanned LIDAR Scanning (ULS) systems by their different platforms [14]. Airborne LiDAR scanning ALS operates on an airborne platform. ALS point clouds are more expensive to acquire and usually do not contain spectral information. The International Society for Photogrammetry and Remote Sensing (ISPRS) dataset is a typical ALS benchmark dataset. Multispectral airborne LIDAR is an ALS system that uses different wavelengths to acquire data [7]. Multispectral LiDAR has excellent performs in extracting vegetation and shadows, but the data are not readily available. Our proposed model will experiment on ALS point cloud.

The associate editor coordinating the review of this manuscript and approving it for publication was Cheng Hu.
A. BACKGROUND

To effectively handle the unstructured and irregularly distributed ALS point cloud, state-of-the-art deep neural networks are designed, which can directly manipulate raw point clouds data. Qi et al. [15] first proposed PointNet that learns point feature directly handle unordered points. PointNet used a set of multi-layer perceptron (MLP) and then aggregated them to a global feature used max-pooling. However, PointNet neglected points-to-points spatial neighboring relations, which contained fine-grained information for object segmentation.

After PointNet, many network architectures have been proposed to learn in-depth features hierarchically, to enhance the ability of local model structure, such as PointNet++ and PointCNN [16], [17]. PointNet++ is a layered neural network based on PointNet. The three critical layers of PointNet++ are the sampling layer, the grouping layer, and the PointNet layer. The sampling layer uses farthest point sampling (FPS) to select a set of points from the input points, which define the local centroids. The grouping layer constructs a local area and then extracts features. PointNet becomes a subnetwork of the PointNet++ network, and iterative layers extract the features. The network gradually implements point cloud segmentation by abstracting local locations or feature sets to higher levels of representation. Li et al. [17] proposed a novel architecture, named PointCNN, which aimed to compensate desertion of shape information and variance for pointing ordering when directly applying convolution kernels on the point cloud. Duan et al. [18] introduced a structural, relational network SRN-PointNet++, that uses MLP to learn the features of the relationships between different local structural. Chen et al. [19] presented a novel double self-attention convolutional network, called DAPNet, which can be directly applied to processing LiDAR point clouds by combining geometric and contextual features to generate better segmentation results. Jing et al. [20] propose SE-PointNet++ to classify point clouds, and SE is embedded in PointNet++ to strengthen important channels to increase the saliency of features and classify point clouds better. In the field of ALS point cloud classification, Considering the consistency of neighboring point labels, researchers began to use CRF (Conditional Random Field) and MRF (Markov Random Field) frameworks to integrate contextual information into the classification process. Niemeyer pointed out a situation classification method based on conditional random field (CRF) [21], [22], improving the point’s classification effect.

Recently, multispectral point cloud data has been increasingly used in classification studies. Bakula et al. carried out a maximum likelihood landcover classification task by fusing the texture, height, and intensity information of multispectral LiDAR data [23]. Wichman et al. [24] first execution of data fusion by the nearest neighbor method and then analyzed the spectral patterns of several land overlays to perform point cloud classification. Morsy et al. [25] created separation between land-water and vegetation using three normalized difference feature indices three Titan multispectral LiDAR wavelengths. By fusing spectral information and the geometric features of the point cloud, the feature differences between different types of point clouds are fully considered to fill the gap that only uses geometric features to extract. In the Vaihingen dataset, we will experiment with spectral information and point cloud information fusion.

B. CONTRIBUTION

Based on the above analysis, in this paper, we present DSPNet++ to explore the relationship between adjacent points and context through a layered encoder-decoder structure; to learn representative local features in the encoding stage, and achieve point cloud segmentation. We proposed a method to identify disordered point sets with different densities without designing complex artificial features.

In the encoder stage, an attention mechanism is embedded to enhance the convolutional neural network and adaptively learn the relationship between point clouds. In addition, in the decoder stage, a feature extraction layer is established based on the self-attention mechanism, further aggregate context information and realize global connection more directly. The encoder-decoder structure filters irrelevant and redundant information during low-level propagation and realizes point cloud segmentation end-to-end.

In ALS point cloud segmentation, errors in semantic annotation of point clouds in multicategory regions are targeted to enhance the spatial smoothness after segmentation label assignment. The initial semantic annotation results are optimized using conditional random fields for post-processing to obtain more accurate ALS point cloud semantic segmentation results. We use Conditional Random Fields (CRF) on the ISPRS dataset to refine the multiclassification results. The main contributions of the study include the following:

1) A new encoder-decoder convolutional network called DSPNet++ is proposed. It is a deep neural network using double self-attention. Additionally, DSPNet++ network can be applied directly to the different numbers of points without additional preprocessing.

2) We combine the channel attention and non-local attention mechanism to extract contextual features of a point cloud. By adjusting the sampling strategy and updating the relationship graph, the model realizes the dynamic relationship between points, which improves the model superiority in the point cloud segmentation.

3) Besides theoretical analysis, the proposed method was tested on the Vaihingen 3D dataset of the International Society for Photogrammetry and Remote Sensing (ISPRS) and GML(B) 3D dataset, and compared our model with other models to prove its feasibility and superiority in ALS point cloud segmentation.

