DE-Net: A Dual-Encoder Network for Local and Long-Distance Context Information Extraction in Semantic Segmentation of Large-Scale Scene Point Clouds

Zhipeng He, Jing Liu, and Shuai Yang

***Abstract*—Semantic segmentation of large-scale point clouds is essential for applications such as autonomous driving and high-definition mapping. However, this task remains challenging due to the imbalanced distribution of categories in large-scale point cloud data and the similarity in local geometric structures. Most current deep learning–based methods concentrate on designing local feature extraction modules while neglecting the significance of long-distance contextual information. Nevertheless, this contextual information is crucial for accurate object segmentation in large-scale scenes. To address this limitation, we propose a dual-encoder segmentation network called DE-Net. DE-Net effectively learns both the local and long-distance contextual information for each point to achieve accurate point segmentation. DE-Net consists of two main components: dual-encoder modules (DEMs) and gradient-aware pooling modules (GAPM). DEMs extract local geometry and long-distance contextual information for each point using positional and trigonometric encoding to distinguish complex geometric features. GAPMs aggregate global information effectively using dual-distance and *xy* gradient information. In addition, a prediction jitter module was introduced during training to address the issue of class imbalance and improve the network's prediction results. The experimental results on three public benchmarks demonstrate that DE-Net outperforms existing state-of-the-art methods, achieving mean intersection over union scores of 83.5%, 61.8%, and 63.9% on Toronto-3D, WHU-MLS, and S3DIS datasets, respectively.**

***Index Terms*—Deep learning, dual-encoder, semantic segmentation, 3-D point cloud.**

I. Introduction

With the rapid development of LiDAR sensors, point cloud data processing has received extensive attention. As a fundamental task of point cloud processing, semantic segmentation has been widely used in the fields of autonomous driving [1], [2], scene-level understanding [3], and robotics [4]. Unfortunately, the disorder, irregularity, and class imbalance of large-scale point clouds have posed challenges for algorithm design.

Existing research methods can be divided into projection-based, voxel-based, and point-based according to their principles. Among them, the projection-based method converts the point cloud into a 2-D image (such as multiview projection [5], spherical projection [6], and bird's eye view projection [7]. Afterward, pixel features were extracted by convolutional neural networks. However, occlusion and pixel distortion of the projected image limit the further development of this method. The voxel-based approach first voxelised the point cloud and further processed it through 3-D convolutional neural networks [8], [9], [10]. This method can solve the disorder of point clouds. However, the low resolution of voxels can result in a loss of details for small objects. High voxel resolution increases memory consumption and computational costs. Different from these methods, point-based methods are able to work directly with point clouds. As a pioneering work, PointNet [11] generates an alignment matrix of point clouds and features through the T-Net network. It uses the multilayer perceptrons (MLP) to learn the features of points and finally adopts maximum pooling to aggregate global features. However, this method does not consider the extraction of local features of point clouds. To effectively extract the local features of point clouds, some studies use the K-nearest neighbors algorithm to construct neighbors to express local context information [12], [13], such as RandLA-Net [14], SCF-Net [15], and Stratified Transformer [16]. RandLA-Net uses random sampling to downsample point clouds, thus significantly reducing computational costs and memory consumption. In addition, they designed the local spatial encoding module to effectively preserve the useful features of the neighborhood. SCF-Net [15] constructs a *Z*-axis rotation invariance in the polar coordinate system to represent the point cloud and learns the local context features. At the same time, the neighborhood location and volume ratio are used to learn the global context information of the point cloud. Stratified Transformer [16] confines the Point Cloud Transformer within a nonoverlapping local window and effectively uses the Transformer for feature extraction. The

Received 12 April 2024; revised 1 July 2024; accepted 19 August 2024. Date of publication 27 August 2024; date of current version 16 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 42471418 and Grant 42001284 and in part by the Jiangsu Natural Science Foundation under Grant BK20200722. *(Corresponding author: Jing Liu.)*

Zhipeng He and Jing Liu are with the Key Laboratory of Virtual Geographic Environment (Nanjing Normal University), Ministry of Education, Nanjing 210023, China, and also with the Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China (e-mail: 211345011@njnu.edu.cn; jingliugeo@njnu.edu.cn).

Shuai Yang is with the 31682 Troop of People's Liberation Army, Lanzhou 730000, China (e-mail: yangshuai1984726@163.com).

Digital Object Identifier 10.1109/JSTARS.2024.3450708

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

excellent results of these methods show that efficient local feature extraction and global context aggregation play a crucial role in point cloud semantic segmentation [17]. However, there are still some challenges to existing methods.

First, most algorithms do not consider the similarity of local features in large-scale scenes, such as poles, tree trunks, fences, walls, and grounds. Nevertheless, the similarities in the local geometry of different classes of objects are not rare. Therefore, the local similarity in the local features of learning may lead to the misclassification of different categories. A simple solution is to introduce more fine-grained local information [15], such as distance and gradient information in the *xy* plane, which can aid in complex feature discrimination. However, for objects such as poles and tree trunks, the feature and gradient information provided by the *xy* plane is limited. It is worth noting that the long distance of contextual information is beneficial for feature discrimination. This observation motivates us to add an attention mechanism to the local feature extraction process. At present, the popular Transformer mechanism can well represent the long-distance context information of point clouds [18], [19], [20]. Nevertheless, memory consumption and computational costs grow exponentially when directly applying the Transformer mechanism to large-scale scene point clouds.

Second, existing methods of global information aggregation are subject to information loss and boundary points. Previous studies [12], [21] use max pooling or mean pooling to forcibly aggregate global information, resulting in the loss of a portion of the information. This is because the max pooling highlights the foreground features and loses the background information, whereas the mean pooling retains the overall information but ignores the feature differences. On the other hand, the literature [14], [22] uses an attention mechanism to aggregate global information, which can preserve the overall information and feature differences. Yet, the attention mechanism still has some limitations and it is easily affected by boundary points. Especially in the first few point cloud downsampling stages, the abnormal boundary points in the small neighborhood will seriously affect the output of the convolutional layer.

Third, the semantic segmentation of point clouds in large-scale scenes is usually affected by the long-tail distribution. This is because the frequency and size of objects in each category in the scene are inconsistent, resulting in an inconsistent number of points for each category. This issue is also present with widely used point cloud benchmark datasets, such as Toronto-3D [23], WHU-MLS [24], and S3DIS [25]. This brings challenges to the semantic segmentation of point clouds in large-scale scenarios. Common methods to address this challenge are oversampling the tail category and downsampling the head category, or balancing the importance of the sample (using a weighted loss function that imposes a greater penalty on the class with a smaller sample size) [26], [27]. However, with the current state-of-the-art technology, a drop in performance can still be observed in the tail category.

In this article, we focus on solving the problems mentioned above. First, to solve the problem of local feature similarity, we propose a dual-encoder module (DEM). This is a novel point local feature coding module, which can effectively combine local neighborhood and long-distance context information, thus improving the robustness of the network to local feature similarity. In particular, for each 3-D point, we explicitly express the local geometry through position encoding and introduce trigonometric functions to encode long-distance contextual information to help discriminate complex similar local geometry. Second, to improve the aggregation ability of global information, we propose a gradient attention pooling module (GAPM) to mitigate the impact of boundary points, which is a novel attention pooling module that aggregates contextual information by introducing double-distance and *xy* gradient information. Third, to overcome the category imbalance and overfitting of the head class, we introduce the prediction jitter module (PJM) in the training phase to make the network generate more robust prediction results. This jitter is dependent on the number of points in each category. This is manifested in the fact that a smaller jitter is assigned to the tail category and a larger jitter is assigned to the head category. In this way, the feature region differences between categories are narrowed down, resulting in a more balanced representation. In summary, our main contributions are as follows.

1) We proposed a novel DEM that can learn the local feature representation and the long-distance context information representation of each 3-D point. Unlike previous methods, we encode point clouds in two different forms.
2) We proposed a GAPM that enhances the capacity of the global context information for each 3-D point by exploiting relative distances, surface distances in the *xy* plane, and *xy* gradient information.
3) We have introduced the PJM during the training phase, which enables the mitigation of feature discrepancies between different classes.
4) Comprehensive experiments on three benchmarks show that we proposed that DE-Net can be used for semantic segmentation of large point clouds with limited memory, and we have achieved excellent segmentation results.