II. RELATED WORK

Previous researchers usually use 3-dimensional point clouds for projection transformation, represented by Multi-view Convolutional Neural Network (MVCNN)
based on 2-dimensional projection and VoxNet based on 3-dimensional voxels [26], [27]. MVCNN is a pioneering work based on the multiview method in point cloud processing, which obtains multiview features of a point cloud at a certain angle by the convolutional neural network and gets global description through the maximum pooling of multiview features [27]. However, maximum pooling only retains the most significant elements in a particular view, which leads to information loss. As a result, MVCNN does not effectively utilize the local feature information of each view. Huang et al. [28] presented local descriptors from multiview correspondence to enhance network generalization, and Feng et al. [29] further proposed the GVCNN framework to group different view visual descriptors to effectively utilize the multiview feature relationship. Although the multiview approach is convenient for network implementation, it can also lead to the destruction of geometric relationships between point clouds and the loss of a large amount of crucial information, which will affect the experimental results. In contrast, the voxels can better handle the intrinsic geometric relationships of the data and use 3-dimensional convolutional neural networks to learn point cloud features efficiently. However, the resolution increase of voxels is greatly limited by the sparsity of point cloud data and the computational cost.

In recent years, deep learning has gradually become one of the most technologies in point clouds classification. After PointNet++, Zhao et al. [30] designed PointWeb, a network for information exchange and local feature learning of points by adaptive feature adjustment module. Wu et al. [31] proposed PointConv networks and pointed out a novel and efficient method of calculating the weight function. This method dynamically expands the network and significantly improve the model ability. Jiang et al. [32] designed the PointSIFT module, inspired by the SIFI operator, in which the nearest point features are encoded in eight directions to solve the problem that the K-neighborhood search in PointNet++ may be in one direction.

These methods improved the overall segmentation performance in public datasets, but there are slightly different under segmentation problems. The graph convolutional neural network is a kind of neural network which runs directly on the graph structure, and it can catch the dependence in the graph by the information transmission between the nodes in the graph convolutional neural network. It has been more and more widely used in the field of computer vision. Wang et al. [33] first applied graph convolutional networks to point clouds processing and constructed a dynamic graph convolutional neural network DGCNN, using Edge-Conv operation to extract the features of the center point and neighboring points. The neighborhood graph is constructed by k points, the local features related to the neighborhood of each point are aggregated, and the graph of the k-nearest neighbor algorithm is dynamically updated.

Given that the spatially transformed space introduced by DGCNN increases the network complexity, Zhang et al. [34] improved it by using the DenseNet network structure with different dynamically layered features connected and using multilayer perceptrons instead of spatially transformed networks to reduce the network model. Lu et al. [35] construct neighborhood graphs to describe the relationship between neighborhood points and design filters for extracting neighborhood geometric features. In general, the performance of DGCNN based point cloud classification network is superior, but the number of nodes of DGCNN is related to the number of point clouds, and the network structure is relatively fixed. This is a potential problem in the large-scale processing of point clouds. Chen et al. [36] introduced the attention mechanism into GCNN and built the network model GAP-Net. GACNet used a novel graph attention convolution with learnable kernel shapes to dynamically adapt to the structures of the objects [37]. The attention mechanism was combined with a recurrent neural network to encode RNN and propose a context-based convolutional neural network to obtain local features. RSCNN enhances the shape perception and robustness of the model by weighting the adjacent point features and encoding the geometric relationship between points. However, the data structure of GACNet and RSCNN is costly, which limits the classification capability for complex scenes [38].

III. MATH

Usually, some of the features learned by deep learning methods, such as PointNet++, might be ineffective for ALS point cloud segmentation tasks, resulting in decreased segmentation accuracy. Therefore, this work achieves ALS point cloud labeling by adaptively exploring semantic relations and aggregating contextual information between points.

Specifically, we first introduce an attention based local relation learning module to collect local features, emphasize important channels, and suppress invalid predictions. Then the context aggregation module is designed to obtain the long-term dependency between related points and enhance the ability to distinguish points in feature space. In addition, skip links concatenated local point features in different layers conditionally.

A. OVERVIEW

Fig. 1 shows an overview of the proposed method. We designed an ALS point cloud segmentation method DSPNet++, which mainly realizes the end-to-end class recognition of each point by adding attention mechanism to the PointNet++. The DSPNet++ architecture involves encoder network, decoder network, and skip link concatenations. The encoder extracts more multi-scale features of the point cloud, and the encoder recovers semantically stronger feature representations to generate highly accurate point cloud classification. Skip connection is a technique to improve the performance and convergence of deep neural networks, which alleviates the optimization difficulties caused by nonlinearity by propagating linear components through the neural network layers.
We design our network as a hierarchical encoder-decoder architecture. Fig. 2 illustrates the encoder framework. In the encoder module part, the point cloud sampling layer uses the FPS algorithm and sphere sampling to obtain the neighborhood information of the central sampling point; the channel attention mechanism (ECA_block) module encodes the construction of neighborhood point cloud associations to improve the learning capability of the network model for local point cloud features.

Fig. 3 illustrates the decoder framework. In the decoder module, we design a context-guided aggregation module (Non-local block) to obtain global context information adaptively, capture the long-term dependencies between points, and place it in the last feature propagation layer to enrich the point cloud feature information. Repeating this process so that it can achieve ALS point cloud segmentation.

**B. LOCAL RELATION LEARNING**

In the encoder structure, the channel attention mechanism is used to construct the correlation of adjacent point cloud data, protect the point cloud data structure, and improve the learning ability of local point cloud features of the network model. Channel focus is used to learn the relationship between any channel graph, and then it uses the correlation between channels to update specific channels. Finally, it enhances the specific semantic response under channels through the correlation between channels.

Hu et al. introduced the channel threshold attention mechanism and used the idea of squeeze and excitation (SE) to pool the information in a channel directly, hence ignoring the local information in each channel and enriching the extracted high-level features [38]. However, the application in CNN makes the model more complex, and the convergence speed of the model slower. Fig. 4 shows the Efficient Channel Attention (ECA) module. ECA_block achieves local cross-channel interaction without dimensionality reduction by replacing the fully connected layer with a one-dimensional convolution, reducing the model parameters, and improving segmentation accuracy [39].

Supposing that any feature transformation, including convolution, can express as \(X \in \mathbb{R}^{H \times W \times C}\). The global average pooling is performed on the feature map by squeeze operation to obtain the global compressed feature vector of the feature map, only the channel information is retained, in order to achieve the compression of the global spatial information into the channel descriptors. Through the spatial dimension \(X\) in feature \(H \times W\) to generate statistics \(z \in \mathbb{R}^{C}\), the \(p\)-th element of \(z\) can express as:

\[
z_p = F_{sq}(x_p) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_p(i, j)
\]

The \(z_p\) can interpret as the set of local features, and the statistics \(Z\) do not need dimensionality reduction, and the attention of each channel can obtain in the following way,
with the weight size as a measure of attention.

\[ \rho = \sigma(W_k z) \quad (2) \]

The \( W_k \) contains \( k \times c \) parameters, and the elements in Equation (3) use gradient descent and Adam optimization algorithms to minimize the loss function, which is continuously updated as training progresses. The \( W_k \) defined as:

\[
\begin{bmatrix}
  w_1^1 & \cdots & w_i^1 & 0 & \cdots & 0 \\
  0 & w_2^2 & \cdots & w_i^2 & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & w_C^{C-k+1} & \cdots & w_C^C
\end{bmatrix} \quad (3)
\]

\( \sigma(\bullet) \) represents the sigmoid nonlinear activation function:

\[ \sigma(x) = \frac{1}{1 + e^{-x}} \quad (4) \]

Equation (2) avoids complete independence between different channels, and achieves local cross-channel interaction while ensuring efficiency and effectiveness. The weight of \( z_i \) is calculated only by considering the interrelationship between \( z_i \) and its \( k \) neighboring elements.

\[ \rho_i = \sigma(\sum_{j=1}^{k} w_j^i z_j^i) \quad y_j^i \in \Omega_i^k \quad (5) \]

The \( \Omega_i^k \) denotes the set of \( k \) channels adjacent to \( y_j \). In order to reduce the complexity of the model, all channels can share the same parameters.

\[ \rho_i = \sigma(\sum_{j=1}^{k} w_j^i z_j^i) \quad y_j^i \in \Omega_i^k \quad (6) \]

In Equation (6), \( \rho_i \) is explicitly used to model the correlation between features. The more critical the point cloud feature in the \( i \)-th channel, the larger the corresponding \( \rho_i \), which indicates that the model pays more attention to the channel. Equation (6) can be further simplified the one-dimensional convolution operation.

\[ \rho = \sigma(H^{(k)} \ast z) \quad (7) \]

The \( H \) denotes the one-dimensional convolution kernel, and \( k \) is the size of the corresponding convolution kernel, representing the coverage of local cross-channel interactions.

The size of the one-dimensional convolution kernel can be determined adaptively by a function on the number of channels \( C \), where \( x \) represents the odd number closest to \( x \).

\[ k = \varphi(C) = \left\lfloor \frac{1b(C) + 1}{2} \right\rfloor \text{ odd} \quad (8) \]

ECA_ Block increase of network depth. The categories extracted by the channel with more significant weight are stronger distinguishable, while the features extracted from the channel with smaller weight are primarily irrelevant or similar. Deepen the network model through the channel attention mechanism extracts more representative point cloud features.

**C. CONTEXT-GUIDED AGGREGATION**

ECA block focuses on the relationship between local channels, and adaptively learns the importance of different channel features. However, it is limited to neighborhood information, and it cannot fully utilize global information. Based on these limitations, the Non-local captures the long-range dependencies by directly computing the interactions between two points [40]. Fig. 5 shows the Non-local model. The convolutional layer with non-local operations is placed at the end of the model. This allows us to construct a more decadent layer that combines local and global information.