II. Related Works

A. 3-D Point Cloud Descriptors

The 3-D point cloud descriptor encodes characteristic information such as coordinates, color, and intensity, making it the most successful method for representing 3-D point clouds. Existing literature categorizes 3-D point cloud descriptors into local-based, global-based, and hybrid-based descriptors. The local feature descriptor is constructed by selecting the center point to create a local reference. The point cloud is then divided into multiple local areas, and the relationship between the points in each area is encoded [12], [21]. For instance, RandLA-Net [14] introduces a local spatial coding unit to represent and convey the local geometry. This method describes the geometric features of a point based on its positional relationship with adjacent points. On the other hand, global-based feature descriptors encode geometric information across the entire point cloud [19], [20], [28]. The point cloud is first divided into voxels or individual cells. Then, the features of each unit are accumulated to form feature descriptors. These descriptors are mainly used for tasks, such as point cloud classification and shape detection. PCT [29] is inspired by the Transformer in natural language

processing, which puts the coordinates of each point and the corresponding features into a coordinate-based coding module and uses the attention mechanism to extract the global features of the point cloud. Hybrid-based descriptors are used to fuse local and global feature coding information for a comprehensive representation of point cloud features. DS-Net [30] recently proposed a dual-path architecture for encoding local and global information, sequentially separating and fusing the two forms of expression to achieve sufficient cross-scale interaction. In point cloud processing, PVCNN [31] uses two modes of points and voxels to encode the original points, and the relationship between points is modeled through 3-D convolution. Different from these methods, we construct two encoding methods for each 3-D point to achieve effective extraction of local features and long-distance contexts.

B. Point Feature Aggregation

Point cloud local feature aggregation plays a crucial role in the semantic segmentation of point clouds in large-scale scenes. [12], [14], [15], [22] achieves accurate point cloud understanding through efficient local feature extraction and global feature aggregation. Therefore, many subsequent works have been carried out from multiple perspectives, and different local space coding modules have been designed to effectively obtain local space information. For example, Li et al. [32] designed a multiscale convolutional kernel point module to extract initial geometric features from coarse to fine. Xiang et al. [33] added relative angle features and other features as inputs to the model to effectively extract geometric information from the point cloud. Li et al. [34] deeply integrated fine-grained point features and coarse-grained voxel features to enhance feature representation. However, a single pooling method may result in the loss of foreground or background feature information. To solve this problem [12], [21], Qiu et al. [22] proposed a hybrid local feature aggregation method, which combines the salient features obtained by the maximum pooling and the detailed information of the average pooling. In addition, Zhang et al. [35] calculated the relative distance based on the geometric distance and Z coordinate for the point product and concatenated it with features to calculate the attention weight for feature aggregation. Xu et al. [17] designed an adaptive connection unit to connect features in series according to the feature volume ratio as a weight and weighted the updated features to enhance feature perception.

C. Class Imbalance

The long-tail distribution of point cloud data is a common problem. This makes it difficult for the model to learn the features of the tail class, which leads to the deterioration of the segmentation performance of the tail class [36]. Previous studies can be categorized into two methods: data augmentation and weighted loss function. The data-augmentation-based approach [26], [37], [38] increases the number of unbalanced class samples in the training data. This allows the model to have more frequent tail class samples during training, resulting in a more balanced class distribution. Although this approach works well for small datasets, it is likely to overfit the model to the tail class. The method based on the weighted loss function [12], [39] is to set the category weights according to the inverse proportion of the sample size and give a greater penalty to the tail category. For example, Lin et al. [40] proposed the focal loss. An additional factor is introduced on top of the cross-entropy loss function. Its role is to adjust the weights based on the size of the category sample. The weight is reduced for categories with a large sample size, whereas increased for categories with a small sample size. Lee and Kim [26] balanced the class losses by dynamically weighting the loss values and mixed the methods of data augmentation.

III. Methodology

In this section, we first present the problem statement. We then discuss our proposed DE-Net and its three key modules: the dual-encoder module (DEM), the gradient-aware pooling module (GAPM), and the PJM.

Problem statement: Given a point cloud $\mathrm{P} = \{(\mathrm{P_i})\}_{\mathrm{i=1}}^{\mathrm{N}}$ with N points $P_i = (X_i, Y_i, Z_i) \in \mathcal{R}^3$, the goal is to predict the semantic label $L = \{(l_i)\}_{i=1}^{N}$ for each point. We train a deep learning model, denoted as DE-Net, with parameters $h = h(\cdot|\theta)$, by minimizing the disparity between the predicted semantic labels for point coordinates and the ground truth.

A. Network Architecture

Our DE-Net follows the network architecture of RandLA-Net [14], as shown in Fig. 1. In the encoding layers, the DEM, GAPM, and random subsampling are stacked between layers. In the decoding layers, we use an oversampling approach for each encoding layer, facilitating the propagation of features between encoding and decoding layers through skip connections. Finally, the softmax function is used to generate predicted probabilities for each class at each point. It is worth noting that our baseline framework is the same as RandLA-Net, except that it is based on a Pytorch reimplementation instead of TensorFlow. The following sections will provide further information on the specifics of the DE-Net structure.

B. Dual-Encoder Module

The key idea of the method is to use two branches to encode the point cloud in parallel. The positional encoding expresses the local features of the point cloud through absolute position, relative position, relative distance, *xy* plane distance, and *xy* gradient information. The trigonometric encoding is used to extract long-distance contextual features from global point clouds. Each point is embedded in a high dimension to enrich the input features and the K-nearest neighbors are utilized to reflect the overall picture of each local community. Exponential transformations are applied to assess feature similarity, and long-distance contextual features are then propagated between points via a shared MLP. We describe the dual-coding module in detail in the following paragraphs.

1) Positional Encoding Branch: We adopt the local spatial encoding from RandLA-Net to explore point features in the

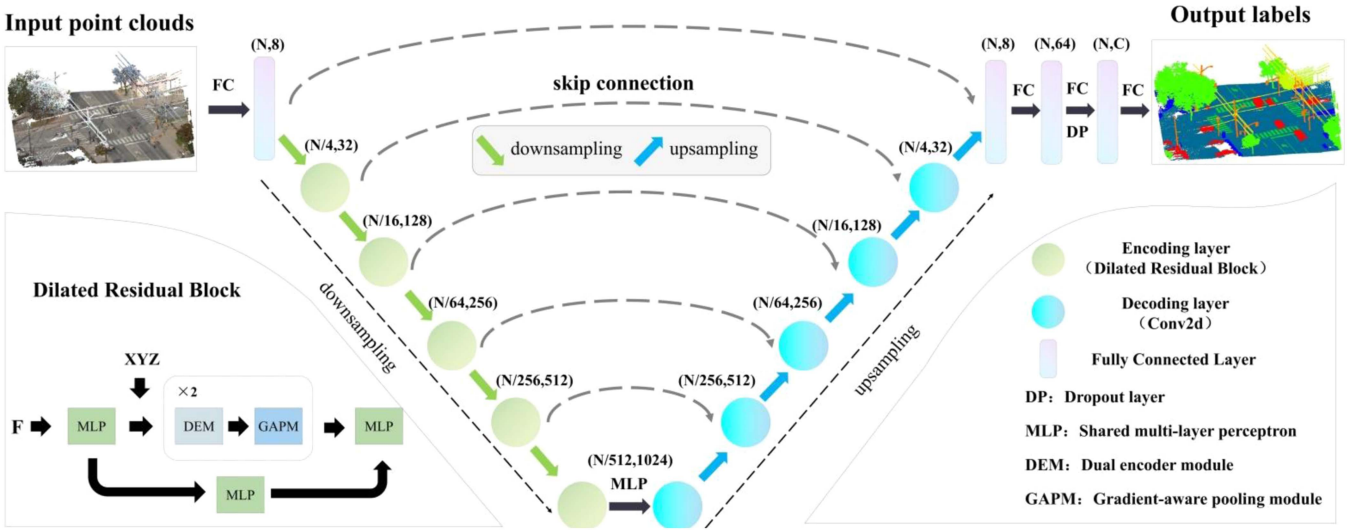

Fig. 1. Overall structure of DE-Net.

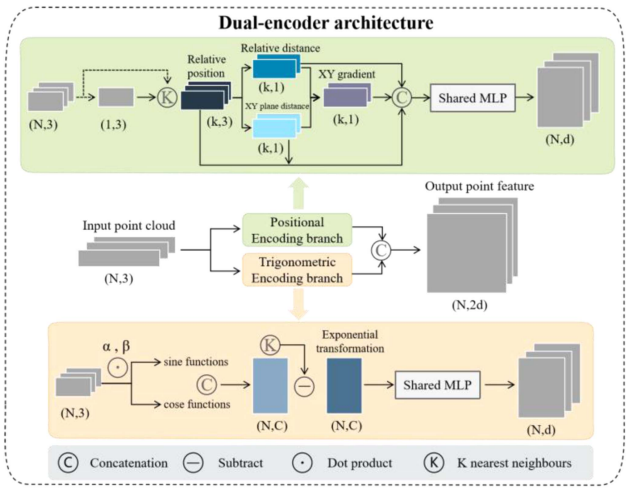

Fig. 2. Dual-encoder module.

local space. The local spatial encoding consists of four components: point coordinates, relative point coordinates, coordinates of K-nearest neighbors, and relative distances. To enhance the expressive power of local point features, we introduced more fine-grained geometric information between the central point and its neighbors based on the local spatial encoding. This includes *xy* plane distance features and *xy* gradient information features, as shown in Fig. 2.