The continuously stacked convolution RNN operation can obtain and combine global information. It can bring richer hierarchical semantic information for feature propagation.

\[ w_{i,j} = \frac{\exp(Q_i \ast K_j^T)}{\sum_{i=1}^{N} \exp(Q_i \ast K_j^T)} \quad i, j = 1, 2, 3 \ldots N \quad (9) \]

Given an input feature \( F \in \mathbb{R}^{N \times C} \), which represents \( N \) points with \( C \)-channel features, firstly, uses \( 1 \times 1 \) convolution to convert \( F_{\text{int}} \) to different embeddings \( Q \in \mathbb{R}^{N \times C/2} \), \( K \in \mathbb{R}^{N \times C/2} \), \( V \in \mathbb{R}^{N \times C/2} \), which denote query, key, and value, respectively. Secondly \( K \) is transposed and left-multiplied by \( Q \) to obtain the affinity matrix \( M \in \mathbb{R}^{N \times N} \) to which softmax operation is applied to calculate the attention weights \( W \in \mathbb{R}^{N \times C/2} \). The \( w_{i,j} \in W \) measures the similarity between \( i \)-th position and \( j \)-th position in input points. Finally, we perform matrix multiplication on attention weights \( W \) and value \( V \) to aggregate contextual information from value points for each query point,
which gives us the augmented feature $F_{out} \in \mathbb{R}^{N \times C}$, which will feed to the next feature propagation layer in the network.

**D. FEATURE PROPAGATION LAYER (DSPNet++)**

In this section, we introduce the architecture of the DSPNet++. The details of the structures can be seen from Table 1, including data flow, the operation’s primary process, and the number of convolutional kernels.

Firstly, the DSPNet++ has four layers of feature abstraction. In each encoder module, the point clouds are divided by sampling and grouping. We use three convolutional layers for feature extraction, and the number of convolutional layers gradually increases. For deeper feature abstraction layers, the number of convolutional kernels is larger than that of the previous layer. After the convolutional layer, we adopt Max pooling to obtain the global features of each group $L_i$. Secondly, we can get enhanced feature of $L_{L-1}$ by the ECA_block modules. The following four encoder layers are feature propagation. Except for the last layer, concatenated the input to the output of the corresponding feature abstraction. We can get the global feature of $L_3$, $L_4$ by the Non-local block modules. Finally, the classifier performs segmentation which is called label $y$ is compared with the obtained Score of all original points.

**E. CONDITIONAL RANDOM FIELD (CRF)**

In dense urban areas, the correct marking of points is challenging due to the height changes of complex objects and object categories (such as buildings, roads, trees, and low vegetation) in the scene. Conditional Random Field (CRF) can be used to optimize point cloud segmentation. Optimized processing can enhance spatial smoothness, consider the context, information, and reduce the impact of noise. CRF randomly provides a powerful probability framework in segmentation. An undirected graph model connects the neighboring points of the point cloud to construct a point cloud as the vertex to establish an undirected graph model. CRF is used to achieve high precision segmentation of airborne LiDAR point clouds in a complex environment with uneven density and noisy points.

$$E(x) = \sum_i \psi^*_i(x_i) + \lambda \sum_{i<j} \psi_p(x_i, x_j) \quad (10)$$

In Equation (10), the first item on the right side of the Equation is the data item. It consists of the loss value of the

---

**TABLE 1. The detailed operations of the DSPNet++.**

| Layer Name | Input | Output | Operations          | Operations |
|------------|-------|--------|---------------------|------------|
| $L_0$      | $L_1$ | Sample and Grouping | MLP | Max pooling ECA_block |
| $L_1$      | $L_2$ | Sample and Grouping | MLP | Max pooling ECA_block |
| $L_2$      | $L_3$ | Sample and Grouping | MLP | Max pooling ECA_block |
| $L_3$, $L_4$ | $L_5$ | Concat Interpolation Conv Non-local block |
| $L_{L-1}$, $L_2$ | $L_1$ | Concat Interpolation Conv Non-local block |
| $L_{L-1}$, $L_2$ | $L_1$ | Concat Interpolation Conv Non-local block |
| $L_4$, $L_5$ | $L_6$ | Concat Interpolation Conv Non-local block |
| $L_5$ | $L_6$ | Conv Non-local block |
| $L_5$ | $L_6$ | Conv Softmax Cross entropy |

---

VOLUME 10, 2022
cross-entropy loss function in the softmax layer of the model, which is used to calculate each point characteristic. These characteristics are calculated according to the probability of the feature class of point i and its adjacent points.

The second item is a binary energy term representing the relationship between label $x_i$ corresponding to point i and label $x_j$ corresponding to point j. It is mainly used for smoothing the category prediction results, and the parameter shows the contribution of the data and smooth terms to the total energy function. Optimal results are obtained when the value of the energy $E(x)$ function is minimum, and the final post-processing results are obtained by iteratively minimizing the energy function.

### IV. EXPERIMENTAL RESULT

In this section, we provide a comprehensive evaluation to demonstrate the excellent performance of our proposed method on the dataset. We implement all tests in the framework of Python3.5, Tensorflow1.12.1, Keras 2.4.3 and train them with NVIDIA GeForce RTX2080 GPU. The number of sampling points is 1024, the number of neighbors k is 32. In this experiment, we employed the Adam optimizer, and the learning rate set 0.001. The batch_size, decay rate, and max_epoch were 8, 1000, and 81, respectively.

#### A. EXPERIMENTAL METRICS

Overall accuracy (OA), F1-score, mIoU (Mean Intersection over Union) are used as accuracy evaluation indexes and they are used to analyze the ALS point cloud segmentation results. The F1-score is an evaluation index that comprehensively considers the precision and the recall. F1-score and mIoU equation is:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (11)$$

$$\text{mIoU} = \frac{N_{TP}}{N_{TP} + N_{FP} + N_{FN}} \quad (12)$$

$$\text{precision} = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad (13)$$

$$\text{recall} = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad (14)$$

In equation (12), $N_{TP}$ is the number of true positives, $N_{FP}$ is the number of true negatives, $N_{FP}$ is the number of false positives, and $N_{FN}$ is the number of false negatives.