The feature of *xy* plane distance refers to the distance between two points projected onto the *xy* horizontal plane. It is acquired by utilizing the coordinates of the central point and its K-nearest neighbors, which can be denoted as follows:

$$XY_{\text{dis}} = \sqrt{\left(X_c - X_i^k\right)^2 + \left(Y_c - Y_i^k\right)^2} \tag{1}$$

where (X_c, Y_c) are the coordinates of the central point P_k, and (X_i^k, Y_i^k) are the coordinates of the K-nearest neighbor point P_k.

The *xy* gradient information feature is acquired by calculating the relative distance between the central point and the K-nearest neighbor points, alongside the distance on the *xy* plane, which can be denoted as follows:

$$\nabla_{xy} = \frac{Z_i^k}{\text{dis}_i^k} \cdot \frac{X_i^k + Y_i^k}{XY_{\text{dis}}} \tag{2}$$

where ∇_{xy} is the *xy* gradient, (X_i^k, Y_i^k, Z_i^k) are the coordinate values of the K-nearest neighbor points P_k, dis_i^k are the relative distance between the central point and its K-nearest neighbors, and XY_{dis} are the *xy* plane distance between the central point and its K-nearest neighbor points. Finally, we extract local spatial features, which can be denoted as follows:

$$\begin{aligned} F_i^g = \ & \text{MLP}\left(P_i \oplus P_i^k \oplus \left(P_i - P_i^k\right) \oplus \|P_i \right. \\ & \left. - P_i^k\| \oplus XY_{\text{dis}} \oplus \nabla_{xy}\right) \end{aligned} \tag{3}$$

where F_i^g represents the local neighborhood features obtained by position encoding, P_i and P_i^k are the absolute *x*-*y*-*z* positions of points, $(P_i - P_i^k)$ is the relative position of the point coordinates, $\|P_i - P_i^k\|$ is the relative distance of the point coordinates, $\oplus$ is the concatenation operation, and MLP is multilayer perceptron.

2) Trigonometric Encoding Branch: To capture the global features of the input point cloud, we use trigonometric functions to encode point coordinates and extract long-distance context information through a weight-shared MLP. Specifically, each point is mapped to a smooth range using trigonometric functions, and the features encoded with sine functions and cosine functions are stacked, increasing the point feature dimension from 3 to C (where $C = 24$) to enrich the input features. Subsequently, relative features between the central point and points in its neighborhood are computed using K-nearest neighbors. Exponential transformation is applied to assess feature similarity. Smaller feature differences are mapped to exponential values close to 1, whereas larger differences are mapped to values close to 0. Finally, further processing is conducted through a weight-shared

MLP. Our triangulation encodes point clouds in the same way as Point-NN [41], which can be denoted as follows:

$$F_i^{xyz}\ (2J) = \sin\left(\frac{\beta \cdot xyz}{\alpha^{6J/C}}\right) \tag{4}$$

$$F_i^{xyz}\ (2J+1) = \cos\left(\frac{\beta \cdot xyz}{\alpha^{6J/C}}\right) \tag{5}$$

where β and α control the magnitude and wavelengths, C represents the output feature dimension, and here we set $C = 24$. J represents the feature channel index, xyz represents the input point coordinates, and F_i^{xyz} represent the high-dimensional features of the trigonometric function of each point i. Then, for each point i, the relative feature distance within its local neighborhood is calculated, and an exponential transformation is applied to the relative feature distance to evaluate the feature similarity, which can be denoted as follows:

$$d_i^k = e^{-\left(\left|F_i^{xyz} - \left(F_i^{xyz}\right)^k\right|_{\text{mean}}\right)} \tag{6}$$

where F_i^{xyz} represents the high-dimensional characteristics of the trigonometric function of each point i, and $(F_i^{xyz})^k$ represents the high-dimensional characteristics of the trigonometric function of each point i in the K neighborhood of point i. d_i^k represents the relative characteristics between the central point i and the point k in the neighborhood. Then, a weight-shared MLP is applied to the calculated d_i^k to automatically learn long-distance contextual information

$$F_{iG}^k = \text{MLP}\left(d_i^k\right) \tag{7}$$

where F_{iG}^k represents the long-distance context information learned by the MLP.

C. Gradient Attention Pooling Module

The previous method [14] performs softmax on the input feature map to calculate the attention weight and multiplies the attention weight and the feature map to obtain the global context information, which is easily affected by boundary points. GAM [42] adds gradient information to the extraction of local neighborhood features, which significantly improves the ability of local feature extraction. In particular, inspired by Hu et al. [42], a GAPM was designed to aggregate global contextual information in our study. Unlike them, we believe that distance is a better representation of the correlation between points. So, we constrain the influence of boundary points by double distance and *xy* gradient information, rather than relative position vectors, as shown in Fig. 3.

We use the GAPM during the coding phase. The *xy* plane distance, relative distance, and *xy* gradient information used here are obtained from the center point and the nearest neighbor point during the position encoding process. Specifically, we stack *xy* plane distances, relative distances, *xy* gradient information, and input feature maps in feature channels. Then, 1×1 convolution is used for secondary feature extraction to realize the constraints of *xy* gradient and distance information on feature aggregation. Finally, the attention weights are calculated by softmax and

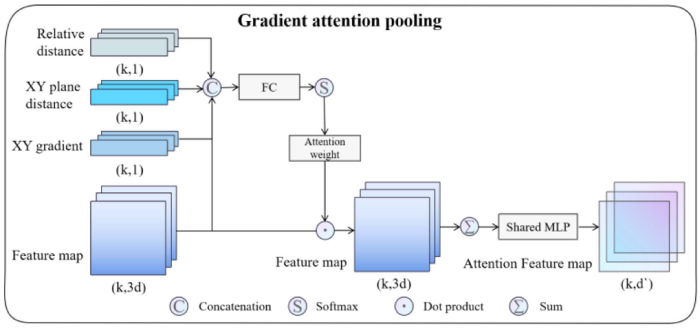

Fig. 3. Gradient attention pooling module.

multiplied by the input feature map. The details are as follows:

$$F_i = F_i^g \oplus F_{iG}^k \oplus F_i^k \tag{8}$$

$$A_{wi} = \text{softmax}\left(\text{Conv2}d\left(F_i \oplus \text{distance} \oplus XY_{\text{dis}} \oplus \nabla_{xy}\right)\right) \tag{9}$$

$$f_i = \text{MLP}\left(\sum_{k=1}^{K}\left(F_i \odot A_{wi}\right)\right) \tag{10}$$

where F_i^k are the features for each point, such as RGB. F_i^g represents the local neighborhood features obtained by position encoding, F_{iG}^k represents the long-distance context information obtained by trigonometric function encoding, and F_i represents the local features of each point after the concatenation of local features, long-distance context information, and color features. A_{wi} represents the attention weight of each point obtained by the convolutional layer and softmax function, f_i represents the global feature description of the point cloud, $\oplus$ is the element concatenation, and $\odot$ is the element product.

D. Prediction Jitter Module

The DEM and the GAPM can help the network to fully learn the fine-grained features of the point cloud. However, due to the category imbalance in the number of samples of each category in large-scale scenes, the network may overfit the features of the head category and compress the features of the tail category in a narrow region, resulting in overconfidence in the prediction results of the head category. Inspired by BLV [43], we introduced the PJM in the training phase to narrow the gap between the feature regions of different classes and prevent the network from overfitting the head class. Fig. 4 shows a more intuitive understanding of the PJM.

Specifically, in the training phase, we obtained the number of points for each category based on the training data, and calculated a vector Q with the same shape as the predicted value using the following equation:

$$Q_k = \log\frac{\sum_{i=0}^{C-1} q_i}{q_k}. \tag{11}$$

The value of each category in the vector Q is inversely proportional to the scale of the number of category points. However, simply scaling the prediction probability based on a fixed class

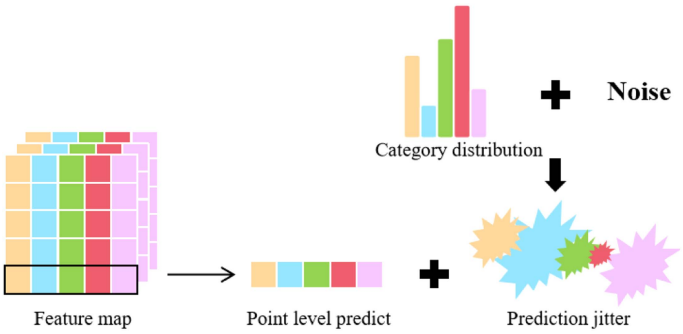

Fig. 4. Prediction jitter module.

frequency can lead to overfitting. Therefore, we take a jitter with a mean of 0 and a variance of four for a fixed class frequency, and add the result of the jitter and the predicted probability, which can be denoted as follows:

$$PJ_k = \frac{Q_k}{\max_{i=0}^{C-1} Q_i} \cdot |\delta(\sigma)| \tag{12}$$

where q_k is the number of points for class k, Q_k indicates an inverse change in the number of category points, and $\delta(\sigma)$ is a Gaussian distribution with a mean of 0 and a variance of σ. Finally, we add the jittered result to the predicted probabilities, which can be denoted as follows:

$$\text{pred}_k^i = \text{pred}_k^i + PJ_k \tag{13}$$

where pred_k^i is the predicted probability for each class in each point and PJ_k are the predicted jitter for each class. By adding this jitter, we aim to achieve a feature representation space that is balanced across classes. It is worth noting that the PJM is discarded during the inference phase to assist the network in obtaining more robust predictions.

IV. Experiments and Analysis

We evaluated the performance of our proposed DE-Net on three large-scale point cloud datasets including Toronto-3D, WHU-MLS, and S3DIS. In addition, we analyzed the three proposed modules and hyperparameters to verify the effectiveness of the designed modules.

A. Description of Datasets

1) Toronto-3D Dataset: Toronto-3D [23] is a street-level point cloud dataset collected by 32-line LiDAR. It was collected from the streets of Toronto, with a total length of about 1 km, and provides nine semantic annotations. It contains 78.2 million points, with an average density of 1000 points per square meter. Each point contains coordinates, color, and intensity information. In the experiment, we used only the coordinates and color information. The data are divided into four parts: L001, L002, L003, and L004. We follow the official recommendation to train DE-Net with L001, L003, and L004 as the training set, and report the test results on L002.

2) WHU-MLS Dataset: WHU-MLS [24] is a point cloud dataset of the outdoor environment of an urban environment, which is collected by mobile LiDAR. It contains more than 200 million points and more than 5000 instance objects, covering categories such as ground, dynamic targets, vegetation, and poles. The dataset has a total of 38 scenarios, of which 28 are used for training and 10 are used for testing. According to the existing research [24], the point cloud segmentation is divided into 17 classes. The coordinates, normal vectors, and intensity information of each point are used as the input of the network.

3) S3DIS Dataset: S3DIS [25] is a large-scale indoor scene dataset collected through Matterport. It consists of 272 rooms (e.g., halls, meeting rooms, classrooms, etc.) in six large-scale areas. Each point contains coordinates and color information. There are 13 semantic categories. We used areas 1–4 and area 6 as training and tested on area 5.

B. Implementation Details and Metrics

In our experiments with the three datasets, we used a grid of rules for the first layer of downsampling. For the Toronto-3D dataset, the first grid size was set to 0.06 m, the hyperparameters were set to 500 iterations, training, batch size was 4, validation batch size was set to 16, and epochs 100. For the WHU-MLS dataset, the first grid size was set to 0.16 m due to its large outdoor scene. For the S3DSI dataset, the first grid size was set to 0.04 m with similar hyperparameters. The initial learning rate for the three datasets was 0.01. This experiment was based on the Pytorch framework and used a Tesla V100 GPU (16GB) to train and update the weights of the network with the help of the Adam optimization algorithm.

To evaluate the segmentation results, we adopted the intersection over union (IoU), mean IoU (mIoU), and overall accuracy (OA) as evaluation metrics. The mathematical formulas of IoU, mIoU, and OA can be expressed as follows:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{14}$$

$$\text{mIoU} = \frac{1}{C} \sum_{i=0}^{C-1} \text{IoU}_i \tag{15}$$

$$\text{OA} = \frac{1}{N} \sum_{i} \text{TP}_i \tag{16}$$

where N is the total number of points, C is the number of segmentation classes, TP is a true positive, FP is a false positive, and FN is a false negative.

C. Experiment Results and Analysis

1) Evaluation on Toronto-3D: Table I shows the quantitative results of DE-Net and other state-of-the-art networks on L002. Experimental results show that DE-Net is superior to all other networks. Notably, DE-Net OA is 3.2% higher than RandLA-Net, and mIoU is 1.7% higher. In addition, DE-Net excelled in specific categories, ranking first in the road mark, car, and fence categories. In addition, DE-Net achieved the top three positions in seven out of eight categories, further validating DE-Net's superior performance.

Fig. 5 shows a visual comparison of DE-Net and RandLA-Net on Toronto-3D. The purple box indicates the difference between RandLA-Net and DE-Net. From the visualization results, the

TABLE I
QUANTITATIVE RESULTS OF DIFFERENT APPROACHES ON TORONTO-3D (L002)

RGB	Method	OA	mIoU	Road	Road mrk	Natural	Buil.	Util. Line	Pole	Car	Fence
No	PointNet++ [12]	92.6	59.5	92.9	0.0	86.1	82.2	60.9	62.8	76.4	14.4
	DGCNN [44]	94.2	61.7	93.9	0.0	91.3	80.4	62.4	62.3	88.3	15.8
	MS-PCNN [45]	90.0	65.9	93.8	3.8	92.5	82.6	67.8	71.9	91.1	22.5
	KPConv [46]	95.4	69.1	94.6	0.1	96.1	91.5	87.7	81.6	85.7	15.7
	MS-TGNet [23]	95.7	70.5	94.4	17.2	95.7	88.8	76.0	73.9	94.2	23.6
Yes	RandLA-Net [14]	94.4	81.8	96.7	64.2	96.9	94.2	**88.0**	77.8	93.4	42.9
	KPConv[46]	94.8	81.5	97.4	63.9	97.0	**94.4**	87.8	83.8	94.1	33.7
	ResDLPS-Net [47]	96.5	80.3	95.8	59.8	96.1	90.9	86.8	79.9	89.4	43.3
	BAAF-Net [22]	94.2	81.2	96.8	67.3	96.8	92.2	86.8	82.3	93.1	34.0
	BAF-LAC [48]	95.2	82.2	96.6	64.7	96.4	92.8	86.1	**83.9**	93.7	43.5
	MFA [49]	97.0	79.9	96.8	70.0	96.1	92.3	86.3	80.4	91.5	29.4
	LACV-Net [50]	97.4	82.7	97.1	66.9	97.3	93.0	87.3	83.4	93.4	43.1
	NeiEA-Net [17]	97.0	80.9	97.1	66.9	**97.3**	93.0	87.3	83.4	93.4	43.1
	SFL-Net [51]	97.6	81.9	**97.7**	70.7	95.8	91.7	87.4	78.8	92.3	40.8
	Baseline	96.3	79.5	95.5	56.3	97.2	92.5	87.1	81.2	89.0	37.8
	DE-Net (ours)	**97.6**	**83.5**	97.4	**70.7**	96.8	93.2	87.5	80.6	**94.4**	**47.4**

Note: The bold numbers indicate the approach having the highest accuracy.

TABLE II
QUANTITATIVE RESULTS OF DIFFERENT APPROACHES ON WHU-MLS

Method	mIoU	Tree	Nd. way	Building	Box	Light	Tel. pole	Mun. pole	Low veg.	Board
		roadway	Rd. mark.	vehicle	pedestrian	trff. light	detector	fence	wire	
PointNet++ [12]	41.1	83.3	42.0	72.7	6.6	59.1	30.8	7.8	33.1	13.9
		80.0	29.5	76.7	38.9	25.0	11.0	56.3	32.7	
PointConv [52]	46.4	85.6	48.9	73.5	28.2	59.7	35.7	20.0	32.4	16.0
		82.0	30.6	76.2	53.8	28.7	27.6	52.6	36.5	
RandLA-Net [14]	47.1	86.8	55.6	76.3	32.4	61.2	43.1	15.2	37.4	18.1
		80.6	29.3	74.9	50.7	21.6	22.2	55.8	39.2	
Han's method [24]	52.8	84.5	**58.4**	77.1	45.4	**71.8**	**49.9**	**26.5**	34.1	20.2
		83.6	38.1	79.1	60.8	31.0	31.3	57.9	47.2	
Baseline	58.6	91.5	49.9	88.3	**59.4**	64.5	11.2	20.6	32.9	45.1
		88.4	48.6	**91.3**	77.6	58.9	**54.4**	**64.4**	50.1	
DE-Net (ours)	**61.8**	**92.2**	52.3	**88.6**	56.4	66.0	2.3	23.4	**39.3**	**54.8**
		90.9	**53.3**	90.4	**79.1**	**71.9**	52.0	64.2	**73.2**	

Note: The bold numbers indicate the approach having the highest accuracy.

segmentation effect of DE-Net on road, road mark, pole, car, and fence is better than that of RandLA-Net, especially the segmentation effect of road mark, car, and fence. This is because the dual-coding module can combine local features and long-distance context information to distinguish regions with similar local structures.