#### B. DATASET AND DATA PREPROCESSING

We use the ALS dataset to evaluate the segmentation performance of the proposed method. The experiments use the Vaihingen3D semantic dataset provided by the International Society for Photogrammetry and Remote Sensing (ISPRS), and the study area located in Vaihingen Baden-Wurttemberg, Germany.

It is located 25 kilometres northwest of Stuttgart, near the river Enz. The vegetation in this area is closely related to the urban environment and overlaps each other. Many types of land objects, such as fences, trees, shrubs, and external walls, have complex and irregular shapes, and buildings are dense and complex. ALS point cloud dataset has a density of 4 points/m$^3$. The ground resolution of multispectral aerial imagery is 8 cm, and each pixel size is 7680 × 13824 pixels. It provides internal and external orientation elements. Precisely, the ALS point cloud divide 1,165,598 points into two areas for training and testing. In total, there are 753,876 training sites and 411,722 test sites. The dataset contains nine different categories, namely, powerless (Pow), low vegetation (Low_veg), impervious surface (Imp_sur), car (Car), fence/hedge (Fe/He), roof (Roof), facade (Facade), shrub (Shrub) and tree (Tree).

| TABLE 2. The number of categories in the ISPRS data set and the proportion (%) of each training and test set category. |
| --- | --- | --- | --- |
| CLASS | Training Set | Test Set |
| --- | --- | --- |
| Pow | 546 | 0.07 | 0.15 |
| Low_veg | 180850 | 24.00 | 98690 | 23.97 |
| Imp_sur | 193725 | 25.70 | 101986 | 24.77 |
| Car | 4614 | 0.61 | 3708 | 0.90 |
| Fe/He | 12070 | 1.60 | 7422 | 1.80 |
| Roof | 152045 | 20.17 | 109048 | 26.49 |
| Facade | 27250 | 3.61 | 11224 | 2.75 |
| Shrub | 47605 | 6.31 | 24818 | 6.03 |
| Tree | 135173 | 17.93 | 54226 | 13.17 |
| All | 753876 | - | 411722 | - |

#### C. ABLATION STUDY

In qualitative evaluating section, the overall accuracy and average F1 Score are used to evaluate the experimental results quantitatively. We conduct ablation experiments to verify the efficacy of each module in our framework.

Firstly, the ECA_block module only uses the layer to perform local learning in the feature propagation layer, and the evaluation of the ECA_block module shows that the architectural accuracy of the ECA_block is significantly improved. The channel attention mechanism is generated through fast
1D convolution, and its kernel size is adaptively determined through the nonlinear mapping of channel size. The size of $K$ represents the range of local cross-channel interaction, that is, how many close neighbors participate in the attention mechanism of a channel. We also set different $K$ values for experiments. By comparing the different $K$ of the model, we can see that the ECA_block module is effective for ALS point cloud segmentation in Table 3. Satisfactory performance achieved by methods (1)-(5), which have better AvgF1 Score and OA Relative to the baseline model (PointNet++ network). The proposed model achieved higher segmentation accuracies with an OA 83.4%, OF 67.8% when the $K = 6$.

Compared with PointNet++, the feature extraction of neighborhood points is improved to produce better segmentation results.

Table 4 shows the segmentation results of our proposed module. In the first row, the evaluation of the ECA_block module by using local learning in the feature propagation layer shows that the accuracy is significantly improved. Compared with the PointNet++ model, OA and F1-score increase by 2.2% and 2.3%, respectively. In addition, the power line increased from 57.9% to 62.4%, with a lower number of power lines, which is often incorrectly categorized as other. SPNet++ can effectively capture the structure and pattern in the local neighborhood.

In the bottom row of Table 4, comparing the performance of DSP++ with PointNet++, the OA and F1-score are improved by 3.6% and 3.0%, respectively. This can be explained as that the combined structure of the local and global encoder-decoder can effectively represent the intrinsic characteristics of the ALS point cloud. Channel attention mechanism can improve local learning in point clouds segmentation by collecting local features and combining with context aggregation module to form a coding and decoding model with better results.

Fig. 7 illustrates the segmentation results. We visualized the low_Veg and Imp_surf features. It can be seen from the results of wrong segmentation that DSPNet++ had significant suppression of error segmentation and compared with SPNet++, the segmentation has better effect at the feature edge.