2) Evaluation on WHU-MLS: Table II shows the quantitative results of DE-Net and other networks on the dataset on WHU-MLS. Experimental results show that the mIoU of DE-Net is 3.2% higher than that of baseline. In addition, DE-Net ranked first in 9 out of 17 categories, further demonstrating DE-Net's superior performance in complex segmentation tasks.

Fig. 6 shows a visual comparison of DE-Net and RandLA-Net on WHU-MLS. For better visualizing differences, we select some categories for visualization. The black box indicates the difference between baseline and DE-Net. From the visualization results, DE-Net has a good effect on wire, low vegetation, road mark, board, and Nd. way category. This is because the Baseline only makes use of local features and is susceptible to local similarities and boundary points. However, DE-Net improves the segmentation accuracy of these categories through the DEM and the GAPM.

3) Evaluation on S3DIS: Table III shows the quantitative results of DE-Net in S3DIS area 5. It can be seen that DE-Net obtained a better mIoU of 63.9% compared to other methods. The most significant improvement is in the sofa category, which is 15.1% higher than RandLA-Net. Experimental results show that the segmented IoU for similar objects (such as tables and chairs) is increased by 6.6% and 3.4%, respectively, compared to the baseline.

Fig. 7 shows a visual comparison of DE-Net and RandLA-Net on semantic segmentation of the S3DIS dataset. The black circles represent the difference between the different networks and the ground truth. From the visualization results, DE-Net

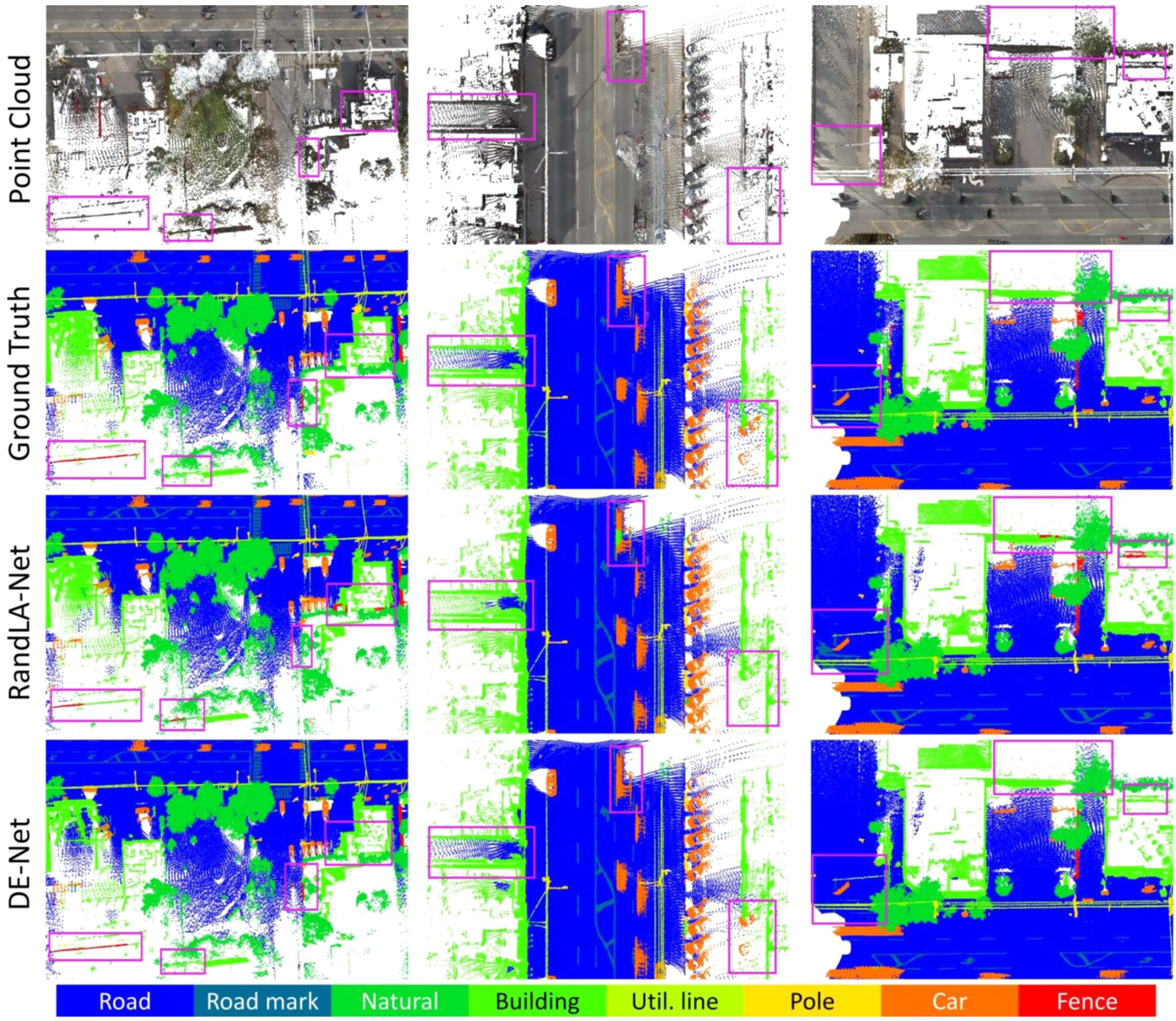

Fig. 5. Comparison of visualization results on the Toronto-3D dataset.

TABLE III
QUANTITATIVE RESULTS OF DIFFERENT APPROACHES ON S3DIS

Method	OA	mIoU	Ceil.	Floor	Wall	Beam	Col.	Wind.	Door	Table	Chair	Sofa	Book.	Board	Clut.
PointNet[11]	-	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	59.0	52.6	5.9	40.3	26.4	33.2
PointCNN[21]	85.9	57.3	92.3	98.2	79.4	0.0	17.6	22.8	**62.1**	74.4	80.6	31.7	66.7	62.1	**56.7**
SPGraph[53]	86.4	58.0	89.4	96.9	78.1	0.0	**42.8**	48.9	61.6	**84.7**	75.4	69.8	52.6	2.1	52.2
HPEIN[54]	87.2	61.9	91.5	98.2	81.4	0.0	23.3	**65.3**	40.0	75.5	87.7	58.5	67.8	65.6	49.4
RandLA-Net[14]	87.2	62.4	91.1	95.6	80.2	0.0	24.7	62.3	47.7	76.2	83.7	60.2	**71.1**	65.7	53.8
PCT[29]	-	61.3	92.5	98.4	80.6	0.0	19.4	61.6	48.0	76.6	85.2	46.2	67.7	**67.9**	52.3
DPFA-Net[55]	88.0	55.2	93.0	**98.6**	80.2	0.0	14.7	55.8	42.8	72.3	73.5	27.3	55.9	53.0	50.5
LGGCM[56]	88.8	63.3	**94.8**	98.3	**81.5**	0.0	35.9	63.3	43.5	80.2	**88.4**	68.8	55.8	64.6	47.8
Baseline	87.1	61.7	92.0	94.1	80.3	0.0	20.8	59.0	46.9	73.7	78.3	72.1	68.2	65.1	51.7
DE-Net (ours)	88.0	**63.9**	93.3	96.9	81.0	0.0	25.1	62.8	47.1	80.0	81.7	**75.3**	69.6	66.3	53.1

Note: The bold numbers indicate the approach having the highest accuracy.

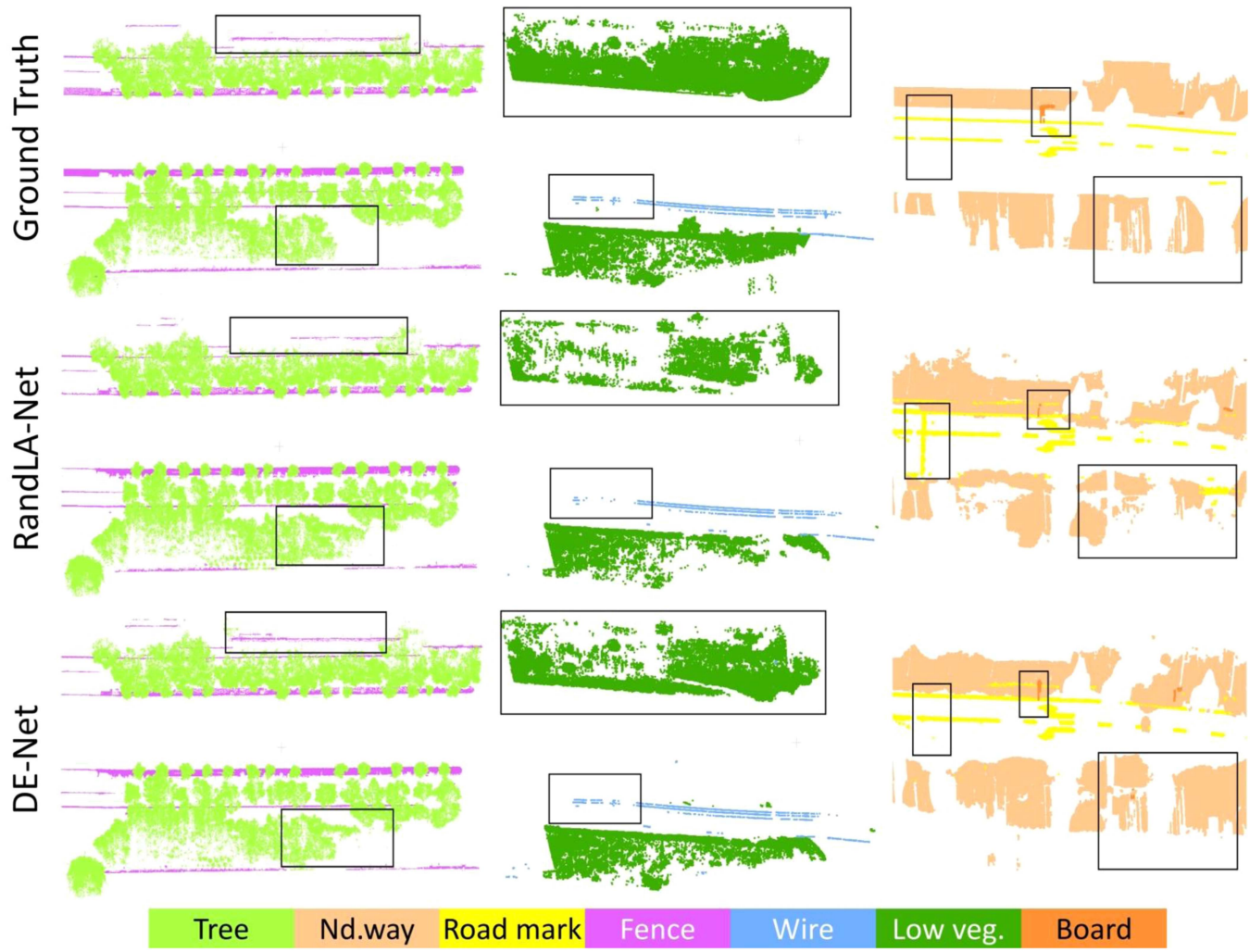

Fig. 6. Comparison of visualization results on the WHU-MLS dataset.

TABLE IV
QUANTITATIVE RESULTS OF DIFFERENT POINT CLOUD CODING MODULES ON TORONTO-3D

Method	OA	mIoU	Road	Road mrk	Natural	Buil.	Util. Line	Pole	Car	Fence
Baseline (B0)	96.3	79.5	95.5	56.3	97.2	92.5	87.1	81.2	89.0	37.8
P	97.3	80.7	97.0	69.4	96.9	92.2	86.1	80.1	92.0	32.1
T	97.0	80.3	96.5	64.7	96.3	93.4	87.3	76.6	92.2	35.6
P+T	97.1	81.9	96.6	61.7	96.8	93.2	87.8	80.7	94.8	44.0

P denotes the positional coding branch, T denotes the trigonometric coding branch, and P+T denotes the dual-encoder module.

has a better effect on chair, sofa, table, and floor segmentation than RandLA-Net. This indicates that DE-Net also has good robustness to point cloud semantic segmentation in large-scale indoor scenes.

D. Ablation Experiments

To further verify the effectiveness of the DE-Net proposed in this article, we designed an ablation experiment to analyze the impact of each module in the DE-Net on the network performance. All of the following experiments were performed on the Toronto-3D (RGB) dataset.

1) Branch Ablations: We first compared the positional coding branch (P-branch), the triangular coding branch (T-branch), and the dual-coding module (P+T). In these three experiments, we replaced the encoder part of the baseline network with three branches to prevent the influence of other modules. As shown in Table IV, the OA, mIoU, and most types of IoU of dual-encoded modules are superior to single-branch networks. It is worth noting that the DEM reduces the segmentation accuracy of the

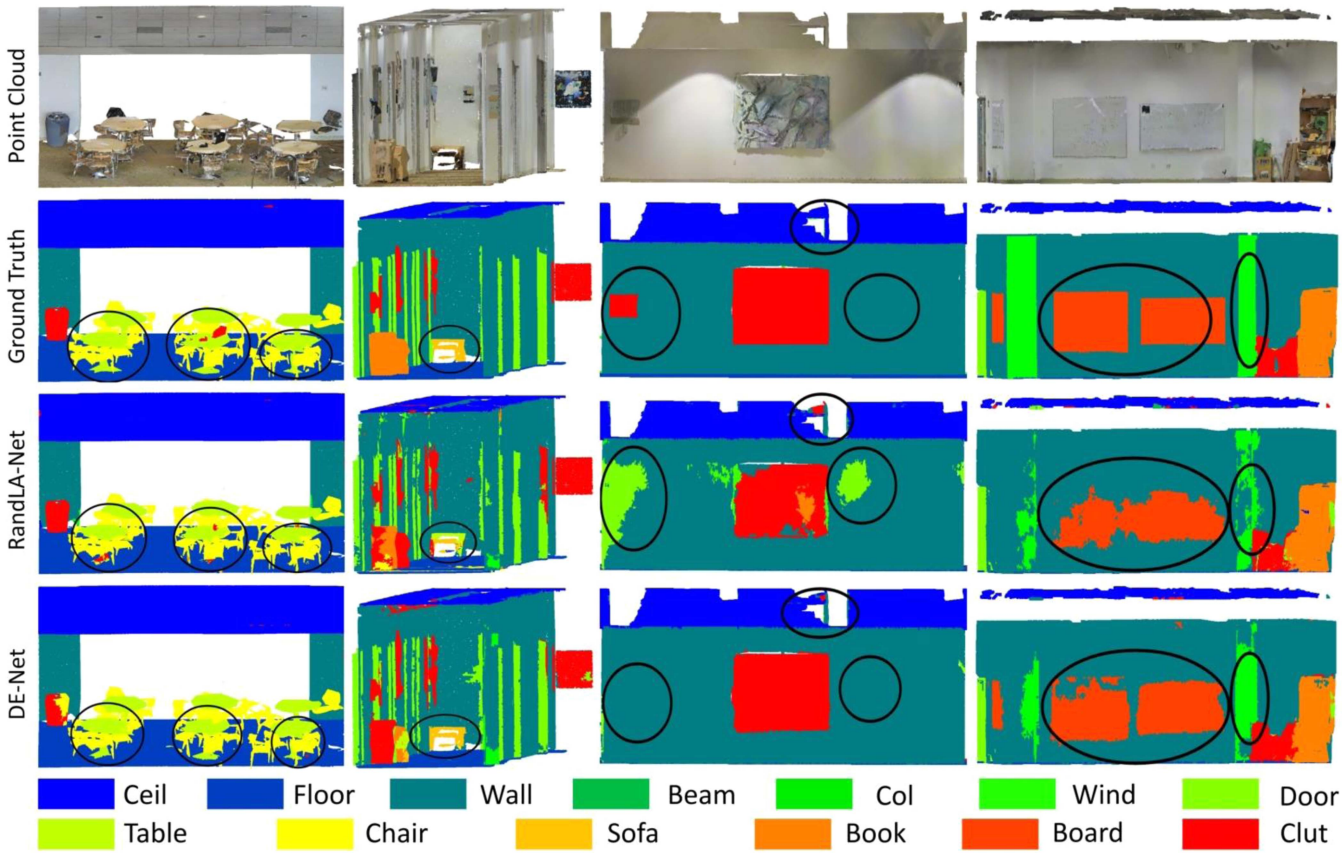

Fig. 7. Comparison of visualization results on the S3DIS dataset.