V. DISCUSSION
A. COMPARISONS WITH OTHER METHODS
Table 5 shows the segmentation results between our method and the different segmentation methods submitted on the
TABLE 4. $F_1$ scores of each category and overall accuracy (OA) of different segmentation methods. (Method 1 SPNet++ represents the use of the ECA_block module in the sampling layer only, and method 2 DSPNet++ represents the use of the model shown in Figure 3.)

| Method  | Pow | Low_veg | Imp_surf | Car | Fe/He | roof | facade | shrub | tree | OA (%) | Avg-$F_1$ (%) |
|---------|-----|---------|----------|-----|-------|------|--------|-------|------|--------|-------------|
| PointNet++ | 57.9 | 79.6 | 90.6 | 66.1 | 31.5 | 91.6 | 54.3 | 41.6 | 77.0 | 81.2 | 65.5 |
| SPNet++  | 62.4 | 80.3 | 90.8 | 67.7 | 31.5 | 92.3 | 53.2 | 41.8 | 80.2 | 83.4 | 67.8 |
| DSPNet++ | 70.6 | 81.2 | 91.6 | 68.8 | 32.4 | 93.4 | 53.3 | 44.0 | 81.7 | 84.8 | 68.5 |

TABLE 5. $F_1$ scores of each category and overall accuracy (OA) of different segmentation methods. (Comparison with the ISPRS competition approach.)

| Method | Pow | Low_veg | Imp_surf | Car | Fe/He | Roof | Facade | Shrub | Tree | OA (%) |
|--------|-----|---------|----------|-----|-------|------|--------|-------|------|--------|
| BJI_W  | 13.8 | 78.5 | 90.5 | 56.4 | 36.3 | 92.2 | 53.2 | 43.3 | 78.4 | 81.5 |
| UM     | 46.1 | 79.0 | 89.1 | 47.7 | 52.2 | 92.0 | 52.7 | 40.9 | 77.9 | 80.8 |
| WhuY2  | 31.9 | 80.0 | 88.9 | 40.8 | 24.5 | 93.1 | 49.1 | 41.1 | 77.3 | 81.0 |
| WhuY3  | 37.1 | 81.4 | 90.1 | 63.4 | 23.9 | 93.4 | 47.5 | 39.9 | 78.0 | 82.3 |
| LUH    | 59.6 | 77.5 | 91.1 | 73.1 | 34.0 | 94.2 | 56.3 | 46.6 | 83.1 | 81.6 |
| RIT_1  | 37.5 | 77.9 | 91.5 | 73.4 | 18.0 | 94.0 | 49.3 | 45.9 | 82.5 | 81.6 |
| DSPNet++ | 70.6 | 84.2 | 94.8 | 69.7 | 38.7 | 93.4 | 53.3 | 45.6 | 80.6 | 84.8 |

ISPRS website, including BJI_W, UM, WhuY2, WhuY3, LUH,RIT_1 (https://www2.isprs.org/commissions/comm2/wg4/results/vaihingen-3d-semantic). To further prove the advantages of our proposed method in this paper, we also compared our method with the recently published point cloud segmentation methods by using deep learning, PointNet [16], PointNet++ [17], PointCNN [18], and PointnetSIF [33]. As shown in Table 7, the overall accuracy, $F_1$-score are used to evaluate the experimental results. For the ISPRS benchmark dataset, the overall accuracy is 84.8%, and the accuracies of impervious surfaces and roof classes are 91.6% and 93.4%, respectively.

As shown in Table 5, compared with the traditional machine learning of LUH, the paper uses the encoder-decoder model to segment the ALS point cloud, which can better achieve the segmentation effect. Compared with the RIT_1 model that segmentation point clouds directly, this paper considers the enhanced multilevel feature dependencies between point clouds. Compared with WhuY3 using feature maps for segmentation, DSPNet++ uses local and global attention mechanisms. Considering the deep features between point cloud features, DSPNet++ is higher by 2.5% than WhuY3 in overall accuracy.

As shown in Table 6, the accuracy of the PointNet network is the lowest, 75.1%, and it lacks neighborhood point cloud information, so the learning ability of the model is poor and needs to be improved. Our method DSPNet++ greatly improves the performance of the baseline (PointNet++). Comparing with the PointnetSIF network, the proposed model DSPNet++ shows a 2.6% increase in accuracy. Achieve the best accuracy in Pow, Low_veg, Imp_surf, Fence_hedge, roof, and tree categories. This result shows that the improvement strategy based on PointNet++ is feasible.

B. VISUALIZATION OF WRONG SCORE RESULTS

Fig. 8 shows the misclassification results and compares them with some current models. The red areas represent the misclassified features, and it is evident that the red areas are decreasing. Circles indicate the diverse region of the car, shrub, roof, and tree classes. Combined with the misclassification graph, it can be seen that low_veg, imp_surf, roof, and three points are classified. The misclassification of the facade is obvious. There is overlap between shrubs and low vegetation, which can easily lead to misclassification. The vegetation near the house is easy to mix differently with each other. The number of power is relatively small. However, our model DSPNet++ can achieve better accuracy.

C. EFFECTIVENESS OF CRF OPTIMIZATION

In order to verify the effectiveness of the CRF optimization algorithm, experiments were performed on the ISPRS benchmark data with and without CRF. The $F_1$ Score of each class and overall accuracy of these two experiments are listed in Table 8. Table 8 shows the result of a different approach.

DSPNet++ means the result of optimization without CRF. DSPNet++-CRF indicates the result of using CRF optimization.