TABLE V
ABLATION STUDY FOR MAIN COMPONENTS OF DE-NET

Method	OA	mIoU	Road	Road mrk	Natural	Buil.	Util. Line	Pole	Car	Fence
Baseline (B0)	96.3	79.5	95.5	56.3	97.2	92.5	87.1	81.2	89.0	37.8
+PJM (B1)	96.6	80.0	95.9	59.2	96.5	93.5	87.4	82.5	92.4	32.1
+PJM +GAPM (B2)	96.7	81.3	97.6	72.9	97.1	93.0	87.6	81.5	93.5	27.0
+PJM +GAPM +DEM (DE-Net)	97.6	83.5	97.4	70.7	96.8	93.2	87.5	80.6	94.4	47.4

DEM denotes the dual-encoder module, GAPM denotes the gradient-aware pooling module, and PJM denotes the prediction jitter module.

Road mark. This is because there is an overfit. The dual-encoding module improves the feature extraction ability of the network and is overconfident in the prediction results, so we improve this problem with the PJM. Second, due to the problem of category imbalance in the dataset, road marks only account for 2.3% of the total number of points. The misclassification of a few points will cause a large difference in IoU.

2) Core Module Ablations: We compared the impact of the three core modules (DEM, GAPM, and PJM) on network performance. As shown in Table V, from B0 to B1, OA increased by 0.3%, and mIoU increased by 0.5%. We observed an improvement in the segmentation performance for tail categories (road mark, util line, pole, and car) with a low number of points, which is consistent with our hypothesis. It is worth noting that the segmentation performance of the fence category decreases. This is because they resemble the walls of buildings. This shows that the PJM module can improve the category imbalance but it is still affected by the local spatial similarity. From B1 to B2, the network performance is improved, with OA increasing by 1.1% and mIoU increasing by 1.3%. We noticed a 13.7% increase in IoU for the road mark. This is mainly because the weights are directly calculated on the feature map, and the obtained feature information is isolated, whereas the introduction of distance and *xy* gradient information can improve the network's understanding of the context information and make the boundary segmentation of the road mark more accurate. Therefore, the introduction of distance and *xy* gradient information is crucial to improve the prediction ability of boundary points. From B2 to

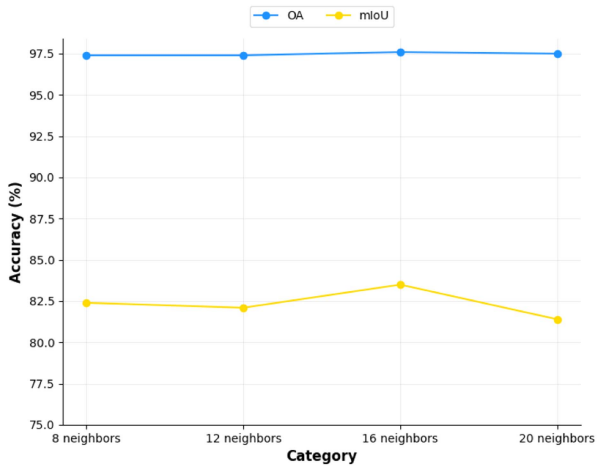

Fig. 8. Effect of different K values on the segmentation result.

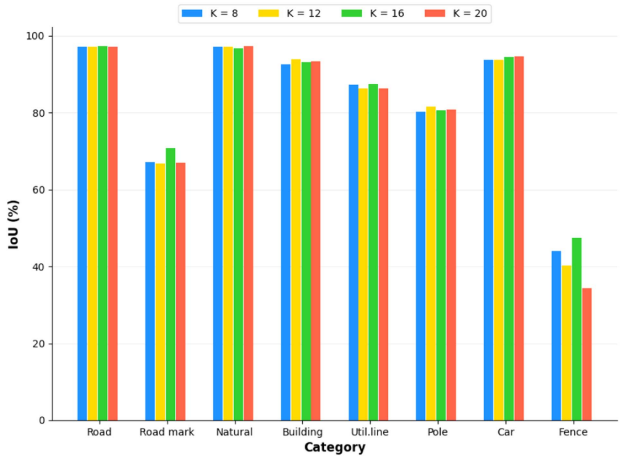

Fig. 9. IoU of each category with different K values.

DE-Net, OA increased by 0.9% and mIoU increased by 2.2%. In particular, the IoU of the fence category was increased by 20.4%, indicating that the dual-encoding module helps to distinguish between two classes of objects with similar characteristics, fence and building. In addition, with the help of the PJM, the robustness of the network is improved, so that the IoU of the Road mark only shows a slight difference.

3) Effectiveness of the Number of Nearest Neighbors: The DEM uses the K-nearest neighbors algorithm to encode the features of the local neighborhood and the long-distance context information. The size of the parameter K determines the size of the local neighborhood, which directly affects the segmentation performance of the network. In Figs. 8 and 9, we explore the effect of different K values on the segmentation results. The results were obtained by DE-Net in the L002 scene of Toronto3D. We take $K = 16$ as the baseline, and when the K value decreases, the local spatial geometry cannot be accurately captured due to the decrease in the number of local points, leading to reduced mIoU by 1.1%–1.4%. However, when the K value increases, the number of local points increases, and it is easy to introduce a variety of different types of points during local feature extraction, resulting in a decrease of 2.1% in mIoU. It is worth noting that when the K value changes from 12 to 8, the mIoU increases by 0.3%. This is because DEM effectively aggregates local features and long-distance contextual information. In addition, this also shows that our DEM is robust to the selection of hyperparameters.

TABLE VI
DIFFERENT VARIANCES ON THE EXPERIMENTAL RESULTS

σ	w/o PJM	3	4	5	6
Baseline (B0)	79.5	80.7	80.0	80.3	80.8
DE-Net (ours)	82.6	83.5	83.5	83.0	83.6

TABLE VII
COMPARISON OF DIFFERENT MODEL SIZE AND EFFICIENCY

Method	Parameter (million)	FLOPs (G)	Inference time (s)	mIoU (%)
RandLA-Net [14]	4.99	3.13	262.3	81.8
BAF-LAC [48]	11.64	7.71	266.4	82.2
KPConv [46]	24.37	—	—	81.5
Baseline	5.51	4.27	253.8	79.5
DE-Net (ours)	6.23	8.65	268.9	83.5

4) Exploration of Variance in PJM Modules: Since the PJM has a unique hyperparameter variance σ, we further explore the optimal variance σ to generalize it to other tasks. Table VI shows the effects of different variance σ on the experimental results of baseline and DE-Net in the Toronto-3D dataset. PJM improved baseline by $+1.3\%$ when $\sigma = 6$ and DE-Net by $+1.0\%$ when $\sigma = 6$. Although the choice of σ is different, its impact on the final performance is quite small (the difference between maximum and minimum mIoU is within $+1\%$), indicating that our PJM is somewhat robust to hyperparameter choice.

E. Model Complexity and Accuracy Analysis

To analyze the efficiency and computational requirements of DE-Net, we calculated the number and complexity of parameters that can be trained by the network. In addition, we compared the inference time and accuracy on Toronto-3D with some networks. As can be seen from Table VII, the number of parameters of DE-Net is only a quarter of that of KPConv, but it has large FLOPs. We believe that the reason for the large FLOPs of DE-Net is that it involves high-dimensional embedding and exponential transform operations, resulting in a large amount of computation. In contrast, we recommend that DE-Net consume a certain amount of computational resources to achieve optimal segmentation performance.

V. DISCUSSION

Our DE-Net has achieved good results in some challenging categories. For example, the road mark, car, and fence in Toronto-3D. Through detailed ablation experiments, we believe that this can be attributed to the similarity between the road mark and the road, fence, and building walls. Although RandLA-Net

has a local spatial coding module to achieve efficient point cloud segmentation and satisfactory segmentation results, it still lacks the aggregation of local features and long-distance context information, and cannot produce accurate and complete segmentation results. In contrast, DE-Net has developed an efficient dual-encoding module that can effectively aggregate point-by-point local and long-distance discriminative features from the input point cloud. This enables the network to extract complete object structure features in complex large-scale scenes and produce accurate segmentation results with local similarities.

In addition, the light and trff. light categories in the WHU-MLS dataset have achieved good segmentation results. This is due to the fact that both light and trff. light belongs to poles and ancillary structures with similar local geometries. DE-Net uses a DEM to perceive the pole and its ancillary structure as a whole, enabling efficient segmentation. However, RandLA-Net uses only the local space encoder module to extract local features, which can easily lead to confusion.

Unlike outdoor scenes, in S3DIS indoor scenes, the table and chair may overlap in the horizontal direction when projected. In DE-Net, there are not only *xy* plane distances as features, but also a combination of absolute coordinates, relative distances, gradients, and other features, which are fused to overcome the limitations that a single feature may bring. For example, *Z*-axis information for coordinates can help distinguish the height of tables and chairs and *xy* gradient information can help distinguish whether the surface of a table and chair is flat or curved.