It can be seen in Table 7, the overall accuracy of DSPNet++-CRF is 85.4%, which is higher than that of DSPNet++. The optimization is particularly effective in categories with low initial classification, such as fence, facade, and shrubs, indicating that CRF effectively refined point cloud segmentation results.
TABLE 6. F1 scores of each category and overall accuracy (OA) of different segmentation methods. (Comparison with mainstream deep learning methods.)

| Method          | Pow | Low_veg | Imp_surf | Car  | Fe/He | Roof  | Facade | Shrub | Tree  | OA (%) |
|-----------------|-----|---------|----------|------|-------|-------|--------|-------|-------|--------|
| PointNet [15]   | 43.6| 80.4    | 87.8     | 47.3 | 21.6  | 81.9  | 28.7   | 38.5  | 64.5  | 75.1   |
| PointNet++ [16] | 57.9| 79.6    | 90.6     | 66.1 | 31.5  | 91.6  | 54.3   | 41.6  | 77.0  | 81.2   |
| PointNetSIF [32]| 55.7| 80.7    | 90.9     | 77.8 | 30.5  | 92.5  | 56.9   | 44.4  | 79.6  | 82.2   |
| Point-CNN [17]  | 61.5| 82.7    | 91.8     | 75.8 | 35.9  | 92.7  | 57.8   | 49.1  | 78.1  | 83.3   |
| DSPNet++        | 70.6| 84.2    | 94.8     | 69.7 | 38.7  | 93.4  | 53.3   | 45.6  | 80.6  | 84.8   |

TABLE 7. F1 Score of each class and overall accuracy (OA) with and without CRF optimization on ISPRS benchmark dataset.

| Method          | Pow | Low_veg | Imp_surf | Car  | Fe/He | Roof  | Facade | Shrub | Tree  | OA (%) |
|-----------------|-----|---------|----------|------|-------|-------|--------|-------|-------|--------|
| DSPNet++        | 70.6| 81.2    | 90.6     | 68.8 | 32.4  | 93.4  | 53.3   | 44.0  | 80.6  | 84.8   |
| DSPNet++-CRF    | 71.2| 85.1    | 94.9     | 72.2 | 39.2  | 94.8  | 45.3   | 46.3  | 81.3  | 85.4   |

**FIGURE 8.** The comparison of error maps between the new model and other methods.

**D. PROVING THE GENERALIZABILITY OF MODEL ON GML(B) DATA**

We perform generalization experiments on the GML(B) dataset to further investigate the feasibility of our segmentation method. The GML(B) dataset is also an ALS point cloud dataset acquired by the airborne laser scanning system ALTM 2050 (Optech, Toronto, ON, Canada). In this dataset, four semantic classes are predefined (ground, building, tree and low vegetation) with 3D coordinates on each point. We chunked the training data by $10 \times 10$ m, overlapping each block by $1/3$, normalizing each block according to the coordinates of the centroids, and testing the data in the same way, but without overlap between blocks.

Table 8 shows the segmentation result. The result of the local area is shown in Fig 8. SPNet++ means PointNet++ improves the encoder and uses ECA_block in the propagation layer, increasing the accuracy value by 3.2%. Relative to that of the PointNet++ network, the accuracy of our proposed method improves by 3.9%, while the AvgF1 Score improves by 20.2%. Fig 9, the ground objects calibrated by
TABLE 8. F1 Score of each class and overall accuracy (OA) of different segmentation methods.

| Method          | Ground (%) | Building (%) | Tree (%) | Low veg (%) | OA (%) | AvgF1 (%) |
|-----------------|------------|--------------|----------|-------------|--------|-----------|
| PointNet++ [15] | 96.8       | 68.2         | 95.1     | 33.4        | 93.3   | 73.4      |
| SPNet++         | 97.7       | 75.7         | 94.8     | 38.0        | 95.1   | 76.6      |
| DSPNet++        | 98.8       | 86.5         | 95.4     | 42.1        | 97.2   | 82.6      |

FIGURE 9. Segmentation results of PointNet++ and our proposed method in the local region.

the black circle, shows that PointNet++ can easily misclassify Low_average into Tree, and our model can effectively suppress the misclassification results closer to the actual value, which proves the effectiveness of the model. It further proves that the combination of local and global attention mechanisms can significantly enhance the model’s ability to learn target features. Single local learning has a slight improvement on the accuracy model. Combining ECA_block with Non_local can effectively improve the segmentation performance of the network and obtain better segmentation results.

VI. CONCLUSION

This study proposes a new end-to-end encoder-decoder model for ALS point cloud segmentation, and designs a new feature extraction module based on a self-attention mechanism, termed DSPNet++. This model can fully combine ECA_blocks, and Non-local blocks can effectively capture broader contextual features. In addition, our DSPNet++ directly processes the original ALS point cloud. It avoids the information loss during data preprocessing, which fully learns the local and global significance structure of the ALS point cloud to ensure that important information is transmitted as efficiently as possible, which improves the ALS point cloud segmentation task. The CRF optimization algorithm ensures the consistency of point-by-point semantic label assignment, improves the experimental results, and increases segmentation accuracy. The study also confirms the feasibility and effectiveness of DSPNet++ for the ALS point cloud segmentation task by conducting experiments on two different data sets. We demonstrate the superiority of the method through extensive experiments.

REFERENCES

[1] B. Yang, W. Xu, and Z. Dong, “Automated extraction of building outlines from airborne laser scanning point clouds,” *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 6, pp. 1399–1403, Nov. 2013.
[2] L. T. Ene, E. Næsset, T. Gobakken, E. W. Mauya, O. M. Bollandsås, T. G. Gregoire, G. Ståhl, and E. Zahabu, “Large-scale estimation of aboveground biomass in miombo woodlands using airborne laser scanning and national forest inventory data,” *Remote Sens. Environ.*, vol. 186, pp. 626–636, Dec. 2016.
[3] P. E. Axelsson, “DEM generation from laser scanner data using adaptive TIN models,” *Int. Arch. Photogramm. Remote Sens.*, vol. 33, no. B4, pp. 110–117, 2000.
[4] M. Hebel, M. Arens, and U. Stilla, “Change detection in urban areas by object-based analysis and on-the-fly comparison of multi-view ALS data,” *ISPRS J. Photogramm. Remote Sens.*, vol. 86, pp. 52–64, Dec. 2013.
[5] A. S. Antonarakis, K. S. Richards, and J. Brasington, “Object-based land cover classification using airborne LiDAR,” *Remote Sens. Environ.*, vol. 112, no. 6, pp. 2988–2998, Jun. 2008.
[6] M. Weinmann, B. Jutzi, and C. Mallet, “Feature relevance assessment for the semantic interpretation of 3D point cloud data,” *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 5, no. W2, p. 1, Oct. 2013.
[7] J. Niemeyer, F. Rottensteiner, and U. Soergel, “Contextual classification of LiDAR data and building object detection in urban areas,” *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 152–165, Jan. 2014.
[8] G. Hu, Z. Zhou, J. Cao, and H. Huang, “Highly accurate 3D reconstruction based on a precise and robust binocular camera calibration method,” *IET Image Process.*, vol. 14, no. 14, pp. 3588–3595, Oct. 2020.
F. Lu, H. Peng, H. Wu, J. Yang, X. Yang, R. Cao, L. Zhang, R. Yang, and B. Zhou, “InstanceFusion: Real-time instance-level 3D reconstruction using a single RGBD camera,” Comput. Graph. Forum, vol. 39, no. 7, pp. 433–445, Nov. 2020.

F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, “The ISPRS benchmark on urban object classification and 3D building reconstruction,” ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci., vol. I-3, pp. 293–298, Jul. 2012.

T. Hackel, N. Savinov, L. Ladicky, D. J. Wegner, K. Schindler, and M. Pollefeys, “Semantic3D-Net: A new large-scale point cloud classification benchmark,” ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., vol. IV-W1, pp. 91–98, May 2017.

J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in Proc. IEEE/CVF Int. Conf. Vis. (ICCV), Oct. 2019, pp. 9297–9307.

M. Wieser, M. Hollaus, G. Mandlbarger, P. Gila, and N. Pfeifer, “ULS LiDAR supported analyses of laser beam penetration from different ALS systems into vegetation,” ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci., vol. III-3, pp. 233–239, Jun. 2016.

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. (CVPR), Jul. 2017, pp. 652–660.

C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2017, arXiv:1706.02413.

Y. Li, B. Bu, M. Sun, W. Wu, X. Di, and B. Chen, “PointCNN: Convolution on X-transformed points,” in Proc. Adv. Neural Inf. Process. System., vol. 31, 2018, pp. 820–830.

Y. Duan, Y. Zheng, J. Lu, J. Zhou, and Q. Tian, “Structural relational reasoning of point clouds,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 949–958.

L. Chen, W. Chen, Z. Xu, H. Huang, S. Wang, Q. Zhu, and H. Li, “DAPNet: A double self-attention convolutional network for point cloud semantic labeling,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 14, pp. 9680–9691, 2021.

Z. Jing, H. Guan, P. Zhao, D. Li, Y. Yu, Y. Zang, H. Wang, and J. Li, “Multispectral LiDAR point cloud classification using SE-PointNet++,” Remote Sens., vol. 13, no. 13, p. 2516, Jun. 2021.

Y. Lu and C. Rastussenn, “Simplified Markov random fields for efficient semantic labeling of 3D point clouds,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Oct. 2012, pp. 2690–2697.

J. Niemeyer, F. Rottensteiner, U. Soergel, and C. Heipke, “Hierarchical higher order CRF for the classification of airborne LiDAR point clouds in urban areas,” Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci., vol. XLI-B3, pp. 655–662, Jun. 2016.

K. Bakula, P. Kupidura, and L. Jekowicki, “Testing of land cover classification from multispectral airborne laser scanning data,” ISPRS-Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci., vol. XLII-B7, pp. 161–169, Jun. 2016.

V. Wichmann, M. Bremer, J. Lindenberger, M. Rutzinger, C. Georges, and F. Pettrini-Monteferri, “Evaluating the potential of multispectral airborne LiDAR for topographic mapping and land cover classification,” in Proc. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., vol. II-3/W5, 2015, pp. 113–119.

S. Morsi, A. Shaker, and A. El-Rabbany, “Multispectral LiDAR data for land cover classification of urban areas,” Sensors, vol. 17, no. 5, p. 958, Apr. 2017.

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., Aug. 2015, pp. 945–953.

D. Maturana and S. Scherer, “VoxNet: A 3D convolutional neural network for real-time object recognition,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Dec. 2015, pp. 922–928.

H. Huang, E. Kalogerakis, S. Chaudhuri, D. Ceylan, V. G. Kim, and E. Yumer, “Learning local shape descriptors from part correspondences with multiview convolutional networks,” ACM Trans. Graph., vol. 37, no. 1, pp. 1–14, Jan. 2018.

Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, “OVCNN: Group-view convolutional neural networks for 3D shape recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 264–272.