In addition, through ablation experiments of the DEM, GAPM, and PJM modules, we found that the DEM and GAPM modules required approximately 2 GB of additional computational resources during the training phase, whereas the PJM model required approximately 3 GB of additional computational resources. However, the PJM modules can be discarded during the prediction phase, so the PJM modules can help train models with better performance and do not require excessive computing resources once deployed. It is important to note that a necessary condition for PJM is that the number of points in each category is known. In the unsupervised point cloud semantic segmentation task, this requirement is difficult to meet. Therefore, it is worth exploring to design of a PJM suitable for unsupervised point cloud semantic segmentation tasks.

VI. Conclusion

In this study, we find that the local geometric space similarity in large-scale scenes and the boundary points in the process of global feature aggregation of point clouds are the key factors affecting the accurate segmentation of point clouds. Therefore, we propose a point cloud semantic segmentation method DE-Net for large-scale scenes. The point features are extracted in parallel through positional coding and trigonometric coding. The influence of outliers in feature aggregation is constrained by *xy* gradient and distance information. The data feature differences are balanced by jitter to the prediction results during training. The mIoU of DE-Net in three benchmark datasets including Toronto-3D, WHU-MLS, and S3DIS were 83.5%, 61.8%, and 63.9%, respectively. Ablation experiments and comparisons with state-of-the-art methods show that our proposed network has clear advantages.

References

[1] J. Li, H. Dai, H. Han, and Y. Ding, "MSeg3D: Multi-modal 3D semantic segmentation for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 21694–21704.

[2] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1907–1915.

[3] F. Verdoja, D. Thomas, and A. Sugimoto, "Fast 3D point cloud segmentation using supervoxels with geometry and color for 3D scene understanding," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2017, pp. 1285–1290.

[4] A. Mousavian, C. Eppner, and D. Fox, "6-DOF GraspNet: Variational grasp generation for object manipulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2901–2910.

[5] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg, "Deep projective 3D semantic segmentation," in *Proc. 17th Int. Conf. Comput. Anal. Images Patterns*, 2017, pp. 95–107.

[6] C. Xu et al., "Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, vol. 16, pp. 1–19.

[7] E. E. Aksoy, S. Baci, and S. Cavdar, "Salsanet: Fast road and vehicle segmentation in LiDAR point clouds for autonomous driving," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 926–932.

[8] J. Huang and S. You, "Point cloud labeling using 3D convolutional neural network," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 2670–2675.

[9] H.-Y. Meng, L. Gao, Y.-K. Lai, and D. Manocha, "Vv-net: Voxel vae net with group convolutions for point cloud segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8500–8508.

[10] X. Zhu et al., "Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9939–9948.

[11] 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. 652–660.

[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," (in English), in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5105–5114.

[13] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu, "PointSIFT: A SIFT-like network module for 3D point cloud semantic segmentation," 2018, *arXiv:1807.00652*.

[14] Q. Hu et al., "Learning semantic segmentation of large-scale point clouds with random sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8338–8354, Nov. 2022.

[15] S. Fan, Q. Dong, F. Zhu, Y. Lv, P. Ye, and F.-Y. Wang, "SCF-Net: Learning spatial contextual features for large-scale point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14499–14508.

[16] X. Lai et al., "Stratified transformer for 3D point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8490–8499.

[17] Y. Xu et al., "NeiEA-NET: Semantic segmentation of large-scale point cloud scene via neighbor enhancement and aggregation," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 119, 2023, Art. no. 103285.

[18] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, "Spherical transformer for LiDAR-based 3D recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17545–17555.

[19] C. Park, Y. Jeong, M. Cho, and J. Park, "Fast point transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16928–16937.

[20] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16259–16268.

[21] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on *X*-transformed points," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 828–838.

[22] S. Qiu, S. Anwar, and N. Barnes, "Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1757–1767.

[23] W. Tan et al., "Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 202–203.

[24] X. Han, Z. Dong, and B. Yang, "A point-based deep learning network for semantic segmentation of MLS point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 175, pp. 199–214, 2021.
[25] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1534–1543.
[26] D. Lee and J. Kim, "Resolving class imbalance for LiDAR-based object detector by dynamic weight average and contextual ground truth sampling," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 682–691.
[27] O. Shrout, Y. Ben-Shabat, and A. Tal, "GraVoS: Voxel selection for 3D point-cloud detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 21684–21693.
[28] P.-S. Wang, "OctFormer: Octree-based transformers for 3D point Clouds," *ACM Trans. Graph.*, vol. 42, 2023, Art. no. 155.
[29] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCTt: Point cloud transformer," *Comput. Vis. Media*, vol. 7, pp. 187–199, 2021.
[30] M. Mao et al., "Dual-stream network for visual recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 25346–25358.
[31] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel CNN for efficient 3D deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 965–975.
[32] Y. Li, X. Li, Z. Zhang, F. Shuang, Q. Lin, and J. Jiang, "DenseKPNET: Dense kernel point convolutional neural networks for point cloud semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5702913.
[33] X. Xiang, L. Wang, W. Zong, and G. Li, "Extraction of local structure information of point clouds through space-filling curve for semantic segmentation," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 114, 2022, Art. no. 103027.
[34] H. Li et al., "MVPNet: A multi-scale voxel-point adaptive fusion network for point cloud semantic segmentation in urban scenes," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 122, 2023, Art. no. 103391.
[35] K. Zhang et al., "A dual attention neural network for airborne LiDAR point cloud semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5704617.
[36] S. Su et al., "PUPS: Point cloud unified panoptic segmentation," in *Proc. 37th AAAI Conf. Artif. Intell., 35th Conf. Innov. Appl. Artif. Intell., 13th Symp. Educ. Adv. Artif. Intell.*, 2023, pp. 2339–2347, doi: 10.1609/aaai.v37i2.25329.
[37] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018, Art. no. 3337.
[38] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "RPVNet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16004–16013.
[39] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1887–1893.
[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.
[41] R. Zhang, L. Wang, Y. Wang, P. Gao, H. Li, and J. Shi, "Starting from non-parametric networks for 3D point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 5344–5353.
[42] H. Hu et al., "GAM: Gradient attention module of optimization for point clouds analysis," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 835–843.
[43] Y. Wang et al., "Balancing logit variation for long-tailed semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 19561–19573.
[44] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
[45] 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.
[46] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6411–6420.
[47] J. Du et al., "ResDLPS-Net: Joint residual-dense optimization for large-scale point cloud semantic segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 182, pp. 37–51, 2021.
[48] H. Shuai, X. Xu, and Q. Liu, "Backward attentive fusing network with local aggregation classifier for 3D point cloud semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 4973–4984, 2021.
[49] J. Chen, Y. Zhao, C. Meng, and Y. Liu, "Multi-feature aggregation for semantic segmentation of an urban scene point cloud," *Remote Sens.*, vol. 14, no. 20, 2022, Art. no. 5134.
[50] Z. Zeng, Y. Xu, Z. Xie, W. Tang, J. Wan, and W. Wu, "Large-scale point cloud semantic segmentation via local perception and global descriptor vector," *Expert Syst. Appl.*, vol. 246, 2024, Art. no. 123269.
[51] X. Li, Z. Zhang, Y. Li, M. Huang, and J. Zhang, "SFL-NET: Slight filter learning network for point cloud semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5703914.
[52] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9621–9630.
[53] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2018, pp. 4558–4567.
[54] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia, "Hierarchical point-edge interaction network for point cloud semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10432–10440.
[55] J. Chen, B. Kakillioglu, and S. Velipasalar, "Background-aware 3-D point cloud segmentation with dynamic point feature aggregation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5703112.
[56] 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.*, vol. 35, no. 4, pp. 4798–4812, Apr. 2024.

Zhipeng He received the M.S. degree in surveying and mapping engineering from the Nanjing Normal University, Nanjing, China, in 2024.

His research interests include LiDAR remote sensing, intelligent point cloud processing, and 3-D point cloud semantic segmentation.

Jing Liu received the Ph.D. degree in natural resources from ITC, University of Twente, Enschede, The Netherlands, in 2019, and the Ph.D. degree in geospatial sciences from RMIT University, Melbourne, VIC, Australia, in 2019.

She is currently an Associate Professor with the School of Geography, Nanjing Normal University, Nanjing, China. Her research interests include LiDAR remote sensing, vegetation remote sensing, and intelligent point cloud processing.

Shuai Yang received the M.S. degree in cartography and geography information systems from Peking University, Beijing, China, in 2014.

His research interests include cartography, point cloud processing, and virtual geographic environments.